

BAB IV

HASIL DAN PEMBAHASAN

Sebelum membahas lebih lanjut pembahasan RUP, semua iterasi yang dilakukan pada penelitian dapat dilihat di <https://github.com/agusgokasi/emonik> tiap iterasinya tercatat sebagai branch pada github tersebut.

A. Tahap *Inception*

Pada tahap *inception* dilakukan proses RUP yaitu berdasarkan iterasi *inception* I1 yang sudah dibahas dalam BAB III subbab iterasi *inception* I1. Iterasi pertama dalam *inception* I1 berdasarkan *development plan* yaitu 35% *Business Modelling*, 20% *Requirements* dan 15% *Analysis & Design*. Artefak yang dibahas dalam tiap proses diharapkan sudah mencapai target *development plan* yaitu pada proses *Business Modelling* diharapkan telah mencapai 35% targetnya dengan dibahasnya artefak *Business Rules*, *Business Vission*, *Business architecture document*, *Business actor* dan *Business use case*, sedangkan pada proses *Requirements* diharapkan telah mencapai 20% targetnya dengan dibahasnya artefak *Requirements Management Plan* dan *Software Requirements Specification*. Selanjutnya pada proses *Analysis & Design* diharapkan telah mencapai 15% targetnya dengan dibahasnya artefak *Use case realization*.

B. Tahap *Elaboration*

Pada tahap *elaboration* dilakukan proses RUP yaitu berdasarkan iterasi *elaboration* yang terdapat 2 iterasi yaitu iterasi E1 dan E2. Pada subbab ini akan dilakukan pembahasan tiap iterasi pada tahap *elaboration*.

1. Iterasi *elaboration* E1

Iterasi selanjutnya pada tahap *elaboration* E1 berdasarkan *development plan* yaitu dibahasnya 35% *Business Modelling*, 35% *Requirements*, 25% *Analysis & Design* dan 10% *Implementation*.

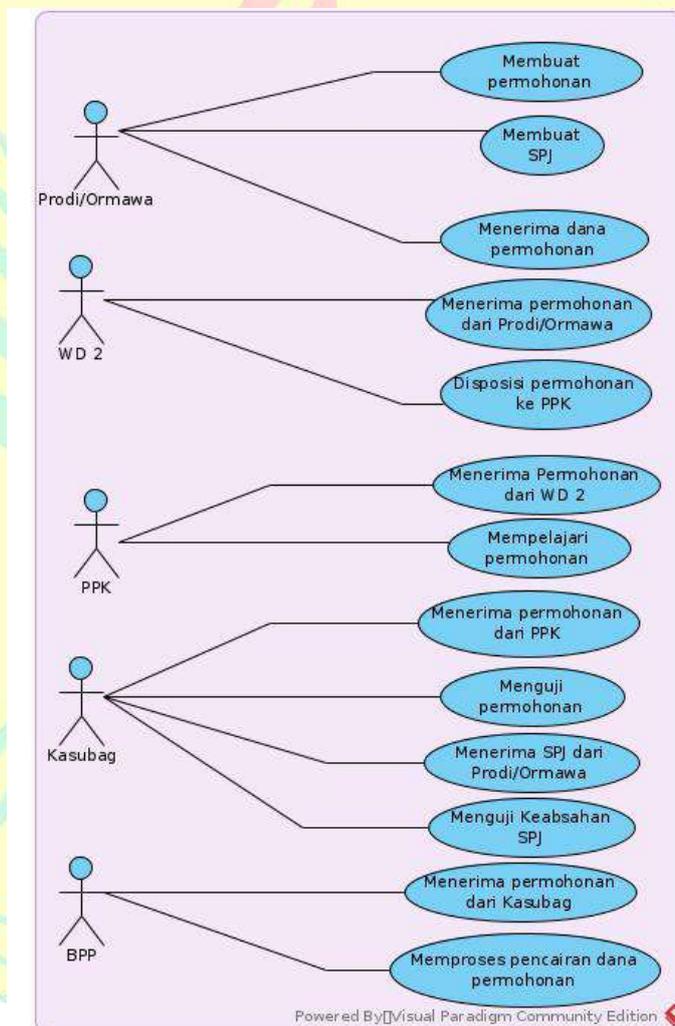
a. *Business Modelling*

Target pada proses *Business Modelling* ini diharapkan sudah mencapai 35% pada iterasi *elaboration* E1 ini dan mencapai target 70% pada target

development plan. Pada proses *Business Modelling* ini akan dibahas artefak lanjutan *Business use case* berdasarkan iterasi sebelumnya dan pembahasan artefak *Business use case* model.

1. *Business use case*

Target pada artefak ini adalah melanjutkan *business use case* iterasi sebelumnya dengan berdasarkan kegiatan SOP Keuangan. Pada iterasi ditambahkan beberapa *business use case* untuk memperjelas alur pengajuan permohonan dana. Perubahan tersebut bisa dilihat pada gambar berikut.



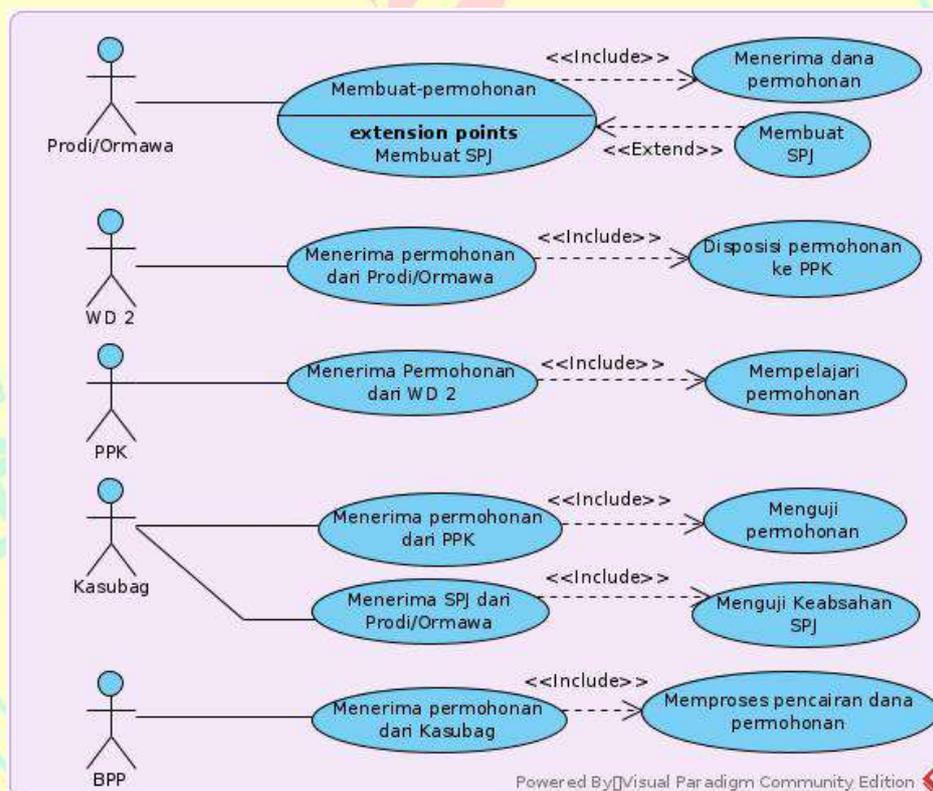
Gambar 4.1 : Penambahan *Business use case* pada iterasi E1

Pada gambar 4.1 sudah terlihat bagaimana jalannya proses pengajuan permohonan dari mulai Prodi/Ormawa membuat permohonan, dilanjutkan WD 2 menerima permohonan dari Prodi/Ormawa dan mendisposisikan

permohonannya ke PPK, kemudian PPK akan mempelajari permohonan tersebut dan dilanjutkan ke kasubag, selanjutnya kasubag akan menguji permohonannya sebelum dilanjutkan ke BPP, BPP akan menerima permohonan yang sah dan akan memproses pencairan dana permohonan tersebut, setelah dana permohonan tersebut cair maka BPP akan menyerahkan dana kegiatan ke Prodi/Ormawa yang mengajukan permohonan tersebut.

2. Business use case model

Target pada artefak ini untuk memperjelas konteks bisnis dari sistem yang akan dibangun. *Business use case model* disusun berdasarkan *business use case* gambar 4.1 yang dapat dilihat pada gambar berikut.



Gambar 4.2 : *Business use case model* pada iterasi E1

b. Requirements

Target pada proses *Requirements* ini diharapkan sudah mencapai 35% pada iterasi *elaboration* E1 ini dan mencapai target 55% pada target *development plan*. Pada proses *Requirements* ini akan dibahas artefak *Actor*

(*human*), *Use Case*, *Use Case Model*, *Supplementary specification* dan penambahan *Software Requirements Specification* dari iterasi sebelumnya.

1. *Actor (human)*

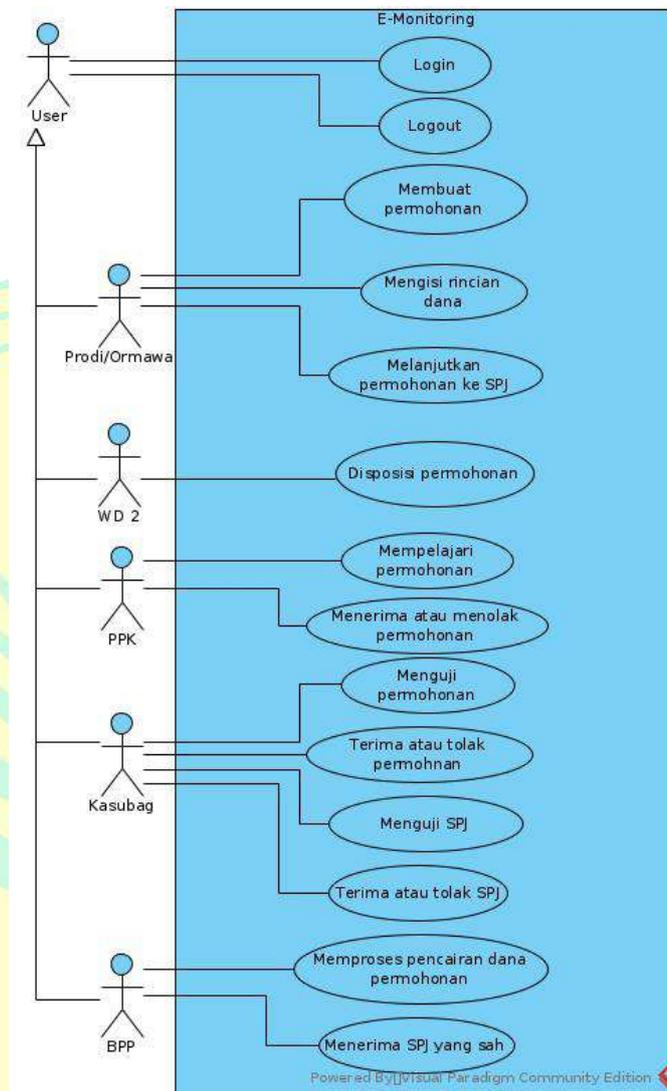
Target pada artefak ini mendefinisikan serangkaian peran yang koheren yang dapat dimainkan oleh pengguna sistem saat berinteraksi dengannya. Instance aktor dapat dimainkan oleh individu atau sistem eksternal. Artefak ini hampir sama dengan artefak *Business actor* tetapi ada beberapa penambahan atau perubahan sesuai peran yang dimainkan oleh pengguna sistem. Berikut adalah *Actor (human)* yang terlibat pada sistem ini:

- a) Prodi/Ormawa sebagai pemohon dalam sistem dan dapat membuat permohonan termasuk melakukan SPJ.
- b) WD 2 mempunyai peran untuk disposisi permohonan.
- c) PPK mempunyai peran untuk menilai.
- d) BPP mempunyai peran untuk menerima permohonan yang sah dan memproses pencairan dana kegiatannya.
- e) Kasubag mempunyai peran untuk menguji permohonan.

2. *Use Case*

Target pada artefak ini mendefinisikan sekumpulan contoh kasus penggunaan, di mana setiap contoh adalah urutan tindakan yang dilakukan sistem yang menghasilkan hasil nilai yang dapat diamati untuk aktor tertentu. Pada iterasi sebelumnya sudah pernah dibahas *use case* awal sebagai dasar pembuatan sistem, pada iterasi E2 ini akan dibahas *use case* lanjutannya untuk dijelaskan *use case* yang relevan untuk dianalisis dan dijelaskan secara rinci. *Use case* pada iterasi ini dapat dilihat pada gambar 4.3.

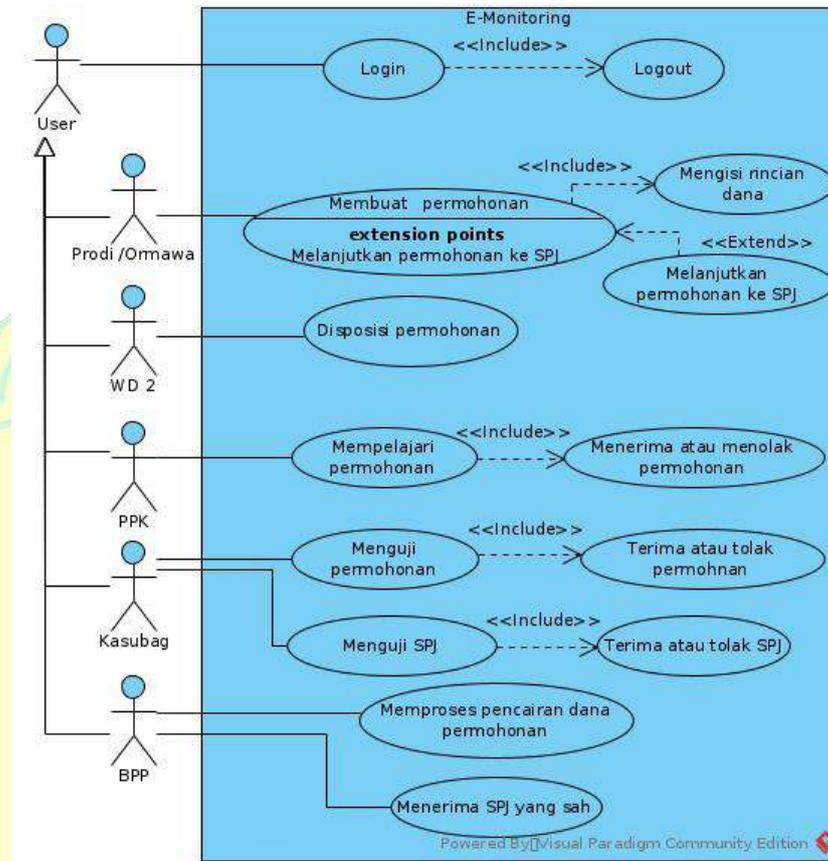
Pada gambar 4.3 melanjutkan *use case* awal yang ada di iterasi sebelumnya, dengan penambahan beberapa *use case* pada iterasi E1. penjelasan gambar 4.3 hampir sama dengan penjelasan pada artefak *business use case* pada proses *business modelling* pada iterasi E1 namun terdapat perbedaan dimana semua aktor digeneralisasi menjadi user yang dapat melakukan tindakan *login* dan *logout* pada sistem.



Gambar 4.3 : Use Case pada iterasi E1

3. Use Case Model

Target pada artefak ini model fungsi sistem yang dimaksudkan dan lingkungannya, dan berfungsi sebagai kontrak antara pelanggan dan pengembang. *Use Case Model* digunakan sebagai masukan penting untuk kegiatan dalam analisis, desain, dan pengujian. *Use Case Model* disusun berdasarkan *use case* gambar 4.3 yang dapat dilihat pada gambar berikut.



Gambar 4.4 : Use Case Model pada iterasi E1

4. Supplementary specification

Target pada artefak ini untuk Spesifikasi tambahan mencakup persyaratan sistem yang tidak langsung ditangkap dalam *use case* di *use case model*. Berikut ini tabel Spesifikasi tambahan persyaratan sistem.

Tabel 4.1 : Spesifikasi tambahan persyaratan sistem

No	Nama	Spesifikasi	Fungsi
1	<i>Framework bootstrap</i>	Version 3	<i>Framework</i> untuk mempermudah proses implementasi <i>front-end</i> .
2	<i>Framework Laravel</i>	Version 5.8	<i>Framework</i> untuk mempermudah proses implementasi <i>back-end</i> .

5. Software Requirements Specification

Pada iterasi E1 ini dilakukan sedikit penambahan yaitu mulai digunakannya github sebagai tempat code program dan dokumentasi artefak.

Tabel 4.2 : Penambahan Daftar *Software*

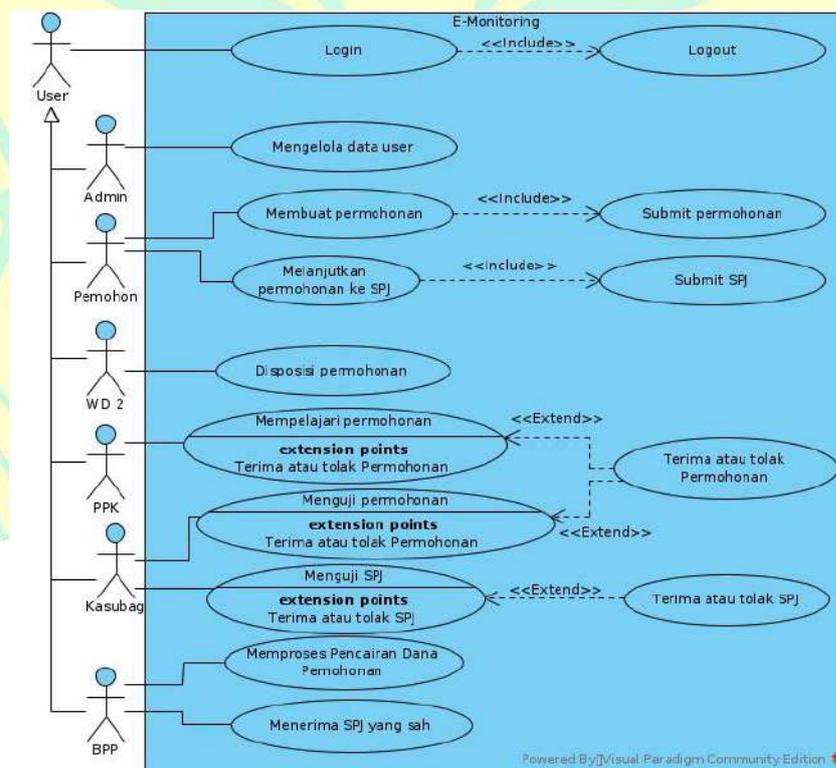
No	Nama	Spesifikasi	Fungsi
1	<i>Github</i>	<i>Free version</i>	Untuk manajemen dan <i>back up</i> kode program dan dokumentasi setiap artefak.

c. *Analysis & Design*

Target pada proses *Analysis & Design* ini diharapkan sudah mencapai 25% pada iterasi *elaboration* E1 ini dan mencapai target 40% pada target *development plan*. Pada proses *Analysis & Design* ini akan dibahas artefak *Use Case Realization* penyelesaian dari iterasi sebelumnya dan *Design model*.

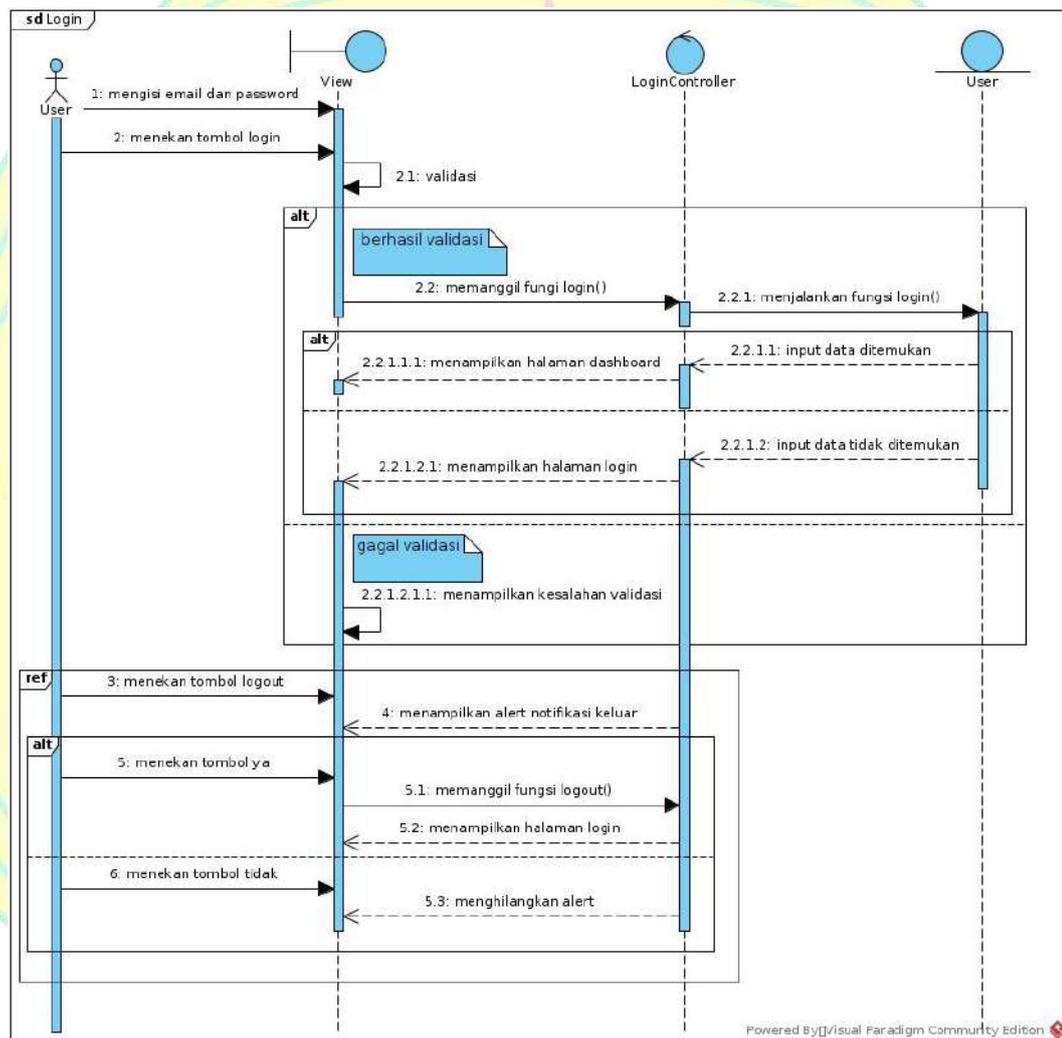
1. *Use Case Realization*

Pada iterasi E1 ini diharapkan artefak ini telah selesai dengan dilakukan penambahan proses pengajuan permohonan dan SPJ serta penambahan role Admin untuk mengelola data *user*. *Use case Realization* pada iterasi ini dapat dilihat pada gambar 4.5.

Gambar 4.5 : *Use case Realization* Final

2. Design model

Target pada artefak ini model objek yang menggambarkan realisasi *use case*, dan berfungsi sebagai abstraksi model implementasi dan kode sumbernya. Pada iterasi ini akan dibahas *sequence diagram login* sebagai dasar pembuatan sistem. *Sequence diagram login* dapat dilihat pada gambar berikut.



Gambar 4.6 : *Sequence diagram login*

Pada gambar 4.6 terdapat *Sequence diagram login* yang menjelaskan bagaimana proses kegiatan saat mulai melakukan aktivitas pada sistem. Pertama, *user* mengisi *email* dan *password* pada formulir di halaman *login*. Setelah selesai mengisi formulir, *user* akan menekan tombol *login*, kemudian sistem akan melakukan validasi dari *input* yang dimasukkan. Jika validasinya

berhasil, maka sistem akan menampilkan halaman *dashboard*, jika gagal maka akan kembali ke halaman *login* dan menampilkan info kesalahannya. Setelah *user* selesai beraktifitas didalam sistem, *user* dapat melakukan *logout* dengan menekan tombol *logout* pada menu yang tersedia. Setelah *user* menekan tombol *logout*, akan muncul notifikasi peringatan, jika *user* menekan ya, maka *user* tersebut akan segera *logout*, jika tidak maka akan tetap pada halaman yang *user* tempati.

d. *Implementation*

Target pada proses *Implementation* ini diharapkan sudah mencapai 10% pada iterasi *elaboration* E1 ini. Pada proses *Implementation* ini akan dibahas artefak *Integration build plan*.

1. *Integration build plan*

Target pada artefak ini memberikan rencana detail untuk integrasi dalam sebuah iterasi. Tujuan dari *Integration build plan* adalah untuk menentukan urutan implementasi komponen dan subsistem, build yang akan dibuat saat mengintegrasikan sistem, dan bagaimana penilaiannya. *Integration build plan* direncanakan dalam iterasi saat ini dan dimodifikasi sesuai kebutuhan selama iterasi. Berikut adalah tabel urutan pembuatan sistem yang dimulai dari sistem *login*, pembuatan *role user*, pengajuan permohonan dan SPJ.

Tabel 4.3 : *Integration build plan* pada iterasi E1

No	<i>Build plan</i>	Target
1	Pembuatan sistem <i>login</i> .	Iterasi E2-C1
2	Pembuatan <i>role user</i> .	Iterasi C1
3	Pembuatan sistem pengajuan permohonan	Iterasi C2
4	Pembuatan sistem SPJ	Iterasi C3
5	Perbaikan <i>bugs</i>	Iterasi T1

2. Iterasi *elaboration* E2

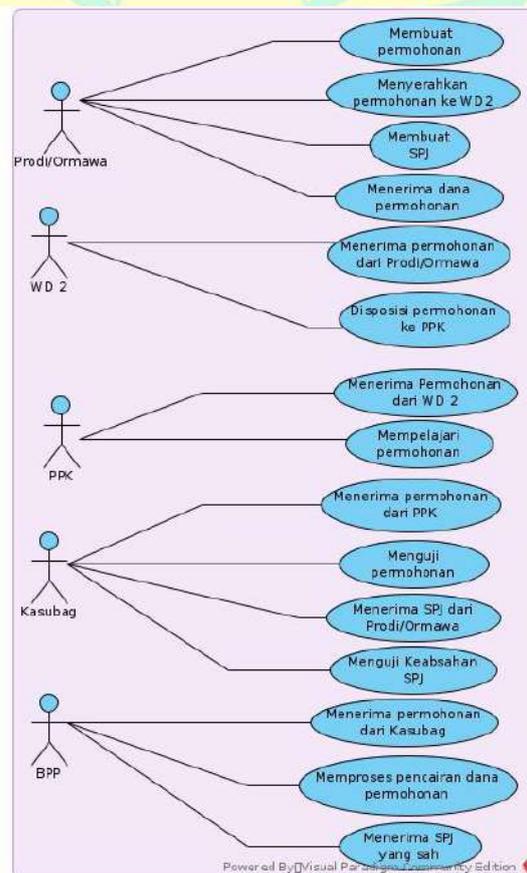
Iterasi selanjutnya pada tahap *elaboration* E2 berdasarkan *development plan* yaitu dibahasnya 20% *Business Modelling*, 35% *Requirements*, 30% *Analysis & Design*, 10% *Implementation* dan 10% *test*.

a. *Business Modelling*

Target pada proses *Business Modelling* ini diharapkan sudah mencapai 20% pada iterasi *elaboration* E2 ini dan mencapai target 90% pada target *development plan* yang artinya sudah hampir selesai semua proses pada *business modelling* ini dan akan menyesuaikan penambahan sesuai kebutuhan. Pada proses *Business Modelling* ini akan dibahas artefak lanjutan *Business use case* dan lanjutan pembahasan artefak *Business use case model* berdasarkan iterasi sebelumnya.

1. *Business use case*

Target pada artefak ini adalah melanjutkan *business use case* iterasi sebelumnya dengan berdasarkan kegiatan SOP Keuangan. Pada iterasi E2 ditambahkan beberapa *business use case* untuk memperjelas alur pengajuan permohonan dana. Perubahan tersebut bisa dilihat pada gambar berikut.

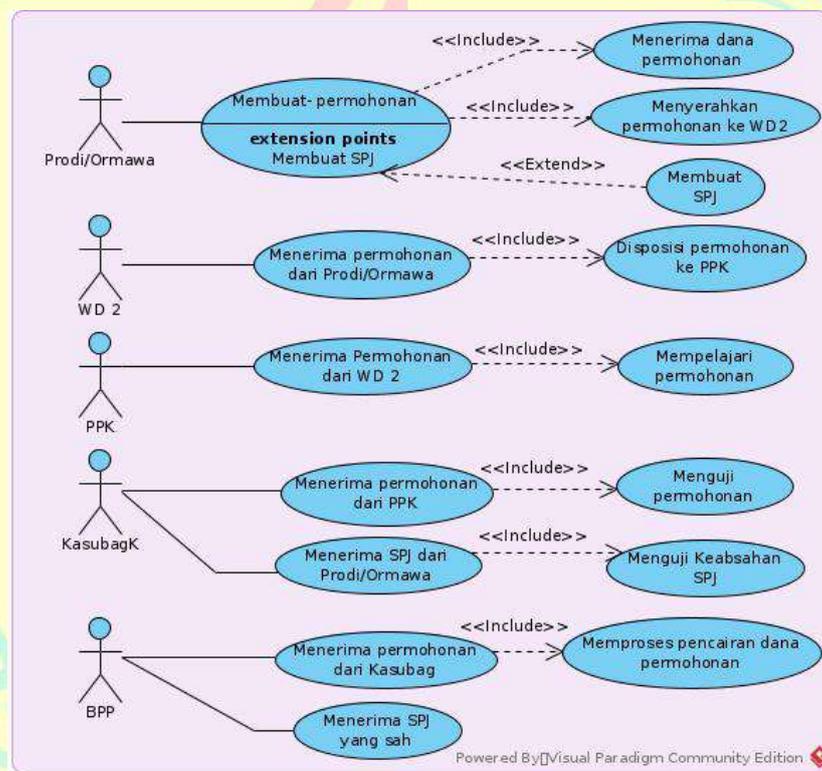


Gambar 4.7 : Penambahan *Business use case* pada iterasi E2

Pada gambar 4.7 ditambahkan beberapa *business use case* untuk lebih memperjelas alur SOP keuangan, yaitu ditambahkan pada prodi/ormawa Menyerahkan permohonan ke WD 2 dan pada BPP menerima SPJ yang sah.

2. *Business use case model*

Melanjutkan artefak ini dari terasi sebelumnya, *Business use case model* pada iterasi E2 ini disusun berdasarkan *business use case* gambar 4.7 yang dapat dilihat pada gambar berikut.



Gambar 4.8 : *Business use case model* pada iterasi E2

b. *Requirements*

Target pada proses *Requirements* ini diharapkan sudah mencapai 35% pada iterasi *elaboration* E2 ini dan mencapai target 90% pada target *development plan* yang artinya sudah hampir selesai semua proses pada *Requirements* ini dan akan menyesuaikan penambahan sesuai kebutuhan. Pada proses *Requirements* ini akan dibahas artefak *Stakeholder request*, *Glossary*, *Vission*, lanjutan *Use Case* dan lanjutan *Use Case Model* dari iterasi sebelumnya.

1. *Stakeholder request*

Target pada artefak ini berisi semua jenis permintaan yang mungkin dimiliki oleh *stakeholder* (pelanggan, pengguna akhir, staf pemasaran, dan sebagainya) pada sistem yang akan dikembangkan. Ini juga dapat berisi referensi ke semua jenis sumber eksternal yang harus dipatuhi oleh sistem. Berikut adalah tabel *Stakeholder request*.

Tabel 4.4 : *Stakeholder request*

No	<i>Request</i>	<i>Stakeholder</i>
1	Memerlukan notifikasi di <i>website</i> tiap ada sesuatu (permohonan/SPJ) baru yang masuk.	BPP FMIPA UNJ
2	Perlu notifikasi <i>email</i> atau <i>whatsapp</i> tiap ada sesuatu yang baru masuk.	Kasubag Keuangan dan Akuntansi FMIPA UNJ
3	Daya serap anggaran <i>convert</i> ke <i>excel</i> .	BPP FMIPA UNJ
4	Perlu dipertimbangkan kemudahan dalam akses sistem yang dikembangkan agar <i>user</i> tidak banyak beraktifitas di sistem.	Kasubag Keuangan dan Akuntansi FMIPA UNJ

2. *Glossary*

Target pada artefak ini untuk mendefinisikan istilah penting yang digunakan oleh proyek. Berikut adalah tabel *Glossary*.

Tabel 4.5 : *Glossary*

No	<i>Glossary</i>	<i>Deskripsi</i>
1	<i>Framework</i>	wadah atau struktur dari sebuah <i>website</i> yang akan dibangun.
2	Laravel	<i>framework</i> berbasis PHP yang sifatnya <i>open source</i> , dan menggunakan konsep MVC (<i>model-view-controller</i>).

3	MVC	sebuah konsep yang diperkenalkan oleh penemu <i>Smalltalk</i> untuk mengenkapsulasi data bersama dengan pemrosesan (<i>model</i>), mengisolasi dari proses manipulasi (<i>controller</i>) dan tampilan (<i>view</i>) untuk direpresentasikan pada sebuah <i>user interface</i> .
---	-----	--

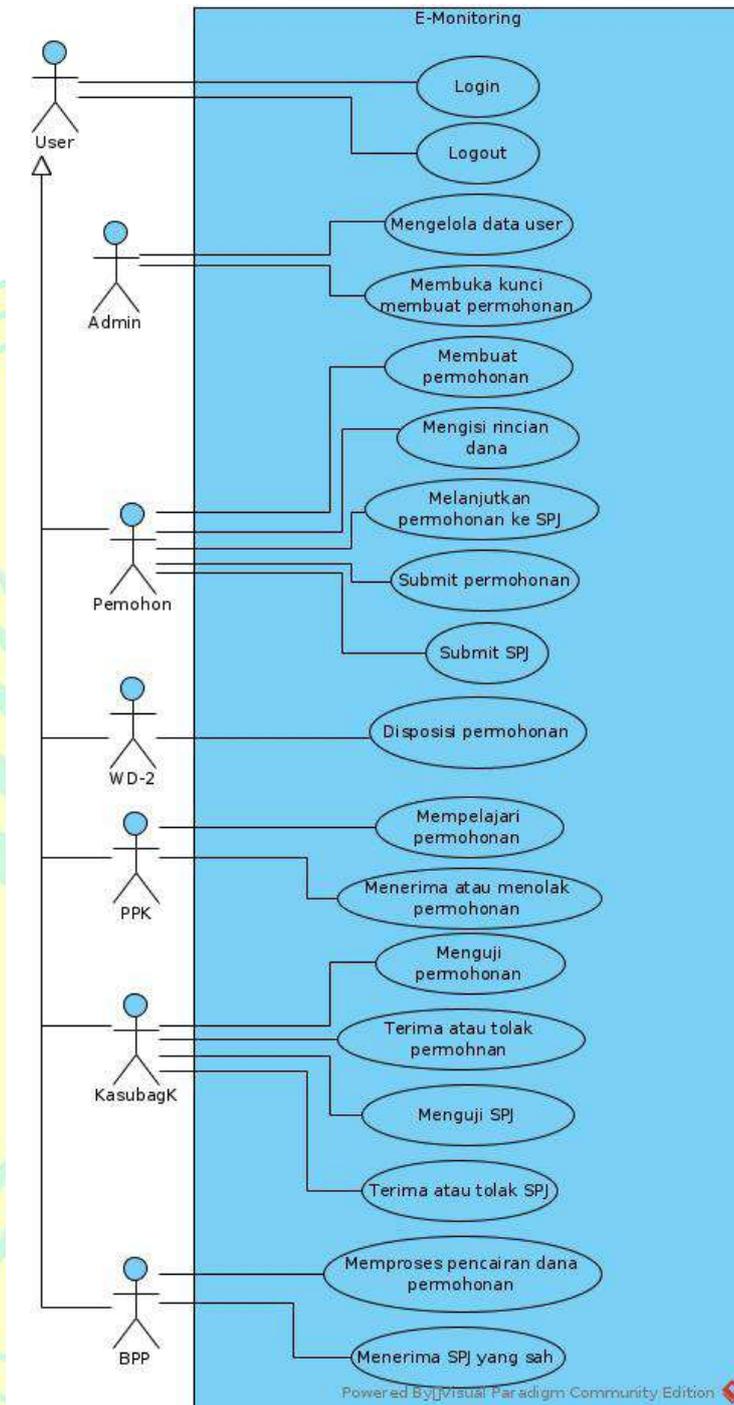
3. *Vission*

Target pada artefak ini mendefinisikan pandangan *stakeholder* dari produk yang akan dikembangkan, ditentukan dalam hal kebutuhan dan fitur utama *stakeholder*. Artefak ini juga merupakan bagian dari *business Vission* yaitu sistem *e-monik* berdasarkan SOP Keuangan yang terdiri dari Sistem monitoring pencairan dana kegiatan mahasiswa dan Sistem monitoring SPJ.

4. *Use Case*

Melanjutkan artefak *use case* pada iterasi sebelumnya dengan berdasarkan kegiatan SOP Keuangan. Pada iterasi E2 ditambahkan beberapa *use case* untuk memperjelas alur pengajuan permohonan dana dan ditambahkan aktor Admin untuk mengelola data *user*. Perubahan tersebut bisa dilihat pada gambar 4.9.

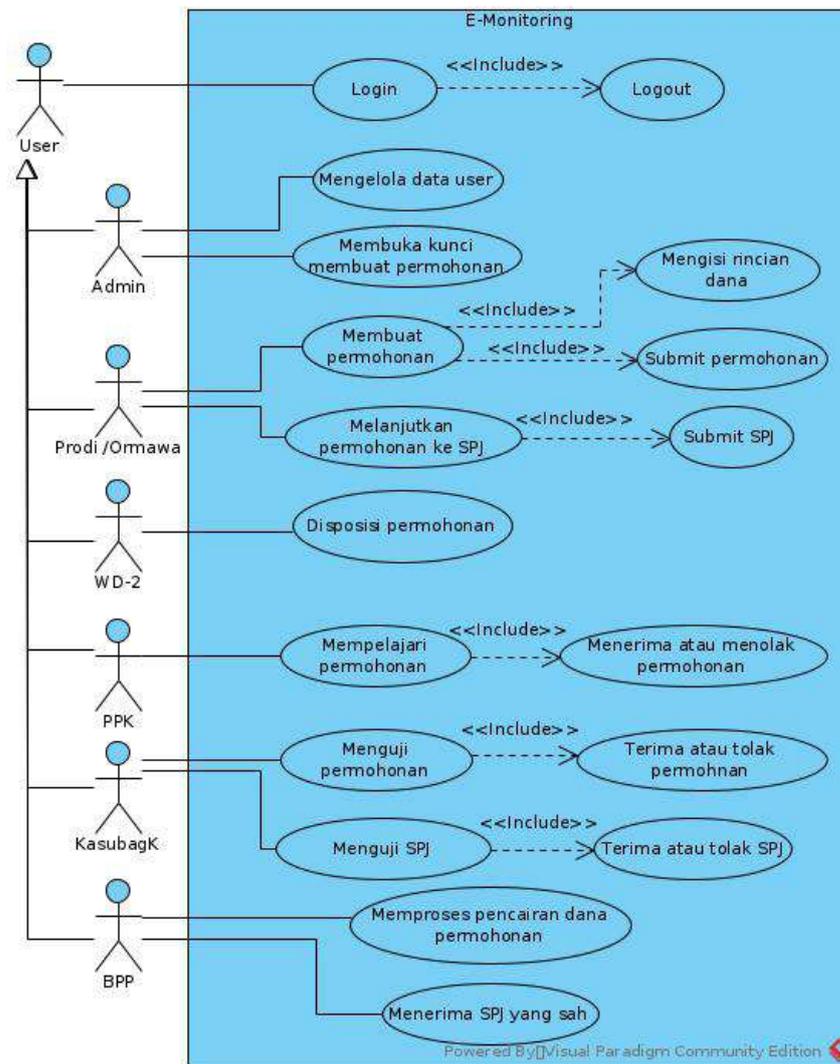
Penjelasan penambahan pada gambar 4.9 adalah penambahan pada aktor Admin yang mempunyai *use case* Mengelola data *user* dan Membuka kunci membuat permohonan. Aktor prodi/ormawa pada *use case* iterasi E2 ini diubah menjadi pemohon agar terkesan seperti *use case* yang terjadi dalam sistem. Selain perubahan nama Aktor, Pemohon juga mendapatkan *use case* baru yaitu *submit* permohonan dan *submit* SPJ pada sistem ini.



Gambar 4.9 : Use Case pada iterasi E2

5. Use Case Model

Melanjutkan dari iterasi sebelumnya, *Use Case Model* pada iterasi E2 ini disusun berdasarkan *use case* gambar 4.9 yang dapat dilihat pada gambar berikut.



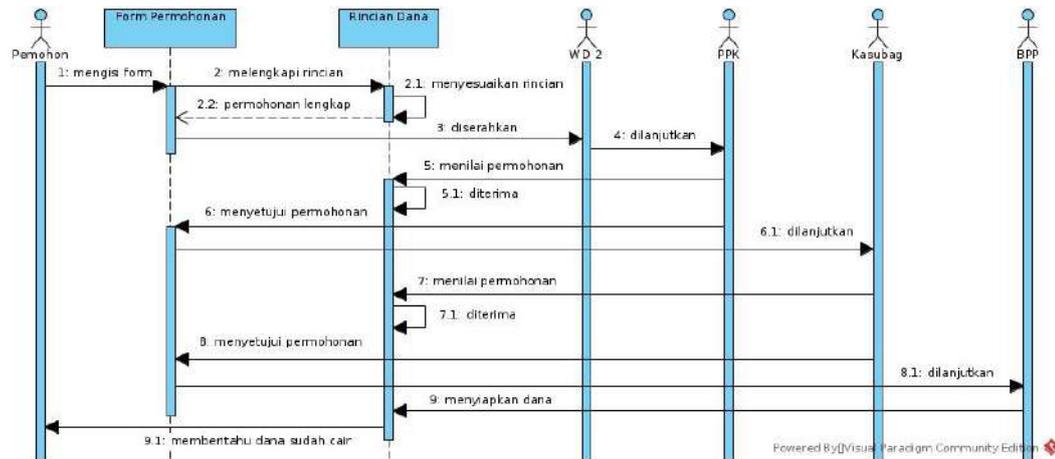
Gambar 4.10 : Use Case Model pada iterasi E2

c. Analysis & Design

Target pada proses *Analysis & Design* ini diharapkan sudah mencapai 30% pada iterasi *elaboration* E2 ini dan mencapai target 70% pada target *development plan*. Pada proses *Analysis & Design* ini akan dibahas artefak lanjutan *Design model* dari iterasi sebelumnya, *Design Class* dan *Data Model*.

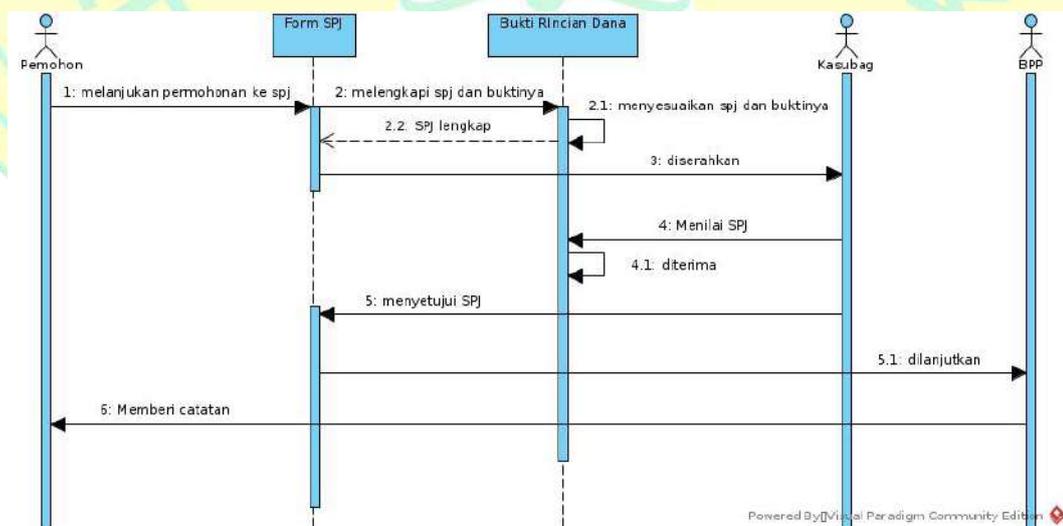
1. Design model

Melanjutkan dari iterasi sebelumnya. Pada iterasi ini akan dibahas *sequence diagram* permohonan dan SPJ secara umum dalam pembuatan sistem. *Sequence diagram* tersebut dapat dilihat pada gambar berikut.



