

## Menu Dan Panduan Penggunaan

### PERANCANGAN PROGRAM e – REDEMPTION BERBASIS ANDROID

#### 1.1 Pendahuluan

Mahasiswa merupakan insan akademik yang tengah berkembang dalam intelektualnya dan penguasaan akademik. Mahasiswa juga merupakan aset bangsa yang harus dikembangkan pemikirannya untuk mengembangkan bakat, minat dan kemampuan, kepedulian sosial kepada masyarakat dan kegiatan akademik maupun non akademik lainnya (Fikri, et al., 2020). Saat ini dibutuhkan program aplikasi untuk mahasiswa-mahasiswa yang telat dalam perkuliahan agar mendapatkan sebuah penebusan (*redemption*), supaya menimbulkan efek jera kepada mahasiswa yang sering membuat masalah, dan alur untuk melakukan *redemption* seperti setiap mahasiswa diharuskan kekampus terlebih dahulu untuk mendapatkan kertas *redeem* dan tugas dari pengawas dan setiap pengawas memiliki kertas yang berbeda setiap prodi, setelah itu mahasiswa diharuskan mengisi data *redemption* tersebut untuk memulai *redeem* tersebut, lalu setelah pekerjaan mahasiswa selesai pengawas akan memberikan point - point kepada mahasiswa yang telah melakukan pekerjaannya, lalu pengawas akan memberikan data tersebut kepada admin Kampus setelah itu admin akan mengupload data tersebut ke website Kampus.

#### 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka perumusan masalah dalam pembuatan sistem ini adalah "Bagaimana cara meningkatkan layanan *redemption* terhadap ,admin,pengawas dan mahasiswa berbasis android pada program E-Redemption?".

#### 1.3 Batasan Masalah

Batasan masalah dalam pengerjaan skripsi ini antara lain:

- a. Data dan kasus yang digunakan dalam penelitian ini adalah data dari para Mahasiswa yang terdaftar di Kampus.
- b. Sistem ini hanya untuk mengurangi penilaian jelek terhadap seorang mahasiswa.
- c. Database yang dibangun menggunakan MYSQL.

- d. Teknologi yang digunakan untuk membuat Program menggunakan Android Studio dan Pembuatan API menggunakan Laravel.
- e. Sistem dapat dijalankan dengan koneksi internet.
- b. Metode Pengembangan Sistem

Dalam pengembangan sistem, digunakan pendekatan model waterfall , Waterfall adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (Muhammad & Ariani, 2016). Pendekatan waterfall melingkupi fase-fase sebagai berikut:

1. Analisis Kebutuhan  
Melakukan analisis yang dilakukan dalam permasalahan yang akan diselesaikan dengan melakukan survei kepada mahasiswa aktif, dengan menyebarkan kuesioner melalui google form dan wawancara terhadap Admin.
2. Desain Sistem  
Melakukan proses desain dan juga perbaikan terhadap Aplikasi apabila masih terdapat ketidaksesuaian desain atau rancangan.
3. Alat Pemrograman  
Proses pengkodean program dari desain sistem yang telah dirancang sebelumnya, dan proses pembuatan sistem ini menggunakan framework Laravel dengan menggunakan database MYSQL dan Android Studio untuk pembuatan aplikasi.
4. Pengujian  
Setelah sistem menjadi suatu Aplikasi yang siap pakai, maka sistem harus dilakukan uji coba terlebih dahulu sebelum digunakan. Tahap ini dilakukan untuk mengetahui apakah Aplikasi tersebut terdapat kesalahan dalam sistem ataupun kesalahan dalam bisnis prosesnya.

## 2.1 Penelitian Terdahulu

PENELITIAN Sejenis					
NO	Tahun	Artikel	Data	Metode	Hasil
1	2017	Aplikasi Pengolahan Data Pelanggaran Siswa Pada SMK Yayasan Pendidikan Teknologi (YPT) 1 Purbalingga Terintegrasi Dengan SMS Gateway	Berdasarkan data dari SMK Yayasan Pendidikan Teknologi (YPT) 1 Purbalingga didapatkan Tabel point dan jenis pelanggaran siswa	Metode Analisa dan Data flow Diagram	Menangani pelanggaran siswa dan cara mengirim laporan pelanggaran pada orang tua/wali dapat dilakukan oleh Guru BK melalui SMS Gateway dengan cepat, akurat dan lebih mudah
2	2018	Sistem Informasi Poin Pelanggaran Siswa (SIMPONIS)	Berdasarkan pengelolaan data dari pihak yang berwenang adalah petugas PKS (Patroli Keamanan Sekolah) Pihak tersebut akan mencatat pelanggaran dan memberikan poin sesuai dengan peraturan	Studi ini menggunakan pendekatan kualitatif dengan obyek studinya adalah SMP Negeri 2 Cepogo Boyolali. Metode penelitian yang digunakan adalah metode penelitian tindakan	Adapun hasilnya sistem dapat mempercepat dan juga mempermudah proses perhitungan poin pelanggaran siswa sehingga dapat mempermudah kinerja guru BK dan juga para wali kelas dalam menangani

## 2.2 Android

Aplikasi mobile yang berbasis android di jaman sekarang ini telah mencapai kemajuan yang sangat besar terutama aplikasi yang berhubungan dengan pendataan mahasiswa dalam sebuah seminar. Sistem aplikasi yang berhubungan dengan pendataan mahasiswa merupakan salah satu bagian yang sangat penting dalam pengelolaan data. Penggunaan sistem