Gambar 4.11 : *Sequence diagram* permohonan

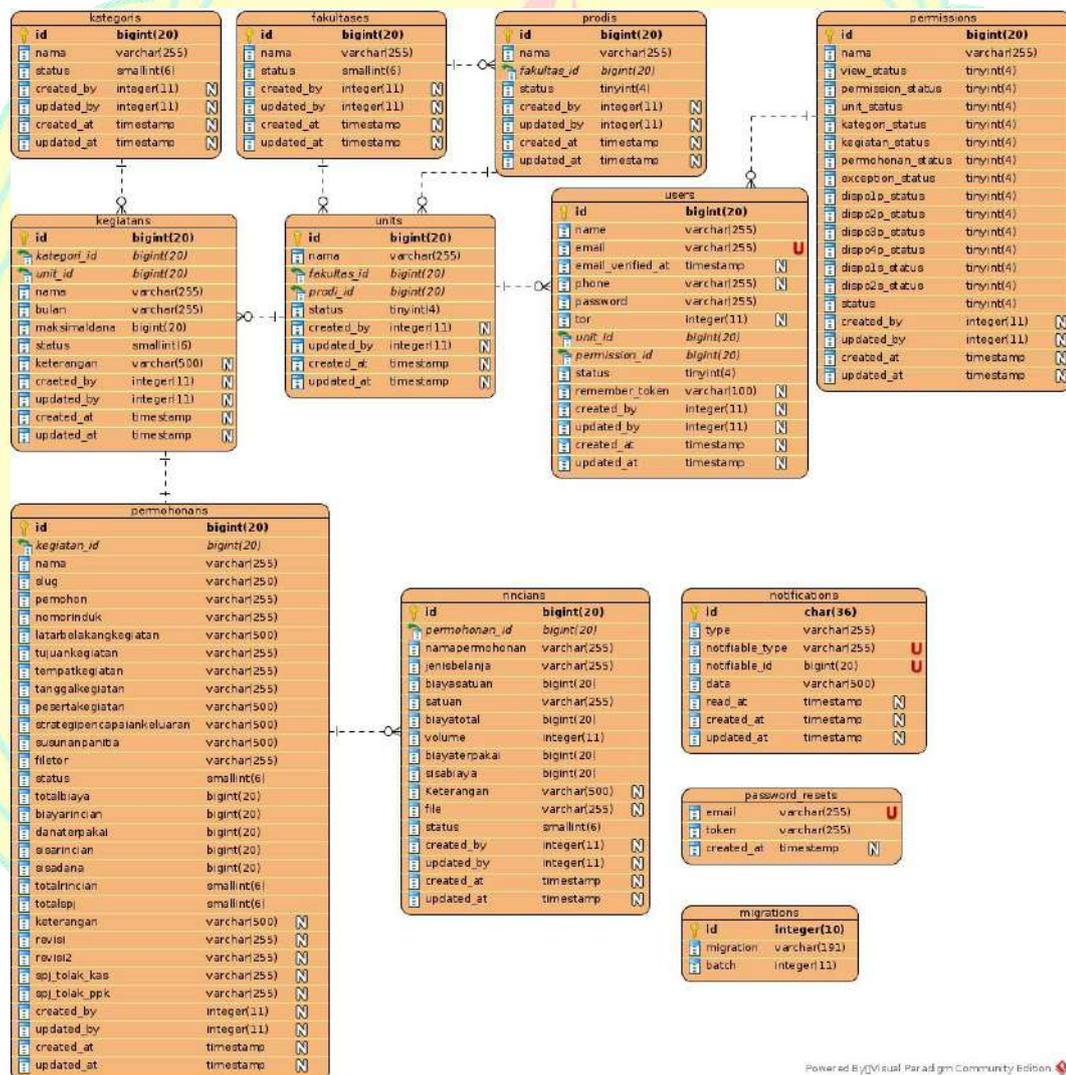
Pada gambar 4.11 menjelaskan jalannya alur permohonan secara umum yang terjadi pada sistem, dimulai dari pemohon mengisi form permohonan, kemudian melengkapi rincian dan menyesuaikannya, setelah selesai maka pemohon akan menyerahkan permohonan tersebut ke WD 2. Setelah WD 2 menerima permohonan, maka akan dilanjutkan ke PPK. Permohonan akan dinilai oleh PPK, jika diterima maka akan dilanjutkan ke Kasubag. Kasubag akan menilai lagi permohonan tersebut, setelah selesai maka akan dilanjutkan ke BPP. BPP akan menyiapkan dana permohonan tersebut dan jika sudah cair, akan memberitahu pemohon tentang pencairan dana permohonan.



Gambar 4.12 : *Sequence diagram* SPJ

3. Data Model

Target pada artefak ini membuat ERD (*entity relationship diagram*) sistem. *Data Model* adalah bagian dari model implementasi yang menggambarkan representasi logis dan fisik dari data persisten dalam sistem. Ini juga mencakup perilaku apa pun yang ditentukan dalam database, seperti prosedur tersimpan, pemicu, batasan, dan sebagainya. Berikut adalah gambar ERD (*entity relationship diagram*) Sistem.



Gambar 4.14 : Entity Relationship Diagram Sistem

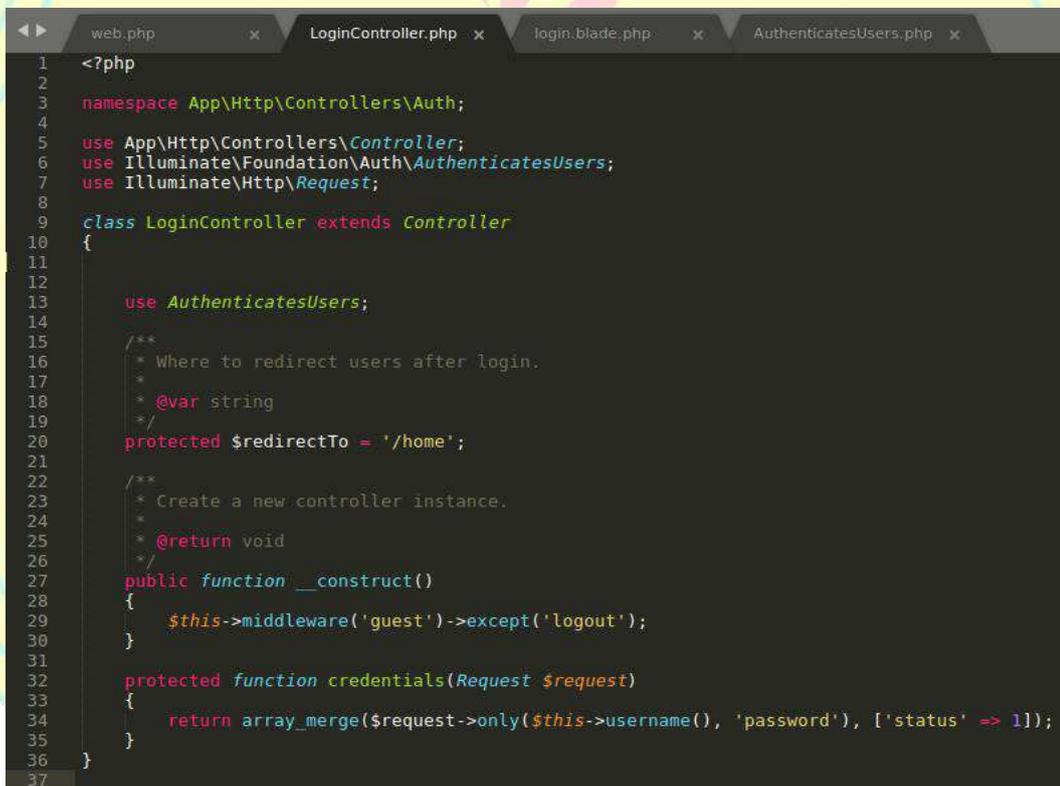
d. Implementation

Target pada proses *Implementation* ini diharapkan sudah mencapai 10% pada iterasi *elaboration* E2 ini dan mencapai target 20% pada target

development plan. Pada proses *Implementation* ini akan dibahas artefak *Component* dan *Implementation Model*.

1. *Component*

Target pada artefak ini untuk mewakili bagian dari kode perangkat lunak (*source*, *binary* atau *executable*), atau *file* yang berisi informasi (misalnya, *startup file* atau *ReadMe file*). Pada iterasi ini sudah dilakukan proses implementasi kode pada proses *login* sesuai gambar *sequence diagram login* pada iterasi sebelumnya. Berikut adalah gambar sampel implementasi kode pada beberapa *file* sehingga dapat menghasilkan sistem *login* pada sistem.



```

1 <?php
2
3 namespace App\Http\Controllers\Auth;
4
5 use App\Http\Controllers\Controller;
6 use Illuminate\Foundation\Auth\AuthenticatesUsers;
7 use Illuminate\Http\Request;
8
9 class LoginController extends Controller
10 {
11
12     use AuthenticatesUsers;
13
14     /**
15      * Where to redirect users after login.
16      *
17      * @var string
18      */
19     protected $redirectTo = '/home';
20
21     /**
22      * Create a new controller instance.
23      *
24      * @return void
25      */
26     public function __construct()
27     {
28         $this->middleware('guest')->except('logout');
29     }
30
31     protected function credentials(Request $request)
32     {
33         return array_merge($request->only($this->username(), 'password'), ['status' => 1]);
34     }
35 }
36
37

```

Gambar 4.15 : Sampel kode *LoginController*

Pada gambar 4.15 terlihat sampel kode *LoginController* yang digunakan untuk memproses kegiatan login pada sistem. Selain *Controller*, dibutuhkan implementasi kode untuk menampilkan halaman *login*. Berikut adalah sampel kode *view login* yang berfungsi untuk menampilkan halaman dan login dan menghiasnya.

```

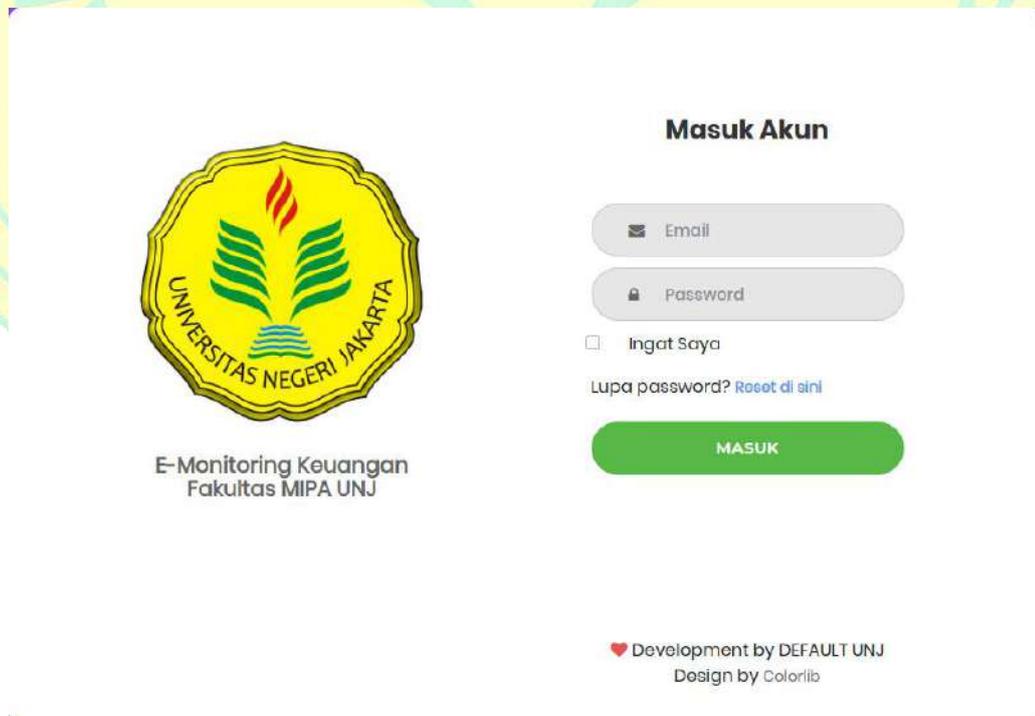
1 @extends('layouts.app')
2
3 @section('content')
4 <div class="limiter">
5 <div class="container-login100" >
6 <div class="wrap-login100" >
7 <div class="login100-pic js-tilt" data-tilt>
8 
9 <a href="{{ route('home') }}"><h5 style="text-align:center;"><strong>E-Monitoring Keuangan
10 Fakultas MIPA UNJ</strong></h5></a>
11 </div>
12 <form class="login100-form" method="POST" action="{{ route('login') }}">
13 {{ csrf_field() }}
14 <span class="login100-form-title">
15 Masuk Akun
16 </span>
17
18 <div class="wrap-input100">
19 <input class="input100 form-control"{{ $errors->has('email') ? ' is-invalid' : '' }}
20 placeholder="Email" id="email" type="email" name="email" value="{{ old('email') }}" required
21 autocomplete="email" autofocus>
22 <span class="focus-input100"></span>
23 <span class="symbol-input100">
24 <i class="fa fa-envelope" aria-hidden="true"></i>
25 </span>
26 </div>
27 @error('email')
28 <span class="small text-danger" role="alert">
29 <strong>{{ $message }}</strong>
30 </span>
31 @enderror
32
33 <div class="wrap-input100">
34 <input class="input100 form-control"{{ $errors->has('password') ? ' is-invalid' : '' }} type=
35 "password" placeholder="Password" id="password" type="password" name="password" required
36 autofocus>
37 <span class="focus-input100"></span>
38 <span class="symbol-input100">
39 <i class="fa fa-lock" aria-hidden="true"></i>
40 </span>
41 </div>
42 @error('password')

```

Gambar 4.16 : Sampel kode Halaman *Login*

2. Implementation Model

Tampilan desain halaman *login* dengan menggunakan *template* dari *Colorlib* (Aigars, 2018). Berikut ini gambar tampilan Halaman *Login*.



Gambar 4.17 : Tampilan Halaman *Login*

e. *Test*

Target pada proses *Test* ini diharapkan sudah mencapai 10% pada iterasi *elaboration* E2 ini. Pada proses *Test* ini akan dibahas artefak *Test Plan*.

1. *Test Plan*

Target pada artefak ini menentukan *planning* testing dan UAT pada iterasi agar dapat berjalan sesuai waktu yang ditentukan. *Testing* dapat dilakukan saat implementasi sistem sudah selesai, karena pada iterasi ini masih dalam proses implementasi dan belum selesai, maka akan dibuat *Test Plan* yang dapat dilihat pada tabel berikut.

Tabel 4.6 : *Test Plan*

No	<i>Test Plan</i>	Target
1	<i>Alpha Testing.</i>	Iterasi C3
2	<i>Beta Testing.</i>	Iterasi C3-T1
3	<i>User Acceptance Test (UAT).</i>	Iterasi T1-T2

C. Tahap *Construction*

Pada tahap *construction* dilakukan proses RUP yaitu berdasarkan iterasi *construction* yang terdapat 3 iterasi yaitu iterasi C1, C2 dan C3. Pada subbab ini akan dilakukan pembahasan tiap iterasi pada tahap *construction*.

1. Iterasi *construction* C1

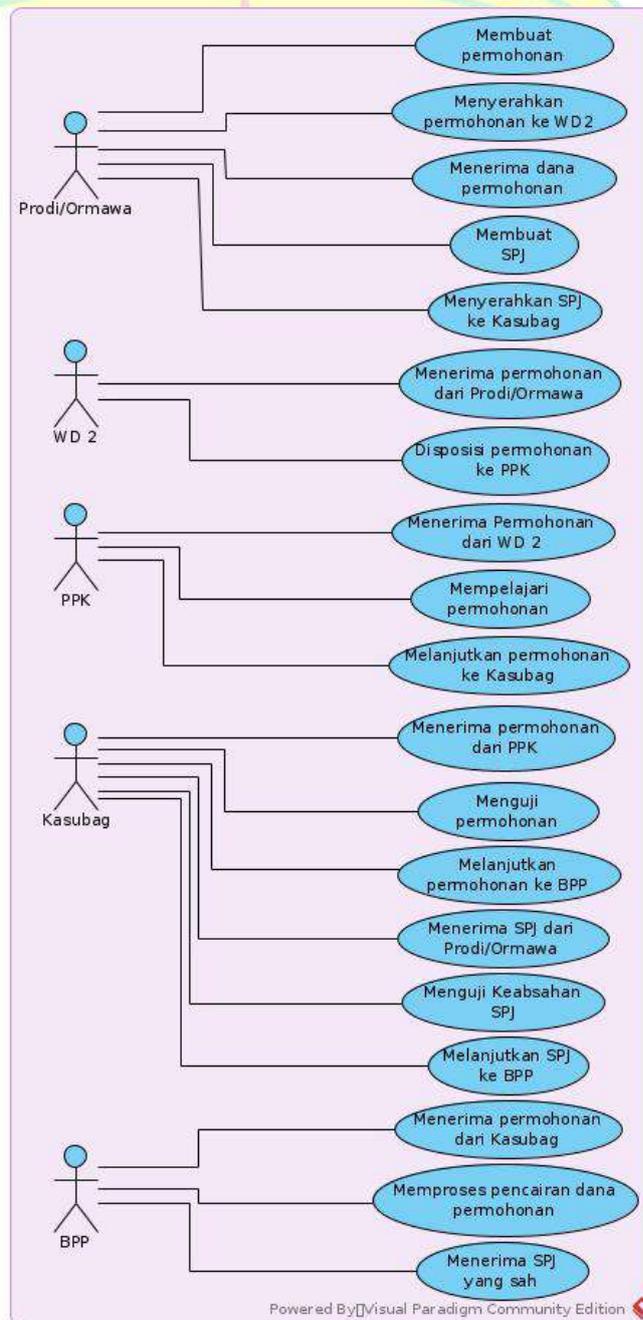
Iterasi selanjutnya pada tahap *construction* C1 berdasarkan *development plan* yaitu dibahasnya 10% *Business Modelling*, 10% *Requirements*, 20% *Analysis & Design*, 30% *Implementation*, 10% *Test* dan 10% *Deployment*.

a. *Business Modelling*

Target pada proses *Business Modelling* ini diharapkan sudah mencapai 10% pada iterasi *construction* C1 ini dan mencapai target 100% pada target *development plan* yang artinya sudah selesai semua proses pada *business modelling*. Pada proses *Business Modelling* ini akan dibahas artefak lanjutan *Business use case* dan lanjutan pembahasan artefak *Business use case model* berdasarkan iterasi sebelumnya.

1. *Business use case*

Target pada artefak ini adalah melanjutkan *business use case* iterasi sebelumnya dengan berdasarkan kegiatan SOP Keuangan. Pada iterasi ditambahkan beberapa *business use case* untuk memperjelas alur pengajuan permohonan dana dan diharapkan perubahan ini adalah versi final dari *business use case* yang bisa dilihat pada gambar berikut.

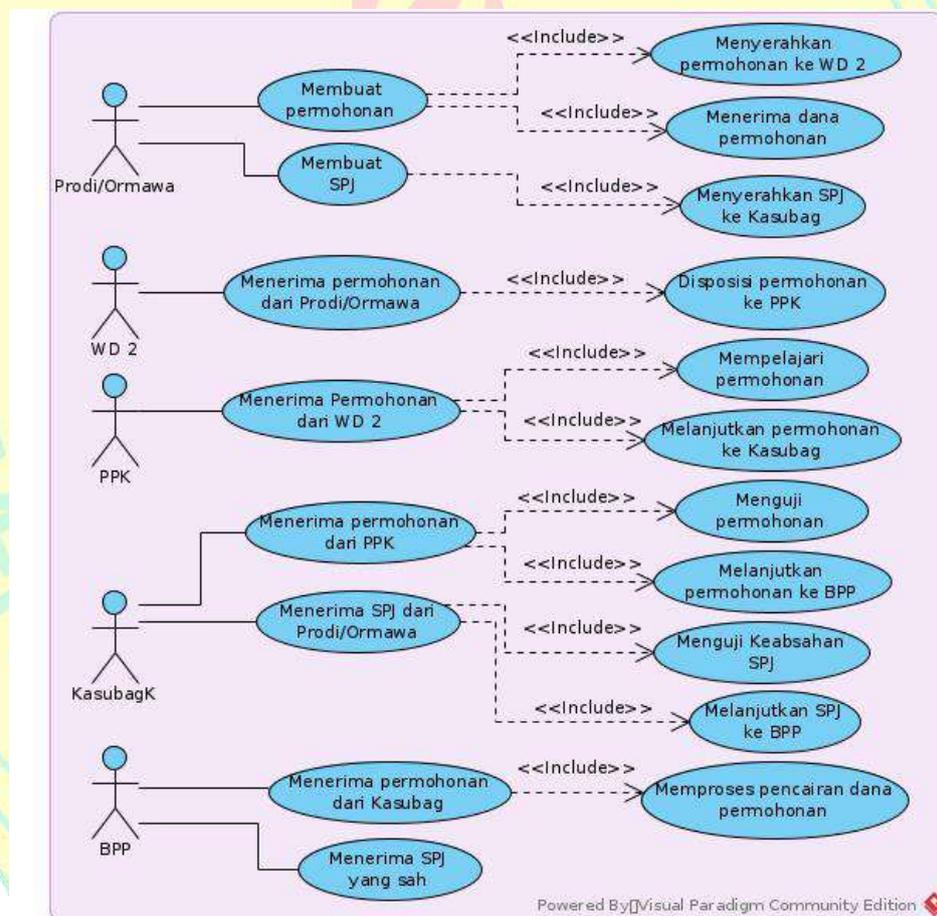


Gambar 4.18 : *Business use case* Final

Pada gambar 4.18 ditambahkan beberapa *business use case* untuk lebih memperjelas alur SOP keuangan, yaitu ditambahkan pada prodi/ormawa Menyerahkan SPJ ke Kasubag, pada PPK Melanjutkan permohonan ke Kasubag dan pada Kasubag Melanjutkan permohonan ke BPP dan Melanjutkan SPJ ke BPP.

2. *Business use case model*

Melanjutkan artefak ini dari terasi sebelumnya, *Business use case model* pada iterasi C1 ini disusun berdasarkan *business use case* gambar 4.18 yang dapat dilihat pada gambar berikut.



Gambar 4.19 : *Business use case model* Final

b. *Requirements*

Target pada proses *Requirements* ini diharapkan sudah mencapai 10% pada iterasi *construction* C1 ini dan mencapai target 100% pada target *development plan* yang artinya sudah selesai semua proses pada *Requirements*.

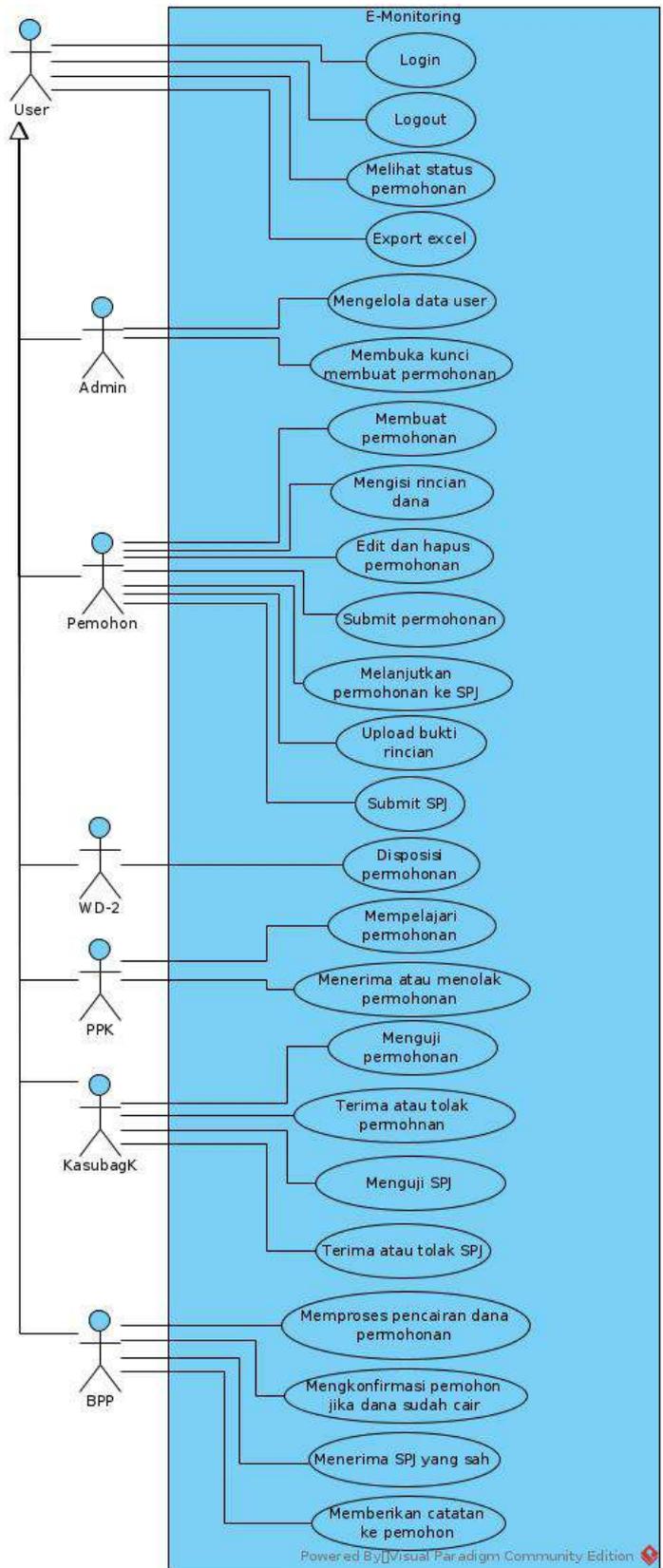
Pada proses *Requirements* ini akan dibahas artefak lanjutan *Use Case* dan lanjutan *Use Case Model* dari iterasi sebelumnya.

1. *Use Case*

Melanjutkan artefak *use case* pada iterasi sebelumnya dengan berdasarkan kegiatan SOP Keuangan. Pada iterasi C1 ditambahkan beberapa *use case* untuk memperjelas alur pengajuan permohonan dana dan diharapkan perubahan ini adalah versi final dari *use case* yang dapat dilihat pada gambar 4.20.

Penjelasan penambahan pada gambar 4.20 adalah penambahan pada aktor *User* yang mempunyai *use case* Melihat status permohonan dan *Export Excel*. Aktor pemohon menambahkan *use case* yaitu Edit dan hapus permohonan dan *Upload* bukti rincian. Aktor BPP ditambahkan *use case* Mengkonfirmasi pemohon jika dana sudah cair dan Memberikan catatan ke pemohon.

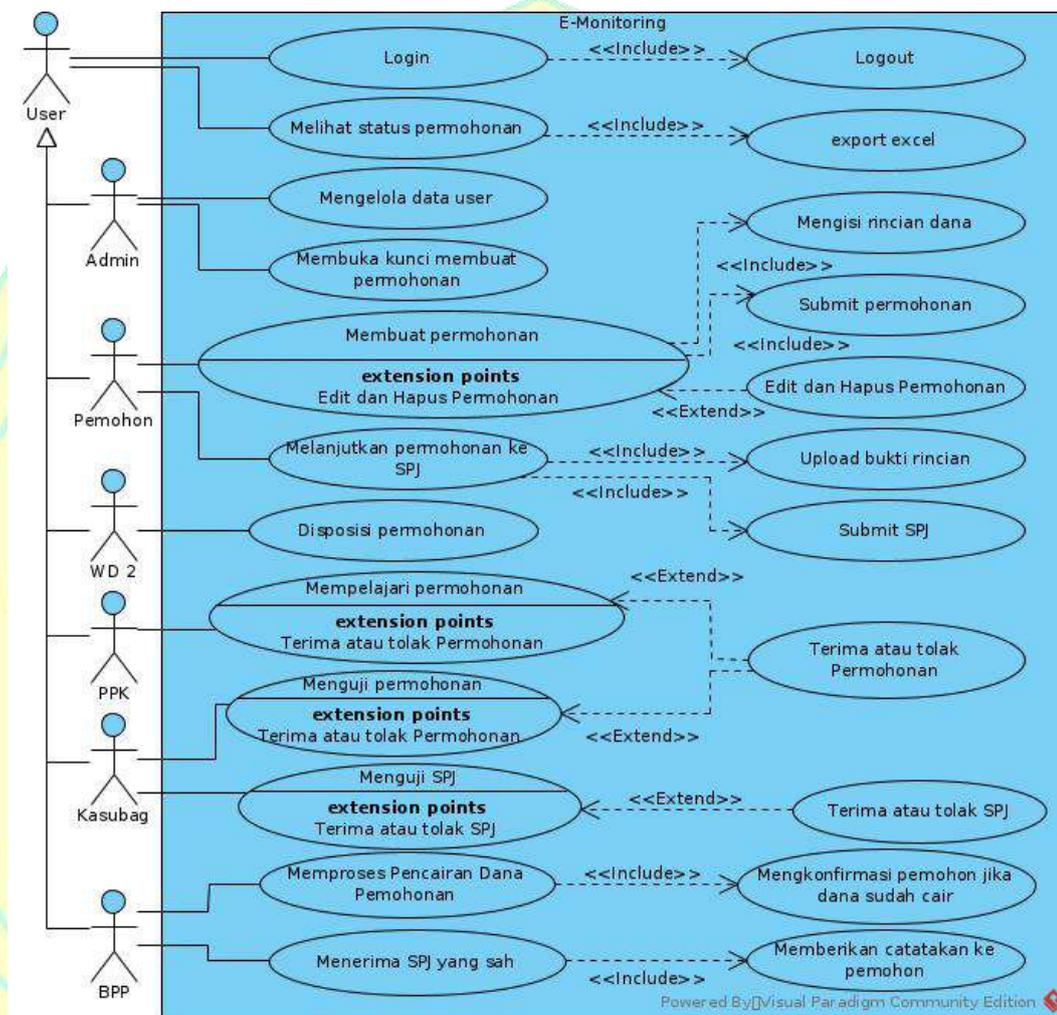




Gambar 4.20 : Use Case Final

2. Use Case Model

Melanjutkan dari iterasi sebelumnya, *Use Case Model* pada iterasi C1 ini disusun berdasarkan *use case* gambar 4.20 yang dapat dilihat pada gambar berikut.



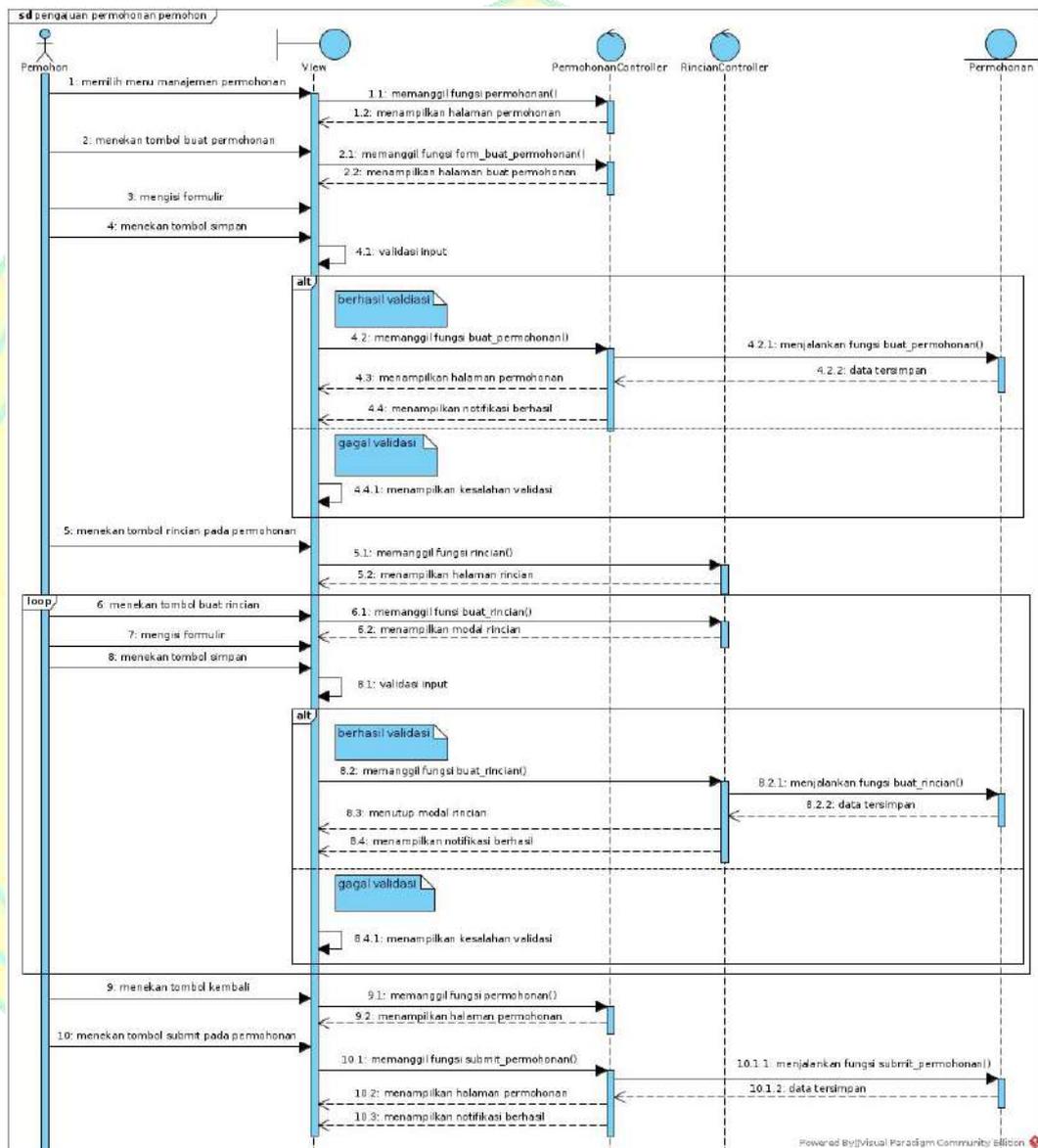
Gambar 4.21 : Use Case Model Final

c. Analysis & Design

Target pada proses *Analysis & Design* ini diharapkan sudah mencapai 20% pada iterasi *construction* C1 ini dan mencapai target 90% pada target *development plan* yang artinya sudah hampir selesai semua proses pada *Analysis & Design* ini dan akan menyesuaikan penambahan sesuai kebutuhan. Pada proses *Analysis & Design* ini akan dibahas artefak lanjutan *Design model* dari iterasi sebelumnya.

1. Design model

Melanjutkan dari iterasi sebelumnya. Pada iterasi ini akan dibahas *sequence diagram* permohonan dengan lengkap dalam pembuatan sistem. *Sequence diagram* tersebut dapat dilihat pada gambar berikut.



Gambar 4.22 : *Sequence diagram* pengajuan permohonan pemohon

Pada gambar 4.22 menjelaskan alur pengajuan permohonan dengan lengkap pada sistem, dimulai dari pemohon memilih menu manajemen permohonan, lalu sistem akan memanggil fungsi `permohonan()` yang akan diproses menjadi tampilan halaman permohonan.

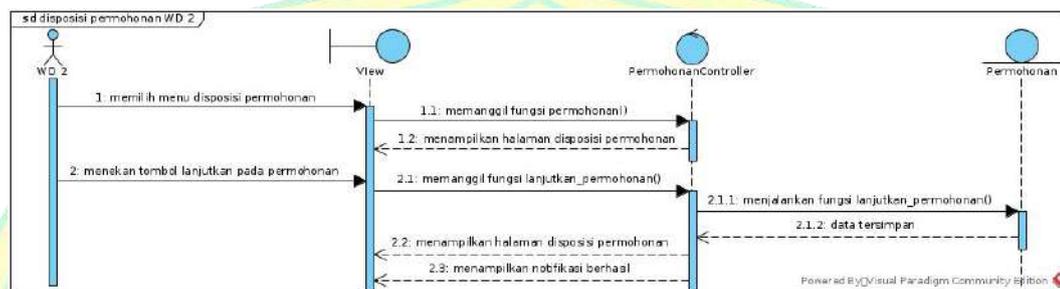
Selanjutnya pemohon menekan tombol buat permohonan, lalu sistem akan memanggil fungsi `form_buat_permohonan()` yang akan diproses menjadi tampilan halaman buat permohonan. Kemudian pemohon akan mengisi formulir tersebut, setelah selesai pemohon akan menekan tombol simpan, kemudian dilanjutkan dengan melakukan validasi input. Jika berhasil validasi, sistem akan memanggil fungsi `buat_permohonan()` dan menjalankan fungsi tersebut agar data input pemohon disimpan, setelah data tersimpan maka sistem akan menampilkan halaman permohonan dan menampilkan notifikasi berhasil, jika gagal validasi maka akan menampilkan kesalahan validasi.

Setelah pemohon selesai membuat draf permohonan, pemohon harus melengkapi rincian dana permohonan terlebih dahulu sebelum bisa melanjutkan permohonan tersebut, pemohon akan menekan tombol rincian pada permohonan, lalu sistem akan memanggil fungsi `rincian()` yang akan diproses menjadi tampilan halaman rincian.

Pada proses pengisi rincian pemohon akan terus melakukan pengulangan proses input rincian sampai semua rincian yang dibutuhkan pemohon tersebut sudah lengkap. Pemohon menekan tombol buat rincian, lalu sistem akan memanggil fungsi `buat_rincian()` yang akan diproses menjadi tampilan modal rincian. Kemudian pemohon akan mengisi formulir tersebut, setelah selesai pemohon akan menekan tombol simpan, kemudian dilanjutkan dengan melakukan validasi input. Jika berhasil validasi, sistem akan memanggil fungsi `buat_rincian()` dan menjalankan fungsi tersebut agar data input pemohon disimpan, setelah data tersimpan maka sistem akan menutup modal rincian dan menampilkan notifikasi berhasil, jika gagal validasi maka akan menampilkan kesalahan validasi.

Setelah pemohon selesai mengisi semua rincian yang dibutuhkan, maka pemohon akan menekan tombol kembali, lalu sistem akan memanggil fungsi `permohonan()` yang akan diproses menjadi tampilan halaman permohonan. Selanjutnya pemohon akan menekan tombol submit pada permohonan, lalu sistem akan memanggil fungsi

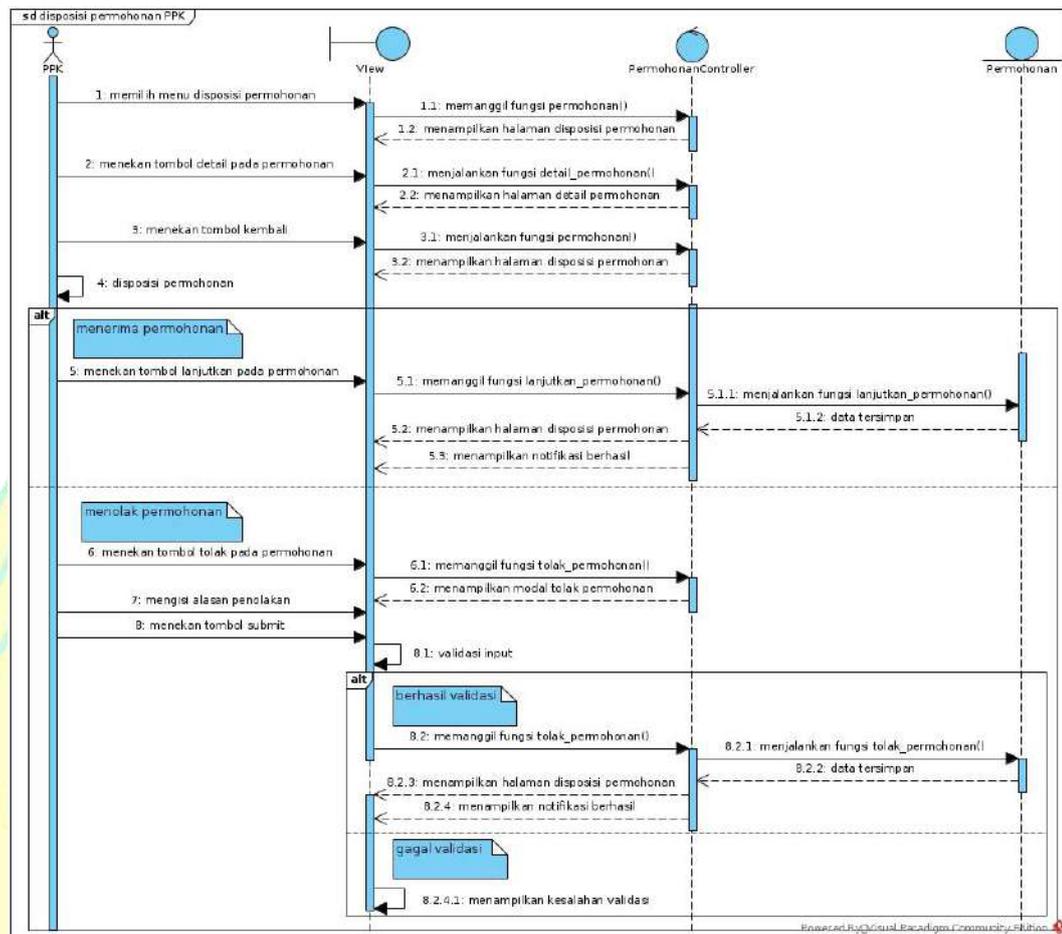
submit_permohonan() dan menjalankan fungsi tersebut agar data permohonan disimpan, setelah data tersimpan maka sistem akan menampilkan halaman permohonan dan menampilkan notifikasi berhasil. Setelah selesai proses mengajukan permohonan, maka akan dilanjutkan proses disposisi permohonan oleh WD2.



Gambar 4.23 : *Sequence diagram* disposisi permohonan WD 2

Pada gambar 4.23 menjelaskan alur disposisi permohonan WD 2 dengan lengkap pada sistem, dimulai dari WD 2 memilih menu disposisi permohonan, lalu sistem akan memanggil fungsi permohonan() yang akan diproses menjadi tampilan halaman disposisi permohonan. Selanjutnya WD 2 menekan tombol lanjutkan pada permohonan, lalu sistem akan memanggil fungsi lanjutkan permohonan dan menjalankan fungsi tersebut agar data permohonan disimpan, setelah data tersimpan maka sistem akan menampilkan halaman disposisi permohonan dan menampilkan notifikasi berhasil. Setelah selesai dsposisi permohonan WD 2, maka akan dilanjutkan proses disposisi permohonan oleh PPK.

Pada gambar 4.24 menjelaskan alur disposisi permohonan PPK dengan lengkap pada sistem, dimulai dari PPK memilih menu disposisi permohonan, lalu sistem akan memanggil fungsi permohonan() dan akan menampilkan halaman disposisi permohonan. Selanjutnya PPK menekan tombol detail pada permohonan untuk memeriksa permohonan, lalu sistem akan menjalankan fungsi detail_permohonan() yang akan diproses menjadi tampilan halaman detail permohonan. Setelah selesai, PPK menekan tombol kembali, lalu sistem akan menjalankan fungsi permohonan() dan menampilkan halaman disposisi permohonan.

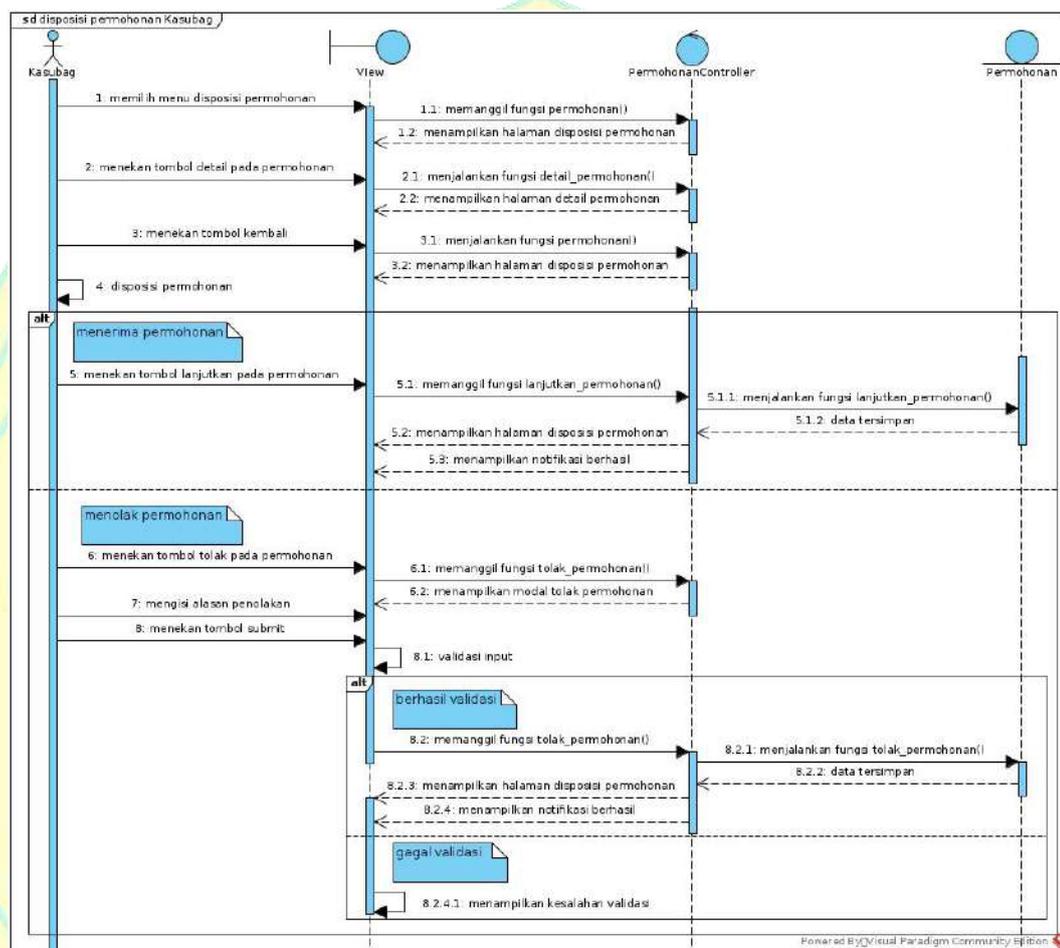


Gambar 4.24 : *Sequence diagram* disposisi permohonan PPK

Selanjutnya PPK akan melakukan disposisi permohonan. Jika menerima permohonan, PPK akan menekan tombol lanjutkan pada permohonan, lalu sistem akan memanggil fungsi `lanjutkan_permohonan()` dan menjalankan fungsi tersebut agar data permohonan disimpan, setelah data tersimpan maka sistem akan menampilkan halaman disposisi permohonan dan menampilkan notifikasi berhasil.

Jika menolak permohonan, PPK akan menekan tombol tolak pada permohonan, lalu sistem akan memanggil fungsi `tolak_permohonan()` dan sistem akan menampilkan modal tolak permohonan, PPK mengisi alasan penolakan dan jika sudah selesai akan menekan tombol submit kemudian dilanjutkan dengan validasi input. Jika berhasil validasi, sistem akan memanggil fungsi `tolak_permohonan()` dan menjalankan fungsi tersebut agar data input disimpan, setelah data tersimpan maka sistem akan

menampilkan halaman disposisi permohonan dan menampilkan notifikasi berhasil, jika gagal validasi maka akan menampilkan kesalahan validasi. Setelah selesai dsposisi permohonan PPK, maka akan dilanjutkan proses disposisi permohonan oleh Kasubag.



Gambar 4.25 : *Sequence diagram* disposisi permohonan Kasubag

Pada gambar 4.25 menjelaskan alur disposisi permohonan Kasubag dengan lengkap pada sistem, proses disposisi permohonan oleh kasubag hampir sama dengan proses disposisi permohonan PPK, hanya terdapat perbedaan Aktor yang melakukannya saja, jika sebelumnya PPK yang melakukan disposisi, maka pada proses ini dilakukan oleh Kasubag. Karena terdapat persamaan penjelasan, penulis memutuskan untuk tidak menulis ulang penjelasannya dan dilanjutkan dengan proses selanjutnya sekaligus proses disposisi permohonan terakhir yang dilakukan oleh BPP.



Gambar 4.26 : *Sequence diagram* disposisi permohonan BPP

Pada gambar 4.26 menjelaskan alur disposisi permohonan BPP dengan lengkap pada sistem, proses disposisi permohonan oleh BPP hampir sama dengan proses disposisi permohonan WD 2, hanya terdapat perbedaan Aktor yang melakukannya saja, jika sebelumnya WD 2 yang melakukan disposisi, maka pada proses ini dilakukan oleh BPP. Karena terdapat persamaan penjelasan, penulis memutuskan untuk tidak menulis ulang penjelasannya dan selesailah semua proses disposisi permohonan.

d. *Implementation*

Target pada proses *Implementation* ini diharapkan sudah mencapai 30% pada iterasi *construction* C1 ini dan mencapai target 50% pada target *development plan*. Pada proses *Implementation* ini akan dibahas artefak *Integration build plan*, *Component* dan *Implementation Model*.

1. *Integration build plan*

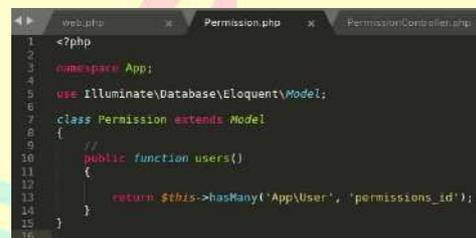
Melanjutkan dari iterasi yang pernah dibuat. Pada iterasi ini dilakukan penyesuaian *build plan* karena perkiraannya berubah dari jadwal sebelumnya. Penyesuaian tersebut dapat dilihat pada tabel berikut.

Tabel 4.7 : Penyesuaian *Integration build plan* pada iterasi C1

No	<i>Build plan</i>	Target
1	Pembuatan sistem <i>login</i> .	Iterasi E2
2	Pembuatan <i>role user</i> .	Iterasi C1
3	Pembuatan sistem pengajuan permohonan	Iterasi C1
4	Pembuatan sistem SPJ	Iterasi C2
5	Perbaikan <i>bugs</i>	Iterasi C3-T1

2. Component

Melanjutkan dari iterasi sebelumnya. Pada iterasi ini sudah dilakukan proses implementasi kode pada proses pembuatan *role user* dan dibuatnya sistem pengajuan permohonan yang dimulai dari pemohon membuat permohonan sampai disposisi permohonan terakhir sehingga permohonan tersebut disetujui. Berikut adalah gambar sampel implementasi kode pada beberapa *file* sehingga dapat menghasilkan sistem pembagian *role user*.



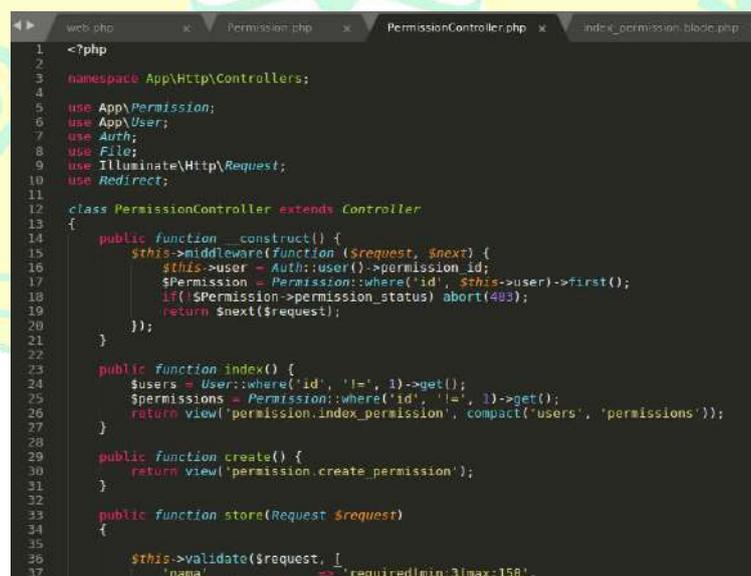
```

1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Permission extends Model
8 {
9     //
10    public function users()
11    {
12        return $this->hasMany('App\User', 'permissions_id');
13    }
14 }
15
16

```

Gambar 4.27 : Sampel kode *PermissionModel*

Pada gambar 4.27 terlihat sampel kode *PermissionModel* yang digunakan untuk mengambil data *user* dan *permission* pada sistem. Selain *Model*, dibutuhkan *PermissionController* untuk memproses kegiatan pada pembuatan *role user* dan implementasi kode *view* untuk menampilkan halaman *permission*. Berikut adalah sampel kode *PermissionController* dan sample kode halaman *permission*.



```

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Permission;
6 use App\User;
7 use Auth;
8 use File;
9 use Illuminate\Http\Request;
10 use Redirect;
11
12 class PermissionController extends Controller
13 {
14     public function __construct() {
15         $this->middleware(function ($request, $next) {
16             $this->user = Auth::user()->permission id;
17             $permission = Permission::where('id', $this->user)->first();
18             if (!$permission->permission_status) abort(403);
19             return $next($request);
20         });
21     }
22
23     public function index() {
24         $users = User::where('id', '!=', 1)->get();
25         $permissions = Permission::where('id', '!=', 1)->get();
26         return view('permission.index_permission', compact('users', 'permissions'));
27     }
28
29     public function create() {
30         return view('permission.create_permission');
31     }
32
33     public function store(Request $request)
34     {
35
36         $this->validate($request, [
37             'nama' => 'required|min:3|max:150',

```

Gambar 4.28 : Sampel kode *PermissionController*

```

1 | extends('layouts.backend')
2 |
3 | @section('content')
4 | <!-- Breadcrumbs -->
5 | <ol class="breadcrumb">
6 |   <li class="breadcrumb-item">
7 |     <a href="{{ route('home') }}">Dashboard</a>
8 |   </li>
9 |   <li class="breadcrumb-item active">
10 |     Permission
11 |   </li>
12 | </ol>
13 |
14 | @include('layouts.pesan')
15 |
16 | <a class="btn btn-primary" href="{{ route('permissionsCreate') }}">
17 |   <i class="fa fa-plus"></i>
18 |   Buat Permission
19 | </a>
20 | <br><br>
21 | <div class="card border border-info">
22 |   <div class="card-body">
23 |     <div class="card-header">
24 |       <i class="fas fa-table">&nbsp;&nbsp;Permission</i>
25 |     </div>
26 |
27 |     <div class="card-body">
28 |       <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
29 |         <thead>
30 |           <tr>
31 |             <th></th>
32 |             <th>No</th>
33 |             <th>Nama Permission</th>
34 |             <th>List Permission</th>
35 |             <th>Status</th>
36 |             <th>Options</th>
37 |           </tr>
38 |         </thead>
39 |         <tbody>
40 |           @if($permissions->count())
41 |             @foreach($permissions as $key => $permission)
42 |               <tr>

```

Gambar 4.29 : Sampel kode Halaman *Permission*

Setelah selesai membuat *permission* pada sistem, selanjutnya membuat manajemen *user*. Berikut adalah gambar sampel implementasi kode pada beberapa *file* sehingga dapat menghasilkan sistem pembagian manajemen *user*.

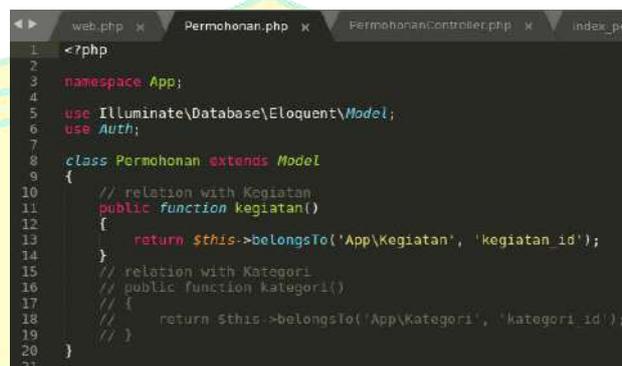
```

1 | <?php
2 |
3 | namespace App;
4 |
5 | use Illuminate\Notifications\Notifiable;
6 | use Illuminate\Contracts\Auth\MustVerifyEmail;
7 | use Illuminate\Foundation\Auth\User as Authenticatable;
8 |
9 | class User extends Authenticatable
10 | {
11 |     use Notifiable;
12 |
13 |     /**
14 |      * The attributes that are mass assignable.
15 |      *
16 |      * @var array
17 |      */
18 |     protected $fillable = [
19 |         'name',
20 |         'email',
21 |         'phone',
22 |         'tor',
23 |         'unit_id',
24 |         'permission_id',
25 |         'status',
26 |         'password',
27 |     ];
28 |
29 |     /**
30 |      * The attributes that should be hidden for arrays.
31 |      *
32 |      * @var array
33 |      */
34 |     protected $hidden = [
35 |         'password', 'remember_token',
36 |     ];

```

Gambar 4.30 : Sampel kode *UserModel*

Setelah selesai membuat manajemen *user* pada sistem, selanjutnya membuat beberapa kode pada alur permohonan. Berikut adalah gambar sampel implementasi kode pada beberapa *file* sehingga dapat menghasilkan sistem pengajuan permohonan.



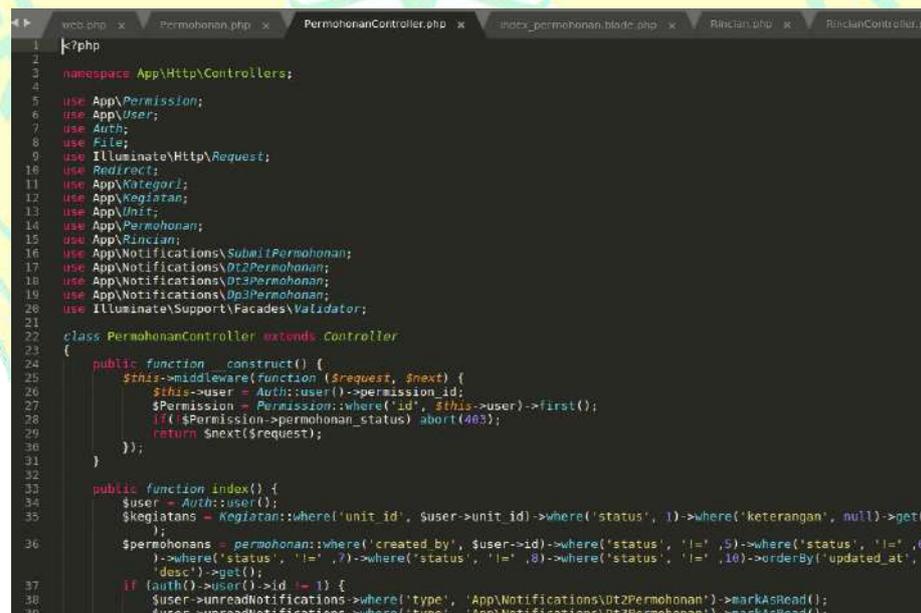
```

1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6 use Auth;
7
8 class Permohonan extends Model
9 {
10     // relation with Kegiatan
11     public function kegiatan()
12     {
13         return $this->belongsTo('App\Kegiatan', 'kegiatan_id');
14     }
15     // relation with Kategori
16     public function kategori()
17     {
18         return $this->belongsTo('App\Kategori', 'kategori_id');
19     }
20 }
21

```

Gambar 4.33 : Sampel kode *PermohonanModel*

Pada gambar 4.33 terlihat sampel kode *PermohonanModel* yang digunakan untuk mengambil data permohonan pada sistem. Selain *Model*, dibutuhkan *PermohonanController* untuk memproses kegiatan pada pembuatan permohonan dan implementasi kode *view* untuk menampilkan halaman manajemen permohonan. Berikut adalah sampel kode *PermohonanController* dan sample kode halaman manajemen permohonan.



```

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Permission;
6 use App\User;
7 use Auth;
8 use File;
9 use Illuminate\Http\Request;
10 use Redirect;
11 use App\Kategori;
12 use App\Kegiatan;
13 use App\Unit;
14 use App\Permohonan;
15 use App\Rincian;
16 use App\Notifications\SubmitPermohonan;
17 use App\Notifications\Dt2Permohonan;
18 use App\Notifications\Dt3Permohonan;
19 use App\Notifications\Dt3Permohonan;
20 use Illuminate\Support\Facades\Validator;
21
22 class PermohonanController extends Controller
23 {
24     public function __construct() {
25         $this->middleware(function ($request, $next) {
26             $this->user = Auth::user()->permission_id;
27             $permission = Permission::where('id', $this->user)->first();
28             if ($permission->permohonan_status) abort(403);
29             return $next($request);
30         });
31     }
32
33     public function index() {
34         $user = Auth::user();
35         $kegiatan = Kegiatan::where('unit_id', $user->unit_id)->where('status', 1)->where('keterangan', null)->get();
36         $permohonans = permohonan::where('created_by', $user->id)->where('status', '!=', 5)->where('status', '!=', 6)->where('status', '!=', 7)->where('status', '!=', 8)->where('status', '!=', 10)->orderBy('updated_at', 'desc')->get();
37         if (auth()->user()->id == 1) {
38             $user->unreadNotifications->where('type', 'App\Notifications\Dt2Permohonan')->markAsRead();
39             $user->unreadNotifications->where('type', 'App\Notifications\Dt3Permohonan')->markAsRead();
40         }
41     }
42
43     public function submit(Request $request) {
44         $validator = Validator::make($request->all(), [
45             'kegiatan_id' => 'required|integer|exists:kegiatan,id',
46             'kategori_id' => 'required|integer|exists:kategori,id',
47             'keterangan' => 'required|string|max:255',
48             'status' => 'required|integer|in:1,2,3,4,5,6,7,8,9,10',
49             'created_by' => 'required|integer|exists:users,id',
50             'unit_id' => 'required|integer|exists:unit,id',
51             'keterangan' => 'required|string|max:255',
52             'status' => 'required|integer|in:1,2,3,4,5,6,7,8,9,10',
53             'created_by' => 'required|integer|exists:users,id',
54             'unit_id' => 'required|integer|exists:unit,id'
55         ]);
56         if ($validator->fails()) {
57             return redirect()->back()->withErrors($validator);
58         }
59         $permohonan = permohonan::create([
60             'kegiatan_id' => $request->kegiatan_id,
61             'kategori_id' => $request->kategori_id,
62             'keterangan' => $request->keterangan,
63             'status' => $request->status,
64             'created_by' => $request->created_by,
65             'unit_id' => $request->unit_id
66         ]);
67         $permohonan->save();
68         $permohonan->notify(new SubmitPermohonan($permohonan));
69         return redirect()->back()->with('message', 'Permohonan berhasil diajukan');
70     }
71
72     public function detail($id) {
73         $permohonan = permohonan::find($id);
74         if (!$permohonan) {
75             return redirect()->back()->with('message', 'Permohonan tidak ditemukan');
76         }
77         $kegiatan = Kegiatan::find($permohonan->kegiatan_id);
78         $kategori = Kategori::find($permohonan->kategori_id);
79         $unit = Unit::find($permohonan->unit_id);
80         $user = User::find($permohonan->created_by);
81         $rincian = Rincian::where('permohonan_id', $id)->get();
82         return view('permohonan.detail', compact('permohonan', 'kegiatan', 'kategori', 'unit', 'user', 'rincian'));
83     }
84
85     public function edit($id) {
86         $permohonan = permohonan::find($id);
87         if (!$permohonan) {
88             return redirect()->back()->with('message', 'Permohonan tidak ditemukan');
89         }
90         $kegiatan = Kegiatan::find($permohonan->kegiatan_id);
91         $kategori = Kategori::find($permohonan->kategori_id);
92         $unit = Unit::find($permohonan->unit_id);
93         $user = User::find($permohonan->created_by);
94         $rincian = Rincian::where('permohonan_id', $id)->get();
95         return view('permohonan.edit', compact('permohonan', 'kegiatan', 'kategori', 'unit', 'user', 'rincian'));
96     }
97
98     public function update($id, Request $request) {
99         $permohonan = permohonan::find($id);
100        if (!$permohonan) {
101            return redirect()->back()->with('message', 'Permohonan tidak ditemukan');
102        }
103        $kegiatan = Kegiatan::find($permohonan->kegiatan_id);
104        $kategori = Kategori::find($permohonan->kategori_id);
105        $unit = Unit::find($permohonan->unit_id);
106        $user = User::find($permohonan->created_by);
107        $rincian = Rincian::where('permohonan_id', $id)->get();
108        $validator = Validator::make($request->all(), [
109            'kegiatan_id' => 'required|integer|exists:kegiatan,id',
110            'kategori_id' => 'required|integer|exists:kategori,id',
111            'keterangan' => 'required|string|max:255',
112            'status' => 'required|integer|in:1,2,3,4,5,6,7,8,9,10',
113            'created_by' => 'required|integer|exists:users,id',
114            'unit_id' => 'required|integer|exists:unit,id'
115        ]);
116        if ($validator->fails()) {
117            return redirect()->back()->withErrors($validator);
118        }
119        $permohonan->update([
120            'kegiatan_id' => $request->kegiatan_id,
121            'kategori_id' => $request->kategori_id,
122            'keterangan' => $request->keterangan,
123            'status' => $request->status,
124            'created_by' => $request->created_by,
125            'unit_id' => $request->unit_id
126        ]);
127        $permohonan->save();
128        $permohonan->notify(new SubmitPermohonan($permohonan));
129        return redirect()->back()->with('message', 'Permohonan berhasil diperbarui');
130    }
131
132     public function destroy($id) {
133         $permohonan = permohonan::find($id);
134         if (!$permohonan) {
135             return redirect()->back()->with('message', 'Permohonan tidak ditemukan');
136         }
137         $permohonan->delete();
138         return redirect()->back()->with('message', 'Permohonan berhasil dihapus');
139     }
140 }

```

Gambar 4.34 : Sampel kode *PermohonanController*

```

1 | @extends('layouts.backend')
2 |
3 | @section('content')
4 |     <!-- Breadcrumbs -->
5 |     <ol class="breadcrumb">
6 |         <li class="breadcrumb-item">
7 |             <a href="{{ route('home') }}">Dashboard</a>
8 |         </li>
9 |         <li class="breadcrumb-item active">
10 |             Manajemen Permohonan
11 |         </li>
12 |     </ol>
13 |
14 |     @include('layouts.pesan')
15 |     @if($kegiatan->count()==0)
16 |         <button class="btn btn-primary" data-toggle="modal"
17 |             data-target="#m-k" ui-toggle-class="bounce"
18 |             ui-target="#animate">
19 |             <i class="fa fa-plus"></i>
20 |             Buat Permohonan
21 |         </button>
22 |     @include('permohonan._ck')
23 |     @else
24 |         @if(!$user->tor)
25 |             <a class="btn btn-primary" href="{{ route('permohonanCreate') }}">
26 |                 <i class="fa fa-plus"></i>
27 |                 Buat Permohonan
28 |             </a>
29 |         @else
30 |             <button class="btn btn-primary" data-toggle="modal"
31 |                 data-target="#m-c" ui-toggle-class="bounce"
32 |                 ui-target="#animate">
33 |                 <i class="fa fa-plus"></i>
34 |                 Buat Permohonan
35 |             </button>
36 |         @include('permohonan._cp')
37 |     @endif
38 | <br><br>
39 | <div class="card border border-info">
40 |     <div class="card mb-3">
41 |         <div class="card-header">
42 |

```

Gambar 4.35 : Sampel kode Halaman Permohonan

```

1 | <?php
2 |
3 | namespace App;
4 |
5 | use Illuminate\Database\Eloquent\Model;
6 |
7 | class Rincian extends Model
8 | {
9 |
10 |     protected $dates = ['created_at', 'updated_at'];
11 |     // relation with Permohonan
12 |     public function permohonan()
13 |     {
14 |         return $this->belongsTo('App\Permohonan', 'permohonan_id');
15 |     }
16 | }
17 |

```

Gambar 4.36 : Sampel kode *RincianModel*

Pada gambar 4.36 terlihat sampel kode *RincianModel* yang digunakan untuk mengambil data rincian pada permohonan dalam sistem. Selain *Model*, dibutuhkan *RincianController* untuk memproses kegiatan pada pembuatan rincian pada permohonan dan implementasi kode *view*

untuk menampilkan halaman *single* permohonan dan modal membuat rincian. Berikut adalah sample kode *RincianController* dan beberapa sampel kode untuk membuat rincian.

```

1 |<?php
2 |
3 | namespace App\Http\Controllers;
4 |
5 | use App\Permission;
6 | use App\User;
7 | use Auth;
8 | use File;
9 | use Illuminate\Http\Request;
10 | use Redirect;
11 | use App\Kategori;
12 | use App\Kegiatan;
13 | use App\Unit;
14 | use App\Permohonan;
15 | use App\Rincian;
16 | use Illuminate\Support\Facades\Validator;
17 |
18 | class RincianController extends Controller
19 | {
20 |     public function __construct() {
21 |         $this->middleware(function ($request, $next) {
22 |             $this->user = Auth::user()->permission_id;
23 |             $Permission = Permission::where('id', $this->user)->first();
24 |             if(!$Permission->permohonan_status) abort(403);
25 |             return $next($request);
26 |         });
27 |     }
28 |
29 |     public function create(Request $request, $id)
30 |     {
31 |         $permohonans = Permohonan::findOrFail($id);
32 |         $validator = Validator::make($request->all(), [
33 |             'jenisbelanja' => 'required|min:3|max:150',
34 |             'volume' => 'required|numeric|min:1|max:999',
35 |             'biayasatuan' => 'required|numeric|min:1|max:100000000',
36 |             // 'biayasatuan' => 'required|numeric|max:100000000',
37 |             'satuan' => 'required|min:3|max:150',
38 |         ], [
39 |             'jenisbelanja.required'=>'jenis belanja harus diisi',
40 |             'jenisbelanja.min'=>'jenis belanja minimal 3 huruf',
41 |             'jenisbelanja.max'=>'jenis belanja maksimal 150 huruf',
42 |         ]);

```

Gambar 4.37 : Sampel kode *RincianController*

```

1 | @extends('layouts.backend')
2 |
3 | @section('content')
4 |     <!-- Breadcrumbs -->
5 |     <ol class="breadcrumb">
6 |         <li class="breadcrumb-item">
7 |             <a href="{{ route('home') }}">Dashboard</a>
8 |         </li>
9 |         <li class="breadcrumb-item">
10 |             <a href="{{ route('permohonan') }}">Permohonan</a>
11 |         </li>
12 |         <li class="breadcrumb-item active">
13 |             {{ $permohonan->nama }}
14 |         </li>
15 |     </ol>
16 |
17 | @include('layouts.pesan')
18 |
19 | <div class="col-md-12 border border-dark" style="border-radius: 40px">
20 |     <div class="text-center" style="margin-top: 30px">{{ $permohonan->nama }}</div>
21 |     <div class="nav-tabs-navigation d-flex justify-content-center">
22 |         <div class="nav-tabs-wrapper">
23 |             <ol class="nav nav-tabs nav-pills nav-fill" role="tablist" id="tabMenu">
24 |                 <li class="nav-item nav-justified">
25 |                     <a class="flex-sm-fill text-sm-center nav-link active" data-toggle="tab" href="#home"
26 |                         role="tab"><i class="fas fa-home"> Detail</i></a>
27 |                 </li>
28 |                 <li class="nav-item nav-justified">
29 |                     <a class="flex-sm-fill text-sm-center nav-link" data-toggle="tab" href="#profile"
30 |                         role="tab"><i class="fas fa-info"> Rincian Biaya</i></a>
31 |                 </li>
32 |                 <li class="nav-item nav-justified">
33 |                     <a class="flex-sm-fill text-sm-center nav-link" data-toggle="tab" href="#report"
34 |                         role="tab"><i class="fa fa-file-download"> File TOR</i></a>
35 |                 </li>
36 |                 <li if ($permohonan->status == 0 || $permohonan->status==0)
37 |                     @if ($permohonan->created_by == Auth::user()->id)
38 |                         <li class="nav-item nav-justified" if (empty($tabMenu) || $tabMenu == 'messages') <li class="nav-item nav-justified" if (empty($tabMenu) || $tabMenu == 'messages')

```

Gambar 4.38 : Sampel kode *SinglePermohonan*

```

1 | <div id="r-c" class="modal fade" data-backdrop="true">
2 | <div class="modal-dialog modal-md id="animate">
3 | <div class="modal-content">
4 | <div class="modal-header">
5 | <h3 class="modal-title">Buat Rincian</h3>
6 | </div>
7 | <form method="post" role="form" action="{{ route('rincianCreate', ['id' => $permohonan->id]) }}" enctype
8 | = "multipart/form-data" id="myForm">
9 | {{ csrf_field() }}
10 | <div class="modal-body text-left p-lg">
11 | <div class="form-group row">
12 | <label class="col-sm-5 col-form-label text-md-left">{{ __('Nama Rincian Biaya') }}</label>
13 | <div class="col-md-7">
14 |
15 | <input id="jenisbelanja" type="text" placeholder="Masukkan Rincian Biaya" class="
16 | form-control {{ $errors->has('jenisbelanja') ? 'is-invalid' : '' }}" name="jenisbelanja"
17 | value="{{ old('jenisbelanja') }}" required autofocus>
18 |
19 | @if ($errors->has('jenisbelanja'))
20 | <span class="invalid-feedback" role="alert">
21 | <strong>{{ $errors->first('jenisbelanja') }}</strong>
22 | </span>
23 | @endif
24 | </div>
25 | </div>
26 | <div class="form-group row">
27 | <label class="col-sm-5 col-form-label text-md-left">{{ __('Biaya Satuan (Rp)') }}</label>
28 | <div class="col-md-7">
29 |
30 | <input id="biayasatuan" type="text" placeholder="Masukkan Biaya Satuan" class="
31 | form-control {{ $errors->has('biayasatuan') ? 'is-invalid' : '' }}" name="biayasatuan"
32 | value="{{ old('biayasatuan') }}" required autofocus>
33 |
34 | @if ($errors->has('biayasatuan'))
35 | <span class="invalid-feedback" role="alert">
36 | <strong>{{ $errors->first('biayasatuan') }}</strong>
37 | </span>
38 | @endif
39 | </div>
40 | </div>

```

Gambar 4.39 : Sampel kode Modal Buat Rincian

Setelah selesai membuat alur pembuatan permohonan dan rincian pada sistem, selanjutnya membuat beberapa kode pada alur disposisi permohonan. Berikut adalah gambar sampel implementasi kode pada beberapa *file* sehingga dapat menghasilkan alur sistem disposisi permohonan.

```

1 | <?php
2 |
3 | namespace App\Http\Controllers;
4 |
5 | use App\Permission;
6 | use App\User;
7 | use Auth;
8 | use Files;
9 | use Illuminate\Http\Request;
10 | use Redirect;
11 | use App\Kategori;
12 | use App\Kejelasan;
13 | use App\Unit;
14 | use App\Permohonan;
15 | use App\Rincian;
16 | use App\Notifications\SubmitPermohonan;
17 | use App\Notifications\Dis1Permohonan;
18 | use App\Notifications\Dis2Permohonan;
19 | use App\Notifications\Dis3Permohonan;
20 | use App\Notifications\Dis4Permohonan;
21 | use App\Notifications\Dis5Permohonan;
22 | use App\Notifications\Dis6Permohonan;
23 | use App\Notifications\Dis7Permohonan;
24 | use App\Notifications\Dis8Permohonan;
25 | use App\Notifications\Dis9Permohonan;
26 | use App\Notifications\Dis10Permohonan;
27 | use App\Notifications\Dis11Permohonan;
28 | use App\Notifications\Dis12Permohonan;
29 | use Illuminate\Support\Facades\Validator;
30 |
31 | class DisposisiController extends Controller
32 | {
33 |     public function disp1()
34 |     {
35 |         if (Auth::user()->permissionsGroup->disp1p_status) {
36 |             $permohonans = permohonan::where('status', 1)->orderBy('updated_at', 'desc')->get();
37 |             $users = User::where('id', '!=', 1)->get();
38 |             if (Auth::user()->user()->permissionsGroup->disp1p_status == 1) {
39 |                 foreach ($users as $user) {
40 |                     $user->unreadNotifications->where('type', 'App\Notifications\SubmitPermohonan')->markAsRead();
41 |                 }
42 |             }
43 |             return view('disposisi.index.disposisi', compact('permohonans'));
44 |         }
45 |     }
46 | }

```

Gambar 4.40 : Sampel kode *DisposisiController*

Pada gambar 4.40 terlihat sampel kode *DisposisiController* yang digunakan untuk memproses disposisi data permohonan dalam sistem. Selain *Controller*, dibutuhkan *view index* dan *single* pada disposisi untuk untuk menampilkan halaman *index* dan *single* disposisi. Berikut adalah sample kode *view index* dan *single* disposisi.

```

1 | @extends('layouts.backend')
2 |
3 | @section('content')
4 | <!-- Breadcrumbs -->
5 | <ol class="breadcrumb">
6 |   <li class="breadcrumb-item">
7 |     <a href="{{ route('home') }}">Dashboard</a>
8 |   </li>
9 |   <li class="breadcrumb-item active">
10 |     Disposisi Permohonan
11 |   </li>
12 | </ol>
13 |
14 | @include('layouts.pesan')
15 | <div class="card border border-info">
16 |   <div class="card ab-3">
17 |     <div class="card-header">
18 |       <i class="fas fa-table">&nbsp;Disposisi Permohonan</i>
19 |     </div>
20 |
21 |     <div class="card-body">
22 |       <div class="table-responsive">
23 |         <table class="table table-bordered" id="dataTable" width="100%" cellpadding="0">
24 |           <thead>
25 |             <tr>
26 |               <th>No</th>
27 |               <th>Nama Permohonan</th>
28 |               <th>Dibuat Oleh</th>
29 |               <th>Tanggal Dibuat</th>
30 |               <th>Tanggal Diadit</th>
31 |               <th>Status</th>
32 |               <th>Detail</th>
33 |               <th>Options</th>
34 |               <th>Keterangan</th>
35 |             </tr>
36 |           </thead>
37 |           <tbody>
38 |             @if($permohonans->count())
39 |               @foreach($permohonans as $key => $permohonan)
40 |                 <tr>
41 |                   <td>{{ $key }}</td>
42 |                   <td><small>{{ $permohonan->nama }}</small></td>

```

Gambar 4.41 : Sampel kode Halaman Disposisi Permohonan

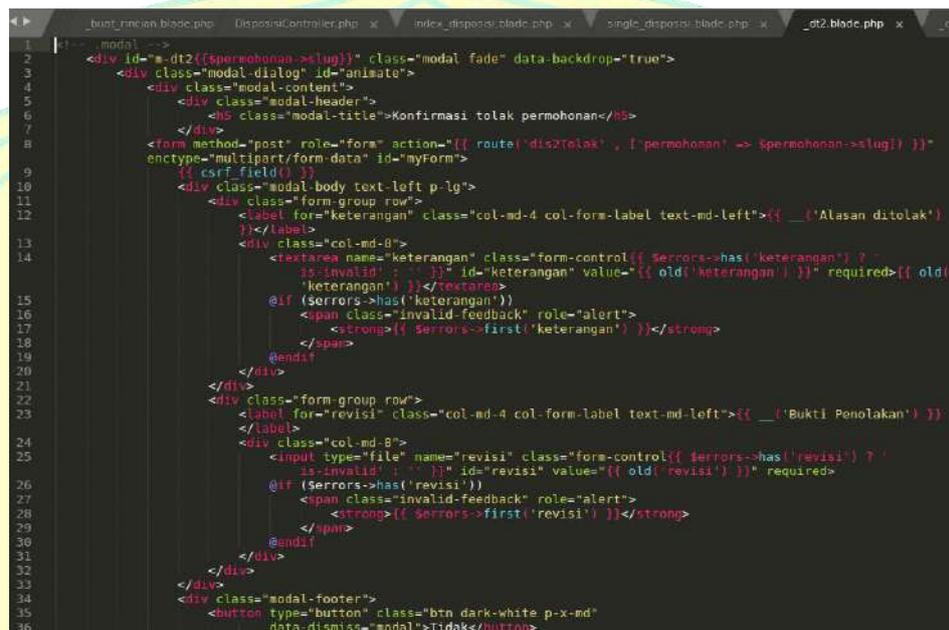
```

1 | @extends('layouts.backend')
2 |
3 | @section('content')
4 | <!-- Breadcrumbs -->
5 | <ol class="breadcrumb">
6 |   <li class="breadcrumb-item">
7 |     <a href="{{ route('home') }}">Dashboard</a>
8 |   </li>
9 |   <li class="breadcrumb-item">
10 |     @if($permohonan->status==1)
11 |       <a href="{{ route('dis1') }}">Disposisi Permohonan</a>
12 |     @elseif($permohonan->status==2)
13 |       <a href="{{ route('dis2') }}">Disposisi Permohonan</a>
14 |     @elseif($permohonan->status==3)
15 |       <a href="{{ route('dis3') }}">Disposisi Permohonan</a>
16 |     @elseif($permohonan->status==4)
17 |       <a href="{{ route('dis4') }}">Disposisi Permohonan</a>
18 |     @endif
19 |   </li>
20 |   <li class="breadcrumb-item active">
21 |     {{ $permohonan->nama }}
22 |   </li>
23 | </ol>
24 |
25 | @include('layouts.pesan')
26 |
27 | <div class="col-md-12 border border-dark" style="border-radius: 40px">
28 |
29 |   <h2 class="text-center" style="margin-top: 30px">{{ $permohonan->nama }}</h2>
30 |   <div class="nav-tabs navigation d-flex justify-content-center">
31 |     <div class="nav-tabs-wrapper">
32 |       <ul class="nav nav-tabs nav-pills nav-fill" role="tablist" id="tabMenu">
33 |         <li class="nav-item nav-justified">
34 |           <a class="flex-sm-fill text-sm-center nav-link active" data-toggle="tab" href="#home"
35 |             role="tab"><i class="fas fa-home"> Detail</i></a>
36 |         </li>
37 |         <li class="nav-item nav-justified">
38 |           <a class="flex-sm-fill text-sm-center nav-link" data-toggle="tab" href="#profile"
39 |             role="tab"><i class="fas fa-info"> Rincian Blaya</i></a>
40 |         </li>
41 |         <li class="nav-item nav-justified">

```

Gambar 4.42 : Sampel kode *Single Disposisi*

Pada saat implementasi kode disposisi permohonan, diperlukan tambahan beberapa kode Modal tolak pada beberapa Aktor yaitu Modal Tolak Permohonan pada PPK dan Modal Tolak Permohonan pada Kasubag. Berikut adalah sampel kode Modal Tolak Permohonan pada PPK dan Modal Tolak Permohonan pada Kasubag.

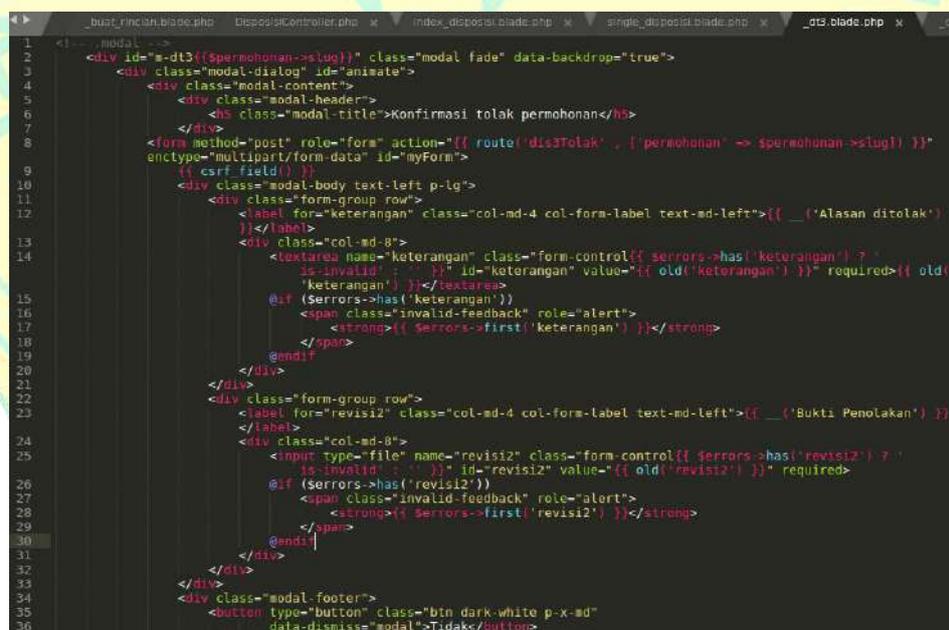


```

1 | <!-- modal -->
2 | <div id="m-dt2({{ $permohonan->slug }})" class="modal fade" data-backdrop="true">
3 | <div class="modal-dialog" id="animate">
4 | <div class="modal-content">
5 | <div class="modal-header">
6 | <h3 class="modal-title">Konfirmasi tolak permohonan</h3>
7 | </div>
8 | <form method="post" role="form" action="{ route('dis2Tolak' , [ $permohonan->slug ] ) }"
9 | enctype="multipart/form-data" id="myForm">
10 | {{ csrf_field() }}
11 | <div class="modal-body text-left p-lg">
12 | <div class="form-group row">
13 | <label for="keterangan" class="col-md-4 col-form-label text-md-left">{{ __( 'Alasan ditolak' ) }}</label>
14 | <div class="col-md-8">
15 | <textarea name="keterangan" class="form-control{{ $errors->has('keterangan') ? ' is-invalid' : '' }}" id="keterangan" value="{{ old('keterangan') }}" required>{{ old('keterangan') }}</textarea>
16 | @if ( $errors->has('keterangan') )
17 | <span class="invalid-feedback" role="alert">
18 | <strong>{{ $errors->first('keterangan') }}</strong>
19 | </span>
20 | @endif
21 | </div>
22 | </div>
23 | <div class="form-group row">
24 | <label for="revisi" class="col-md-4 col-form-label text-md-left">{{ __( 'Bukti Penolakan' ) }}</label>
25 | <div class="col-md-8">
26 | <input type="file" name="revisi" class="form-control{{ $errors->has('revisi') ? ' is-invalid' : '' }}" id="revisi" value="{{ old('revisi') }}" required>
27 | @if ( $errors->has('revisi') )
28 | <span class="invalid-feedback" role="alert">
29 | <strong>{{ $errors->first('revisi') }}</strong>
30 | </span>
31 | @endif
32 | </div>
33 | </div>
34 | <div class="modal-footer">
35 | <button type="button" class="btn dark-white p-x-md"
36 | data-dismiss="modal">Tidak</button>

```

Gambar 4.43 : Sampel kode Modal Tolak Permohonan pada PPK



```

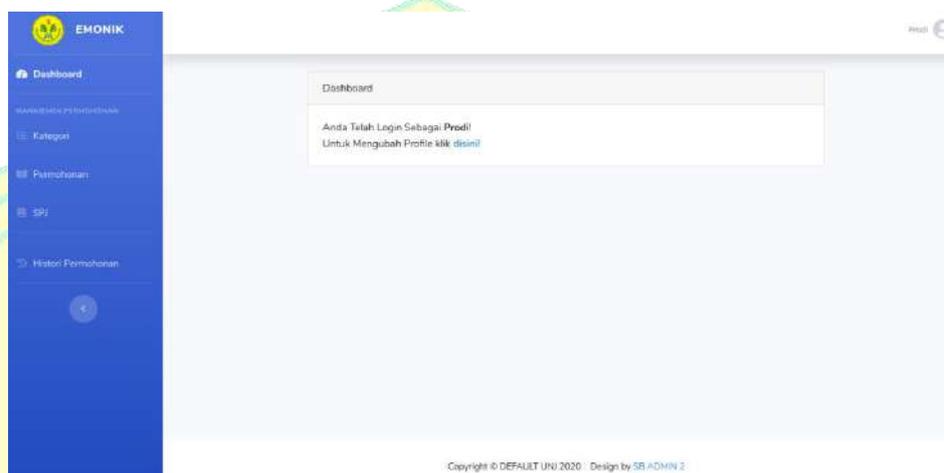
1 | <!-- modal -->
2 | <div id="m-dt3({{ $permohonan->slug }})" class="modal fade" data-backdrop="true">
3 | <div class="modal-dialog" id="animate">
4 | <div class="modal-content">
5 | <div class="modal-header">
6 | <h3 class="modal-title">Konfirmasi tolak permohonan</h3>
7 | </div>
8 | <form method="post" role="form" action="{ route('dis3Tolak' , [ $permohonan->slug ] ) }"
9 | enctype="multipart/form-data" id="myForm">
10 | {{ csrf_field() }}
11 | <div class="modal-body text-left p-lg">
12 | <div class="form-group row">
13 | <label for="keterangan" class="col-md-4 col-form-label text-md-left">{{ __( 'Alasan ditolak' ) }}</label>
14 | <div class="col-md-8">
15 | <textarea name="keterangan" class="form-control{{ $errors->has('keterangan') ? ' is-invalid' : '' }}" id="keterangan" value="{{ old('keterangan') }}" required>{{ old('keterangan') }}</textarea>
16 | @if ( $errors->has('keterangan') )
17 | <span class="invalid-feedback" role="alert">
18 | <strong>{{ $errors->first('keterangan') }}</strong>
19 | </span>
20 | @endif
21 | </div>
22 | </div>
23 | <div class="form-group row">
24 | <label for="revisi2" class="col-md-4 col-form-label text-md-left">{{ __( 'Bukti Penolakan' ) }}</label>
25 | <div class="col-md-8">
26 | <input type="file" name="revisi2" class="form-control{{ $errors->has('revisi2') ? ' is-invalid' : '' }}" id="revisi2" value="{{ old('revisi2') }}" required>
27 | @if ( $errors->has('revisi2') )
28 | <span class="invalid-feedback" role="alert">
29 | <strong>{{ $errors->first('revisi2') }}</strong>
30 | </span>
31 | @endif
32 | </div>
33 | </div>
34 | <div class="modal-footer">
35 | <button type="button" class="btn dark-white p-x-md"
36 | data-dismiss="modal">Tidak</button>

```

Gambar 4.44 : Sampel kode Modal Tolak Permohonan pada Kasubag

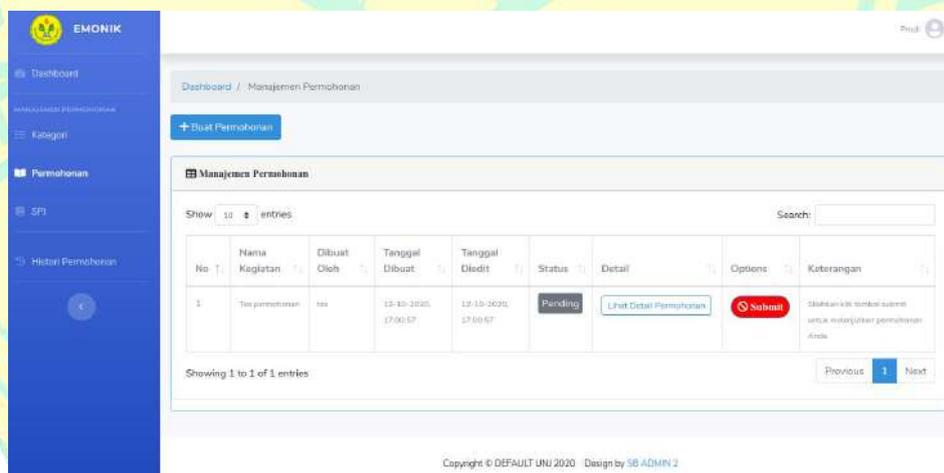
3. Implementation Model

Tampilan desain halaman *dashboard* dengan menggunakan *template* dari *SB Admin 2* (Miller, 2014). Berikut ini gambar tampilan Halaman *Dashboard* pada pemohon.



Gambar 4.45 : Tampilan Halaman *Dashboard* Pemohon

Pertama akan dijelaskan proses pengajuan permohonan oleh pemohon dimulai dari pemohon membuat permohonan dengan memilih menu Permohonan dalam kelompok menu Manajemen Permohonan.



Gambar 4.46 : Tampilan Halaman Menu Permohonan

Selanjutnya pada menu Permohonan, pemohon menekan tombol buat permohonan yang akan menampilkan formulir untuk membuat permohonan, lalu pemohon akan mengisi formulir tersebut, jika sudah selesai maka pemohon bisa menekan tombol simpan untuk menyimpan.

Dashboard / Permohonan / Buat Permohonan

Buat Permohonan

Nama Kegiatan:

Kategori:

Nama Permohonan:

Nomor Induk:

Latar Belakang Kegiatan:

Tujuan Kegiatan:

Tempat Kegiatan:

Tanggal Kegiatan:

Gambar 4.47 : Tampilan Halaman Formulir Buat Permohonan

Tempat Kegiatan:

Tanggal Kegiatan:

Peserta Kegiatan:

Strategi Pencapaian Keluaran:

Susunan Panitia:

File TOR:

Copyright © DEFAULT UNJ 2020 Design by SB ADMIN 2

Gambar 4.48 : Tampilan Tombol Simpan Formulir Buat Permohonan
Setelah pemohon menekan tombol simpan, maka akan muncul notifikasi berhasil membuat permohonan.

Dashboard / Manajemen Permohonan

Permohonan berhasil dibuat!

[+ Buat Permohonan](#)

Manajemen Permohonan

Show: 10 entries

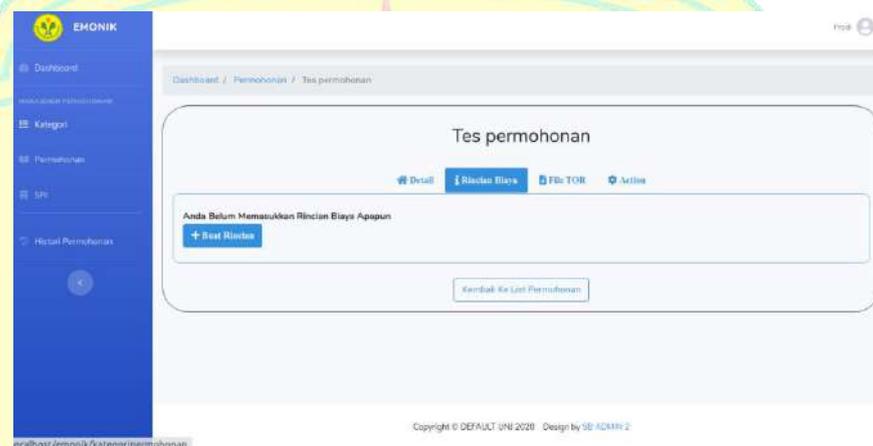
No	Nama Kegiatan	Dibuat Oleh	Tanggal Dibuat	Tanggal Disetujui	Status	Detail	Options	Keterangan
1	tes permohonan	tes	12-10-2020 17:00:57	13-10-2020 17:50:57	Pending	Link Detail Permohonan	Refresh	Silahkan klik tombol submit untuk mengaktifkan permohonan Anda

Showing 1 to 1 of 1 entries

Previous 1 Next

Gambar 4.49 : Tampilan Notifikasi Permohonan berhasil dibuat

Setelah pemohon selesai membuat draf permohonan, pemohon harus melengkapi rincian pada draf permohonan tersebut agar bisa melakukan *submit* permohonan. Pemohon bisa melengkapi rincian permohonan dengan menekan tombol lihat detail permohonan pada permohonan yang sudah dibuat yang akan menampilkan halaman detail dari permohonan tersebut dan menekan menu rincian biaya pada halaman tersebut.

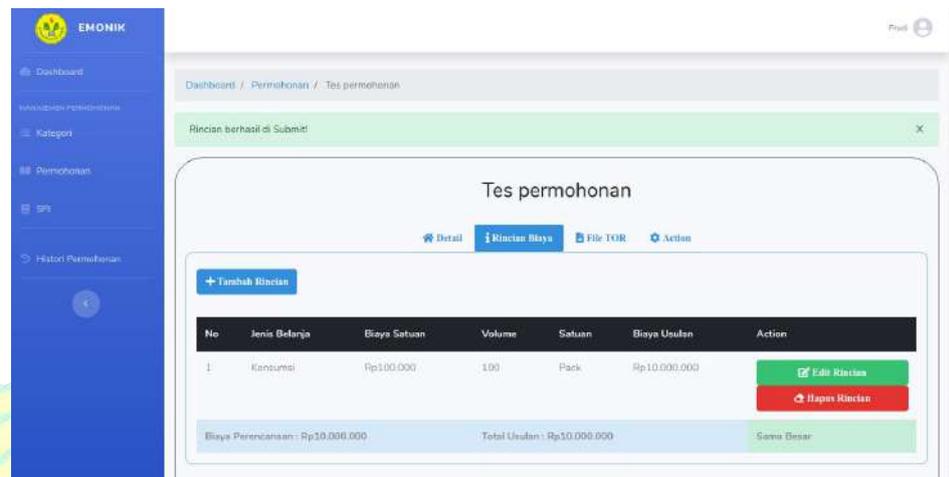


Gambar 4.50 : Tampilan Halaman Menu Rincian Biaya

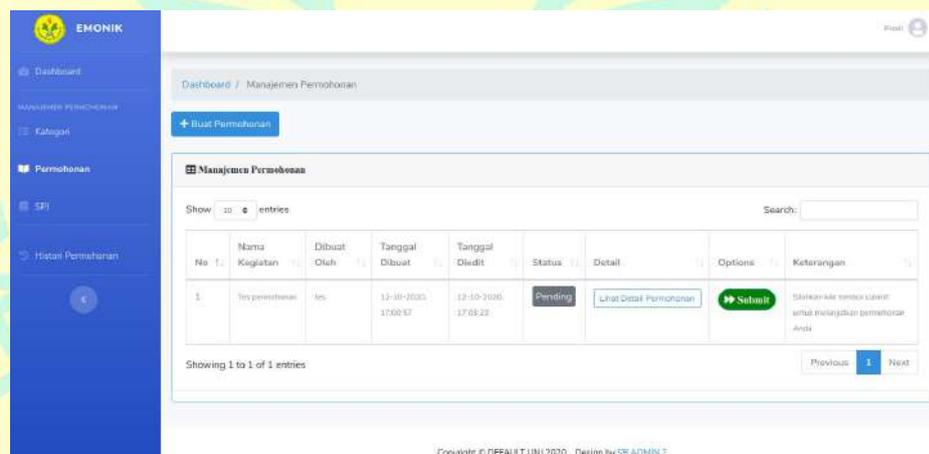
Pada menu Rincian Biaya, pemohon bisa membuat rincian biaya dengan menekan tombol Buat Rincian yang akan menampilkan Modal Buat Rincian.

Gambar 4.51 : Tampilan Modal Buat Rincian

Setelah pemohon selesai mengisi formulir Buat Rincian, pemohon bisa menekan tombol simpan pada formulir tersebut, setelah pemohon menekan tombol simpan maka akan muncul notifikasi berhasil.



Gambar 4.52 : Tampilan Halaman Rincian dengan Notifikasi Berhasil
Setelah pemohon selesai mengisi Rincian Biaya, pemohon dapat menekan tombol kembali pada halaman itu untuk kembali ke halaman manajemen permohonan untuk melakukan *submit* permohonan.

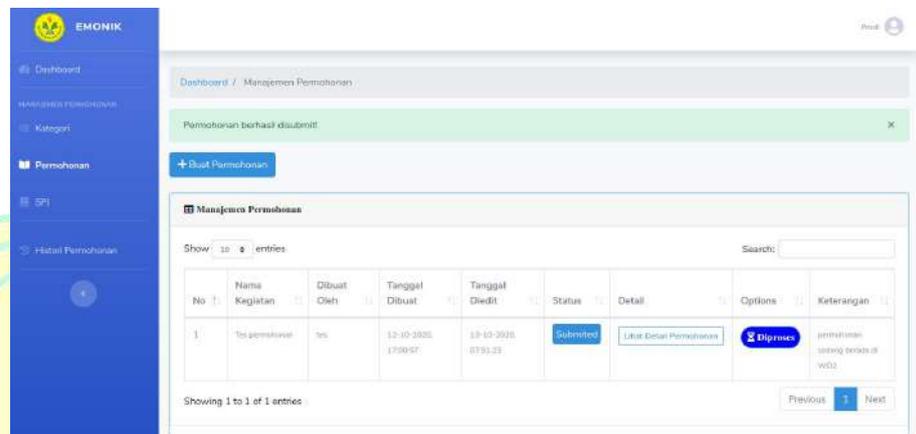


Gambar 4.53 : Tampilan Menu *Submit* pada Permohonan
Kemudian pemohon menekan tombol *submit* yang akan menampilkan konfirmasi *submit* permohonan, pemohon dapat menekan tombol submit pada konfirmasi tersebut.



Gambar 4.54 : Tampilan Konfirmasi *Submit* Permohonan

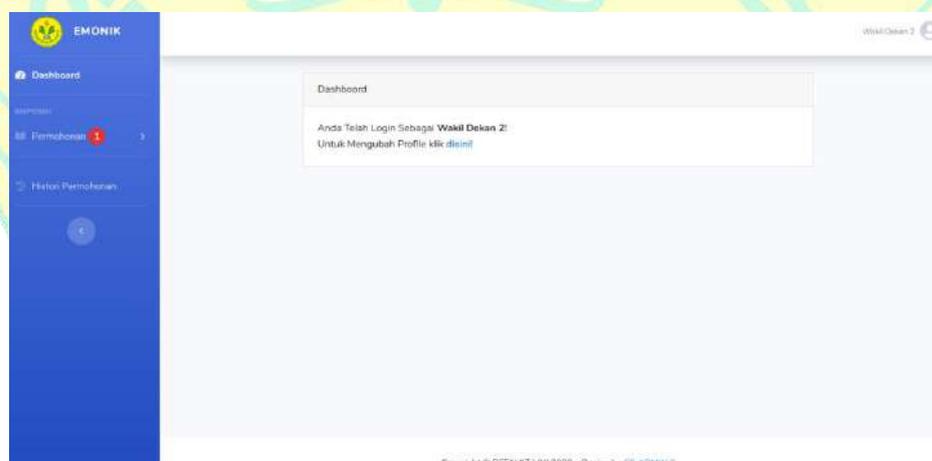
Setelah pemohon menekan tombol *submit* pada konfirmasi *submit* permohonan, maka akan muncul notifikasi permohonan berhasil di *submit*.



Gambar 4.55 : Tampilan Notifikasi Permohonan berhasil di *submit*

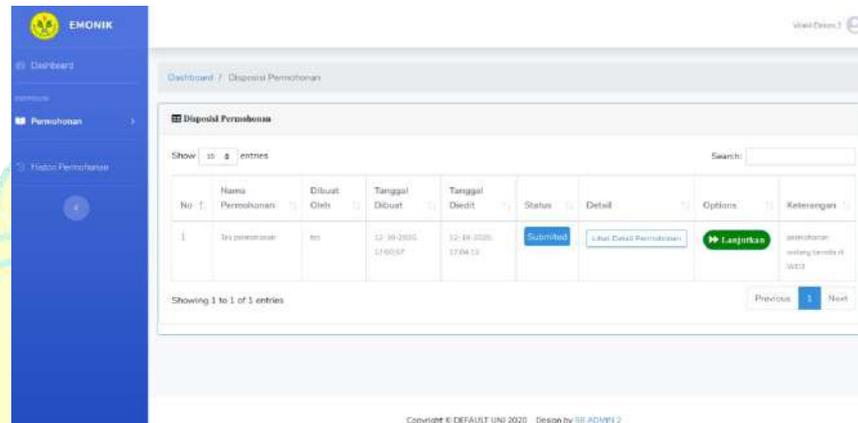
Permohonan sudah berhasil diajukan dan akan diproses oleh petugas keuangan yang bertugas untuk mendisposisi permohonan tersebut sampai permohonannya disetujui. Pemohon menunggu sampai permohonannya disetujui dan akan menerima notifikasi setiap disposisi permohonan sampai permohonannya disetujui.

Selanjutnya setelah proses pengajuan oleh pemohon selesai, akan dilanjutkan dengan beberapa proses disposisi permohonan sampai permohonan disetujui. Pertama proses disposisi oleh WD 2, WD 2 akan menerima notifikasi masuknya permohonan pada sistem.



Gambar 4.56 : Tampilan Halaman *Dashboard* WD 2

Kemudian WD 2 menekan menu disposisi permohonan yang akan menampilkan halaman disposisi permohonan pada WD 2. Pada halaman tersebut terdapat permohonan yang harus dilanjutkan atau didisposisikan.



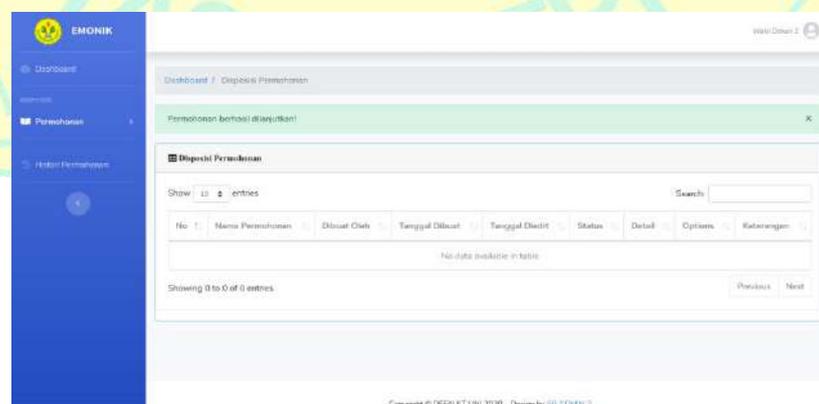
Gambar 4.57 : Tampilan Halaman Menu Disposisi Permohonan WD 2

Selanjutnya WD 2 akan menekan tombol lanjutkan untuk melanjutkan permohonan tersebut. Setelah menekan tombol lanjutkan, akan muncul Konfirmasi Lanjutkan Permohonan.



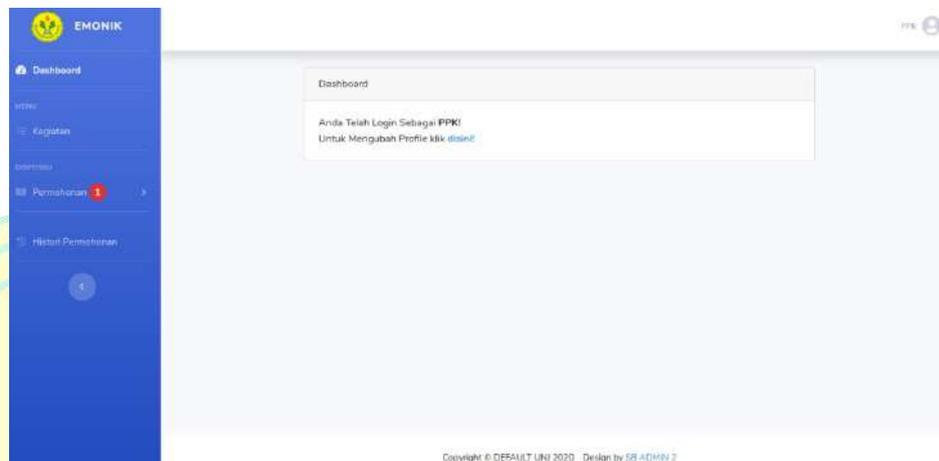
Gambar 4.58 : Tampilan Konfirmasi Lanjutkan Permohonan WD 2

Setelah WD 2 menekan tombol Lanjutkan pada Konfirmasi Lanjutkan Permohonan, maka akan muncul notifikasi permohonan berhasil di lanjutkan.



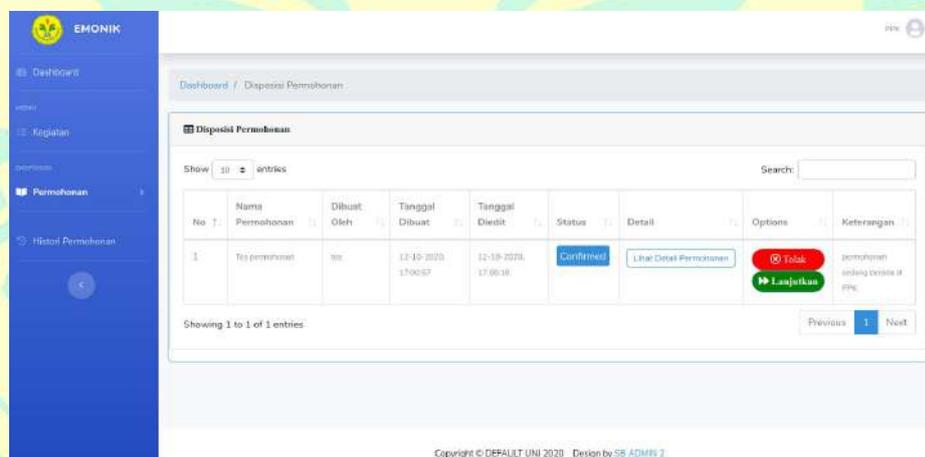
Gambar 4.59 : Tampilan Notifikasi Permohonan berhasil di lanjutkan WD 2

Setelah proses disposisi permohonan oleh WD 2 selesai, akan dilanjutkan dengan proses disposisi permohonan oleh PPK, PPK akan menerima notifikasi masuknya permohonan pada sistem.



Gambar 4.60 : Tampilan Halaman *Dashboard* PPK

Kemudian PPK menekan menu disposisi permohonan yang akan menampilkan halaman disposisi permohonan pada PPK. Pada halaman tersebut terdapat permohonan yang harus dilanjutkan atau didisposisikan.



Gambar 4.61 : Tampilan Halaman Menu Disposisi Permohonan PPK

Selanjutnya PPK akan memeriksa permohonan yang masuk dan memilih menekan tombol lanjutkan untuk melanjutkan permohonan tersebut atau menekan tombol tolak untuk menolak pengajuan permohonan tersebut. Setelah menekan tombol lanjutkan, akan muncul Konfirmasi Lanjutkan Permohonan. Setelah menekan tombol tolak maka akan muncul konfirmasi tolak permohonan.

Konfirmasi tolak permohonan

Alasan ditolak

Bukti Penolakan No file chosen

Tidak

Gambar 4.62 : Tampilan Konfirmasi Tolak Permohonan

Dalam penulisan ini penulis tidak akan menekan tombol tolak pada konfirmasi tolak permohonan, permohonan tersebut akan terus dilanjutkan sampai permohonan tersebut selesai.

Konfirmasi Lanjutkan Permohonan!

Anda yakin ingin melanjutkan?
[Tes permohonan]
*Melanjutkan ke Kasubag (disposisi 3)!

Tidak

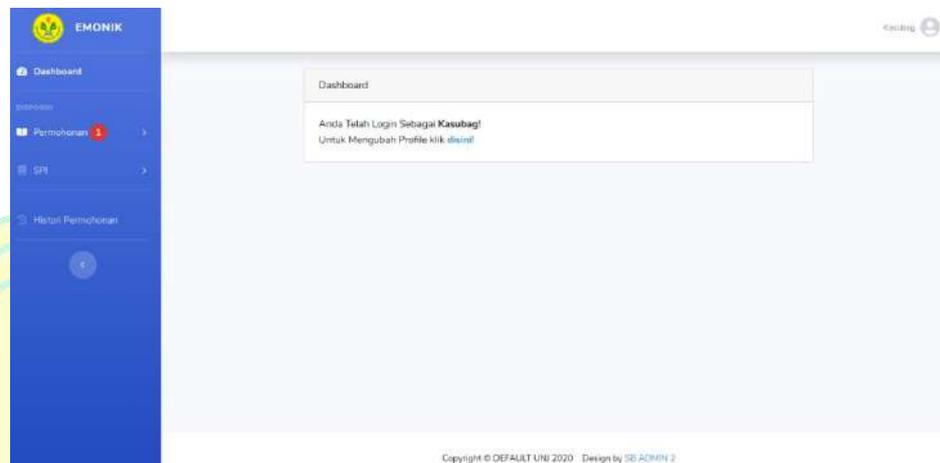
Gambar 4.63 : Tampilan Konfirmasi Lanjutkan Permohonan PPK

Setelah PPK menekan tombol Lanjutkan pada Konfirmasi Lanjutkan Permohonan, maka akan muncul notifikasi permohonan berhasil di lanjutkan.

The screenshot shows the EMONIK dashboard interface. On the left is a blue sidebar with navigation options: Dashboard, Registrasi, and Permohonan. The main content area displays a notification: 'Permohonan berhasil dilanjutkan!'. Below the notification is a table titled 'Disposisi Permohonan'. The table has columns for No., Nama Permohonan, Dibuat Oleh, Tanggal Dibuat, Tanggal Diedit, Status, Detail, Options, and Keterangan. The table is currently empty, showing 'No data available in table'. At the bottom of the dashboard, there is a footer: 'Copyright © DEFAULT UNI 2020 Design by SB ADMIN'.

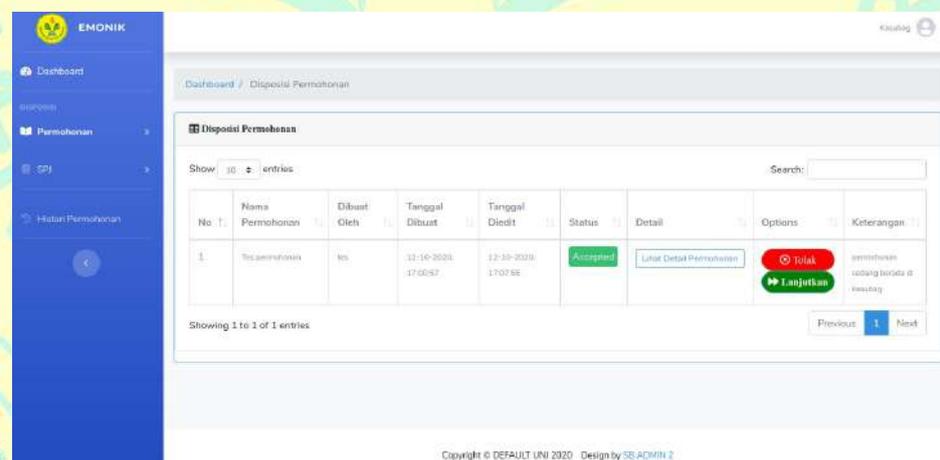
Gambar 4.64 : Tampilan Notifikasi Permohonan berhasil di lanjutkan PPK

Setelah proses disposisi permohonan oleh PPK selesai, akan dilanjutkan dengan proses disposisi permohonan oleh Kasubag, Kasubag akan menerima notifikasi masuknya permohonan pada sistem.



Gambar 4.65 : Tampilan Halaman *Dashboard* Kasubag

Kemudian Kasubag menekan menu disposisi permohonan yang akan menampilkan halaman disposisi permohonan pada Kasubag. Pada halaman tersebut terdapat permohonan yang harus dilanjutkan atau didisposisikan.



Gambar 4.66 : Tampilan Halaman Menu Disposisi Permohonan Kasubag

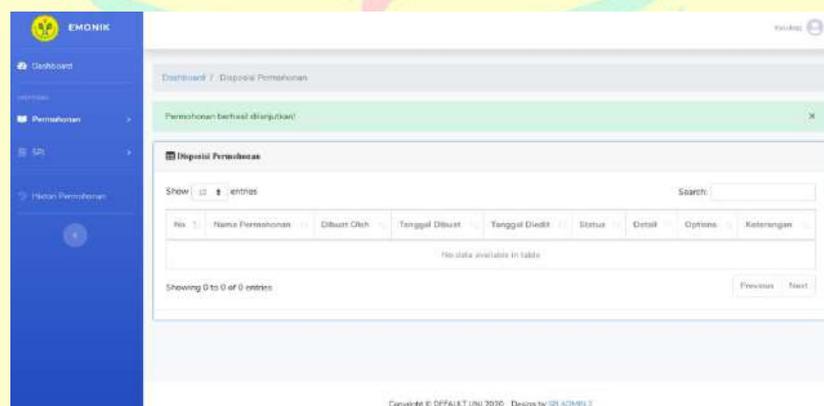
Selanjutnya Kasubag akan memeriksa permohonan yang masuk dan memilih menekan tombol lanjutkan untuk melanjutkan permohonan tersebut atau menekan tombol tolak untuk menolak pengajuan permohonan tersebut. Setelah menekan tombol lanjutkan, akan muncul Konfirmasi Lanjutkan Permohonan.

Konfirmasi Lanjutkan Permohonan!

Anda yakin ingin melanjutkan?
 [Tes permohonan]
 *Melanjutkan ke BPP (disposisi 4)

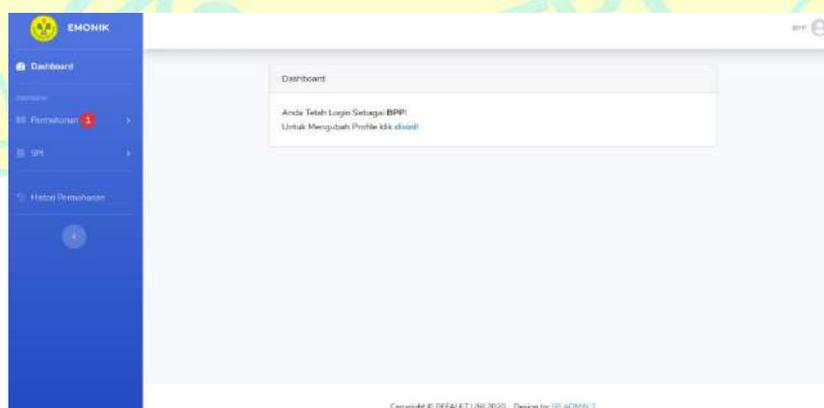
Tidak

Gambar 4.67 : Tampilan Konfirmasi Lanjutkan Permohonan Kasubag
 Pada proses disposisi Kasubag tidak akan ditampilkan lagi Tampilan Konfirmasi Tolak Permohonan, karena tampilannya sama dengan tampilan konfirmasi pada PPK. Setelah PPK menekan tombol Lanjutkan pada Konfirmasi Lanjutkan Permohonan, maka akan muncul notifikasi permohonan berhasil di lanjutkan.



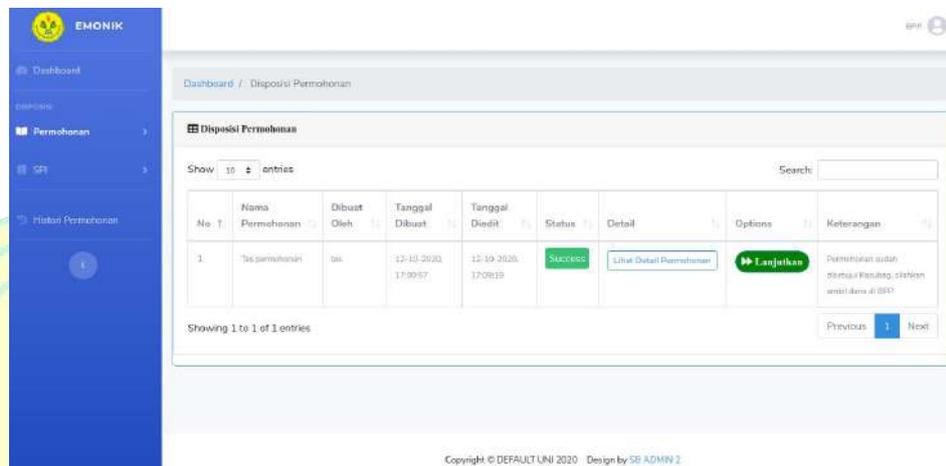
Gambar 4.68 : Tampilan Notifikasi Permohonan berhasil di lanjutkan Kasubag

Setelah proses disposisi permohonan oleh Kasubag selesai, akan dilanjutkan dengan proses disposisi permohonan oleh BPP, BPP akan menerima notifikasi masuknya permohonan pada sistem.



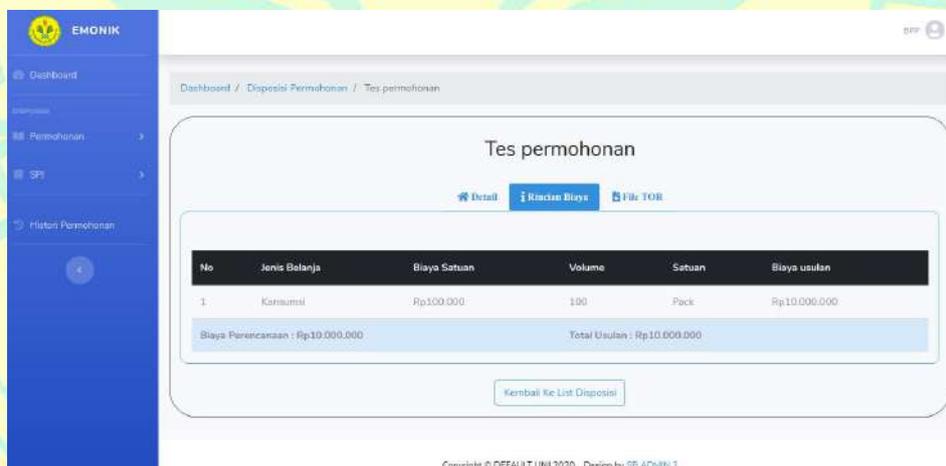
Gambar 4.69 : Tampilan Halaman *Dashboard* BPP

Kemudian BPP menekan menu disposisi permohonan yang akan menampilkan halaman disposisi permohonan pada BPP. Pada halaman tersebut terdapat permohonan yang harus dilanjutkan atau didisposisikan.



Gambar 4.70 : Tampilan Halaman Menu Disposisi Permohonan BPP

Pada Halaman Menu Disposisi Permohonan, jika petugas permohonan menekan tombol detail permohonan dan pada tab rincian akan menampilkan usulan rincian biaya yang diusulkan pemohon.



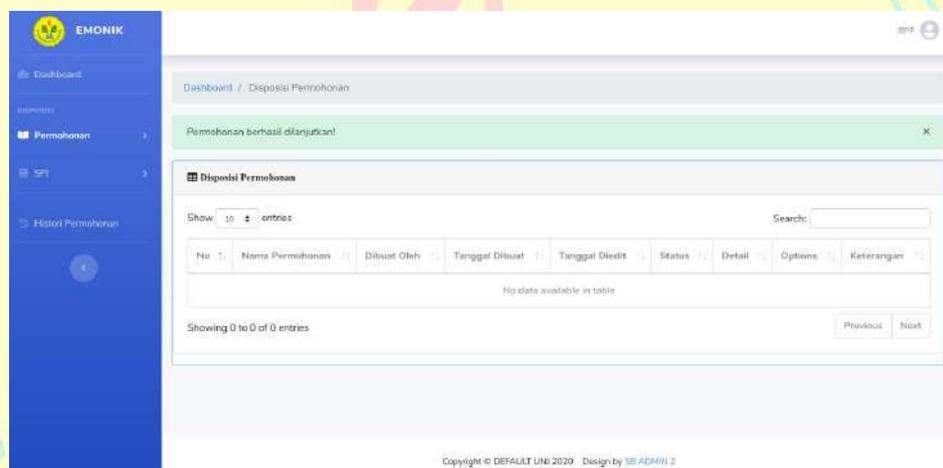
Gambar 4.71 : Tampilan Halaman Menu Rincian Biaya pada Disposisi Permohonan

Selanjutnya setelah pemohon mengambil dana kegiatan, BPP akan menekan tombol lanjutkan untuk melanjutkan permohonan tersebut. Setelah menekan tombol lanjutkan, akan muncul Konfirmasi Lanjutkan Permohonan.



Gambar 4.72 : Tampilan Konfirmasi Lanjutkan Permohonan BPP

Setelah BPP menekan tombol Lanjutkan pada Konfirmasi Lanjutkan Permohonan, maka akan muncul notifikasi permohonan berhasil di lanjutkan.



Gambar 4.73 : Tampilan Notifikasi Permohonan berhasil di lanjutkan BPP

Kemudian setelah BPP menerima Notifikasi Permohonan berhasil di lanjutkan, maka permohonan yang diajukan pemohon telah selesai dan akan dilanjutkan ke SPJ yang akan dibahas pada iterasi selanjutnya.

e. *Test*

Target pada proses *Test* ini diharapkan sudah mencapai 10% pada iterasi *construction C1* ini dan mencapai target 20% pada target *development plan*. Pada proses *Test* ini akan dibahas artefak lanjutan *Test Plan* dari iterasi sebelumnya.

1. *Test Plan*

Melanjutkan iterasi sebelumnya. Pada iterasi ini mengubah *planning* testing dan UAT yang sudah ada pada iterasi agar dapat berjalan sesuai dengan iterasi. Penyesuaian *Test Plan* dapat dilihat pada tabel berikut.

Tabel 4.8 : Penyesuaian *Test Plan*

No	<i>Test Plan</i>	Target
1	<i>Alpha Testing.</i>	Iterasi C2
2	<i>Beta Testing.</i>	Iterasi C2-C3
3	<i>User Acceptance Test (UAT).</i>	Iterasi C3-T2

f. *Deployment*

Target pada proses *Deployment* ini diharapkan sudah mencapai 10% pada iterasi *construction* C1 ini. Pada proses *Deployment* ini akan dibahas artefak *Deployment Plan*.

1. *Deployment Plan*

Target pada artefak ini menentukan *planning* rilis produk pada iterasi agar dapat berjalan sesuai waktu yang ditentukan. *Deployment* dapat dilakukan saat implementasi sistem sudah selesai, karena pada iterasi ini masih dalam proses implementasi dan belum selesai, maka akan dibuat *Deployment Plan* yang dapat dilihat pada tabel berikut.

Tabel 4.9 : *Deployment Plan*

No	<i>Deployment Plan</i>	Target
1	Rilis produk <i>Alpha</i> .	Iterasi C2
2	Rilis produk <i>Beta</i> .	Iterasi C2-C3
3	Rilis produk <i>Final</i> .	Iterasi C3-T2

2. *Iterasi construction C2*

Iterasi selanjutnya pada tahap *construction* C2 berdasarkan *development plan* yaitu dibahasnya 10% *Analysis & Design*, 20% *Implementation*, 10% *Test* dan 10% *Deployment*.

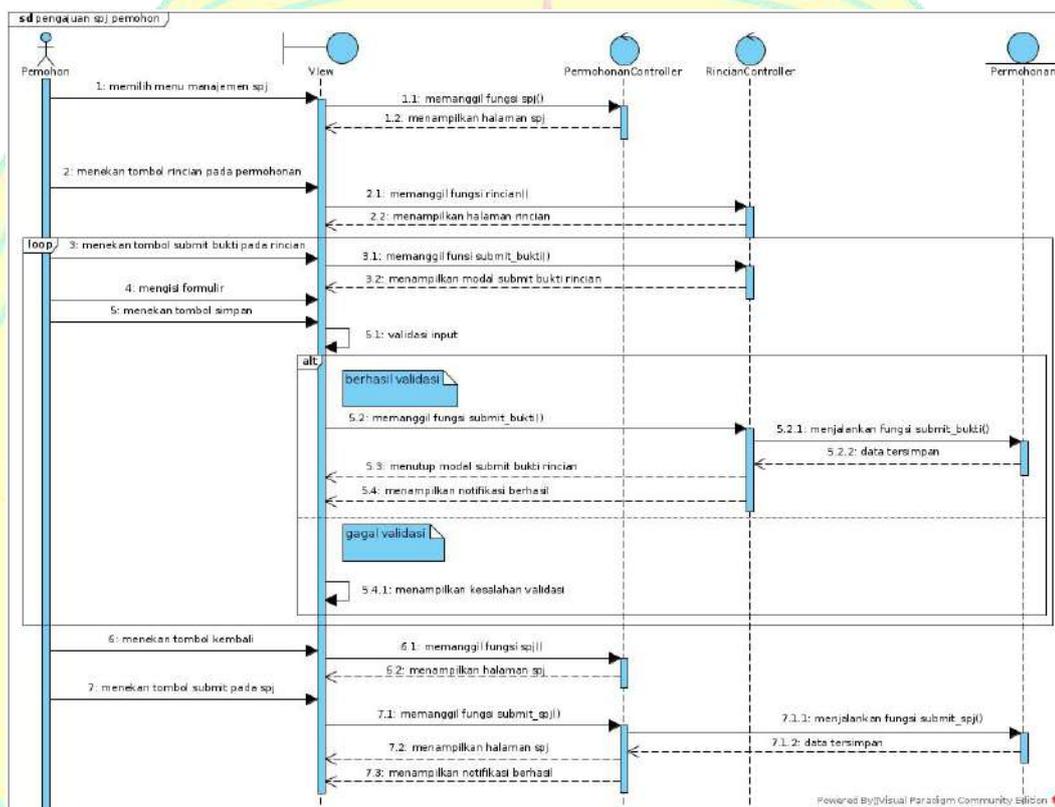
a. *Analysis & Design*

Target pada proses *Analysis & Design* ini diharapkan sudah mencapai 10% pada iterasi *construction* C2 ini dan mencapai target 100% pada target *development plan* yang artinya selesai semua proses pada *Analysis & Design* ini dan akan menyesuaikan penambahan sesuai kebutuhan. Pada proses

Analysis & Design ini akan dibahas artefak lanjutan *Design model* dari iterasi sebelumnya.

1. *Design model*

Melanjutkan dari iterasi sebelumnya. Pada iterasi ini akan dibahas *sequence diagram* SPJ dengan lengkap dalam pembuatan sistem. *Sequence diagram* tersebut dapat dilihat pada gambar berikut.



Gambar 4.74 : *Sequence diagram* pengajuan SPJ pemohon

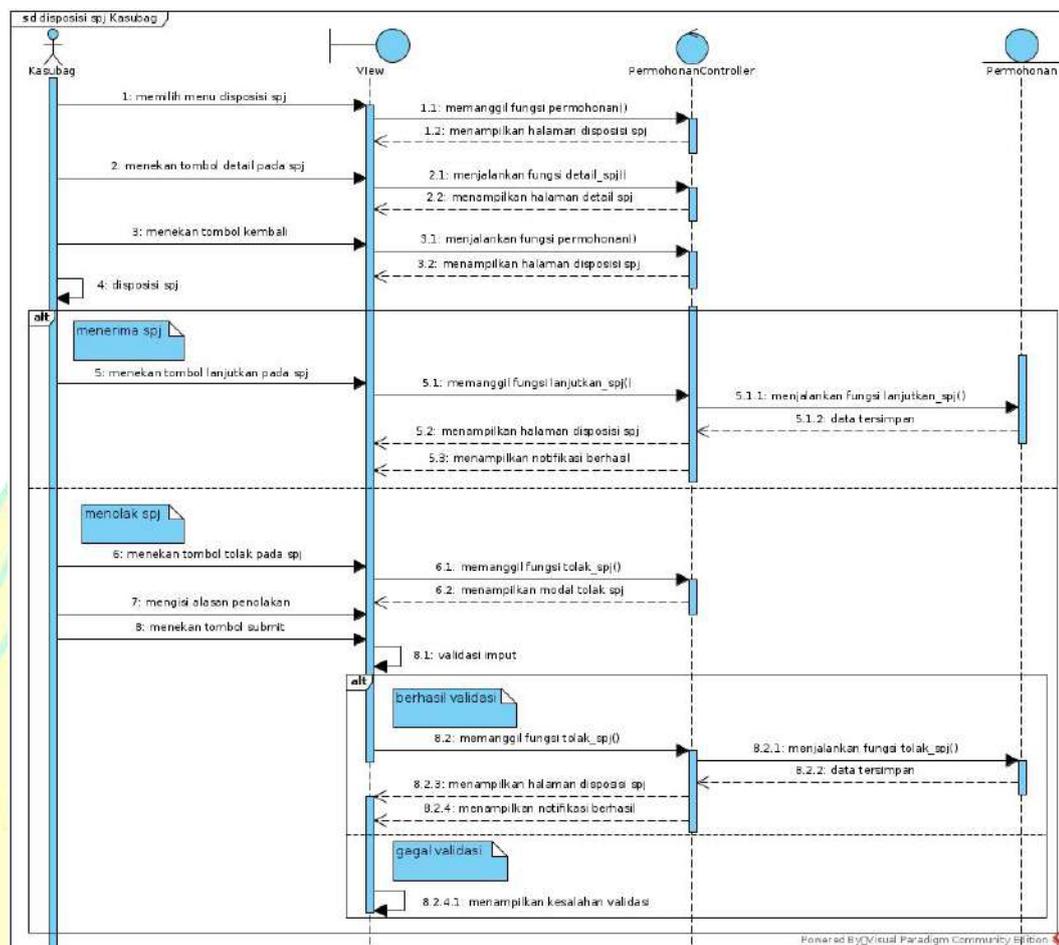
Pada gambar 4.74 menjelaskan alur pengajuan SPJ dengan lengkap pada sistem, dimulai dari pemohon memilih menu manajemen SPJ, lalu sistem akan memanggil fungsi `spj()` yang akan diproses menjadi tampilan halaman SPJ.

Selanjutnya pemohon mengisi bukti rincian pada permohonan yang akan dilakukan SPJ-nya terlebih dahulu sebelum bisa melanjutkan SPJ tersebut, pemohon akan menekan tombol rincian pada permohonan, lalu sistem akan memanggil fungsi `rincian()` yang akan diproses menjadi tampilan halaman rincian.

Pada proses pengisian bukti rincian, pemohon akan terus melakukan pengulangan proses input bukti rincian sampai semua bukti rincian yang dibutuhkan pemohon tersebut sudah lengkap. Pemohon menekan tombol submit bukti pada rincian, lalu sistem akan memanggil fungsi `submit_bukti()` yang akan diproses menjadi tampilan modal submit bukti rincian. Kemudian pemohon akan mengisi formulir tersebut, setelah selesai pemohon akan menekan tombol simpan, kemudian dilanjutkan dengan melakukan validasi input. Jika berhasil validasi, sistem akan memanggil fungsi `submit_bukti()` dan menjalankan fungsi tersebut agar data input pemohon disimpan, setelah data tersimpan maka sistem akan menutup modal submit bukti rincian dan menampilkan notifikasi berhasil, jika gagal validasi maka akan menampilkan kesalahan validasi.

Setelah pemohon selesai mengisi semua bukti rincian yang dibutuhkan, maka pemohon akan menekan tombol kembali, lalu sistem akan memanggil fungsi `permohonan()` yang akan diproses menjadi tampilan halaman SPJ. Selanjutnya pemohon akan menekan tombol submit pada SPJ, lalu sistem akan memanggil fungsi `submit_spj()` dan menjalankan fungsi tersebut agar data permohonan disimpan, setelah data tersimpan maka sistem akan menampilkan halaman SPJ dan menampilkan notifikasi berhasil. Setelah selesai proses mengajukan SPJ, maka akan dilanjutkan proses disposisi permohonan oleh Kasubag.

Pada gambar 4.75 menjelaskan alur disposisi SPJ Kasubag dengan lengkap pada sistem, dimulai dari Kasubag memilih menu disposisi SPJ, lalu sistem akan memanggil fungsi `spj()` dan akan menampilkan halaman disposisi SPJ. Selanjutnya Kasubag menekan tombol detail pada SPJ untuk memeriksa SPJ, lalu sistem akan menjalankan fungsi `detail_spj()` yang akan diproses menjadi tampilan halaman detail SPJ. Setelah selesai, Kasubag menekan tombol kembali, lalu sistem akan menjalankan fungsi `spj()` dan menampilkan halaman disposisi SPJ.

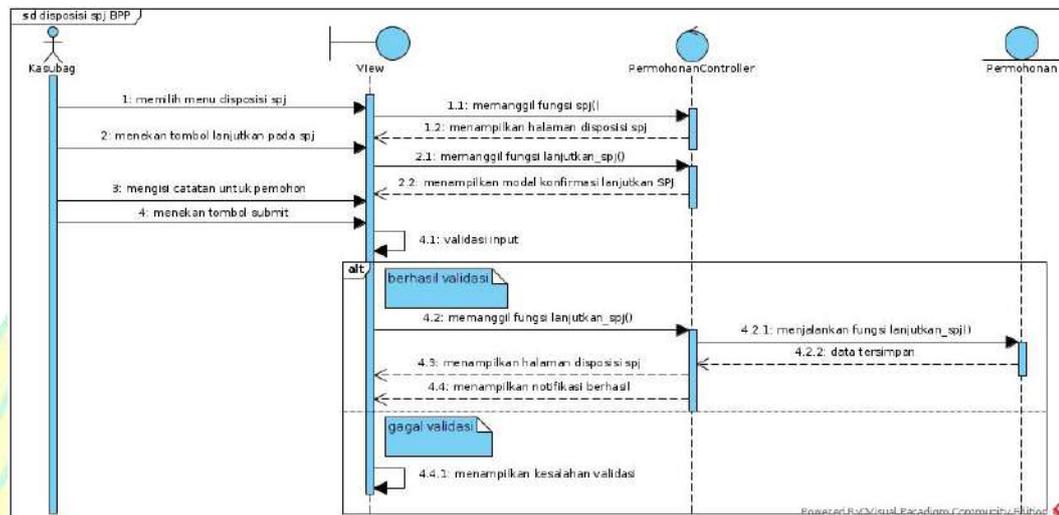


Gambar 4.75 : *Sequence diagram* disposisi SPJ Kasubag

Selanjutnya Kasubag akan melakukan disposisi SPJ. Jika menerima SPJ, Kasubag akan menekan tombol lanjutkan pada SPJ, lalu sistem akan memanggil fungsi `lanjutkan_spj()` dan menjalankan fungsi tersebut agar data SPJ disimpan, setelah data tersimpan maka sistem akan menampilkan halaman disposisi SPJ dan menampilkan notifikasi berhasil.

Jika menolak SPJ, Kasubag akan menekan tombol tolak pada SPJ, lalu sistem akan memanggil fungsi `tolak_spj()` dan sistem akan menampilkan modal tolak SPJ, Kasubag mengisi alasan penolakan dan jika sudah selesai akan menekan tombol submit kemudian dilanjutkan dengan validasi input. Jika berhasil validasi, sistem akan memanggil fungsi `tolak_spj()` dan menjalankan fungsi tersebut agar data input disimpan, setelah data tersimpan maka sistem akan menampilkan halaman disposisi SPJ dan menampilkan

notifikasi berhasil, jika gagal validasi maka akan menampilkan kesalahan validasi. Setelah selesai dsposisi permohonan Kasubag, maka akan dilanjutkan proses disposisi SPJ oleh BPP.



Gambar 4.76 : *Sequence diagram* disposisi SPJ BPP

Pada gambar 4.76 menjelaskan alur disposisi SPJ BPP dengan lengkap pada sistem, dimulai dari BPP memilih menu disposisi SPJ, lalu sistem akan memanggil fungsi `spj()` dan akan menampilkan halaman disposisi SPJ.

Selanjutnya BPP akan menekan tombol lanjutkan SPJ, lalu sistem akan memanggil fungsi `lanjutkan_spj()` dan akan menampilkan modal konfirmasi lanjutkan SPJ, kemudian BPP akan mengisi catatan untuk pemohon. Setelah BPP selesai mengisi catatan pemohon, BPP akan menekan tombol submit untuk melanjutkan disposisi SPJ, lalu sistem akan melakukan validasi input.

Jika validasi berhasil maka sistem akan memanggil fungsi `lanjutkan_spj()` menjalankan fungsi tersebut agar data SPJ disimpan, setelah data tersimpan maka sistem akan menampilkan halaman disposisi SPJ dan menampilkan notifikasi berhasil. Jika gagal validasi maka akan menampilkan kesalahan validasi. Setelah selesai disposisi SPJ BPP, maka selesailah semua proses disposisi SPJ.

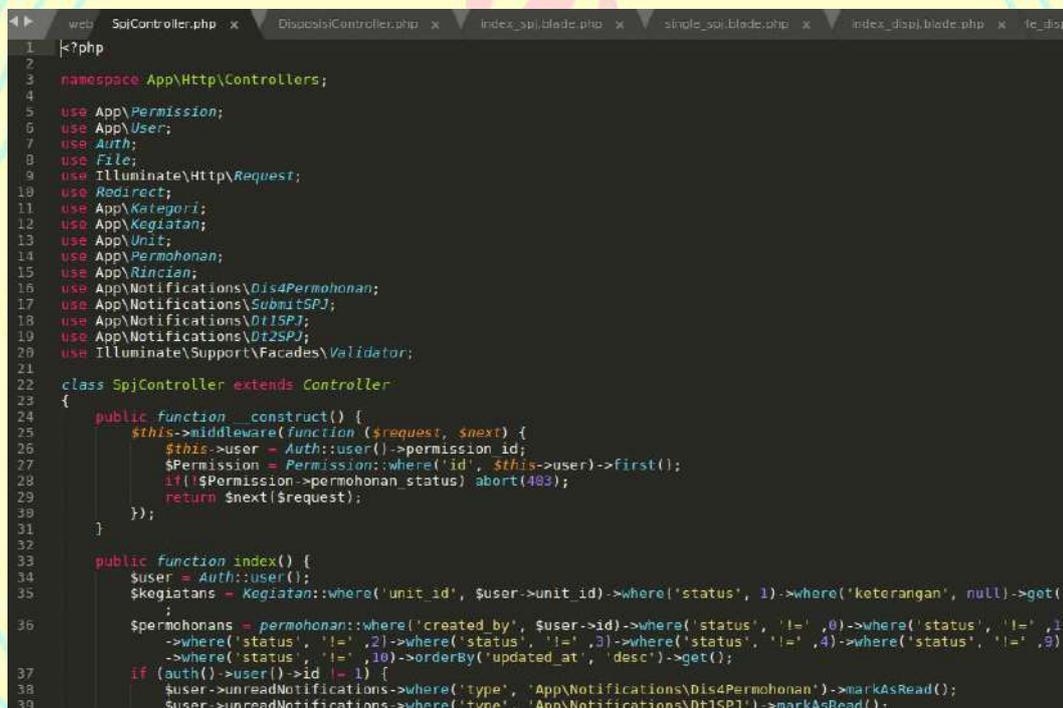
b. Implementation

Target pada proses *Implementation* ini diharapkan sudah mencapai 20% pada iterasi *construction* C2 ini dan mencapai target 70% pada target

development plan. Pada proses *Implementation* ini akan dibahas artefak *Component* dan *Implementation Model*.

1. *Component*

Melanjutkan dari iterasi sebelumnya. Pada iterasi ini sudah dilakukan proses implementasi kode pada proses pembuatan sistem pengajuan SPJ yang dimulai dari pemohon melanjutkan SPJ permohonannya sampai disposisi SPJ terakhir sehingga SPJ tersebut disetujui. Berikut adalah gambar sampel implementasi kode pada beberapa *file* sehingga dapat menghasilkan sistem pembagian pengajuan SPJ.



```

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Permission;
6 use App\User;
7 use Auth;
8 use File;
9 use Illuminate\Http\Request;
10 use Redirect;
11 use App\Kategori;
12 use App\Kegiatan;
13 use App\Unit;
14 use App\Permohonan;
15 use App\Rincian;
16 use App\Notifications\Dis4Permohonan;
17 use App\Notifications\SubmitSPJ;
18 use App\Notifications\Dt1SPJ;
19 use App\Notifications\Dt2SPJ;
20 use Illuminate\Support\Facades\Validator;
21
22 class SpjController extends Controller
23 {
24     public function __construct() {
25         $this->middleware(function ($request, $next) {
26             $this->user = Auth::user()->permission id;
27             $Permission = Permission::where('id', $this->user)->first();
28             if(!$Permission->permohonan status) abort(403);
29             return $next($request);
30         });
31     }
32
33     public function index() {
34         $user = Auth::user();
35         $kegiatan = Kegiatan::where('unit_id', $user->unit_id)->where('status', 1)->where('keterangan', null)->get();
36
37         $permohonan = permohonan::where('created by', $user->id)->where('status', '!=', 0)->where('status', '!=', 1)
38             ->where('status', '!=', 2)->where('status', '!=', 3)->where('status', '!=', 4)->where('status', '!=', 9)
39             ->where('status', '!=', 10)->orderBy('updated_at', 'desc')->get();
40
41         if (auth()->user()->id != 1) {
42             $user->unreadNotifications->where('type', 'App\Notifications\Dis4Permohonan')->markAsRead();
43             $user->unreadNotifications->where('type', 'App\Notifications\Dt1SPJ')->markAsRead();
44         }
45     }
46 }

```

Gambar 4.77 : Sampel kode *SpjController*

Pada gambar 4.77 terlihat sampel kode *SpjController* yang digunakan untuk mengambil memproses kegiatan SPJ pada sistem. Karena *Model* yang digunakan untuk *SpjController* menggunakan model permohonan yang telah ditampilkan pada iterasi sebelumnya, maka pada iterasi ini tidak akan ditampilkan lagi modelnya. Selain *SpjController*, dibutuhkan implementasi kode *view* untuk menampilkan halaman manajemen SPJ. Berikut adalah sampel kode halaman manajemen SPJ.

```

1 | extends('layouts.backend')
2 |
3 | @section('content')
4 | <!-- Breadcrumbs -->
5 | <ol class="breadcrumb">
6 | <li class="breadcrumb-item">
7 | <a href="{{ route('home') }}">Dashboard</a>
8 | </li>
9 | <li class="breadcrumb-item active">
10 | Manajemen SPJ
11 | </li>
12 | </ol>
13 |
14 | @include('layouts.pesan')
15 |
16 | <div class="card border border-info">
17 | <div class="card mb-3">
18 | <div class="card-header">
19 | <i class="fas fa-table">&nbsp;&nbsp;&nbsp;Manajemen SPJ</i>
20 | </div>
21 | <div class="card-body">
22 | <div class="table-responsive">
23 | <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
24 | <thead>
25 | <tr>
26 | <th>No</th>
27 | <th>Nama Permohonan</th>
28 | <th>Dibuat Oleh</th>
29 | <th>Tanggal Dibuat</th>
30 | <th>Tanggal Diodit</th>
31 | <th>Status</th>
32 | <th>Detail</th>
33 | <th>Options</th>
34 | <th>Keterangan</th>
35 | </tr>
36 | </thead>
37 | <tbody>
38 | @if($permohonans->count())
39 | @foreach($permohonans as $key => $permohonan)
40 | <tr>
41 | <td>{{ $key }}</td>
42 |

```

Gambar 4.78 : Sampel kode Halaman SPJ

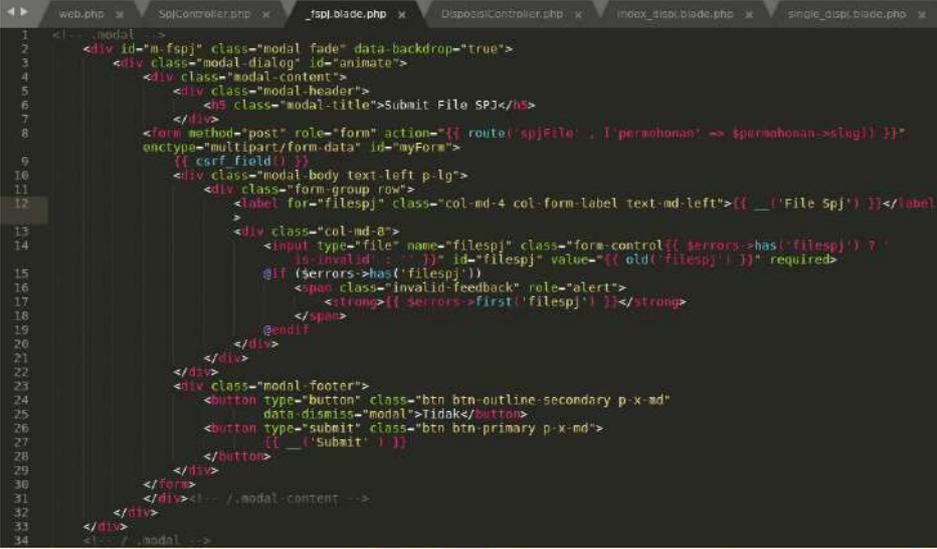
Setelah membuat halaman manajemen SPJ, dilanjutkan dengan membuat halaman *single* SPJ dan modal submit bukti untuk melengkapi SPJ. Karena sampel kode *RincianModel* dan *RincianController* sudah ditampilkan pada iterasi sebelumnya, maka pada iterasi ini tidak akan ditampilkan lagi. Implementasi kode *view* untuk menampilkan halaman *single* SPJ dan modal membuat submit bukti rincian. Berikut adalah sample kode *single* SPJ dan beberapa sampel kode untuk membuat submit bukti rincian.

```

1 | extends('layouts.backend')
2 |
3 | @section('content')
4 | <!-- Breadcrumbs -->
5 | <ol class="breadcrumb">
6 | <li class="breadcrumb-item">
7 | <a href="{{ route('home') }}">Dashboard</a>
8 | </li>
9 | <li class="breadcrumb-item">
10 | <a href="{{ route('spj') }}">Managemnt SPJ</a>
11 | </li>
12 | <li class="breadcrumb-item active">
13 | {{ $permohonan->nama }}
14 | </li>
15 | </ol>
16 |
17 | @include('layouts.pesan')
18 |
19 | <div class="col-md-12 border border-dark" style="border-radius: 40px">
20 | <div class="text-center" style="margin-top: 30px">{{ $permohonan->nama }}</div>
21 | <hr>
22 | <div class="nav-tabs-navigation d-flex justify-content-center">
23 | <div class="nav-tabs-wrapper">
24 | <ol class="nav nav-tabs nav-pills nav-fill" role="tablist" id="tabMenu">
25 | <li class="nav-item nav-justified">
26 | <a class="flex-sm-fill text-sm-center nav-link active" data-toggle="tab" href="#home"
27 | role="tab"><i class="fas fa-home"> Detail</i></a>
28 | </li>
29 | <li class="nav-item nav-justified">
30 | <a class="flex-sm-fill text-sm-center nav-link" data-toggle="tab" href="#profile"
31 | role="tab"><i class="fas fa-info"> Rincian Biaya</i></a>
32 | </li>
33 | <li class="nav-item nav-justified">
34 | <a class="flex-sm-fill text-sm-center nav-link" data-toggle="tab" href="#report"
35 | role="tab"><i class="fa fa-file-download"> File TOR</i></a>
36 | </li>
37 | <li class="nav-item nav-justified">
38 | <a class="flex-sm-fill text-sm-center nav-link" data-toggle="tab" href="#spj"
39 | role="tab"><i class="fa fa-file-upload"> File SPJ</i></a>
40 | </li>
41 | </ol>
42 | @if($permohonan->status == 0 || $permohonan->status == 0)

```

Gambar 4.79 : Sampel kode *SingleSPJ*



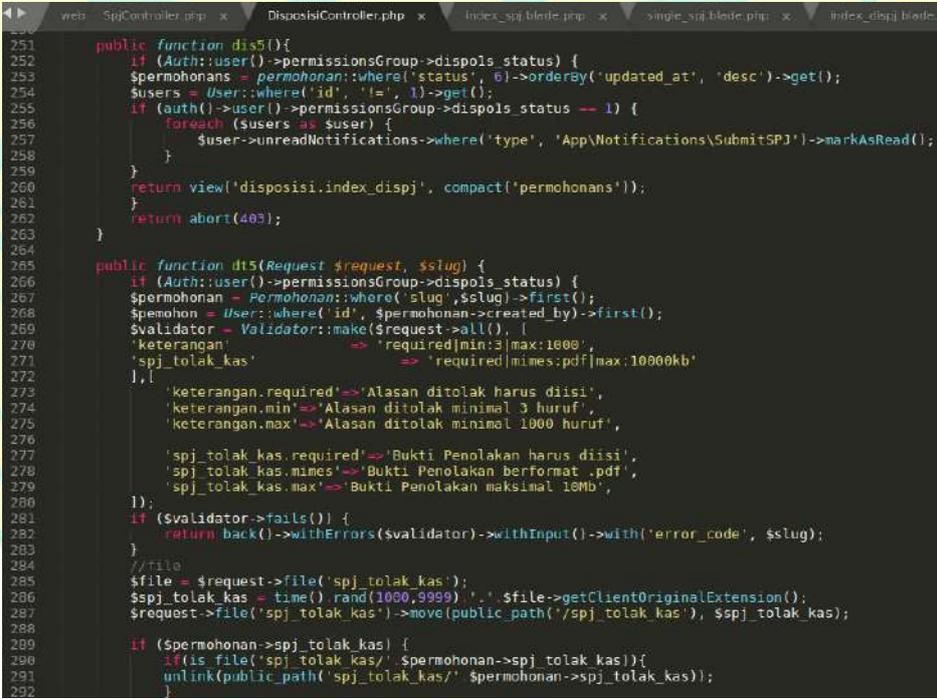
```

1 <!-- modal -->
2 <div id="m-fspj" class="modal fade" data-backdrop="true">
3 <div class="modal-dialog" id="animate">
4 <div class="modal-content">
5 <div class="modal-header">
6 <h5 class="modal-title">Submit File SPJ</h5>
7 </div>
8 <form method="post" role="form" action="{{ route('spjFile', ['permohonan' => $permohonan->slug]) }}"
9 enctype="multipart/form-data" id="myForm">
10 {{ csrf_field() }}
11 <div class="modal-body text-left p-lg">
12 <div class="form-group row">
13 <label for="filespj" class="col-md-4 col-form-label text-md-left">{{ __('File Spj') }}</label>
14 <div class="col-md-8">
15 <input type="file" name="filespj" class="form-control{{ $errors->has('filespj') ?
16 ' is-invalid' : '' }}" id="filespj" value="{{ old('filespj') }}" required>
17 <span class="invalid-feedback" role="alert">
18 <strong>{{ $errors->first('filespj') }}</strong>
19 </span>
20 </div>
21 </div>
22 </div>
23 <div class="modal-footer">
24 <button type="button" class="btn btn-outline-secondary p-x-md"
25 data-dismiss="modal">Tidaks</button>
26 <button type="submit" class="btn btn-primary p-x-md">
27 {{ __('Submit') }}
28 </button>
29 </div>
30 </form>
31 </div><!-- /.modal-content -->
32 </div><!-- /.modal -->
33
34

```

Gambar 4.80 : Sampel kode Modal Submit Bukti Rincian

Setelah selesai membuat alur pengajuan SPJ permohonan dan submit bukti rincian pada sistem, selanjutnya membuat beberapa kode pada alur disposisi SPJ. Berikut adalah gambar sampel implementasi kode pada beberapa *file* sehingga dapat menghasilkan alur sistem disposisi SPJ.



```

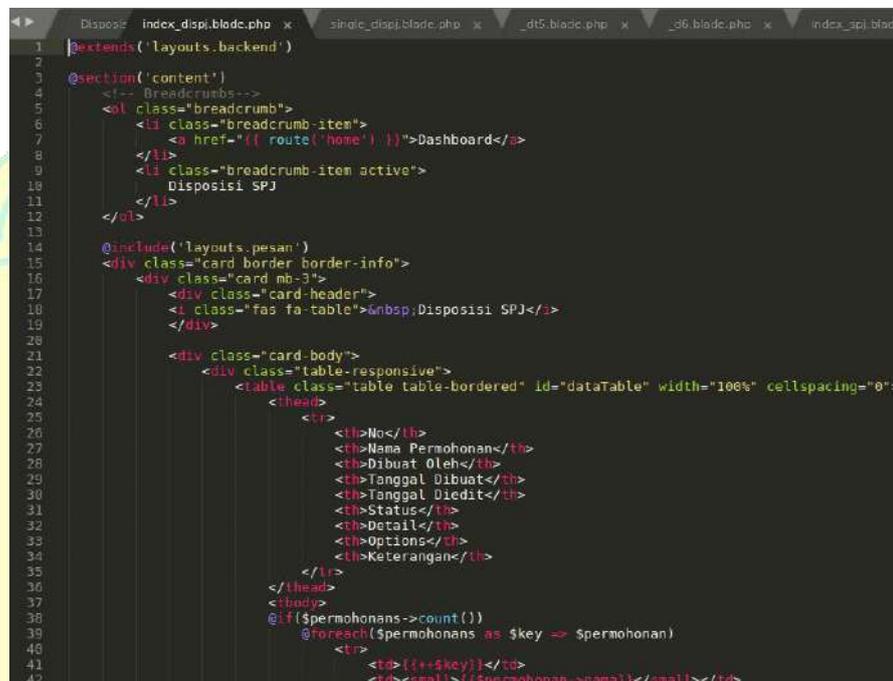
251 public function dis5(){
252     if (Auth::user()->permissionsGroup->dispols status) {
253         $permohonans = permohonan::where('status', 6)->orderBy('updated_at', 'desc')->get();
254         $users = User::where('id', '!=', 1)->get();
255         if (auth()->user()->permissionsGroup->dispols status == 1) {
256             foreach ($users as $user) {
257                 $user->unreadNotifications->where('type', 'App\Notifications\SubmitSPJ')->markAsRead();
258             }
259         }
260         return view('disposisi.index_dispj', compact('permohonans'));
261     }
262     return abort(403);
263 }
264
265 public function dt5(Request $request, $slug) {
266     if (Auth::user()->permissionsGroup->dispols status) {
267         $permohonan = Permohonan::where('slug', $slug)->first();
268         $permohon = User::where('id', $permohonan->created_by)->first();
269         $validator = Validator::make($request->all(), [
270             'keterangan' => 'required|min:3|max:1000',
271             'spj_tolak_kas' => 'required|mimes:pdf|max:1000kb'
272         ], [
273             'keterangan.required' => 'Alasan ditolak harus diisi',
274             'keterangan.min' => 'Alasan ditolak minimal 3 huruf',
275             'keterangan.max' => 'Alasan ditolak minimal 1000 huruf',
276
277             'spj_tolak_kas.required' => 'Bukti Penolakan harus diisi',
278             'spj_tolak_kas.mimes' => 'Bukti Penolakan berformat pdf',
279             'spj_tolak_kas.max' => 'Bukti Penolakan maksimal 10Mb',
280         ]);
281         if ($validator->fails()) {
282             return back()->withErrors($validator)->withInput()->with('error_code', $slug);
283         }
284         //file
285         $file = $request->file('spj_tolak_kas');
286         $spj_tolak_kas = time() . rand(1000,9999) . '.' . $file->getClientOriginalExtension();
287         $request->file('spj_tolak_kas')->move(public_path('/spj_tolak_kas'), $spj_tolak_kas);
288
289         if ($permohonan->spj_tolak_kas) {
290             if (is_file('spj_tolak_kas/' . $permohonan->spj_tolak_kas)) {
291                 unlink(public_path('spj_tolak_kas/' . $permohonan->spj_tolak_kas));
292             }

```

Gambar 4.81 : Sampel kode Fungsi SPJ pada *DisposisiController*

Pada gambar 4.81 terlihat sampel kode Fungsi SPJ pada *DisposisiController* yang digunakan untuk memproses disposisi data SPJ

dalam sistem. Selain *Controller*, dibutuhkan *view index* dan *single* pada disposisi untuk untuk menampilkan halaman *index* dan *single* disposisi SPJ. Berikut adalah sample kode *view index* dan *single* disposisi.

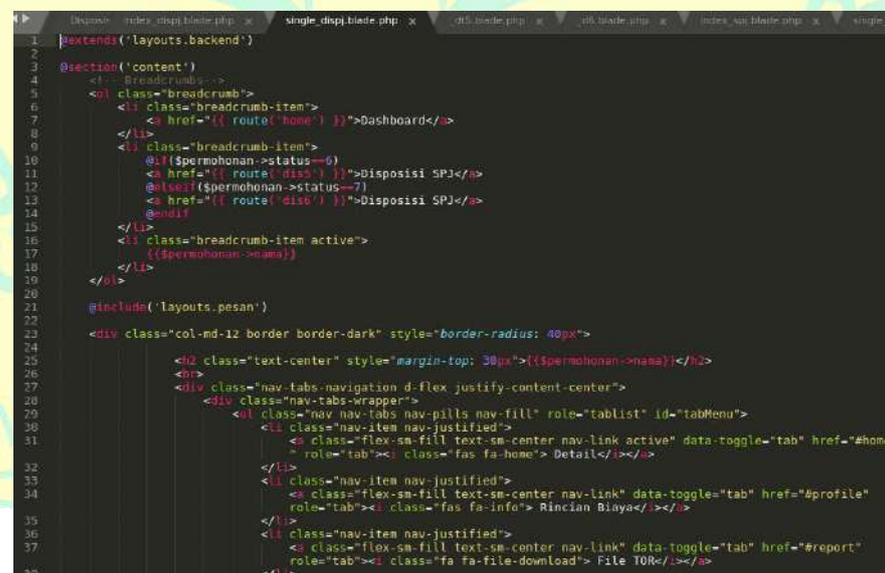


```

1 | @extends('layouts.backend')
2 |
3 | @section('content')
4 | <!-- Breadcrumbs-->
5 | <ol class="breadcrumb">
6 |   <li class="breadcrumb-item">
7 |     <a href="{{ route('home') }}">Dashboard</a>
8 |   </li>
9 |   <li class="breadcrumb-item active">
10 |     Disposisi SPJ
11 |   </li>
12 | </ol>
13 |
14 | @include('layouts.pesan')
15 | <div class="card border border-info">
16 |   <div class="card mb-3">
17 |     <div class="card-header">
18 |       <i class="fas fa-table">&nbsp;&nbsp;&nbsp;Disposisi SPJ</i>
19 |     </div>
20 |
21 |     <div class="card-body">
22 |       <div class="table-responsive">
23 |         <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
24 |           <thead>
25 |             <tr>
26 |               <th>No</th>
27 |               <th>Nama Permohonan</th>
28 |               <th>Dibuat Oleh</th>
29 |               <th>Tanggal Dibuat</th>
30 |               <th>Tanggal Diedit</th>
31 |               <th>Status</th>
32 |               <th>Detail</th>
33 |               <th>Options</th>
34 |               <th>Keterangan</th>
35 |             </tr>
36 |           </thead>
37 |           <tbody>
38 |             @if($permohonans->count())
39 |               @foreach($permohonans as $key => $permohonan)
40 |                 <tr>
41 |                   <td>{{++$key}}</td>
42 |                   <td><small>{{ $permohonan->nama}}</small></td>

```

Gambar 4.82 : Sampel kode Halaman Disposisi SPJ



```

1 | @extends('layouts.backend')
2 |
3 | @section('content')
4 | <!-- Breadcrumbs-->
5 | <ol class="breadcrumb">
6 |   <li class="breadcrumb-item">
7 |     <a href="{{ route('home') }}">Dashboard</a>
8 |   </li>
9 |   <li class="breadcrumb-item">
10 |     @if($permohonan->status==6)
11 |       <a href="{{ route('disp') }}">Disposisi SPJ</a>
12 |     @elseif($permohonan->status==7)
13 |       <a href="{{ route('disp') }}">Disposisi SPJ</a>
14 |     @endif
15 |   </li>
16 |   <li class="breadcrumb-item active">
17 |     {{ $permohonan->nama}}
18 |   </li>
19 | </ol>
20 |
21 | @include('layouts.pesan')
22 |
23 | <div class="col-md-12 border border-dark" style="border-radius: 40px">
24 |   <div class="text-center" style="margin-top: 30px">{{ $permohonan->nama}}</div>
25 |   <div class="nav-tabs navigation d-flex justify-content-center">
26 |     <div class="nav nav-tabs nav-pills nav-fill" role="tablist" id="tabMenu">
27 |       <li class="nav-item nav-justified">
28 |         <a class="flex-sm-fill text-sm-center nav-link active" data-toggle="tab" href="#home"
29 |           role="tab"><i class="fas fa-home"> Detail</i></a>
30 |       </li>
31 |       <li class="nav-item nav-justified">
32 |         <a class="flex-sm-fill text-sm-center nav-link" data-toggle="tab" href="#profile"
33 |           role="tab"><i class="fas fa-info"> Rincian Biaya</i></a>
34 |       </li>
35 |       <li class="nav-item nav-justified">
36 |         <a class="flex-sm-fill text-sm-center nav-link" data-toggle="tab" href="#report"
37 |           role="tab"><i class="fa fa-file-download"> File TOR</i></a>
38 |       </li>

```

Gambar 4.83 : Sampel kode *Single Disposisi Spj*

Pada saat implementasi kode disposisi SPJ, diperlukan tambahan Modal Tolak SPJ pada Kasubag. Berikut adalah sampel kode Modal Tolak SPJ pada Kasubag.

```

1 <div id="m-d5({{ $permohonan->slug }})" class="modal fade" data-backdrop="true">
2 <div class="modal-dialog" id="animate">
3 <div class="modal-content">
4 <div class="modal-header">
5 <h5 class="modal-title">Konfirmasi tolak SPJ</h5>
6 </div>
7 <form method="post" role="form" action="{{ route('dis5Tolak' , [ $permohonan => $permohonan->slug ] ) }}"
8 enctype="multipart/form-data" id="myForm">
9 {{ csrf_field() }}
10 <div class="modal-body text-left p-lg">
11 <div class="form-group row">
12 <label for="keterangan" class="col-md-4 col-form-label text-md-left">{{ __('Alasan ditolak') }}</label>
13 <div class="col-md-8">
14 <textarea name="keterangan" class="form-control{{ $errors->has('keterangan') ? '
15 is-invalid : '' }}" id="keterangan" value="{{ old('keterangan') }}" required>{{ old
16 'keterangan' }}</textarea>
17 @if ($errors->has('keterangan'))
18 <span class="invalid-feedback" role="alert">
19 <strong>{{ $errors->first('keterangan') }}</strong>
20 </span>
21 @endif
22 </div>
23 </div>
24 <div class="form-group row">
25 <label for="spj_tolak_kas" class="col-md-4 col-form-label text-md-left">{{ __('Bukti
26 Penolakan') }}</label>
27 <div class="col-md-8">
28 <input type="file" name="spj_tolak_kas" class="form-control{{ $errors->has('spj_tolak_kas
29 ' ) ? ' is-invalid : '' }}" id="spj_tolak_kas" value="{{ old('spj_tolak_kas') }}"
30 required>
31 @if ($errors->has('spj_tolak_kas'))
32 <span class="invalid-feedback" role="alert">
33 <strong>{{ $errors->first('spj_tolak_kas') }}</strong>
34 </span>
35 @endif
36 </div>
37 </div>
38 <div class="modal-footer">
39 <button type="button" class="btn dark-white p-x-md"

```

Gambar 4.84 : Sampel kode Modal Tolak SPJ pada Kasubag

Selain ditambahkan kode Modal Tolak SPJ pada Kasubag, diperlukan tambahan Modal Konfirmasi SPJ pada BPP. Berikut adalah sampel kode Modal Konfirmasi SPJ pada BPP.

```

1 <div id="m-d6({{ $permohonan->slug }})" class="modal fade" data-backdrop="true">
2 <div class="modal-dialog" id="animate">
3 <div class="modal-content">
4 <div class="modal-header">
5 <h5 class="modal-title">Konfirmasi Lanjutkan SPJ</h5>
6 </div>
7 <form method="post" role="form" action="{{ route('dis6Submit' , [ $permohonan => $permohonan->slug ] ) }}"
8 enctype="multipart/form-data" id="myForm">
9 {{ csrf_field() }}
10 <div class="modal-body text-left p-lg">
11 <div class="form-group row">
12 <label for="keterangan" class="col-md-4 col-form-label text-md-left">{{ __('Catatan Pemohon') }}</label>
13 <div class="col-md-8">
14 <textarea name="keterangan" class="form-control{{ $errors->has('keterangan') ? '
15 is-invalid : '' }}" id="keterangan" value="{{ old('keterangan') }}" required>{{ old
16 'keterangan' }}</textarea>
17 @if ($errors->has('keterangan'))
18 <span class="invalid-feedback" role="alert">
19 <strong>{{ $errors->first('keterangan') }}</strong>
20 </span>
21 @endif
22 </div>
23 </div>
24 </div>
25 <div class="modal-footer">
26 <button type="button" class="btn dark-white p-x-md"
27 data-dismiss="modal">Tidakk</button>
28 <button type="submit" class="btn btn-success p-x-md"
29 {{ __('Lanjutkan') }}</button>
30 </div>
31 </div>
32 </form>
33 </div>
34 </div>
35 </div>

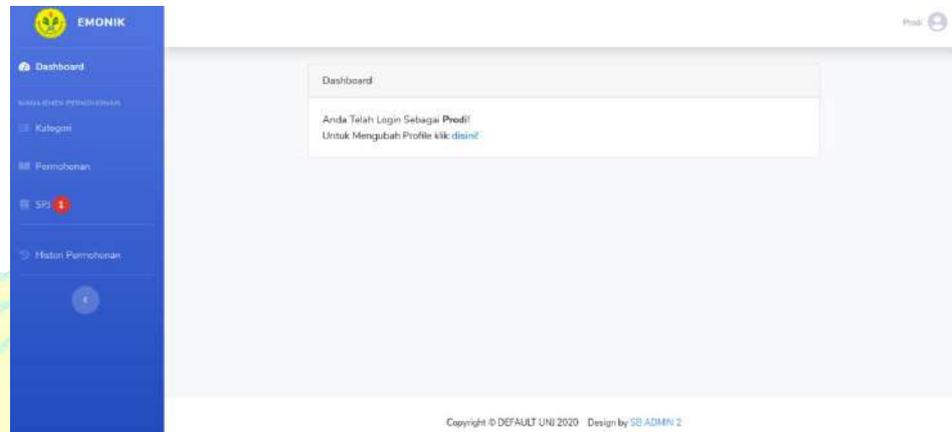
```

Gambar 4.85 : Sampel kode Modal Konfirmasi SPJ pada BPP

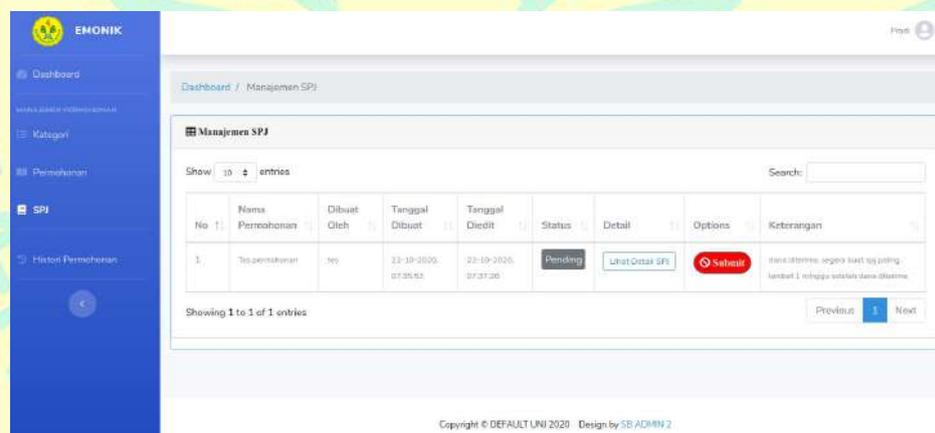
2. Implementation Model

Pada iterasi ini akan dijelaskan proses pengajuan SPJ permohonan oleh pemohon. Setelah permohonan selesai dan pemohon telah menerima dana kegiatan permohonannya, pemohon harus menyelesaikan SPJ-nya paling

lambat 7 hari setelah penerimaan dana. Berikut adalah tampilan halaman *Dashboard* pemohon dengan notifikasi SPJ.

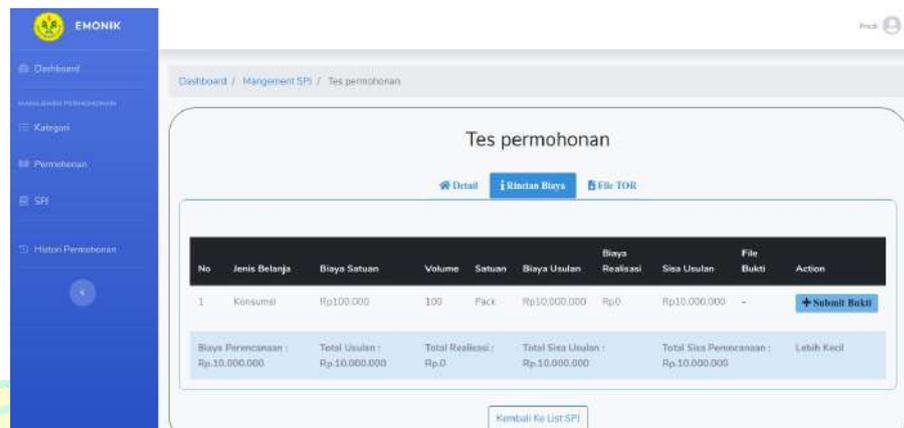


Gambar 4.86 : Tampilan Notifikasi SPJ pada *Dashboard* Pemohon Pertama akan dijelaskan proses pengajuan SPJ permohonan oleh pemohon dimulai dari pemohon melengkapi bukti SPJ permohonan dengan memilih menu SPJ dalam kelompok menu Manajemen Permohonan.



Gambar 4.87 : Tampilan Halaman Menu SPJ

Selanjutnya pada menu SPJ, pemohon menekan tombol lihat detail permohonan pada permohonan yang sudah dibuat yang akan menampilkan halaman detail dari SPJ permohonan tersebut dan menekan menu rincian biaya pada halaman tersebut.

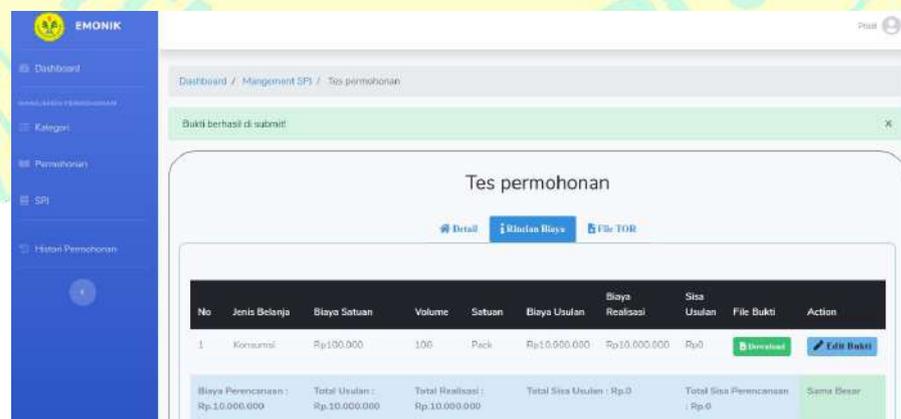


Gambar 4.88 : Tampilan Halaman Menu Submit Bukti Rincian

Pada menu Submit Bukti Rincian, pemohon bisa melengkapi bukti rincian biaya dengan menekan tombol Submit Bukti yang akan menampilkan Modal Submit Bukti.

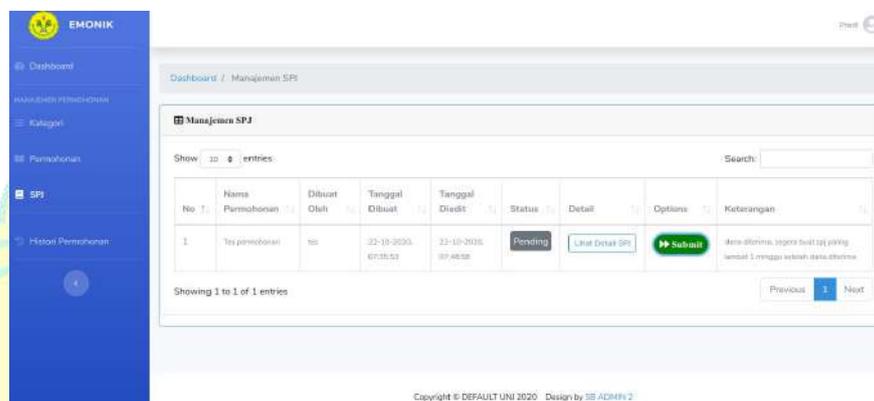
Gambar 4.89 : Tampilan Modal Submit Bukti Rincian

Setelah pemohon selesai mengisi formulir Submit Bukti Rincian, pemohon bisa menekan tombol submit pada formulir tersebut, setelah pemohon menekan tombol submit maka akan muncul notifikasi berhasil.



Gambar 4.90 : Tampilan Notifikasi Berhasil pada Submit Bukti Rincian

Setelah pemohon selesai mengisi Bukti Rincian Biaya, pemohon dapat menekan tombol kembali pada halaman itu untuk kembali ke halaman manajemen SPJ untuk melakukan *submit* permohonan.



Gambar 4.91 : Tampilan Menu *Submit* pada SPJ

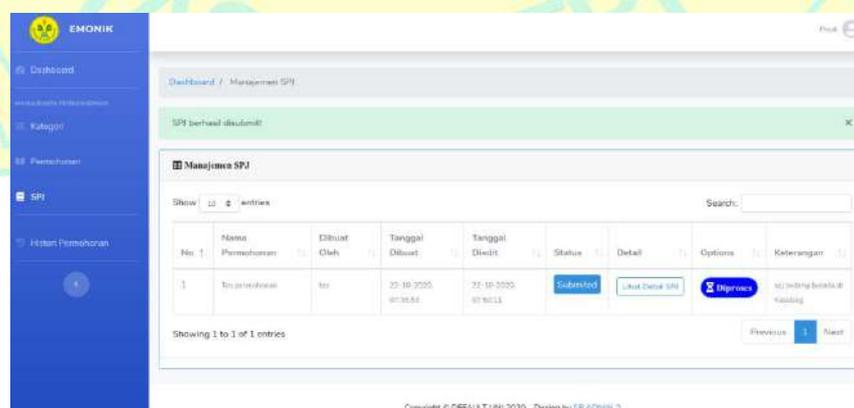
Kemudian pemohon menekan tombol *submit* yang akan menampilkan konfirmasi *submit* SPJ, pemohon dapat menekan tombol submit pada konfirmasi tersebut.

The 'Submit Bukti' form contains the following fields and buttons:

- Biaya Realisasi (Rp): 10.000.000
- File Bukti: Choose File tes.pdf
- Buttons: Tidak, Submit

Gambar 4.92 : Tampilan Konfirmasi *Submit* SPJ

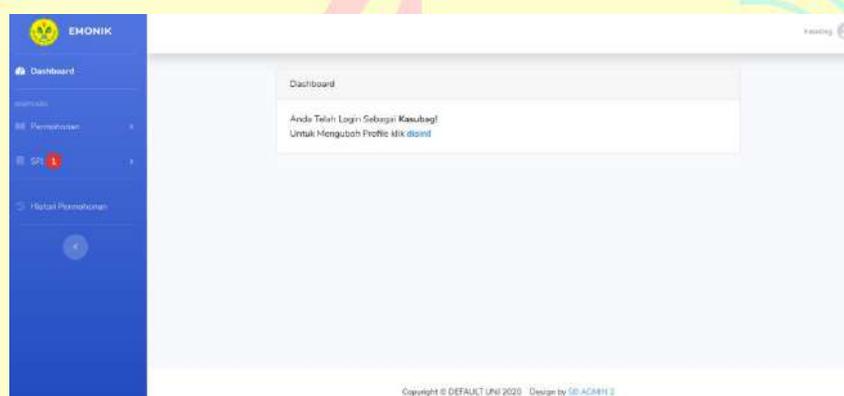
Setelah pemohon menekan tombol *submit* pada konfirmasi *submit* SPJ, maka akan muncul notifikasi SPJ berhasil di *submit*.



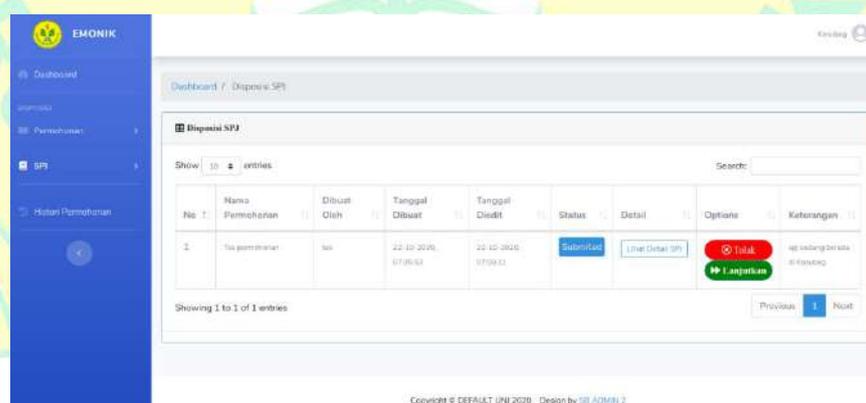
Gambar 4.93 : Tampilan Notifikasi SPJ berhasil di *submit*

SPJ sudah berhasil diajukan dan akan diproses oleh petugas keuangan yang bertugas untuk mendisposisi SPJ tersebut sampai SPJ-nya disetujui. Pemohon menunggu sampai SPJ-nya disetujui dan akan menerima notifikasi setiap disposisi permohonan sampai permohonannya disetujui.

Selanjutnya setelah proses pengajuan SPJ oleh pemohon selesai, akan dilanjutkan dengan beberapa proses disposisi permohonan sampai permohonan disetujui. Pertama proses disposisi oleh Kasubag, Kasubag akan menerima notifikasi masuknya permohonan pada sistem.

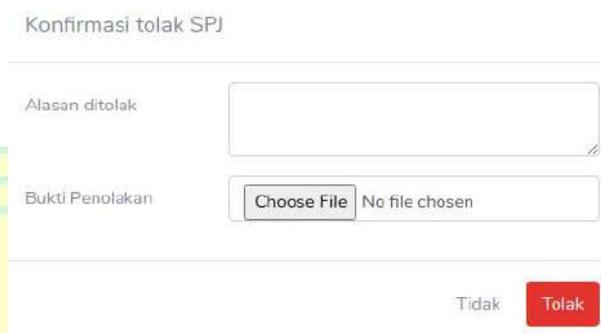


Gambar 4.94 : Tampilan Notifikasi diposisi SPJ pada *Dashboard* Kasubag
Kemudian Kasubag menekan menu disposisi SPJ yang akan menampilkan halaman disposisi SPJ pada Kasubag. Pada halaman tersebut terdapat SPJ yang harus dilanjutkan atau didisposisikan.



Gambar 4.95 : Tampilan Halaman Menu Disposisi SPJ Kasubag
Selanjutnya Kasubag akan memeriksa SPJ yang masuk dan memilih menekan tombol lanjutkan untuk melanjutkan permohonan tersebut atau menekan tombol tolak untuk menolak pengajuan SPJ tersebut. Setelah

menekan tombol lanjutkan, akan muncul Konfirmasi Lanjutkan SPJ. Setelah menekan tombol tolak maka akan muncul konfirmasi Tolak SPJ.



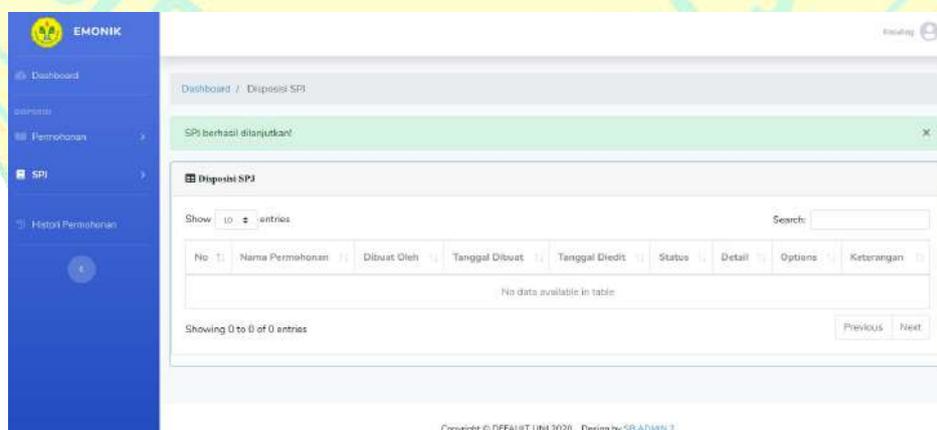
Gambar 4.96 : Tampilan Konfirmasi Tolak SPJ

Dalam penulisan ini penulis tidak akan menekan tombol tolak pada konfirmasi Tolak SPJ, permohonan tersebut akan terus dilanjutkan sampai SPJ tersebut selesai.



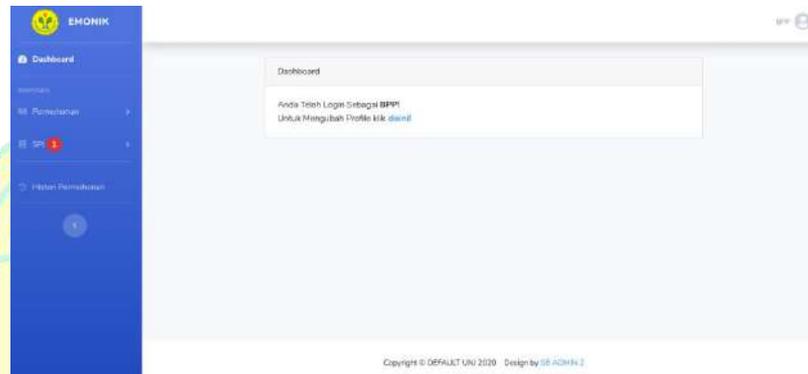
Gambar 4.97 : Tampilan Konfirmasi Lanjutkan SPJ Kasubag

Setelah Kasubag menekan tombol Lanjutkan pada Konfirmasi Lanjutkan SPJ, maka akan muncul notifikasi SPJ berhasil di lanjutkan.

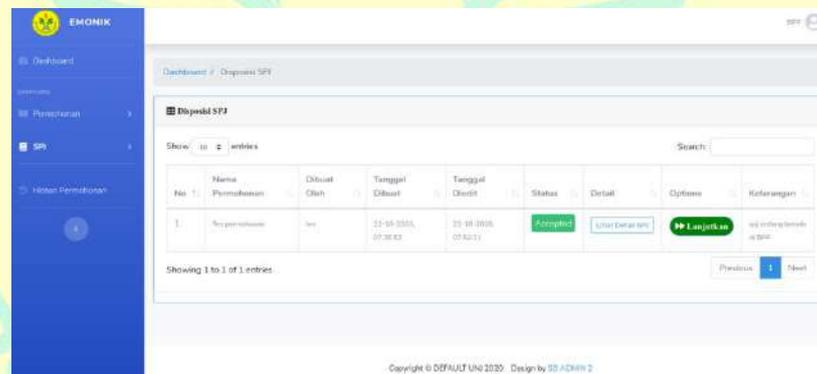


Gambar 4.98 : Tampilan Notifikasi SPJ berhasil di lanjutkan Kasubag

Setelah proses disposisi SPJ oleh Kasubag selesai, akan dilanjutkan dengan proses disposisi SPJ oleh BPP, BPP akan menerima notifikasi masuknya SPJ pada sistem.



Gambar 4.99 : Tampilan Notifikasi diposisi SPJ pada *Dashboard* BPP
Kemudian BPP menekan menu disposisi SPJ yang akan menampilkan halaman disposisi SPJ pada BPP. Pada halaman tersebut terdapat SPJ yang harus dilanjutkan atau didisposisikan.



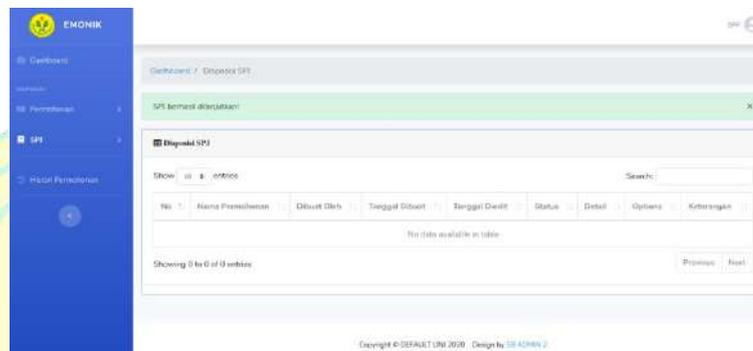
Gambar 4.100 : Tampilan Halaman Menu Disposisi SPJ BPP
Selanjutnya BPP akan menekan tombol lanjutkan untuk melanjutkan SPJ tersebut. Setelah menekan tombol lanjutkan, akan muncul Konfirmasi Lanjutkan SPJ.

Konfirmasi Lanjutkan SPJ!

Catatan Pemohon

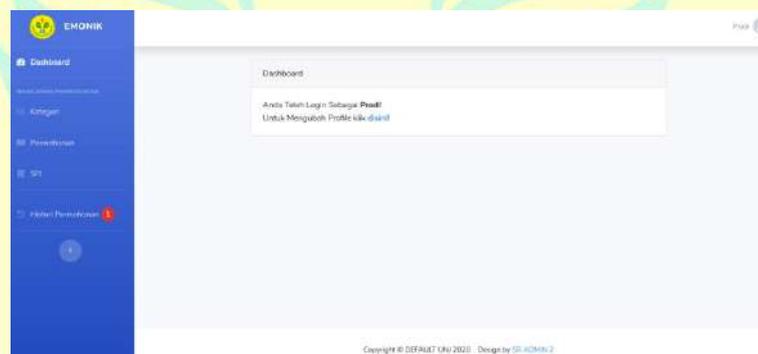
Gambar 4.101 : Tampilan Konfirmasi Lanjutkan SPJ BPP

Setelah BPP selesai mengisi catatan pemohon, selanjutnya menekan tombol Lanjutkan pada Konfirmasi Lanjutkan SPJ, maka akan muncul notifikasi SPJ berhasil di lanjutkan.



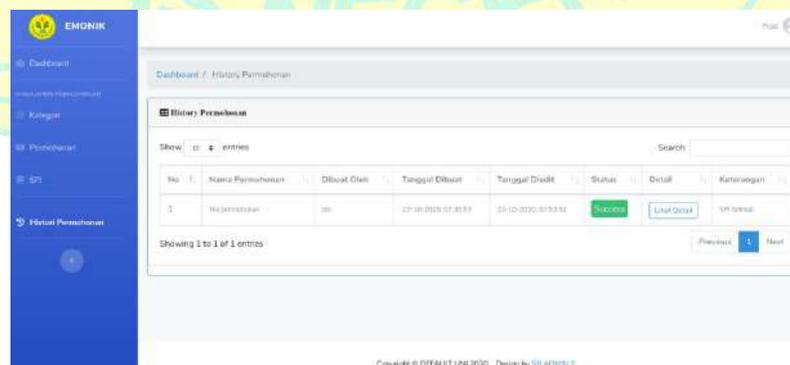
Gambar 4.102 : Tampilan Notifikasi SPJ berhasil di lanjutkan BPP

Kemudian setelah BPP menerima Notifikasi SPJ berhasil di lanjutkan, maka SPJ yang diajukan pemohon telah selesai dan pemohon akan menerima notifikasi SPJ Selesai pada *dashboard*.



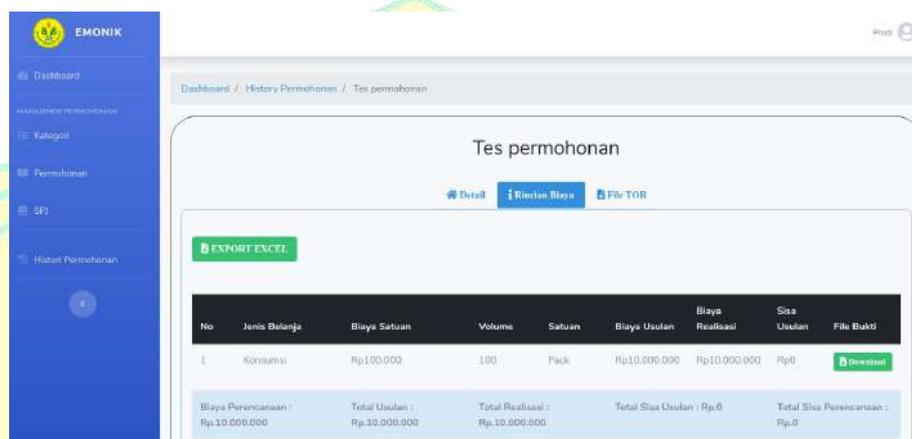
Gambar 4.103 : Tampilan Notifikasi SPJ Selesai pada *Dashboard* Pemohon

Selanjutnya pemohon akan menekan tombol menu Histori Permohonan yang akan menampilkan halaman Histori Permohonan.



Gambar 4.104 : Tampilan Halaman Menu Histori Permohonan

Pada Halaman Menu Histori Permohonan akan menampilkan semua permohonan yang telah selesai dibuat oleh Pemohon. Pemohon menekan tombol lihat detail pada permohonan untuk memeriksa kembali rinciannya dan untuk mengakses fitur *export excel* pada permohonan.



Gambar 4.105 : Tampilan Halaman Menu Rincian pada Histori Permohonan

Untuk mengakses fitur *export excel* pada permohonan yang telah selesai SPJ-nya adalah dengan menekan tombol *export excel* yang ada pada Rincian Biaya. Setelah menekan tombol *export excel* maka akan mengunduh *excel* laporan serap anggaran dana kegiatan permohonan dengan lengkap.

E-Monitoring							
Laporan Keuangan: Tes permohonan							
#	Jenis Belanja	Biaya Satuan	Volume	Satuan	Biaya Usulan	Biaya Realisasi	Sisa Usulan
1	Konsumsi	Rp100.000	100	Pack	Rp10.000.000	Rp0	
Biaya Perencanaan:		Total Usulan:		Total Realisasi:		Total Sisa Usulan:	
Rp10.000.000		Rp10.000.000		Rp10.000.000		Rp0	
						Total Sisa Perencanaan:	
						Rp0	

Gambar 4.106 : Tampilan Laporan Serapan Anggaran pada *Excel*

c. Test

Target pada proses *Test* ini diharapkan sudah mencapai 10% pada iterasi *construction C2* ini dan mencapai target 30% pada target *development plan*. Pada proses *Test* ini akan dibahas artefak lanjutan *Test Plan* dari iterasi sebelumnya.

1. Test Plan

Melanjutkan iterasi sebelumnya. Pada iterasi ini mengubah *planning* testing dan UAT yang sudah ada pada iterasi karena testing dan UAT dapat dilakukan jika pembuatan sistemnya sudah selesai. Penyesuaian *Test Plan* dapat dilihat pada tabel berikut.

Tabel 4.10 : Penyesuaian *Test Plan* pada Iterasi C2

No	<i>Test Plan</i>	Target
1	<i>Alpha Testing.</i>	Iterasi C3
2	<i>Beta Testing.</i>	Iterasi C3-T1
3	<i>User Acceptance Test (UAT).</i>	Iterasi T1-T2

d. Deployment

Target pada proses *Deployment* ini diharapkan sudah mencapai 10% pada iterasi *construction* C2 ini dan mencapai target 20% pada target *development plan*. Pada proses *Deployment* ini akan dibahas artefak lanjutan *Deployment Plan* iterasi sebelumnya.

1. *Deployment Plan*

Melanjutkan iterasi sebelumnya. Pada iterasi ini mengubah *planning deployment* yang sudah ada pada iterasi karena karena pada iterasi ini masih dalam proses implementasi dan baru selesai. Penyesuaian *Deployment Plan* dapat dilihat pada tabel berikut.

Tabel 4.11 : Penyesuaian *Deployment Plan*

No	<i>Deployment Plan</i>	Target
1	Rilis produk <i>Alpha.</i>	Iterasi C3
2	Rilis produk <i>Beta.</i>	Iterasi C3-T1
3	Rilis produk <i>Final.</i>	Iterasi T1-T2

3. Iterasi *construction* C3

Iterasi selanjutnya pada tahap *construction* C3 berdasarkan *development plan* yaitu dibahasnya 20% *Implementation*, 20% *Test* dan 10% *Deployment*.

a. Implementation

Target pada proses *Implementation* ini diharapkan sudah mencapai 20% pada iterasi *construction* C3 ini dan mencapai target 90% pada target *development plan*. Pada proses *Implementation* ini akan dibahas artefak *Build, Component* dan *Implementation Model*.

1. *Build*

Target pada artefak ini dibuat dengan versi operasional dari sistem atau bagian dari sistem yang menunjukkan subset dari kemampuan yang akan disediakan dalam produk akhir. Berikut adalah tabel pencapaian target *build* pada iterasi C3.

Tabel 4.12 : Pencapaian *Integration build plan* pada iterasi C3

No	<i>Build</i>	Target	Penyelesaian
1	Pembuatan sistem <i>login</i> .	Iterasi E2	Selesai
2	Pembuatan <i>role user</i> .	Iterasi C1	Selesai
3	Pembuatan sistem pengajuan permohonan	Iterasi C1	Selesai
4	Pembuatan sistem SPJ	Iterasi C2	Selesai
5	Perbaikan <i>bugs</i>	Iterasi C3-T1	Belum Selesai

2. *Component*

Melanjutkan dari iterasi sebelumnya. Pada iterasi ini sudah dilakukan perbaikan kode pada proses perbaikan *bugs* sistem. Berikut adalah gambar perubahan kode pada beberapa *file* yang diperbaiki sehingga sistem dapat berjalan seperti keinginan penulis.

```

core/app/Http/Controllers/FakultasController.php
@@ -25,7 +25,7 @@ public function __construct() {
25 25     }
26 26
27 27     public function index() {
28 28     - $fakultases = Fakultas::orderBy('updated_at', 'desc')->get();
28 28     + $fakultases = Fakultas::get();
29 29     - return view('fakultas.index_fakultas', compact('fakultases'));
30 30     }
31 31

```

Gambar 4.107 : Perubahan kode pada *FakultasController*

```

1 core/app/Http/Controllers/HomeController.php
2
3 namespace App\Http\Controllers;
4
5 + use Illuminate\Support\Facades\Hash;
6 use Illuminate\Http\Request;
7 use App\User;
8 use Auth;

```

Gambar 4.108 : Perubahan kode pada *HomeController*

```

28 28 public function getUnits($id) {
29 29 - $prodis = Prodi::where('fakultas_id', $id)->pluck("nama","id");
29 29 + $prodis = Prodi::where('fakultas_id', $id)->where('status', 1)->pluck("nama","id");
30 30 return json_encode($prodis);
31 31 }
32 32
33 33 public function index() {
34 34 - $fakultases = Fakultas::get();
34 34 + $fakultases = Fakultas::where('status', 1)->get();
35 35 $prodis = Prodi::where('status', 1)->get();
36 36 - $units = Unit::where('status', 1)->get();
36 36 + $units = Unit::get();
37 37 return view('unit.index_unit', compact('units', 'fakultases', 'prodis'));
38 38 }
39 39
@@ -62,14 +62,19 @@ public function update(Request $request, $id) {
62 62 $units = Unit::findOrFail($id);
63 63 $validator = Validator::make($request->all(), [
64 64 'nama' => 'required|string|max:150',
65 65 - 'fakultas' => 'required',
66 66 - 'prodi' => 'required',
67 67 'status' => 'required|in:1,0',
68 68 ], [
69 69 'nama.required' => 'Nama harus diisi',
70 70 'nama.max' => 'Nama maksimal 150 huruf',
71 71 - 'fakultas.required' => 'Fakultas harus diisi',
72 72 - 'prodi.required' => 'Prodi harus diisi',
73 73 'status.required' => 'status harus diisi',
74 74 ]);

```

Gambar 4.109 : Perubahan kode pada *UnitController*

```

1 core/resources/views/prodi/_edit_prodi.blade.php
2
3 <div class="col-md-6">
4 <select name="fakultas" id="fakultas" class="form-control">
5 <option value="" {!! ($prodi->fakultas_id) ? "selected='selected'" : "" !!}>Tidak Punya Fakultas</option>
6 - <option value="" {!! ($prodi->fakultas_id) ? "selected='selected'" : "" !!}>Tidak Punya Fakultas</option>
6 + <!--><option value="" {!! ($prodi->fakultas_id) ? "selected='selected'" : "" !!}>Tidak Punya Fakultas</option>
```

```

7 core/app/Http/Controllers/UserController.php
24 24 @@ -24,12 +24,11 @@ public function __construct() {
25 25     public function index() {
26 26         $users = User::where('id', '!=', 1)->get();
27 -         $permissions = Permission::where('id', '!=', 1)->get();
28 -         return view('user.index_user', compact('users', 'permissions'));
29 +         return view('user.index_user', compact('users'));
30 29     }
31 30     public function create() {
32 -         $permissions = Permission::where('id', '!=', 1)->get();
33 +         $permissions = Permission::where('id', '!=', 1)->where('status', 1)->get();
34 32         $units = Unit::get();
35 33         return view('user.create_user', compact('permissions', 'units'));
36 34     }
37 35     public function store(Request $request)
38 36     {
39 37     }
40 38     public function edit($id) {
41 39         $permissions = Permission::where('id', '!=', 1)->get();
42 +         $permissions = Permission::where('id', '!=', 1)->where('status', 1)->get();
43 40         $units = Unit::get();
44 41         $user = User::findOrFail($id);
45 42         return view('user.edit_user', compact('permissions', 'units', 'user'));
46 43     }
47 44 }

```

Gambar 4.111 : Perubahan kode pada *UsersController*

Selain memperbaiki *bugs*, pada iterasi ini juga dilakukan penghapusan *footer*. Berikut adalah gambar perubahan kode pada beberapa *file* untuk melakukan penghapusan *footer*.

```

4 core/resources/views/auth/login.blade.php
71 71 </div>
72 72 <div class="text-center p-t-140">
73 73 <footer class="sticky-footer">
74 - <div class="text-center">
75 + <div class="text-center">
76 75 <div class="container my-auto">
77 76 <div class="copyright text-center my-auto">
78 77 <span style="color:#255697"><i class="fa fa-heart" aria-hidden="true"></i></span> Development by DEFAULT UNJ
79 78 Design by <a href="https://colorlib.com/wp/template/login-form-v1/">colorlib</a>
80 79 </div>
81 80 </div>
82 - </footer>
83 + </footer>
84 81 </div>
85 82 </form>

```

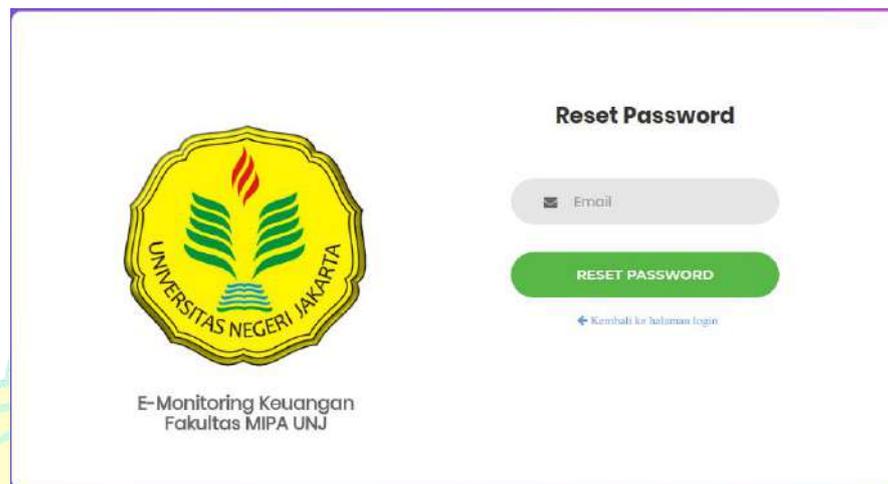
Gambar 4.112 : Perubahan kode pada file *login.blade*

```

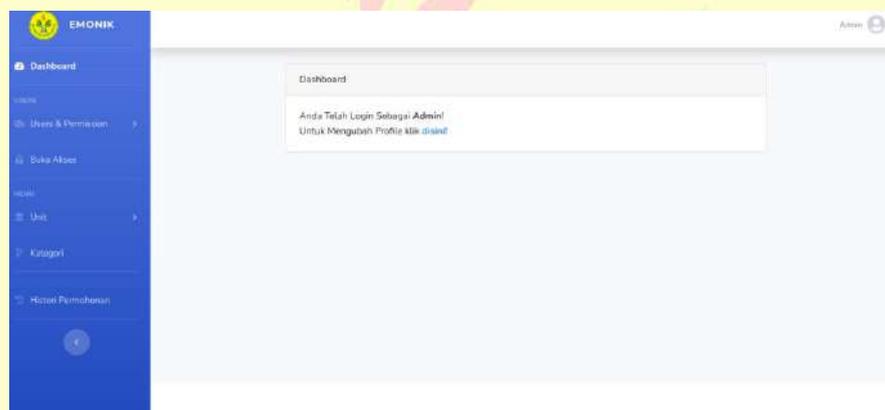
4 core/resources/views/auth/passwords/email.blade.php
80 80 </div>
81 81 <div class="text-center p-t-140">
82 82 <div class="text-center">
83 - <div class="text-center">
84 + <div class="text-center">
85 83 <div class="container my-auto">
86 84 <div class="copyright text-center my-auto">
87 85 <span style="color:#255697"><i class="fa fa-heart" aria-hidden="true"></i></span> Development by DEFAULT UNJ
88 86 Design by <a href="https://colorlib.com/wp/template/login-form-v1/">colorlib</a>
89 87 </div>
90 88 </div>
91 - </div>
92 + </div>
93 89 </form>
94 90 </div>
95 91 </div>
96 92 </form>

```

Gambar 4.113 : Perubahan kode pada file *email.blade*



Gambar 4.117 : Tampilan Halaman *Reset Password* tanpa footer



Gambar 4.118 : Tampilan Halaman *Dashboard* Admin tanpa footer

b. *Test*

Target pada proses *Test* ini diharapkan sudah mencapai 20% pada iterasi *construction C3* ini dan mencapai target 50% pada target *development plan*. Pada proses *Test* ini akan dibahas artefak *Test script* dan *Test log* dari iterasi sebelumnya.

1. *Test script*

Target pada artefak ini adalah membuat *test script*. *Test script* dapat berupa instruksi tekstual terdokumentasi yang dijalankan secara manual atau instruksi yang dapat dibaca komputer yang memungkinkan eksekusi tes otomatis. Berikut adalah tabel *Test script alpha testing* yang dilakukan manual oleh penulis pada tanggal 16 - 22 Oktober 2020.

Tabel 4.13 : Test script alpha testing

No	Test Deskripsi	Result	Notes
1	Fitur <i>login</i> dan <i>logout</i> pada semua aktor.	<i>Pass</i>	-
2	Fitur mengubah data diri pada semua aktor.	<i>Pass</i>	-
3	Fitur mengelola data <i>user</i> dan <i>permission</i> pada aktor Admin.	<i>Fail</i>	Filter data <i>user</i> dan <i>permission</i> tidak aktif.
4	Fitur membuka akses permohonan untuk pemohon pada aktor Admin.	<i>Pass</i>	-
5	Fitur mengelola data unit, fakultas dan prodi pada aktor Admin.	<i>Fail</i>	Filer data unit dan fakultas serta edit prodi tidak aktif
6	Fitur mengelola data kategori pada aktor Admin.	<i>Pass</i>	-
7	Fitur melihat kategori permohonan untuk anda pada aktor Pemohon.	<i>Pass</i>	-
8	Fitur mengelola pengajuan permohonan pada aktor Pemohon.	<i>Pass</i>	-
9	Fitur mengelola data rincian permohonan pada aktor Pemohon.	<i>Pass</i>	-
10	Fitur mengelola pengajuan SPJ pada aktor Pemohon.	<i>Pass</i>	-
11	Fitur mengelola data bukti rincian permohonan untuk SPJ pada aktor Pemohon.	<i>Pass</i>	-
12	Fitur menerima surel yang berkaitan dengan kegiatan pengajuan permohonan dan SPJ pada Pemohon.	<i>Pass</i>	-

No	Test Deskripsi	Result	Notes
13	Fitur mengelola (melanjutkan) disposisi permohonan pada aktor WD 2.	Pass	-
14	Fitur menerima surel yang berkaitan dengan kegiatan disposisi permohonan pada aktor WD 2 dan PPK.	Pass	-
15	Fitur mengelola data kegiatan permohonan pada aktor PPK.	Pass	-
16	Fitur mengelola disposisi permohonan (melanjutkan atau menolak) pada aktor PPK.	Pass	-
17	Fitur mengelola disposisi permohonan (melanjutkan atau menolak) pada aktor Kasubag.	Pass	-
18	Fitur mengelola (melanjutkan atau menolak) disposisi SPJ pada aktor Kasubag.	Pass	-
19	Fitur menerima surel yang berkaitan dengan kegiatan disposisi permohonan dan SPJ pada aktor Kasubag.	Pass	-
20	Fitur mengelola (melanjutkan) disposisi permohonan pada aktor BPP.	Pass	-
21	Fitur mengelola (melanjutkan) disposisi SPJ pada aktor BPP.	Pass	-
22	Fitur menerima surel yang berkaitan dengan kegiatan disposisi permohonan dan SPJ pada aktor BPP.	Pass	-
23	Fitur melihat rekap histori permohonan termasuk <i>export excel</i> pada permohonan yang selesai SPJ-nya pada semua aktor.	Pass	-

2. *Test log*

Target pada artefak ini adalah kumpulan *output* mentah yang diambil selama eksekusi unik dari satu atau beberapa pengujian, biasanya mewakili *output* yang dihasilkan. Berikut adalah tabel *Test log alpha testing* yang merupakan penyelesaian masalah yang terdapat pada artefak *Test Scripts*.

Tabel 4.14 : *Test log alpha testing*

No	<i>Test Deskripsi</i>	<i>Errors</i>	<i>Level Erros</i>	<i>Restore Errors</i>
1	Fitur mengelola data <i>user</i> dan <i>permission</i> pada aktor Admin.	Filter data <i>user</i> dan <i>permission</i> tidak aktif.	<i>Script Failures</i>	22 Oktober 2020
2	Fitur mengelola data unit, fakultas dan prodi pada aktor Admin.	Filer data unit dan fakultas serta edit prodi tidak aktif	<i>Script Failures</i>	22 Oktober 2020

c. *Deployment*

Target pada proses *Deployment* ini diharapkan sudah mencapai 10% pada iterasi *construction C3* ini dan mencapai target 30% pada target *development plan*. Pada proses *Deployment* ini akan dibahas artefak *Realease notes*.

1. *Realease notes*

Artefak ini dibuat untuk menjelaskan rilis tersebut. *Realease notes* mengidentifikasi perubahan dan *bugs* yang diketahui dalam versi unit *build* atau penerapan yang telah tersedia untuk digunakan. Untuk mendeskripsikan fitur baru utama dan perubahan dalam rilis. *Realease notes* juga harus menjelaskan *bugs* dan batasan yang diketahui atau solusi untuk menggunakan produk. pada iterasi *construction C3* telah dibuat produk *alpha* dan telah melakukan *alpha testing* agar produk tersebut dapat berfungsi sesuai dengan yang diharapkan. Berikut adalah tabel *Realease notes* pada iterasi *construction C3*.

Tabel 4.15 : *Realease notes* pada iterasi *construction* C3

No	<i>Product Version</i>	<i>Bugs</i>	<i>What's New</i>	<i>Realease Date</i>
1	Sistem Informasi <i>E-monitoring</i> Keuangan Berbasis <i>Website</i> di FMIPA UNJ <i>alpha version 1.</i>	- Filter data <i>user</i> dan <i>permission</i> tidak aktif. - Filer data unit dan fakultas serta edit prodi tidak aktif.	-	16 Oktober 2020
2	Sistem Informasi <i>E-monitoring</i> Keuangan Berbasis <i>Website</i> di FMIPA UNJ <i>alpha version 2.</i>	-	<i>Bug fixed</i> <i>on alpha</i> <i>version 1.</i>	22 Oktober 2020

D. Tahap *Transition*

Pada tahap *transition* dilakukan proses RUP yaitu berdasarkan iterasi *transition* yang merupakan iterasi terakhir, terdapat 2 iterasi yaitu iterasi T1 dan T2. Pada subbab ini akan dilakukan pembahasan tiap iterasi pada tahap *transition*.

1. Iterasi *transition* T1

Iterasi selanjutnya pada tahap *transition* T1 berdasarkan *development plan* yaitu dibahasnya 10% *Implementation*, 30% *Test* dan 35% *Deployment*.

a. *Implementation*

Target pada proses *Implementation* ini diharapkan sudah mencapai 10% pada iterasi *transition* T1 ini dan mencapai target 100% pada target *development plan* yang artinya selesai semua proses pada proses *Implementation* ini dan akan menyesuaikan penambahan sesuai kebutuhan. Pada proses *Implementation* ini akan dibahas pembahasan lanjutan dari beberapa artefak *Build*, *Component* dan *Implementation Model* yang diharapkan akan selesai pada iterasi ini.

1. Build

Melanjutkan dari iterasi sebelumnya. Pada iterasi ini sudah dilakukan Perbaikan *bugs* pada tabel Pencapaian *Integration build plan Final* yang diharapkan sudah selesai proses pada artefak *build* pada iterasi *transition T1*. Berikut adalah tabel pencapaian target *build Final*.

Tabel 4.16 : Pencapaian *Integration build plan Final*

No	Build	Target	Penyelesaian
1	Pembuatan sistem <i>login</i> .	Iterasi E2	Selesai
2	Pembuatan <i>role user</i> .	Iterasi C1	Selesai
3	Pembuatan sistem pengajuan permohonan	Iterasi C1	Selesai
4	Pembuatan sistem SPJ	Iterasi C2	Selesai
5	Perbaikan <i>bugs</i>	Iterasi C3-T1	Selesai

2. Component

Melanjutkan dari iterasi sebelumnya. Pada iterasi ini dilakukan perbaikan kode pada *beta testing* dan penambahan fitur seperti pengajuan proker. Berikut adalah beberapa gambar pilihan perubahan kode dan penambahan *file* sehingga sistem dapat berjalan seperti keinginan penulis, dan untuk melihat semua perubahan secara lengkap dapat dilihat pada *commit fixed beta test bug* pada github <https://github.com/agusgokasi/emonik>.

```

15 core/app/Http/Controllers/RincianController.php
10 10 use File;
11 11 use Illuminate\Http\Request;
12 12 use Redirect;
13 13 - use App\Kategori;
14 14 + //use App\Kategori;
15 15 use App\Magang;
16 16 use App\Unit;
17 17 use App\Permohonan;
18 18
19 19 public function fileBukti(Request $request, $id)
20 20 {
21 21     return back()-withErrors($validator->withInput(['tab'=>'profile'])-with('error_code', $id);
22 22 }
23 23 //file
24 24 - if(!file_exists($file.'/'.$rincian->file)){
25 25     unlink(public_path($file.'/'.$rincian->file));
26 26 - }
27 27 + //if(!file_exists($file.'/'.$rincian->file)){
28 28 + //unlink(public_path($file.'/'.$rincian->file));
29 29 + //}
30 30 $filename = $request->file('file');
31 31 $file = time().rand(1000,9999).'.', $filename->getClientOriginalExtension();
32 32 $request->file('file')->move(public_path($file), $file);
33 33
34 34 public function editBukti(Request $request, $id)
35 35 {
36 36     return back()-withErrors($validator->withInput(['tab'=>'profile'])-with('error_code', $id);
37 37 }

```

Gambar 4.119 : Perubahan kode pada *RincianController*

```

357 core/app/Http/Controllers/PermohonanController.php |
↑  @@ -8,12 +8,15 @@
8 8 use File;
9 9 use Illuminate\Http\Request;
10 10 use Redirect;
11 - use App\Kategori;
11 + // use App\Kategori;
12 12 use App\Kegiatan;
13 13 use App\Unit;
14 14 use App\Permohonan;
15 15 use App\Rincian;
16 + use App\Notifications\SubmitProker;
16 17 use App\Notifications\SubmitPermohonan;
18 + use App\Notifications\Dis1Permohonan;
19 + use App\Notifications\Dis2Permohonan;
17 20 use App\Notifications\Dt2Permohonan;
18 21 use App\Notifications\Dt3Permohonan;
19 22 use App\Notifications\Dp3Permohonan;
@@ -30,10 +33,107 @@ public function __construct() {
30 33 });
31 34 }
32 35
33 - public function index() {
36 + public function indexProker() {
34 37     $user = Auth::user();

```

Gambar 4.120 : Perubahan kode pada *PermohonanController*

```

95 core/app/Http/Controllers/KegiatanController.php |
↑  @@ -8,9 +8,12 @@
8 8 use File;
9 9 use Illuminate\Http\Request;
10 10 use Redirect;
11 - use App\Kategori;
11 + // use App\Kategori;
12 12 use App\Kegiatan;
13 13 use App\Unit;
14 + use App\Notifications\SubmitProker;
15 + use App\Notifications\TerimaProker;
16 + use App\Notifications\TolakProker;
14 17 use Illuminate\Support\Facades\Validator;
15 18
16 19 class KegiatanController extends Controller
@@ -25,25 +28,85 @@ public function __construct() {
25 28 }
26 29
27 30 public function index() {
28 - $kegiatan = Kegiatan::orderBy('updated_at', 'desc')->get();
29 - $kategori = Kategori::where('status', 1)->get();
30 - $units = Unit::where('status', 1)->get();
31 - return view('kegiatan.index_kegiatan', compact('kegiatan', 'kategori', 'units'));
31 + $kegiatan = Kegiatan::where('status', 1)->orderBy('updated_at', 'desc')->get();
32 + // $kategori = Kategori::where('status', 1)->get();
33 + // $units = Unit::where('status', 1)->get();
34 + return view('kegiatan.index_kegiatan', compact('kegiatan'));

```

Gambar 4.121 : Perubahan kode pada *KegiatanController*

```

21 core/database/migrations/2019_09_28_150232_create_permohonans_table.php
... .. 00 -18,16 +18,15 @@ public function up()
18 18 // $table->integer('kategori_id')->nullable();
19 19 $table->integer('kegiatan_id')->nullable();
20 20 $table->string('nama');
21 - $table->string('slug', 250)->unique();
21 + $table->string('slug', 270)->unique();
22 22 $table->string('pemohon');
23 - $table->string('nomorinduk');
24 - $table->text('latarbelakangkegiatan');
25 - $table->text('tujuankegiatan');
26 - $table->string('tempatkegiatan');
27 - $table->string('tanggalkegiatan');
28 - $table->text('pesertakegiatan');
29 - $table->text('strategipencapaiankeluaran');
30 - $table->text('susunanpanitia');
23 + // $table->string('nomorinduk');
24 + $table->text('latarbelakang');
25 + $table->text('tujuan');
26 + $table->text('ruanglingkup');
27 + $table->text('waktupencapaian');
28 + $table->text('luaran');
29 + $table->text('pembiayaan');
31 30 $table->string('filetor');
32 31 // $table->string('filespj')->nullable();
33 32 $table->smallInteger('status')->default(0);

```

Gambar 4.122 : Perubahan kode pada *DatabasePermohonan*

```

141 core/resources/views/permohonan/proker_permohonan.blade.php
... .. 00 -6,0 +1,141 @@
1 + @extends('layouts.backend')
2 +
3 + @section('content')
4 + <!-- Breadcrumbs -->
5 + <ol class="breadcrumb">
6 + <li class="breadcrumb-item">
7 + <a href="{{ route('home') }}">Dashboard</a>
8 + </li>
9 + <li class="breadcrumb-item active">
10 + Program Kerja
11 + </li>
12 + </ol>
13 +
14 + @include('layouts.pesan')
15 + <button class="btn btn-primary" data-toggle="modal" data-target="#m-c" ui-toggle-class="bounce" ui-target="#animate">
16 + <i class="fa fa-plus"></i>
17 + Buat Proker
18 + </button>
19 + @include('permohonan._proker_buat')
20 + <br><br>
21 + <div class="card border border-info">
22 + <div class="card mb-3">
23 + <div class="card-header">
24 + <i class="fas fa-table">&nbsp;&nbsp;&nbsp;Program Kerja Untuk Anda</i>
25 + </div>

```

Gambar 4.123 : Penambahan *file PermohonanProker*

```

78 core/resources/views/permohonan/_proker_buat.blade.php
...  @0 -0,8 +1,78 @0
1 + <!-- modal -->
2 + <div id="m-0" class="modal fade" data-backdrop="true">
3 +   <div class="modal-dialog" id="animate">
4 +     <div class="modal-content">
5 +       <div class="modal-header">
6 +         <h5 class="modal-title">Buat Proker</h5>
7 +       </div>
8 +       <form method="post" role="form" action="{{ route('prokercreate') }}" enctype="multipart/form-data" id="myForm">
9 +         {{ csrf_field() }}
10 +        <div class="modal-body text-left p-lg">
11 +          <!-- nama -->
12 +          <div class="form-group row">
13 +            <label for="nama" class="col-md-4 col-form-label text-md-left">{{ __('Nama Kegiatan') }}</label>
14 +            <div class="col-md-8">
15 +              <input id="nama" type="text" class="form-control"{{ $errors->has('nama') ? ' is-invalid' : '' }} name="nama">
16 +
17 +              @if ($errors->has('nama'))
18 +                <span class="invalid-feedback" role="alert">
19 +                  <strong>{{ $errors->first('nama') }}</strong>
20 +                </span>
21 +              @endif
22 +            </div>
23 +          </div>
24 +          <!-- hulan -->
25 +          <div class="form-group row">
26 +            <label for="hulan" class="col-md-4 col-form-label text-md-left">{{ __('Hulan') }}</label>

```

Gambar 4.124 : Penambahan file *BuatProker*

```

78 core/resources/views/permohonan/_proker_edit.blade.php
...  @0 -0,8 +1,78 @0
1 + <!-- modal -->
2 + <div id="m-0-{{ $kegiatan->id }}" class="modal fade" data-backdrop="true">
3 +   <div class="modal-dialog" id="animate">
4 +     <div class="modal-content">
5 +       <div class="modal-header">
6 +         <h5 class="modal-title">Edit Proker</h5>
7 +       </div>
8 +       <form method="post" role="form" action="{{ route('prokerupdate', ['id'=>$kegiatan->id]) }}" enctype="multipart
9 +         form-data">
10 +        {{ csrf_field() }}
11 +        <div class="modal-body text-left p-lg">
12 +          <!-- nama -->
13 +          <div class="form-group row">
14 +            <label for="nama" class="col-md-4 col-form-label text-md-left">{{ __('Nama Kegiatan') }}</label>
15 +            <div class="col-md-8">
16 +              <input id="nama" type="text" class="form-control"{{ $errors->has('nama') ? ' is-invalid' : '' }} name="nama">
17 +
18 +              @if ($errors->has('nama'))
19 +                <span class="invalid-feedback" role="alert">
20 +                  <strong>{{ $errors->first('nama') }}</strong>
21 +                </span>
22 +              @endif
23 +            </div>
24 +          </div>
25 +          <!-- hulan -->
26 +          <div class="form-group row">

```

Gambar 4.125 : Penambahan file *EditProker*

```

25 core/resources/views/permohonan/_proker_delete.blade.php
...  @0 -0,8 +1,25 @0
1 + <!-- modal -->
2 + <div id="m-{{ $kegiatan->id }}" class="modal fade" data-backdrop="true">
3 +   <div class="modal-dialog" id="animate">
4 +     <div class="modal-content">
5 +       <div class="modal-header">
6 +         <h5 class="modal-title">Konfirmasi</h5>
7 +       </div>
8 +       <div class="modal-body text-center p-lg">
9 +         <p>
10 +           Anda yakin ingin menghapus?
11 +         <br>
12 +         <strong>[ {{ $kegiatan->nama }} ]</strong> <br>
13 +         <small>*tindakan ini akan menghapus semua id proker yang berhubungan!</small>
14 +       </p>
15 +     </div>
16 +     <div class="modal-footer">
17 +       <button type="button" class="btn btn-outline-secondary p-x-md"
18 +         data-dismiss="modal">Tidak</button>
19 +       <a href="{{ route('prokerDestroy', ['id'=>$kegiatan->id]) }}"
20 +         class="btn btn-danger p-x-md">Ya</a>
21 +     </div>
22 +   </div><!-- /.modal-content -->
23 + </div>
24 + </div>
25 + <!-- /.modal -->

```

Gambar 4.126 : Penambahan file *DeleteProker*

```

25 core/resources/views/permohonan/_proker_submit.blade.php
... 00 -0,0 +1,25 00
1 +
2 + <!-- modal -->
3 + <div id="m-s-{{ $kegiatan->id }}" class="modal fade" data-backdrop="true">
4 + <div class="modal-dialog" id="animate">
5 + <div class="modal-content">
6 + <div class="modal-header">
7 + <h5 class="modal-title">Konfirmasi Submit Proker!</h5>
8 + </div>
9 + <div class="modal-body text-center p-1g">
10 + <p>
11 +     Anda yakin ingin mensubmit?
12 + <br>
13 + <strong>[ {{ $kegiatan->nama }} ]</strong> <br>
14 + </p>
15 + </div>
16 + <div class="modal-footer">
17 + <button type="button" class="btn btn-outline-secondary p-x-md"
18 +     data-dismiss="modal">Tidak</button>
19 + <a href="{{ route('prokerSubmit', ['id'=>$kegiatan->id]) }}"
20 +     class="btn btn-success p-x-md">&nbsp;Submit&nbsp;</a>
21 + </div>
22 + </div> <!-- /.modal-content -->
23 + </div>
24 + </div>
25 + <!-- /.modal -->

```

Gambar 4.127 : Penambahan file *SubmitProker*

```

122 core/resources/views/kegiatan/index_proker.blade.php
... 00 -0,0 +1,122 00
1 + @extends('layouts.backend')
2 +
3 + @section('content')
4 + <!-- Breadcrumbs -->
5 + <ol class="breadcrumb">
6 + <li class="breadcrumb-item">
7 + <a href="{{ route('home') }}">Dashboard</a>
8 + </li>
9 + <li class="breadcrumb-item active">
10 +     Program Kerja
11 + </li>
12 + </ol>
13 +
14 + @include('layouts.pesan')
15 +
16 + <!-- button class="btn btn-primary" data-toggle="modal" data-target="#m-c" ui-toggle-class="bounce" ui-target="#animate">
17 + <i class="fa fa-plus"></i>
18 + Buat Kegiatan
19 + </button>
20 + @include('kegiatan_buat_kegiatan')
21 + <br><br> --}}
22 + <div class="card border border-info">
23 + <div class="card-body">
24 + <div class="card-header">
25 + <i class="fas fa-table">&nbsp;Manajemen Program Kerja</i>
26 + </div>

```

Gambar 4.128 : Penambahan file *ManajemenProker*

```

76 core/resources/views/kegiatan/_terima_proker.blade.php
... 00 -0,0 +1,76 00
1 + <!-- modal -->
2 + <div id="m-to-{{ $kegiatan->id }}" class="modal fade" data-backdrop="true">
3 + <div class="modal-dialog" id="animate">
4 + <div class="modal-content">
5 + <div class="modal-header">
6 + <h5 class="modal-title">Konfirmasi Terima Proker</h5>
7 + </div>
8 + <form method="post" role="form" action="{{ route('prokerTerima', ['id'=>$kegiatan->id]) }}" enctype="multipart
9 + <form method="post" role="form" action="{{ route('prokerTerima', ['id'=>$kegiatan->id]) }}" enctype="multipart
10 + <form method="post" role="form" action="{{ route('prokerTerima', ['id'=>$kegiatan->id]) }}" enctype="multipart
11 + <div class="modal-body text-left p-1g">
12 + <div class="form-group">
13 + <div class="form-group row">
14 + <div class="form-group row">
15 + <div class="form-group row">
16 + <div class="form-group row">
17 + <div class="form-group row">
18 + <div class="form-group row">
19 + <div class="form-group row">
20 + <div class="form-group row">
21 + <div class="form-group row">
22 + <div class="form-group row">
23 + <div class="form-group row">
24 + <div class="form-group row">
25 + <div class="form-group row">
26 + <div class="form-group row">
27 + <div class="form-group row">
28 + <div class="form-group row">
29 + <div class="form-group row">
30 + <div class="form-group row">
31 + <div class="form-group row">
32 + <div class="form-group row">
33 + <div class="form-group row">
34 + <div class="form-group row">
35 + <div class="form-group row">
36 + <div class="form-group row">
37 + <div class="form-group row">
38 + <div class="form-group row">
39 + <div class="form-group row">
40 + <div class="form-group row">
41 + <div class="form-group row">
42 + <div class="form-group row">
43 + <div class="form-group row">
44 + <div class="form-group row">
45 + <div class="form-group row">
46 + <div class="form-group row">
47 + <div class="form-group row">
48 + <div class="form-group row">
49 + <div class="form-group row">
50 + <div class="form-group row">
51 + <div class="form-group row">
52 + <div class="form-group row">
53 + <div class="form-group row">
54 + <div class="form-group row">
55 + <div class="form-group row">
56 + <div class="form-group row">
57 + <div class="form-group row">
58 + <div class="form-group row">
59 + <div class="form-group row">
60 + <div class="form-group row">
61 + <div class="form-group row">
62 + <div class="form-group row">
63 + <div class="form-group row">
64 + <div class="form-group row">
65 + <div class="form-group row">
66 + <div class="form-group row">
67 + <div class="form-group row">
68 + <div class="form-group row">
69 + <div class="form-group row">
70 + <div class="form-group row">
71 + <div class="form-group row">
72 + <div class="form-group row">
73 + <div class="form-group row">
74 + <div class="form-group row">
75 + <div class="form-group row">
76 + <div class="form-group row">

```

Gambar 4.129 : Penambahan file *TerimaProker*

```

68 core/resources/views/kegiatan/_tolak_proker.blade.php
... @0 -0,0 +1,00 @0
1 + <!-- Modal -->
2 + <div id="m-to-{{ $kegiatan->id }}" class="modal fade" data-backdrop="true">
3 + <div class="modal-dialog" id="animate">
4 + <div class="modal-content">
5 + <div class="modal-header">
6 + <h5 class="modal-title">Konfirmasi Tolak Proker</h5>
7 + </div>
8 + <form method="post" role="form" action="{ route('prokerTolak', ['id'=>$kegiatan->id]) }" enctype="multipart
9 + { form_fields }">
10 + <div class="modal-body text-left p-lg">
11 + {!-- nama --}
12 + <div class="form-group row">
13 + <label for="nama" class="col-md-4 col-form-label text-md-left">{{ __('Nama Kegiatan') }}</label>
14 + <div class="col-md-8">
15 + <input id="nama" type="text" class="form-control" {{ $errors->has('nama') ? ' is-invalid' : ''
16 +
17 + <!-- Errors -->
18 + <span class="invalid-feedback" role="alert">
19 + <strong>{{ $errors->first('nama') }}</strong>
20 + </span>
21 + </div>
22 + </div>
23 + {!-- button --}
24 + <div class="form-group row">

```

Gambar 4.130 : Penambahan file *TolakProker*

```

40 core/resources/views/kegiatan/index_kegiatan.blade.php
... @0 -7,22 +7,22 @0
7 + <a href="{ route('home') }">Dashboard</a>
8 + </li>
9 + <li class="breadcrumb-item active">
10 + Kegiatan
11 + </li>
12 + </ol>
13 + </div>
14 + @include('layouts.pesan')
15 +
16 + <button class="btn btn-primary" data-toggle="modal" data-target="#m-c" ui-toggle-class="bounce" ui-target="animate">
17 + <span class="fa fa-plus"></span>
18 + Buat Kegiatan
19 + </button>
20 + @include('kegiatan_buat_kegiatan')
21 + </div>
22 + @include('kegiatan_buat_kegiatan')
23 + </div>
24 + <div class="card border border-info">
25 + <div class="card-body">
26 + <div class="card-header">
27 + <h5 class="fas fa-table">Buat Kegiatan</h5>
28 + <div class="fas fa-table">Kalendar Kegiatan</div>
29 + </div>

```

Gambar 4.131 : Perubahan kode pada *KalendarKegiatan*

```

65 core/app/Notifications/SubmitProker.php
... @0 -0,0 +1,05 @0
1 + <?php
2 +
3 + namespace App\Notifications;
4 +
5 + use Illuminate\Bus\Queueable;
6 + use Illuminate\Notifications\Notification;
7 + use Illuminate\Contracts\Queue\ShouldQueue;
8 + use Illuminate\Notifications\Messages\MailMessage;
9 +
10 + class SubmitProker extends Notification
11 + {
12 +     use Queueable;
13 +
14 +     /**
15 +      * Create a new notification instance.
16 +      *
17 +      * @return void
18 +      */
19 +     public function __construct()
20 +     {
21 +         //
22 +     }
23 +
24 +     /**
25 +      * Get the notification's delivery channels.
26 +      *

```

Gambar 4.132 : Penambahan file *NotifSubmitProker*

```
65 core/app/Notifications/TerimaProker.php
... .. @@ -0,0 +1,65 @@
1 + <?php
2 +
3 + namespace App\Notifications;
4 +
5 + use Illuminate\Bus\Queueable;
6 + use Illuminate\Notifications\Notification;
7 + use Illuminate\Contracts\Queue\ShouldQueue;
8 + use Illuminate\Notifications\Messages\MailMessage;
9 +
10 + class TerimaProker extends Notification
11 + {
12 +     use Queueable;
13 +
14 +     /**
15 +      * Create a new notification instance.
16 +      *
17 +      * @return void
18 +      */
19 +     public function __construct()
20 +     {
21 +         //
22 +     }
23 +
24 +     /**
25 +      * Get the notification's delivery channels.
26 +      *
```

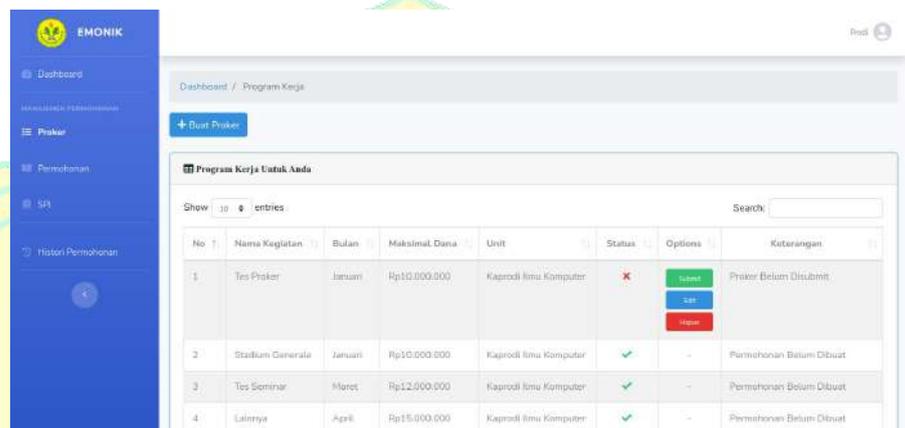
Gambar 4.133 : Penambahan file *NotifTerimaProker*

```
65 core/app/Notifications/TolakProker.php
... .. @@ -0,0 +1,65 @@
1 + <?php
2 +
3 + namespace App\Notifications;
4 +
5 + use Illuminate\Bus\Queueable;
6 + use Illuminate\Notifications\Notification;
7 + use Illuminate\Contracts\Queue\ShouldQueue;
8 + use Illuminate\Notifications\Messages\MailMessage;
9 +
10 + class TolakProker extends Notification
11 + {
12 +     use Queueable;
13 +
14 +     /**
15 +      * Create a new notification instance.
16 +      *
17 +      * @return void
18 +      */
19 +     public function __construct()
20 +     {
21 +         //
22 +     }
23 +
```

Gambar 4.134 : Penambahan file *NotifTolakProker*

3. Implementation Model

Pada iterasi ini sudah dilakukan perbaikan *bugs* sistem untuk *beta testing* serta beberapa penambahan fitur seperti pengajuan proker. Berikut adalah tampilan beberapa halaman untuk fitur baru proker dan kalender kegiatan.



No	Nama Kegiatan	Bulan	Maksimal Dana	Unit	Status	Options	Keterangan
1	Tes Proker	Januari	Rp10.000.000	Kaprodi Ilmu Komputer	✖	Submit, Edit, Hapus	Proker Belum Disubmit
2	Stadlum Generala	Januari	Rp10.000.000	Kaprodi Ilmu Komputer	✔	-	Permohonan Belum Dibuat
3	Tes Seminar	Marct	Rp12.000.000	Kaprodi Ilmu Komputer	✔	-	Permohonan Belum Dibuat
4	Lainnya	April	Rp15.000.000	Kaprodi Ilmu Komputer	✔	-	Permohonan Belum Dibuat

Gambar 4.135 : Tampilan Halaman Menu Proker Pemohon



Buat Proker

Nama Kegiatan: Tes Proker

Bulan: Januari

Usulan Dana (Rp): 10.000.000

Tidak Simpan

Gambar 4.136 : Tampilan Modal Buat Proker



Edit Proker

Nama Kegiatan: Tes Proker

Bulan: Januari

Usulan Dana (Rp): 10.000.000

Tidak Simpan

Gambar 4.137 : Tampilan Modal Edit Proker

Konfirmasi

Anda yakin ingin menghapus?
[Tes Proker]

*tindakan ini akan menghapus semua id proker yang berhubungan!

Tidak

Gambar 4.138 : Tampilan Konfirmasi Hapus Proker

Konfirmasi Submit Proker!

Anda yakin ingin mensubmit?
[Tes Proker]

Tidak

Gambar 4.139 : Tampilan Konfirmasi Submit Proker

EMONIK

Dashboard

Proker & Kegiatan

Manajemen Program Kerja

Dashboard / Program Kerja

Manajemen Program Kerja

Show 1 of 1 entries

No	Nama Kegiatan	Bulan	Maksimal Dana	UNIT	Status	Options	Keterangan
1	Tes Proker	Januari	Rp10.000.000	Kaprosdi Ilmu Komputer	<input type="button" value="Terima"/> <input type="button" value="Tidak"/>		Proker sedang di proses

Showing 1 to 1 of 1 entries

Previous

Gambar 4.140 : Tampilan Halaman Menu Manajemen Proker PPK

Konfirmasi Terima Proker

Nama Kegiatan

Bulan

Maksimal Dana (Rp)

Tidak

Gambar 4.141 : Tampilan Konfirmasi Terima Proker

Konfirmasi Tolak Proker

Nama Kegiatan: Tes Proker

Bulan: Januari

Maksimal Dana (Rp): 10.000.000

Alasan Penolakan:

Tidak Tolak

Gambar 4.142 : Tampilan Konfirmasi Tolak Proker

No	Nama Kegiatan	Bulan	Maksimal Dana	Unit	Status	Keterangan
1	Tes Proker	Januari	Rp10.000.000	Kapredil Ilmu Komputer	✓	Permohonan Belum Dibuat
2	Stadlum Generasi	Januari	Rp10.000.000	Kapredil Ilmu Komputer	✓	Permohonan Belum Dibuat
3	Tes Seminar	Maret	Rp12.000.000	Kapredil Ilmu Komputer	✓	Permohonan Belum Dibuat
4	Lainnya	April	Rp15.000.000	Kapredil Ilmu Komputer	✓	Permohonan Belum Dibuat
5	Tes Lainnya	Mei	Rp8.000.000	Kapredil Ilmu Komputer	✓	Permohonan Belum Dibuat

Gambar 4.143 : Tampilan Halaman Menu Kalender Kegiatan PPK

b. Test

Target pada proses *Test* ini diharapkan sudah mencapai 30% pada iterasi *transition T1* ini dan mencapai target 80% pada target *development plan*. Pada proses *Test* ini akan dibahas artefak *Tes Guidelines*, *Test case*, *Test data* dan *Test log*.

1. Test guideliness

Target pada artefak ini adalah membuat *test guideliness* untuk *beta testing*. Dalam proses pengembangan aplikasi sesuai dengan metode *Rational Unified Process (RUP)*, setelah implementasi dilakukan *beta testing*. *Beta testing* pada sistem dilakukan terhadap dua responden yaitu dosen pembimbing 1 dan perwakilan untuk dosen pembimbing 2 yang dilakukan

tanggal 8 - 9 Desember 2020 yang hasil tesnya dapat dilihat pada *file excel* “Beta Testing.xlsx” pada *branch github* iterasi T1. Setiap responden melakukan pengujian terhadap seluruh fungsi sistem. *Beta testing* yang dilakukan menggunakan data dari kuesioner Testing *Website* EMONIK. Berikut langkah-langkah *Beta testing* Sistem Informasi *E-Monitoring* Keuangan berbasis *website* di Fakultas MIPA UNJ:

1. Admin
 - a. Mengelola (menambah, menyunting, dan menghapus) data *user* dan *permission*.
 - b. Membuka akses dalam membuat permohonan yang diajukan pemohon.
 - c. Mengelola (menambah, menyunting, dan menghapus) data unit, fakultas dan prodi.
 - d. Mengelola (menambah, menyunting, dan menghapus) data kategori.
 - e. Melihat rekap histori permohonan termasuk export excel untuk permohonan yang sudah selesai SPJ-nya.
2. Pemohon
 - a. Melihat kategori permohonan untuk anda.
 - b. Mengelola (menambah, menyunting, dan menghapus) pengajuan permohonan.
 - c. Mengelola (menambah, menyunting, dan menghapus) data rincian permohonan pada permohonan yang diajukan.
 - d. Mengelola (menambah, menyunting, dan menghapus) pengajuan SPJ pada permohonan.
 - e. Mengelola (menambah, menyunting, dan menghapus) data bukti rincian permohonan pada SPJ yang diajukan.
 - f. Menerima surel yang berkaitan dengan kegiatan pengajuan permohonan dan SPJ.
 - g. Melihat rekap histori permohonan termasuk export excel untuk permohonan yang sudah selesai SPJ-nya.

3. Wakil Dekan II
 - a. Mengelola (melanjutkan) disposisi permohonan.
 - b. Menerima surel yang berkaitan dengan kegiatan disposisi permohonan.
 - c. Melihat rekap histori permohonan termasuk export excel untuk permohonan yang sudah selesai SPJ-nya.
4. PPK
 - a. Mengelola (menambah, menyunting, dan menghapus) data kegiatan permohonan.
 - b. Mengelola (melanjutkan atau menolak) disposisi permohonan.
 - c. Menerima surel yang berkaitan dengan kegiatan disposisi permohonan.
 - d. Melihat rekap histori permohonan termasuk export excel untuk permohonan yang sudah selesai SPJ-nya.
5. Kasubag Keuangan
 - a. Mengelola (melanjutkan atau menolak) disposisi permohonan.
 - b. Mengelola (melanjutkan atau menolak) disposisi SPJ.
 - c. Menerima surel yang berkaitan dengan kegiatan disposisi permohonan dan SPJ.
 - d. Melihat rekap histori permohonan termasuk export excel untuk permohonan yang sudah selesai SPJ-nya.
6. BPP
 - a. Mengelola (melanjutkan) disposisi permohonan.
 - b. Mengelola (melanjutkan) disposisi SPJ.
 - c. Menerima surel yang berkaitan dengan kegiatan disposisi permohonan dan SPJ.
 - d. Melihat rekap histori permohonan termasuk export excel untuk permohonan yang sudah selesai SPJ-nya.

Pada *test case* dilakukan pengujian fungsional yang digunakan untuk penilaian seluruh komponen berdasarkan pilihan berikut:

1. S : Setuju
2. TS : Tidak Setuju

Untuk *test data* dilakukan pengujian kebergunaan (usability) digunakan skala likert. Skala likert yaitu skala penilaian dengan pilihan skala nilai pada setiap skala yang digunakan untuk mengukur tingkat persetujuan terhadap sesuatu (Maryuliana & M. I., 2016). Skala yang digunakan dengan nilai 1 s.d. 5 dengan perincian sebagai berikut :

- 1 : Sangat Tidak Setuju
- 2 : Tidak Setuju
- 3 : Cukup
- 4 : Setuju
- 5 : Sangat Setuju

Setelah didapatkan seluruh nilai dari pengguna (user) saat pengujian, nilai tersebut dikalkulasi sesuai dengan sistem penilaian berikut:

- Nilai Total

Nilai total yaitu jumlah keseluruhan yang didapatkan dari setiap pertanyaan yang dapat ditulis menjadi:

$$\text{Nilai Total} = (\text{jumlah} \times \text{skorSS}) + (\text{jumlah} \times \text{skorS}) + (\text{jumlah} \times \text{skorC}) + (\text{jumlah} \times \text{skorTS}) + (\text{jumlah} \times \text{skorSTS})$$

- Presentase Kelayakan

Presentase Kelayakan Presentase kelayakan yaitu persentase nilai rata-rata yang didapatkan dari nilai total dibagi skor yang diharapkan. Skor yang diharapkan didapat dari skor maksimal yang dikalikan dengan jumlah responden. Skor maksimal adalah nilai maksimal dari skala likert dikalikan dengan jumlah pertanyaan. Perhitungan tersebut dapat ditulis menjadi:

$$\text{Presentase kelayakan}(\%) = \frac{\text{Nilai Total}}{\text{Skor Diharapkan}} \times 100\%$$

Persentase kelayakan yang telah didapatkan akan dibandingkan dengan skor pada skala likert. Berikut model skala likert:

1. Sangat Kurang Sesuai = 0% - 20%
2. Kurang Sesuai = 21% - 40%
3. Cukup Sesuai = 41% - 60%
4. Sesuai = 61% - 80%
5. Sangat Sesuai = 81% - 100%

2. *Test case*

Target pada artefak ini adalah membuat *test case*. *Test case* adalah sekumpulan input pengujian, kondisi eksekusi, dan hasil yang diharapkan diidentifikasi untuk tujuan membuat evaluasi beberapa aspek tertentu. Berdasarkan hasil pengujian *beta testing* telah disusun tabel Hasil *Test case beta testing*.

Tabel 4.17 : Hasil *Test case beta testing*

No	<i>Test Deskripsi</i>	Jawaban Responden	
		Setuju	Tidak Setuju
1	Fitur <i>login</i> dan <i>logout</i> pada semua aktor.	2	0
2	Fitur mengubah data diri pada semua aktor.	2	0
3	Fitur mengelola data <i>user</i> dan <i>permission</i> pada aktor Admin.	2	0
4	Fitur membuka akses permohonan untuk pemohon pada aktor Admin.	2	0
5	Fitur mengelola data unit, fakultas dan prodi pada aktor Admin.	2	0
6	Fitur mengelola data kategori pada aktor Admin.	2	0
7	Fitur melihat kategori permohonan untuk anda pada aktor Pemohon.	2	0
8	Fitur mengelola pengajuan permohonan pada aktor Pemohon.	2	0
9	Fitur mengelola data rincian permohonan pada aktor Pemohon.	2	0

No	Test Deskripsi	Jawaban Responden	
		Setuju	Tidak Setuju
10	Fitur mengelola pengajuan SPJ pada aktor Pemohon.	2	0
11	Fitur mengelola data bukti rincian permohonan untuk SPJ pada aktor Pemohon.	2	0
12	Fitur menerima surel yang berkaitan dengan kegiatan pengajuan permohonan dan SPJ pada Pemohon.	2	0
13	Fitur mengelola (melanjutkan) disposisi permohonan pada aktor WD 2.	2	0
14	Fitur menerima surel yang berkaitan dengan kegiatan disposisi permohonan pada aktor WD 2 dan PPK.	2	0
15	Fitur mengelola data kegiatan permohonan pada aktor PPK.	2	0
16	Fitur mengelola disposisi permohonan (melanjutkan atau menolak) pada aktor PPK.	2	0
17	Fitur mengelola disposisi permohonan (melanjutkan atau menolak) pada aktor Kasubag.	2	0
18	Fitur mengelola (melanjutkan atau menolak) disposisi SPJ pada aktor Kasubag.	2	0
19	Fitur menerima surel yang berkaitan dengan kegiatan disposisi permohonan dan SPJ pada aktor Kasubag.	2	0
20	Fitur mengelola (melanjutkan) disposisi permohonan pada aktor BPP.	2	0
21	Fitur mengelola (melanjutkan) disposisi SPJ pada aktor BPP.	2	0

No	Test Deskripsi	Jawaban Responden	
		Setuju	Tidak Setuju
22	Fitur menerima surel yang berkaitan dengan kegiatan disposisi permohonan dan SPJ pada aktor BPP.	2	0
23	Fitur melihat rekap histori permohonan termasuk <i>export excel</i> pada permohonan yang selesai SPJ-nya pada semua aktor.	2	0
Total		46	0
Presentase Jawaban		100%	0%

Berdasarkan hasil *test case*, mendapatkan presentase jawaban 100%. Maka dapat dikatakan pada tahap *beta testing* ini sistem emonik berjalan dengan baik.

3. Test data

Target pada artefak ini adalah membuat *test data*. *Test data* adalah kumpulan nilai input pengujian yang digunakan selama pelaksanaan pengujian, dan hasil yang diharapkan direferensikan untuk tujuan perbandingan selama pelaksanaan pengujian. Berdasarkan hasil pengujian *beta testing* telah disusun tabel Hasil *Test data beta testing*.

Tabel 4.18 : Hasil *Test data beta testing*

No	Test Deskripsi	1	2	3	4	5
1	Fitur <i>login</i> dan <i>logout</i> pada semua aktor mudah dimengerti.	0	0	0	2	0
2	Fitur mengubah data diri pada semua aktor mudah dimengerti.	0	0	0	2	0
3	Fitur mengelola data <i>user</i> dan <i>permission</i> pada aktor Admin mudah dimengerti.	0	0	0	2	0
4	Fitur membuka akses permohonan untuk pemohon pada aktor Admin mudah dimengerti.	0	0	0	1	1

No	Test Deskripsi	1	2	3	4	5
5	Fitur mengelola data unit, fakultas dan prodi pada aktor Admin mudah dimengerti.	0	0	0	2	0
6	Fitur mengelola data kategori pada aktor Admin mudah dimengerti.	0	0	0	2	0
7	Fitur melihat kategori permohonan untuk anda pada aktor Pemohon mudah dimengerti.	0	0	0	2	0
8	Fitur mengelola pengajuan permohonan pada aktor Pemohon mudah dimengerti.	0	0	0	2	0
9	Fitur mengelola data rincian permohonan pada aktor Pemohon mudah dimengerti.	0	0	0	2	0
10	Fitur mengelola pengajuan SPJ pada aktor Pemohon mudah dimengerti.	0	0	0	2	0
11	Fitur mengelola data bukti rincian permohonan untuk SPJ pada aktor Pemohon mudah dimengerti.	0	0	0	2	0
12	Fitur menerima surel yang berkaitan dengan kegiatan pengajuan permohonan dan SPJ pada Pemohon mudah dimengerti.	0	0	0	2	0
13	Fitur mengelola (melanjutkan) disposisi permohonan pada aktor WD 2 mudah dimengerti.	0	0	0	2	0
14	Fitur menerima surel yang berkaitan dengan kegiatan disposisi permohonan pada aktor WD 2 dan PPK mudah dimengerti.	0	0	0	2	0
15	Fitur mengelola data kegiatan permohonan pada aktor PPK mudah dimengerti.	0	0	0	2	0
16	Fitur mengelola disposisi permohonan (melanjutkan atau menolak) pada aktor PPK mudah dimengerti.	0	0	0	2	0
17	Fitur mengelola disposisi permohonan (melanjutkan atau menolak) pada aktor Kasubag mudah dimengerti.	0	0	0	2	0

No	Test Deskripsi	1	2	3	4	5
18	Fitur mengelola (melanjutkan atau menolak) disposisi SPJ pada aktor Kasubag mudah dimengerti.	0	0	0	2	0
19	Fitur menerima surel yang berkaitan dengan kegiatan disposisi permohonan dan SPJ pada aktor Kasubag mudah dimengerti.	0	0	0	2	0
20	Fitur mengelola (melanjutkan) disposisi permohonan pada aktor BPP mudah dimengerti.	0	0	0	2	0
21	Fitur mengelola (melanjutkan) disposisi SPJ pada aktor BPP mudah dimengerti.	0	0	0	2	0
22	Fitur menerima surel yang berkaitan dengan kegiatan disposisi permohonan dan SPJ pada aktor BPP mudah dimengerti.	0	0	0	2	0
23	Fitur melihat rekap histori permohonan termasuk <i>export excel</i> pada permohonan yang selesai SPJ-nya semua aktor mudah dimengerti.	0	0	0	1	1
Total		0	0	0	44	2

$$\text{Nilai Total} = (0 \times 1) + (0 \times 2) + (0 \times 3) + (44 \times 4) + (2 \times 5) = 186$$

$$\text{Persentase kelayakan(\%)} = \frac{186}{230} \times 100\% = 80,8\%$$

Berdasarkan hasil *test data*, didapatkan persentase kelayakan senilai 80,8%. Maka dapat dikatakan pada tahap *beta testing* ini sistem emonik mendapat predikat sesuai untuk aspek kebergunaan sistem. Dengan catatan perlu dilakukan uji coba sistem ke beberapa pengguna agar lebih baik.

4. *Test log*

Target pada artefak ini pada iterasi T1 adalah memperbaiki kekurangan dari sistem dan penambahan fitur proker berdasarkan masukan dari pengujian *beta test*. Berikut adalah tabel *Test log beta testing* yang merupakan penyelesaian masalah yang terdapat pada *beta test*.

Tabel 4.19 : *Test log beta testing*

No	<i>Test Deskripsi</i>	<i>Errors</i>	<i>Level Errors</i>	<i>Restore Errors</i>
1	Fitur pengajuan permohonan kembali ke proses awal setelah permohonannya ditolak.	Permohonan kembali ke tahap awal setelah ditolak.	<i>Script Failures</i>	30 Desember 2020
2	Fitur mengelola pengajuan proker untuk pemohon dan PPK.	Fitur mengelola proker belum ada.	<i>Script Failures</i>	30 Desember 2020

c. *Deployment*

Target pada proses *Deployment* ini diharapkan sudah mencapai 35% pada iterasi *transition* T1 ini dan mencapai target 65% pada target *development plan*. Pada proses *Deployment* ini akan dibahas artefak *Realease notes*.

1. *Realease notes*

Melanjutkan iterasi sebelumnya, pada iterasi *transition* T1 telah dibuat produk *beta* dan telah melakukan *beta testing*. Berikut adalah tabel *Realease notes* pada iterasi *transition* T1.

Tabel 4.20 : *Realease notes* pada iterasi *transition* T1

No	<i>Product Version</i>	<i>Bugs</i>	<i>What's New</i>	<i>Realease Date</i>
1	Sistem Informasi <i>E-monitoring</i> Keuangan Berbasis <i>Website</i> di FMIPA UNJ <i>beta version</i> 1.	- Permohonan kembali ke tahap awal setelah ditolak. - Fitur mengelola proker belum ada.	-	8 Desember 2020

No	Product Version	Bugs	What's New	Realease Date
2	Sistem Informasi <i>E-monitoring</i> Keuangan Berbasis <i>Website</i> di FMIPA UNJ <i>beta version 2.</i>	-	<i>Bug fixed</i> <i>on beta</i> <i>version 1.</i>	30 Desember 2020

2. Iterasi *transition* T2

Iterasi selanjutnya pada tahap *transition* T2 berdasarkan *development plan* yaitu dibahasnya 20% *Test* dan 35% *Deployment*.

a. *Test*

Target pada proses *Test* ini diharapkan sudah mencapai 20% pada iterasi *transition* T2 ini dan mencapai target 100% pada target *development plan* yang artinya selesai semua proses pada proses *Test* ini dan akan menyesuaikan penambahan sesuai kebutuhan. Pada proses *Test* ini akan dibahas artefak *Test guidelines*, *Test case*, *Test data*, *Test result* dan *Test log*.

1. *Test guidelines*

Melanjutkan iterasi sebelumnya, pada iterasi T2 adalah membuat *test guidelines* untuk UAT (*User Acceptance Test*). Dalam proses pengembangan aplikasi sesuai dengan metode *Rational Unified Process* (RUP), setelah implementasi dan beberapa tahap testing dilakukan UAT. UAT pada sistem dilakukan terhadap beberapa *stakeholder* sistem sebagai responden. Setiap responden melakukan pengujian terhadap fungsi sistem sesuai perannya masing-masing. UAT yang dilakukan menggunakan data dari kuesioner UAT *Website* EMONIK. Berikut langkah-langkah *Beta testing* Sistem Informasi *E-Monitoring* Keuangan berbasis *website* di Fakultas MIPA UNJ:

1. Admin

- a. Mengelola (menambah, menyunting, dan menghapus) data *user* dan *permission*.

- b. Membuka akses dalam membuat permohonan yang diajukan pemohon.
 - c. Mengelola (menambah, menyunting, dan menghapus) data unit, fakultas dan prodi.
 - d. Melihat rekap histori permohonan termasuk export excel untuk permohonan yang sudah selesai SPJ-nya.
2. Pemohon
- a. Mengelola (menambah, menyunting, menghapus dan mensubmit) pengajuan program kerja.
 - b. Mengelola (menambah, menyunting, menghapus dan mensubmit) pengajuan permohonan.
 - c. Mengelola (menambah, menyunting, dan menghapus) data rincian permohonan pada permohonan yang diajukan.
 - d. Mengelola (menambah, menyunting, dan menghapus) pengajuan SPJ pada permohonan.
 - e. Mengelola (mensubmit) pengajuan SPJ pada permohonan.
 - f. Mengelola (menambah dan menyunting) data bukti rincian permohonan pada SPJ yang diajukan.
 - g. Mengirim dan menerima surel yang berkaitan dengan kegiatan pengajuan proker, permohonan dan SPJ.
 - h. Melihat rekap histori permohonan termasuk export excel untuk permohonan yang sudah selesai SPJ-nya.
3. Wakil Dekan II
- a. Mengelola (melanjutkan) disposisi permohonan.
 - b. Mengirim dan menerima surel yang berkaitan dengan kegiatan disposisi permohonan.
 - c. Melihat rekap histori permohonan termasuk export excel untuk permohonan yang sudah selesai SPJ-nya.
4. PPK
- a. Mengelola (menerima atau menolak) data pengajuan program kerja.

- b. Mengelola (melanjutkan atau menolak) disposisi permohonan.
 - c. Mengirim dan menerima surel yang berkaitan dengan kegiatan pengajuan proker dan disposisi permohonan.
 - d. Melihat rekap histori permohonan termasuk export excel untuk permohonan yang sudah selesai SPJ-nya.
5. Kasubag Keuangan
- a. Mengelola (melanjutkan atau menolak) disposisi permohonan.
 - b. Mengelola (melanjutkan atau menolak) disposisi SPJ.
 - c. Mengirim dan menerima surel yang berkaitan dengan kegiatan disposisi permohonan dan SPJ.
 - d. Melihat rekap histori permohonan termasuk export excel untuk permohonan yang sudah selesai SPJ-nya.
6. BPP
- a. Mengelola (melanjutkan) disposisi permohonan.
 - b. Mengelola (melanjutkan) disposisi SPJ.
 - c. Mengirim dan menerima surel yang berkaitan dengan kegiatan disposisi permohonan dan SPJ.
 - d. Melihat rekap histori permohonan termasuk export excel untuk permohonan yang sudah selesai SPJ-nya.

Pada *test case* dilakukan pengujian fungsional yang digunakan untuk penilaian seluruh komponen berdasarkan pilihan berikut:

1. S : Setuju
2. TS : Tidak Setuju

Untuk *test data* dilakukan pengujian kebergunaan (usability) digunakan skala likert. Skala likert yaitu skala penilaian dengan pilihan skala nilai pada setiap skala yang digunakan untuk mengukur tingkat persetujuan terhadap sesuatu (Maryuliana & M. I., 2016). Skala yang digunakan dengan nilai 1 s.d. 5 dengan perincian sebagai berikut :

- 1 : Sangat Tidak Setuju
- 2 : Tidak Setuju
- 3 : Cukup

- 4 : Setuju
- 5 : Sangat Setuju

Setelah didapatkan seluruh nilai dari pengguna (user) saat pengujian, nilai tersebut dikalkulasi sesuai dengan sistem penilaian berikut:

- Nilai Total

Nilai total yaitu jumlah keseluruhan yang didapatkan dari setiap pertanyaan yang dapat ditulis menjadi:

$$\text{Nilai Total} = (\text{jumlah} \times \text{skorSS}) + (\text{jumlah} \times \text{skorS}) + (\text{jumlah} \times \text{skorC}) + (\text{jumlah} \times \text{skorTS}) + (\text{jumlah} \times \text{skorSTS})$$

- Presentase Kelayakan

Presentase Kelayakan Presentase kelayakan yaitu persentase nilai rata-rata yang didapatkan dari nilai total dibagi skor yang diharapkan. Skor yang diharapkan didapat dari skor maksimal yang dikalikan dengan jumlah responden. Skor maksimal adalah nilai maksimal dari skala likert dikalikan dengan jumlah pertanyaan. Perhitungan tersebut dapat ditulis menjadi:

$$\text{Persentase kelayakan(\%)} = \frac{\text{NilaiTotal}}{\text{SkorDiharapkan}} \times 100\%$$

Persentase kelayakan yang telah didapatkan akan dibandingkan dengan skor pada skala likert. Berikut model skala likert:

- | | |
|-------------------------|--------------|
| 6. Sangat Kurang Sesuai | = 0% - 20% |
| 7. Kurang Sesuai | = 21% - 40% |
| 8. Cukup Sesuai | = 41% - 60% |
| 9. Sesuai | = 61% - 80% |
| 10. Sangat Sesuai | = 81% - 100% |

2. *Test case*

Melanjutkan iterasi sebelumnya, pada iterasi T2 adalah membuat *test case* untuk UAT (*User Acceptance Test*). Berdasarkan hasil UAT telah disusun beberapa tabel Hasil *Test case* UAT untuk setiap responden yang mengisi form UAT. Berikut adalah hasil *Test case* UAT untuk masing-masing *role user* yang terlibat dalam sistem.

a. Pengujian oleh Admin

Pada uji coba yang dilakukan oleh satu responden admin. Penilaian yang diberikan mengenai Hasil *Test case* UAT pada Admin dengan uji coba fungsional dibuktikan dengan hasil kuesioner UAT pada admin [Lampiran 3] dan ringkasan dari hasil kuesioner tersebut sebagai berikut:

Tabel 4.21 : Hasil *Test case* UAT pada Admin

No	<i>Test Deskripsi</i>	Jawaban Responden	
		Setuju	Tidak Setuju
1	Fitur masuk ke dalam sistem (<i>login</i>) berjalan dengan baik.	1	0
2	Fitur menyunting data diri berjalan dengan baik.	1	0
3	Fitur mengelola (menambah, menyunting, dan menghapus) data <i>user</i> dan <i>permission</i> berjalan dengan baik.	1	0
4	Fitur membuka akses dalam membuat permohonan yang diajukan pemohon berjalan dengan baik.	1	0
5	Fitur mengelola (menambah, menyunting, dan menghapus) data unit, fakultas dan prodi berjalan dengan baik.	1	0
6	Fitur melihat rekap histori permohonan termasuk export excel untuk permohonan yang sudah selesai SPJ-nya berjalan dengan baik.	1	0
7	Fitur keluar sistem (<i>logout</i>) berjalan dengan baik.	1	0
Total		7	0
Presentase Jawaban		100%	0%

Berdasarkan hasil *test case* UAT pada Admin, mendapatkan presentase jawaban 100%. Maka dapat dikatakan sistem informasi *e-monik* ini berjalan dengan baik sesuai dengan yang diharapkan.

b. Pengujian oleh Pemohon

Pada uji coba yang dilakukan oleh satu responden pemohon. Penilaian yang diberikan mengenai Hasil *Test case* UAT pada Pemohon dengan uji coba fungsional dibuktikan dengan hasil kuesioner UAT pada pemohon [Lampiran 4] dan ringkasan dari hasil kuesioner tersebut sebagai berikut:

Tabel 4.22 : Hasil *Test case* UAT pada Pemohon

No	<i>Test Deskripsi</i>	Jawaban Responden	
		Setuju	Tidak Setuju
1	Fitur masuk ke dalam sistem (<i>login</i>) berjalan dengan baik.	1	0
2	Fitur menyunting data diri berjalan dengan baik.	1	0
3	Fitur mengelola (menambah, menyunting, menghapus dan mensubmit) pengajuan program kerja berjalan dengan baik.	1	0
4	Fitur mengelola (menambah, menyunting, menghapus dan mensubmit) pengajuan permohonan berjalan dengan baik.	1	0
5	Fitur mengelola (menambah, menyunting, dan menghapus) data rincian permohonan pada permohonan yang diajukan berjalan dengan baik.	1	0
6	Fitur mengelola (mensubmit) pengajuan SPJ pada permohonan berjalan dengan baik.	1	0
7	Fitur mengelola (menambah dan menyunting) data bukti rincian permohonan pada SPJ yang diajukan berjalan dengan baik.	1	0

No	Test Deskripsi	Jawaban Responden	
		Setuju	Tidak Setuju
8	Fitur mengirim dan menerima surel yang berkaitan dengan kegiatan pengajuan proker, permohonan dan SPJ berjalan dengan baik.	1	0
9	Fitur melihat rekap histori permohonan termasuk export excel untuk permohonan yang sudah selesai SPJ-nya berjalan dengan baik.	1	0
10	Fitur keluar sistem (<i>logout</i>) berjalan dengan baik.	1	0
Total		10	0
Presentase Jawaban		100%	0%

Berdasarkan hasil *test case* UAT pada Pemohon, mendapatkan presentase jawaban 100%. Maka dapat dikatakan sistem informasi *e-monik* ini berjalan dengan baik sesuai dengan yang diharapkan.

c. Pengujian oleh Wakil Dekan II

Pada uji coba yang dilakukan oleh satu responden Wakil Dekan II. Penilaian yang diberikan mengenai Hasil *Test case* UAT pada Wakil Dekan II dengan uji coba fungsional dibuktikan dengan hasil kuesioner UAT pada Wakil Dekan II [Lampiran 5] dan ringkasan dari hasil kuesioner tersebut sebagai berikut:

Tabel 4.23 : Hasil *Test case* UAT pada Wakil Dekan II

No	Test Deskripsi	Jawaban Responden	
		Setuju	Tidak Setuju
1	Fitur masuk ke dalam sistem (<i>login</i>) berjalan dengan baik.	1	0
2	Fitur menyunting data diri berjalan dengan baik.	1	0

No	Test Deskripsi	Jawaban Responden	
		Setuju	Tidak Setuju
3	Fitur mengelola (melanjutkan) disposisi permohonan berjalan dengan baik.	1	0
4	Fitur mengirim dan menerima surel yang berkaitan dengan kegiatan disposisi permohonan berjalan dengan baik.	1	0
5	Fitur melihat rekap histori permohonan termasuk export excel untuk permohonan yang sudah selesai SPJ-nya berjalan dengan baik.	1	0
6	Fitur keluar sistem (<i>logout</i>) berjalan dengan baik.	1	0
Total		6	0
Presentase Jawaban		100%	0%

Berdasarkan hasil *test case* UAT pada Wakil Dekan II, mendapatkan presentase jawaban 100%. Maka dapat dikatakan sistem informasi *e-monik* ini berjalan dengan baik sesuai dengan yang diharapkan.

d. Pengujian oleh PPK

Pada uji coba yang dilakukan oleh satu responden PPK. Penilaian yang diberikan mengenai Hasil *Test case* UAT pada PPK dengan uji coba fungsional dibuktikan dengan hasil kuesioner UAT pada PPK [Lampiran 6] dan ringkasan dari hasil kuesioner tersebut sebagai berikut:

Tabel 4.24 : Hasil *Test case* UAT pada PPK

No	Test Deskripsi	Jawaban Responden	
		Setuju	Tidak Setuju
1	Fitur masuk ke dalam sistem (<i>login</i>) berjalan dengan baik.	1	0

No	Test Deskripsi	Jawaban Responden	
		Setuju	Tidak Setuju
2	Fitur menyunting data diri berjalan dengan baik.	1	0
3	Fitur mengelola (menerima atau menolak) data pengajuan program kerja berjalan dengan baik.	1	0
4	Fitur mengelola (melanjutkan atau menolak) disposisi permohonan berjalan dengan baik.	1	0
5	Fitur mengirim dan menerima surel yang berkaitan dengan kegiatan pengajuan proker dan disposisi permohonan berjalan dengan baik.	1	0
6	Fitur melihat rekap histori permohonan termasuk export excel untuk permohonan yang sudah selesai SPJ-nya berjalan dengan baik.	1	0
7	Fitur keluar sistem (<i>logout</i>) berjalan dengan baik.	1	0
Total		7	0
Presentase Jawaban		100%	0%

Berdasarkan hasil *test case* UAT pada PPK, mendapatkan presentase jawaban 100%. Maka dapat dikatakan sistem informasi *e-monik* ini berjalan dengan baik sesuai dengan yang diharapkan.

e. Pengujian oleh Kasubag

Pada uji coba yang dilakukan oleh satu responden Kasubag Keuangan. Penilaian yang diberikan mengenai Hasil *Test case* UAT pada Kasubag dengan uji coba fungsional dibuktikan dengan hasil kuesioner UAT pada Kasubag Keuangan [Lampiran 7] dan ringkasan dari hasil kuesioner tersebut sebagai berikut:

Tabel 4.25 : Hasil *Test case* UAT pada Kasubag

No	<i>Test Deskripsi</i>	Jawaban Responden	
		Setuju	Tidak Setuju
1	Fitur masuk ke dalam sistem (<i>login</i>) berjalan dengan baik.	1	0
2	Fitur menyunting data diri berjalan dengan baik.	1	0
3	Fitur mengelola (melanjutkan atau menolak) disposisi permohonan berjalan dengan baik.	1	0
4	Fitur mengelola (melanjutkan atau menolak) disposisi SPJ berjalan dengan baik.	1	0
5	Fitur mengirim dan menerima surel yang berkaitan dengan kegiatan disposisi permohonan dan SPJ berjalan dengan baik.	1	0
6	Fitur melihat rekap histori permohonan termasuk export excel untuk permohonan yang sudah selesai SPJ-nya berjalan dengan baik.	1	0
7	Fitur keluar sistem (<i>logout</i>) berjalan dengan baik.	1	0
Total		7	0
Presentase Jawaban		100%	0%

Berdasarkan hasil *test case* UAT pada PPK, mendapatkan presentase jawaban 100%. Maka dapat dikatakan sistem informasi *e-monik* ini berjalan dengan baik sesuai dengan yang diharapkan.

f. Pengujian oleh BPP

Pada uji coba yang dilakukan oleh satu responden BPP. Penilaian yang diberikan mengenai Hasil *Test case* UAT pada BPP dengan uji coba fungsional dibuktikan dengan hasil kuesioner UAT pada BPP [Lampiran 8] dan ringkasan dari hasil kuesioner tersebut sebagai berikut:

Tabel 4.26 : Hasil *Test case* UAT pada BPP

No	<i>Test Deskripsi</i>	Jawaban Responden	
		Setuju	Tidak Setuju
1	Fitur masuk ke dalam sistem (<i>login</i>) berjalan dengan baik.	1	0
2	Fitur menyunting data diri berjalan dengan baik.	1	0
3	Fitur mengelola (melanjutkan atau menolak) disposisi permohonan berjalan dengan baik.	1	0
4	Fitur mengelola (melanjutkan atau menolak) disposisi SPJ berjalan dengan baik.	1	0
5	Fitur mengirim dan menerima surel yang berkaitan dengan kegiatan disposisi permohonan dan SPJ berjalan dengan baik.	1	0
6	Fitur melihat rekap histori permohonan termasuk export excel untuk permohonan yang sudah selesai SPJ-nya berjalan dengan baik.	1	0
7	Fitur keluar sistem (<i>logout</i>) berjalan dengan baik.	1	0
Total		7	0
Presentase Jawaban		100%	0%

Berdasarkan hasil *test case* UAT pada BPP, mendapatkan presentase jawaban 100%. Maka dapat dikatakan sistem informasi *e-monik* ini berjalan dengan baik sesuai dengan yang diharapkan.

3. *Test data*

Melanjutkan iterasi sebelumnya, pada iterasi T2 adalah membuat *test data* untuk UAT (*User Acceptance Test*). Berdasarkan hasil UAT telah disusun beberapa tabel Hasil *Test data* UAT untuk setiap responden yang

mengisi form UAT. Berikut adalah hasil *Test case* UAT untuk masing-masing *role user* yang terlibat dalam sistem.

a. Pengujian oleh Admin

Pada uji coba yang dilakukan oleh satu responden admin. Penilaian yang diberikan mengenai Hasil *Test data* UAT pada Admin dengan uji coba kebergunaan dibuktikan dengan hasil kuesioner UAT pada admin [Lampiran 3] dan ringkasan dari hasil kuesioner tersebut sebagai berikut:

Tabel 4.27 : Hasil *Test data* UAT pada Admin

No	<i>Test Deskripsi</i>	1	2	3	4	5
1	Fitur menyunting data diri mudah dimengerti.	0	0	0	0	1
2	Fitur mengelola (menambah, menyunting, dan menghapus) data <i>user</i> dan <i>permission</i> mudah dimengerti.	0	0	0	0	1
3	Fitur membuka akses dalam membuat permohonan yang diajukan pemohon mudah dimengerti.	0	0	0	0	1
4	Fitur mengelola (menambah, menyunting, dan menghapus) data unit, fakultas dan prodi mudah dimengerti.	0	0	0	0	1
5	Fitur melihat rekap histori permohonan termasuk export excel untuk permohonan yang sudah selesai SPJ-nya mudah dimengerti.	0	0	0	0	1
6	Sistem ini mudah digunakan.	0	0	0	0	1
7	Sistem ini nyaman digunakan.	0	0	0	0	1
Total		0	0	0	0	7

$$\text{Nilai Total} = (0 \times 1) + (0 \times 2) + (0 \times 3) + (0 \times 4) + (7 \times 5) = 35$$

$$\text{Persentase kelayakan(\%)} = \frac{35}{35} \times 100\% = 100\%$$

Berdasarkan hasil *test data*, didapatkan persentase kelayakan senilai 100%. Maka dapat dikatakan sistem informasi *e-monik* ini mendapat predikat

sangat sesuai untuk aspek kebergunaan sistem. Dengan catatan semua sudah oke.

b. Pengujian oleh Pemohon

Pada uji coba yang dilakukan oleh satu responden pemohon. Penilaian yang diberikan mengenai Hasil *Test data* UAT pada Pemohon dengan uji coba kebergunaan dibuktikan dengan hasil kuesioner UAT pada pemohon [Lampiran 4] dan ringkasan dari hasil kuesioner tersebut sebagai berikut:

Tabel 4.28 : Hasil *Test data* UAT pada Pemohon

No	<i>Test Deskripsi</i>	1	2	3	4	5
1	Fitur menyunting data diri mudah dimengerti.	0	0	0	0	1
2	Fitur mengelola (menambah, menyunting, menghapus dan mensubmit) pengajuan program kerja mudah dimengerti.	0	0	0	1	0
3	Fitur mengelola (menambah, menyunting, menghapus dan mensubmit) pengajuan permohonan mudah dimengerti.	0	0	0	1	0
4	Fitur mengelola (menambah, menyunting, dan menghapus) data rincian permohonan pada permohonan yang diajukan mudah dimengerti.	0	0	0	1	0
5	Fitur mengelola (mensubmit) pengajuan SPJ pada permohonan mudah dimengerti.	0	0	0	1	0
6	Fitur mengelola (menambah dan menyunting) data bukti rincian permohonan pada SPJ yang diajukan mudah dimengerti.	0	0	0	1	0
7	Fitur mengirim dan menerima surel yang berkaitan dengan kegiatan pengajuan proker, permohonan dan SPJ mudah dimengerti.	0	0	0	1	0
8	Fitur melihat rekap histori permohonan termasuk export excel untuk permohonan yang sudah selesai SPJ-nya mudah dimengerti.	0	0	0	1	0

No	Test Deskripsi	1	2	3	4	5
9	Sistem ini mudah digunakan.	0	0	0	0	1
10	Sistem ini nyaman digunakan.	0	0	0	1	0
Total		0	0	0	8	2

$$\text{Nilai Total} = (0 \times 1) + (0 \times 2) + (0 \times 3) + (8 \times 4) + (2 \times 5) = 42$$

$$\text{Persentase kelayakan(\%)} = \frac{42}{50} \times 100\% = 84\%$$

Berdasarkan hasil *test data*, didapatkan persentase kelayakan senilai 84%.

Maka dapat dikatakan sistem informasi *e-monik* ini mendapat predikat sangat sesuai untuk aspek kebergunaan sistem. Dengan catatan agar aplikasi ini dapat lebih bermanfaat bagi yang memerlukan, diberikan untuk adanya pengembangan dari sistem.

c. Pengujian oleh Wakil Dekan II

Pada uji coba yang dilakukan oleh satu responden Wakil Dekan II. Penilaian yang diberikan mengenai Hasil *Test data* UAT pada Wakil Dekan II dengan uji coba kebergunaan dibuktikan dengan hasil kuesioner UAT pada Wakil Dekan II [Lampiran 5] dan ringkasan dari hasil kuesioner tersebut sebagai berikut:

Tabel 4.29 : Hasil *Test data* UAT pada Wakil Dekan II

No	Test Deskripsi	1	2	3	4	5
1	Fitur menyunting data diri mudah dimengerti.	0	0	0	0	1
2	Fitur mengelola (melanjutkan) disposisi permohonan mudah dimengerti.	0	0	0	0	1
3	Fitur mengirim dan menerima surel yang berkaitan dengan kegiatan disposisi permohonan mudah dimengerti.	0	0	0	0	1
4	Fitur melihat rekap histori permohonan termasuk export excel untuk permohonan yang sudah selesai SPJ-nya mudah dimengerti.	0	0	0	0	1
5	Sistem ini mudah digunakan.	0	0	0	0	1

No	Test Deskripsi	1	2	3	4	5
6	Sistem ini nyaman digunakan.	0	0	0	0	1
Total		0	0	0	0	6

$$\text{Nilai Total} = (0 \times 1) + (0 \times 2) + (0 \times 3) + (0 \times 4) + (6 \times 5) = 30$$

$$\text{Persentase kelayakan}(\%) = \frac{30}{30} \times 100\% = 100\%$$

Berdasarkan hasil *test data*, didapatkan persentase kelayakan senilai 100%. Maka dapat dikatakan sistem informasi *e-monik* ini mendapat predikat sangat sesuai untuk aspek kebergunaan sistem.

d. Pengujian oleh PPK

Pada uji coba yang dilakukan oleh satu responden PPK. Penilaian yang diberikan mengenai Hasil *Test data* UAT pada PPK dengan uji coba kebergunaan dibuktikan dengan hasil kuesioner UAT pada PPK [Lampiran 6] dan ringkasan dari hasil kuesioner tersebut sebagai berikut:

Tabel 4.30 : Hasil *Test data* UAT pada PPK

No	Test Deskripsi	1	2	3	4	5
1	Fitur menyunting data diri mudah dimengerti.	0	0	0	1	0
2	Fitur mengelola (menerima atau menolak) data pengajuan program kerja mudah dimengerti.	0	0	0	1	0
3	Fitur mengelola (melanjutkan atau menolak) disposisi permohonan mudah dimengerti.	0	0	0	1	0
4	Fitur mengirim dan menerima surel yang berkaitan dengan kegiatan pengajuan proker dan disposisi permohonan mudah dimengerti.	0	0	0	1	0
5	Fitur melihat rekap histori permohonan termasuk export excel untuk permohonan yang sudah selesai SPJ-nya mudah dimengerti.	0	0	0	1	0
6	Sistem ini mudah digunakan.	0	0	0	1	0
7	Sistem ini nyaman digunakan.	0	0	0	1	0
Total		0	0	0	7	0

$$\text{Nilai Total} = (0 \times 1) + (0 \times 2) + (0 \times 3) + (7 \times 4) + (0 \times 5) = 28$$

$$\text{Persentase kelayakan(\%)} = \frac{28}{35} \times 100\% = 80\%$$

Berdasarkan hasil *test data*, didapatkan persentase kelayakan senilai 80%. Maka dapat dikatakan sistem informasi *e-monik* ini mendapat predikat sesuai untuk aspek kebergunaan sistem. Dengan Catatan saat *export excel*, kalau bisa perbaris langsung diberi warna atau langsung berbentuk tabel dengan border.

e. Pengujian oleh Kasubag

Pada uji coba yang dilakukan oleh satu responden Kasubag Keuangan. Penilaian yang diberikan mengenai Hasil *Test data* UAT pada Kasubag dengan uji coba kebergunaan dibuktikan dengan hasil kuesioner UAT pada Kasubag Keuangan [Lampiran 7] dan ringkasan dari hasil kuesioner tersebut sebagai berikut:

Tabel 4.31 : Hasil *Test data* UAT pada Kasubag

No	<i>Test Deskripsi</i>	1	2	3	4	5
1	Fitur menyunting data diri mudah dimengerti.	0	0	0	0	1
2	Fitur mengelola (melanjutkan atau menolak) disposisi permohonan mudah dimengerti.	0	0	0	0	1
3	Fitur mengelola (melanjutkan atau menolak) disposisi SPJ mudah dimengerti.	0	0	0	0	1
4	Fitur mengirim dan menerima surel yang berkaitan dengan kegiatan disposisi permohonan dan SPJ mudah dimengerti.	0	0	0	0	1
5	Fitur melihat rekap histori permohonan termasuk <i>export excel</i> untuk permohonan yang sudah selesai SPJ-nya mudah dimengerti.	0	0	0	0	1
6	Sistem ini mudah digunakan.	0	0	0	0	1
7	Sistem ini nyaman digunakan.	0	0	0	0	1
Total		0	0	0	0	7

$$\text{Nilai Total} = (0 \times 1) + (0 \times 2) + (0 \times 3) + (0 \times 4) + (5 \times 5) = 35$$

$$\text{Persentase kelayakan}(\%) = \frac{35}{35} \times 100\% = 100\%$$

Berdasarkan hasil *test data*, didapatkan persentase kelayakan senilai 100%. Maka dapat dikatakan sistem informasi *e-monik* ini mendapat predikat sangat sesuai untuk aspek kebergunaan sistem.

f. Pengujian oleh BPP

Pada uji coba yang dilakukan oleh satu responden BPP. Penilaian yang diberikan mengenai Hasil *Test data* UAT pada BPP dengan uji coba kebergunaan dibuktikan dengan hasil kuesioner UAT pada BPP [Lampiran 8] dan ringkasan dari hasil kuesioner tersebut sebagai berikut:

Tabel 4.32 : Hasil *Test data* UAT pada BPP

No	<i>Test Deskripsi</i>	1	2	3	4	5
1	Fitur menyunting data diri mudah dimengerti.	0	0	0	1	0
2	Fitur mengelola (melanjutkan) disposisi permohonan mudah dimengerti.	0	0	1	0	0
3	Fitur mengelola (melanjutkan) disposisi SPJ mudah dimengerti.	0	0	1	0	0
4	Fitur mengirim dan menerima surel yang berkaitan dengan kegiatan disposisi permohonan dan SPJ mudah dimengerti.	0	0	1	0	0
5	Fitur melihat rekap histori permohonan termasuk export excel untuk permohonan yang sudah selesai SPJ-nya mudah dimengerti.	0	0	1	0	0
6	Sistem ini mudah digunakan.	0	0	1	0	0
7	Sistem ini nyaman digunakan.	0	0	1	0	0
Total		0	0	6	1	0

$$\text{Nilai Total} = (0 \times 1) + (0 \times 2) + (6 \times 3) + (1 \times 4) + (0 \times 5) = 22$$

$$\text{Persentase kelayakan}(\%) = \frac{22}{35} \times 100\% = 62.8\%$$

Berdasarkan hasil *test data*, didapatkan persentase kelayakan senilai 62.8%. Maka dapat dikatakan sistem informasi *e-monik* ini mendapat predikat sesuai untuk aspek kebergunaan sistem.

4. *Test log*

Target pada artefak ini pada iterasi T2 adalah mendata pelaksanaan UAT oleh beberapa *user*. Berikut adalah tabel *Test log* UAT.

Tabel 4.33 : *Test log* UAT

No	<i>Test Cases</i>	<i>Pass/Fail</i>	<i>Tested By</i>	<i>Date Tested</i>
1	UAT sistem informasi <i>e-monik</i> pada Admin	<i>Pass</i>	Alfria Widiyanti, A,md	28 Januari 2021
2	UAT sistem informasi <i>e-monik</i> pada Pemohon	<i>Pass</i>	Ir. Fariani Hermin Indiyah, M.T.	15 Januari 2021
3	UAT sistem informasi <i>e-monik</i> pada Wakil Dekan II	<i>Pass</i>	Drs. Sudarwanto, M.Si, DEA	28 Januari 2021
4	UAT sistem informasi <i>e-monik</i> pada PPK	<i>Pass</i>	Ika Aryani	28 Januari 2021
5	UAT sistem informasi <i>e-monik</i> pada Kasubag	<i>Pass</i>	Yayuk Fajarwati, S.Sos	28 Januari 2021
6	UAT sistem informasi <i>e-monik</i> pada BPP	<i>Pass</i>	Wiwik Endang Sulistiyowati, S.Pd	28 Januari 2021

5. *Test result*

Target pada artefak ini adalah membuat *test result*. Berdasarkan hasil UAT untuk fungsional dan kebergunaan pada semua *user*, didapatkan bahwa fitur-fitur yang terdapat pada sistem dapat berjalan dengan baik.

Selain itu didapatkan presentase kelayakan sebagai berikut :

- *User Admin* : 100%
- *User Pemohon* : 84%
- *User Wakil Dekan II* : 100%
- *User PPK* : 80%
- *User Kasubag Keuangan* : 100%
- *User BPP* : 62,8%

Dari presentase masing-masing *user* kemudian dihitung total presentase kelayakan yang merupakan rata-rata dari nilai presentase kelayakan semua *user*, dapat dilihat sebagai berikut:

$$\text{Total Persentase kelayakan (\%)} = \left(\frac{100+84+100+80+100+62,8}{6} \right) \% = 87,8\%$$

Berdasarkan perhitungan tersebut didapatkan total presentase kelayakan sebesar 87.8% berada pada rentang tafsiran 81%-100%, maka dapat dikatakan bahwa nilai kebergunaan pada seluruh sistem mendapatkan predikat sangat sesuai.

b. Deployment

Target pada proses *Deployment* ini diharapkan sudah mencapai 35% pada iterasi *transition* T2 dan mencapai 100% pada *development plan* yang artinya selesai semua proses pada proses *Deployment* ini dan akan menyesuaikan penambahan sesuai kebutuhan. Pada proses *Deployment* ini dibahas artefak *Realease notes*, *End user support material* dan *Deployment unit*.

1. *Realease notes*

Melanjutkan iterasi sebelumnya, pada iterasi *transition* T2 telah dibuat produk rilis UAT. Berikut adalah *Realease notes* pada iterasi *transition* T2.

Tabel 4.34 : *Realease notes* pada iterasi *transition* T2

No	<i>Product Version</i>	<i>Bugs</i>	<i>What's New</i>	<i>Realease Date</i>
1	Sistem Informasi <i>E-monitoring</i> Keuangan Berbasis <i>Website</i> di FMIPA UNJ	-	-	15 Januari 2020

2. *End user support material*

Target pada artefak ini yaitu membuat panduan penggunaan sistem (user manual). Target artefak ini telah tercapai dengan dibuatnya panduan penggunaan sistem (user manual) untuk Sistem Informasi E-monitoring Keuangan Berbasis Website di FMIPA UNJ yang telah diserahkan kepada Wakil Dekan II yang dibuktikan dengan Berita Acara Serah Terima Software Wakil Dekan II [Lampiran 9].

3. *Deployment unit*

Target pada artefak ini yaitu membuat *deployment unit*. Target artefak ini telah tercapai dengan sistem yang sudah di *upload hosting* pada server MIPA dapat diakses pada alamat url <http://fmipa.unj.ac.id/emonik/> yang dibuktikan dengan Berita Acara Serah Terima *Software* Wakil Dekan II [Lampiran 9].

E. Iterasi

Pada subbab ini akan dibahas hasil iterasi yang terjadi sebenarnya dengan iterasi yang direncanakan, membuat ringkasan iterasi dari semua iterasi yang direncanakan sampai iterasi yang terjadi sebenarnya dan juga membuat kesimpulan iterasi.

1. Hasil iterasi

Hasil iterasi akan membahas hasil iterasi yang direncanakan dan hasil iterasi yang sebenarnya.

a. Hasil iterasi yang direncanakan

Hasil iterasi yang direncanakan akan menuliskan semua artefak yang dihasilkan secara singkat berdasarkan hasil yang sudah dibahas pada penelitian ini di bab 4 pada subbab tahap *inception*, tahap *elaboration*, tahap *construction* dan tahap *transition* yang sebelumnya sudah diatur dalam tabel *development plan* pada Bab 3 yang terjadi sebanyak 8 iterasi dengan rincian berikut yaitu 1 iterasi pada iterasi *inception*, 2 iterasi pada iterasi *elaboration*, 3 iterasi pada iterasi *construction* dan 2 iterasi pada iterasi *transition*. Berikut adalah tabel Hasil iterasi yang direncanakan.

Tabel 4.35 : Hasil iterasi yang direncanakan

No	Iterasi	Process	Hasil	Tanggal
1	I1	35% <i>Business Modelling</i>	<i>Business Rules, Business Vission, Business architecture document, Business actor dan Business use case</i>	21 September 2020
		20% <i>Requirements</i>	<i>Requirements Management Plan dan Software Requirements Specification</i>	
		15% <i>Analysis & Design</i>	<i>Use case realization</i>	
2.	E1	35% <i>Business Modelling</i>	<i>Business use case dan Business use case model</i>	28 September 2020
		35% <i>Requirements</i>	<i>Actor (human), Use Case, Use Case Model, Supplementary specification dan Software Requirements Specification</i>	
		25% <i>Analysis & Design</i>	<i>Use Case Realization dan Design model</i>	
		10% <i>Implementation</i>	<i>Integration build plan</i>	
3.	E2	20% <i>Business Modelling</i>	<i>Business use case dan Business use case model</i>	5 Oktober 2020
		35% <i>Requirements</i>	<i>Stakeholder request, Glossary, Vission, Use Case dan Use Case Model</i>	
		30% <i>Analysis & Design</i>	<i>Design model, Design Class dan Data Model</i>	

No	Iterasi	Process	Hasil	Tanggal
		10% <i>Implementation</i>	<i>Component</i> dan <i>Implementation Model</i>	
		10% <i>Test</i>	<i>Test Plan</i>	
4.	C1	10% <i>Business Modelling</i>	<i>Business use case</i> dan <i>Business use case model</i>	13 Oktober 2020
		10% <i>Requirements</i>	<i>Use Case</i> dan <i>Use Case Model</i>	
		20% <i>Analysis & Design</i>	<i>Design model</i>	
		30% <i>Implementation</i>	<i>Integration build plan,</i> <i>Component</i> dan <i>Implementation Model</i>	
		10% <i>Test</i>	<i>Test Plan</i>	
		10% <i>Deployment</i>	<i>Deployment Plan</i>	
5.	C2	10% <i>Analysis & Design</i>	<i>Design model</i>	22 Oktober 2020
		20% <i>Implementation</i>	<i>Component</i> dan <i>Implementation Model</i>	
		10% <i>Test</i>	<i>Test Plan</i>	
		10% <i>Deployment</i>	<i>Deployment Plan</i>	
6.	C3	20% <i>Implementation</i>	<i>Build, Component</i> dan <i>Implementation Model</i>	13 Desember 2020
		20% <i>Test</i>	<i>Test script</i> dan <i>Test log</i>	
		10% <i>Deployment</i>	<i>Realease notes</i>	
7.	T1	10% <i>Implementation</i>	<i>Build, Component</i> dan <i>Implementation Model</i>	3 Januari 2021

No	Iterasi	Process	Hasil	Tanggal
		30% Test	<i>Tes Guidelines, Test case, Test data dan Test log</i>	
		35% <i>Deployment</i>	<i>Realease notes</i>	
8.	T2	20% Test	<i>Test guideliness, Test case, Test data, Test result dan Test log</i>	1 Februari
		35% <i>Deployment</i>	<i>Realease notes, End user support material dan Deployment unit</i>	2021

Keterangan :

Iterasi I : *Inception*

Iterasi E : *Elaboration*

Iterasi C : *Contruccion*

Iterasi T : *Transition*

b. Hasil iterasi yang sebenarnya

Hasil iterasi yang sebenarnya akan menuliskan semua artefak dan proses yang dihasilkan pada penelitian ini secara singkat berdasarkan hasil iterasi sesuai dengan keadaan sebenarnya dengan interval pada tiap iterasi berlangsung selama satu minggu. Berikut adalah tabel Hasil iterasi yang direncanakan.

Tabel 4.36 : Hasil iterasi yang sebenarnya

No	Iterasi	Interval	Hasil	Process
1	I1	21 - 27 Sep 2020	<i>Business Rules, Business Vission, Business architecture document, Business actor dan Business use case</i>	40% BM

No	Iterasi	Interval	Hasil	Process
			<i>Requirements Management Plan dan Software Requirements Specification</i>	20% Req
			<i>Use case realization</i>	10% A&D
2	E1	28 Sep - 4 Okt 2020	<i>Business use case dan Business use case model</i>	20% BM
			<i>Actor (human), Use Case, Use Case Model, Supplementary specification dan Software Requirements Specification</i>	30% Req
			<i>Use Case Realization dan Design model</i>	20% A&D
			<i>Integration build plan</i>	5% Imp
3.	E2	5 - 11 Okt 2020	<i>Business use case dan Business use case model</i>	20% BM
			<i>Stakeholder request, Glossary, Vission, Use Case Use Case Model</i>	30% Req
			<i>Design model, Design Class dan Data Model</i>	35% A&D
			<i>Component, Implementation Model</i>	15% Imp
			<i>Test Plan</i>	5% Test
4.	C1	12 - 18 Okt 2020	<i>Business use case dan Business use case model</i>	20% BM
			<i>Use Case dan Use Case Model</i>	20% Req
			<i>Design model</i>	20% A&D
			<i>Integration build plan, Component dan Implementation Model</i>	30% Imp
			<i>Test Plan</i>	5% Test
			<i>Deployment Plan</i>	10% Dep

No	Iterasi	Interval	Hasil	Process
5.	C2	19 - 25 Okt 2020	<i>Design model</i>	15% <i>A&D</i>
			<i>Component, Implementation Model</i>	20% <i>Imp</i>
			<i>Test Plan</i>	5% <i>Test</i>
			<i>Deployment Plan</i>	10% <i>Dep</i>
6.	C3	26 Okt - 1 Nov 2020	Tidak ada aktivitas	-
7.	C4	2 - 8 Nov 2020	Tidak ada aktivitas	-
8.	C5	9 - 15 Nov 2020	Tidak ada aktivitas	-
9.	C6	16 - 22 Nov 2020	Tidak ada aktivitas	-
10.	C7	23 - 29 Nov 2020	Tidak ada aktivitas	-
11.	C8	30 Nov - 6 Des 2020	Tidak ada aktivitas	-
12.	C9	7 - 13 Des 2020	<i>Build, Component dan Implementation Model</i>	10% <i>Imp</i>
			<i>Test script dan Test log</i>	20% <i>Test</i>
			<i>Realease notes</i>	20% <i>Dep</i>
13.	C10	14 - 20 Des 2020	Tidak ada aktivitas	-
14.	C11	21 - 27 Des 2020	Tidak ada aktivitas	-
15.	T1	28 Des 2020 - 3 Jan 2021	<i>Build, Component dan Implementation Model</i>	20% <i>Imp</i>
			<i>Test case, Test data dan Test log</i>	25% <i>Test</i>
			<i>Realease notes</i>	20% <i>Dep</i>

No	Iterasi	Interval	Hasil	Process
16.	T2	4 - 10 Jan 2021	Tidak ada aktivitas	-
17.	T3	11 - 17 Jan 2021	Tidak ada aktivitas	-
18	T4	18 - 24 Jan 2021	Tidak ada aktivitas	-
19	T5	25 - 31 Jan 2021	<i>Tes Guidelines</i>	<i>5% Test</i>
20	T6	1 - 7 Feb 2021	<i>Test guideliness, Test case, Test data, Test result dan Test log</i>	<i>35% Test</i>
			<i>Realease notes</i>	<i>20% Dep</i>
21	T7	8 - 14 Feb 2021	Tidak ada aktivitas	-
22	T8	15-21 Feb 2021	<i>End user support material dan Deployment unit</i>	<i>20% Dep</i>

Keterangan :

Iterasi I : *Inception*

Iterasi E : *Elaboration*

Iterasi C : *Contruction*

Iterasi T : *Transition*

BM : Business Modeling

Req : Requirements

A&D : Analysis & Design

Imp : Implementation

Dep : Deployment

2. Ringkasan iterasi

Ringkasan iterasi akan membahas ringkasan iterasi yang direncanakan dan ringkasan iterasi yang sebenarnya.

a. Ringkasan iterasi yang direncanakan

Penelitian dimulai dari tahap *inception*, pada tahap ini dilakukan satu iterasi yaitu I1 dengan dibahasnya *process 35% Business Modelling* yang menghasilkan artefak *Business Rules* yang membahas pembatasan masalah pada sistem, artefak *Business Vission* yang membahas tujuan dibuatnya sistem, artefak *Business architecture document* yang membahas gambaran menyeluruh tentang sistem yang akan dibuat, artefak *Business actor* yang merepresentasikan peran yang dimainkan seseorang atau sesuatu dalam lingkungan bisnis terkait dengan bisnisnya, dan artefak *Business use case* yang membuat *business use case* dan Hasil *automation scope business case*.

Process Selanjutnya di iterasi I1 adalah 20% *Requirements* yang menghasilkan artefak *Requirements Management Plan* yang membahas Kebutuhan Fungsional Sistem dan Kebutuhan non-Fungsional Sistem, dan artefak *Software Requirements Specification* yang membahas Kebutuhan perangkat lunak (*Software*) dan Kebutuhan perangkat keras (*Hardware*). *Process* Selanjutnya di iterasi I1 adalah 15% *Analysis & Design* yang menghasilkan artefak *Use case realization* yang membuat *Use Case Diagram Awal* dan *Use Case Realization*.

Setelah selesai tahap *inception*, tahap selanjutnya adalah tahap *elaboration*, pada tahap ini dilakukan dua iterasi yaitu E1 dan E2. Pada iterasi E1 dibahas *process 35% Business Modelling* yang menghasilkan artefak *Business use case* yang membuat Penambahan *Business use case* pada iterasi E1, dan artefak *Business use case model* yang membuat *Business use case model* pada iterasi E1.

Process Selanjutnya di iterasi E1 adalah 35% *Requirements* yang menghasilkan artefak *Actor (human)* yang mendefinisikan serangkaian peran yang koheren yang dapat dimainkan oleh pengguna sistem saat berinteraksi dengannya, artefak *Use Case* yang membuat *Use Case* pada iterasi E1, artefak *Use Case Model* yang membuat *Use Case Model* pada iterasi E1, artefak *Supplementary specification* yang membahas Spesifikasi tambahan

persyaratan sistem, dan artefak *Software Requirements Specification* yang membahas Penambahan Daftar *Software*.

Process Selanjutnya di iterasi E1 adalah 25% *Analysis & Design* yang menghasilkan artefak *Use Case Realization* yang membuat *Use case Realization Final*, dan artefak *Design model* yang membuat *Sequence diagram login*. *Process* Selanjutnya di iterasi E1 adalah 10% *Implementation* yang menghasilkan artefak *Integration build plan* yang membahas *Integration build plan* pada iterasi E1.

Selanjutnya pada iterasi E2 dibahas *process* 20% *Business Modelling* yang menghasilkan artefak *Business use case* yang membuat Penambahan *Business use case* pada iterasi E2, dan artefak *Business use case model* yang membahas *Business use case model* pada iterasi E2. *Process* Selanjutnya di iterasi E2 adalah 35% *Requirements* yang menghasilkan artefak *Stakeholder request* dengan membahas *Stakeholder request*, artefak *Glossary* yang membahas *Glossary*, artefak *Vission* yang mendefinisikan pandangan *stakeholder* dari produk yang akan dikembangkan, artefak *Use Case* yang membuat *Use Case* pada iterasi E2, dan artefak *Use Case Model* yang membuat *Use Case Model* pada iterasi E2.

Process Selanjutnya di iterasi E2 adalah 30% *Analysis & Design* yang menghasilkan artefak *Design model* yang membuat *Sequence diagram* permohonan dan *Sequence diagram SPJ*, artefak *Design Class* yang membuat *Class Diagram* Sistem, dan artefak *Data Model* yang membuat *Entity Relationship Diagram* Sistem. *Process* Selanjutnya di iterasi E2 adalah 10% *Implementation* yang menghasilkan artefak *Component* yang membuat implementasi kode pada proses *login*, dan artefak *Implementation Model* yang membuat Tampilan desain halaman *login*. *Process* Selanjutnya di iterasi E2 adalah 10% *Test* yang menghasilkan artefak *Test Plan* yang menentukan *planning* testing dan UAT pada iterasi agar dapat berjalan sesuai waktu yang ditentukan.

Setelah selesai tahap *elaboration*, tahap selanjutnya adalah tahap *construction*, pada tahap ini dilakukan tiga iterasi yaitu C1, C2 dan C3. Pada

iterasi C1 dibahas *process* 10% *Business Modelling* yang menghasilkan artefak *Business use case* yang membuat *Business use case* Final, dan artefak *Business use case model* yang membuat *Business use case model* Final. *Process* Selanjutnya di iterasi C1 adalah 10% *Requirements* yang menghasilkan artefak *Use Case* yang membuat *Use Case* Final, dan artefak *Use Case Model* yang membuat *Use Case Model* Final. *Process* Selanjutnya di iterasi C1 adalah 20% *Analysis & Design* yang menghasilkan artefak *Design model* yang membuat *Sequence diagram* pengajuan permohonan pemohon, *Sequence diagram* disposisi permohonan WD 2, *Sequence diagram* disposisi permohonan PPK, *Sequence diagram* disposisi permohonan Kasubag dan *Sequence diagram* disposisi permohonan BPP.

Process Selanjutnya di iterasi C1 adalah 30% *Implementation* yang menghasilkan artefak *Integration build plan* yang membahas Penyesuaian *Integration build plan* pada iterasi C1, artefak *Component* yang membuat implementasi kode pada proses pembuatan *role user* dan dibuatnya sistem pengajuan permohonan yang dimulai dari pemohon membuat permohonan sampai disposisi permohonan terakhir, dan artefak *Implementation Model* yang Tampilan desain halaman *dashboard* beserta semua menu yang digunakan untuk sistem pengajuan permohonan yang dimulai dari pemohon membuat permohonan sampai disposisi permohonan terakhir. *Process* Selanjutnya di iterasi C1 adalah 10% *Test* yang menghasilkan artefak *Test Plan* yang membahas Penyesuaian *Test Plan*. *Process* Selanjutnya di iterasi C1 adalah 10% *Deployment* yang menghasilkan artefak *Deployment Plan* yang membahas *Deployment Plan*.

Selanjutnya pada iterasi C2 dibahas *process* 10% *Analysis & Design* yang menghasilkan artefak *Design model* yang membuat *Sequence diagram* pengajuan SPJ pemohon, *Sequence diagram* disposisi SPJ Kasubag dan *Sequence diagram* disposisi SPJ BPP. *Process* Selanjutnya di iterasi C2 adalah 20% *Implementation* yang menghasilkan artefak *Component* yang membuat implementasi kode pada proses pembuatan sistem pengajuan SPJ yang dimulai dari pemohon melanjutkan SPJ permohonannya sampai

disposisi SPJ terakhir, dan artefak *Implementation Model* yang membuat Tampilan desain untuk semua menu yang digunakan pada proses pembuatan sistem pengajuan SPJ yang dimulai dari pemohon melanjutkan SPJ permohonannya sampai disposisi SPJ terakhir. *Process* Selanjutnya di iterasi C2 adalah 10% *Test* yang menghasilkan artefak *Test Plan* yang membahas Penyesuaian *Test Plan* pada Iterasi C2. *Process* Selanjutnya di iterasi C2 adalah 10% *Deployment* yang menghasilkan artefak *Deployment Plan* yang membahas Penyesuaian *Deployment Plan*.

Selanjutnya pada iterasi C3 dibahas *process* 20% *Implementation* yang menghasilkan artefak *Build* yang membahas Pencapaian *Integration build plan* pada iterasi C3, artefak *Component* dilakukan perbaikan kode pada proses perbaikan *bugs* sistem, dan artefak *Implementation Model* yang dilakukan perbaikan Tampilan desain pada proses perbaikan *bugs* sistem. *Process* Selanjutnya di iterasi C3 adalah 20% *Test* yang menghasilkan artefak *Test script* yang membahas *Test script alpha testing*, dan artefak *Test log* yang membahas *Test log alpha testing*. *Process* Selanjutnya di iterasi C3 adalah 10% *Deployment* yang menghasilkan artefak *Realease notes* yang membahas *Realease notes* pada iterasi *construction* C3.

Setelah selesai tahap *construction*, tahap selanjutnya adalah tahap *transition*, pada tahap ini dilakukan dua iterasi yaitu T1 dan T2. Pada iterasi T1 dibahas *process* 10% *Implementation* yang menghasilkan artefak *Build* yang membahas Pencapaian *Integration build plan Final*, artefak *Component* yang dilakukan perbaikan kode pada *beta testing* dan penambahan fitur seperti pengajuan proker, dan artefak *Implementation Model* yang dilakukan perbaikan Tampilan desain pada perbaikan *bugs* sistem untuk *beta testing* serta beberapa penambahan fitur seperti pengajuan proker.

Process Selanjutnya di iterasi T1 adalah 30% *Test* yang menghasilkan artefak *Tes Guidelines* yang membuat *test guideliness* untuk *beta testing*, artefak *Test case* yang membahas Hasil *Test case beta testing*, artefak *Test data* yang membahas Hasil *Test data beta testing*, dan artefak *Test log* yang membahas *Test log beta testing*. *Process* Selanjutnya di iterasi T1 adalah 35%

Deployment yang menghasilkan artefak *Realease notes* yang membahas *Realease notes* pada iterasi *transition* T1.

Selanjutnya pada iterasi T2 yang menjadi iterasi terakhir dibahas *process* 20% *Test* yang menghasilkan artefak *Test guideliness* yang membahas *test guideliness* untuk UAT (*User Acceptance Test*), artefak *Test case* yang membahas *test case* untuk UAT pada semua *stakeholder* sistem, artefak *Test data* yang membahas *test data* untuk UAT pada semua *stakeholder* sistem, artefak *Test log* yang membahas *Test log* UAT, dan artefak *Test result* yang didapatkan bahwa fitur-fitur yang terdapat pada sistem dapat berjalan dengan baik, dan didapatkan total presentase kelayakan sebesar 87.8%. *Process* Selanjutnya di iterasi T1 adalah 35% *Deployment* yang menghasilkan artefak *Realease notes* yang membahas *Realease notes* pada iterasi *transition* T2, *End user support material* yang panduan penggunaan sistem (*user manual*) untuk Sistem Informasi *E-monitoring* Keuangan Berbasis *Website* di FMIPA UNJ yang dibuktikan dengan Berita Acara Serah Terima *Software* Wakil Dekan II [Lampiran 9] dan *Deployment unit* yang membahas sistem yang sudah di *upload hosting* pada server MIPA dapat diakses pada alamat url <http://fmipa.unj.ac.id/emonik/> yang dibuktikan dengan Berita Acara Serah Terima *Software* Wakil Dekan II [Lampiran 9].

b. Ringkasan iterasi yang sebenarnya

Ringkasan iterasi yang sebenarnya pada dasarnya hampir sama hasilnya dengan ringkasan iterasi yang direncanakan, hanya saja terdapat perbedaan pada letak iterasi, total iterasi dan presentase proses yang lebih akurat dengan keadaan yang sebenarnya dengan pembahasan yang lebih detail mengenai hasil iterasi yang sesuai dengan interval iterasi setiap 1 minggu sesuai dengan keadaan yang terjadi sebenarnya. Oleh karena itu, pada Ringkasan iterasi yang sebenarnya akan dibahas pelaporan berdasarkan iterasi yang sebenarnya yang dapat dilihat pada tabel berikut.

Tabel 4.37 : Pelaporan iterasi yang sebenarnya

Iterasi	Process					
	<i>BM</i>	<i>Req</i>	<i>A&D</i>	<i>Imp</i>	<i>Test</i>	<i>Dep</i>
I1	0.4	0.2	0.1	-	-	-
E1	0.2	0.3	0.2	0.05	-	-
E2	0.2	0.3	0.35	0.15	0.05	-
C1	0.2	0.2	0.2	0.3	0.05	0.1
C2	-	-	0.15	0.2	0.05	0.1
C3	-	-	-	-	-	-
C4	-	-	-	-	-	-
C5	-	-	-	-	-	-
C6	-	-	-	-	-	-
C7	-	-	-	-	-	-
C8	-	-	-	-	-	-
C9	-	-	-	0.1	0.2	0.2
C10	-	-	-	-	-	-
C11	-	-	-	-	-	-
T1	-	-	-	0.2	0.25	0.2
T2	-	-	-	-	-	-
T3	-	-	-	-	-	-
T4	-	-	-	-	-	-
T5	-	-	-	-	0.05	-
T6	-	-	-	-	0.35	0.2
T7	-	-	-	-	-	-
T8	-	-	-	-	-	0.2

Keterangan :

Iterasi I : *Inception*

Iterasi E : *Elaboration*

Iterasi C : *Construction*

Iterasi T : *Transition*

BM : *Business Modeling*

Req : *Requirements*

A&D : *Analysis & Design*

Imp : *Implementation*

Dep : *Deployment*

3. Kesimpulan iterasi

Kesimpulan iterasi yang sudah dituliskan pada penelitian ini adalah:

1. Pada iterasi yang direncanakan dilakukan total 8 kali iterasi yaitu 1 iterasi pada *Inception*, 2 iterasi pada *Elaboration*, 3 iterasi pada *Construction* dan 2 iterasi pada *Transition*. Setiap iterasi yang direncanakan dilakukan secara berurutan sesuai dengan presentase pada *development plan* yang ada pada bab 3.
2. Pada iterasi yang sebenarnya dilakukan total 22 kali iterasi yaitu 1 iterasi *Inception*, 2 iterasi pada *Elaboration*, 11 iterasi pada *Construction* dan 8 iterasi pada *Transition*. Setiap iterasi yang sebenarnya dilakukan secara berurutan sesuai dengan keadaan yang sebenarnya dengan interval tetap yang berlangsung 1 minggu setiap iterasi.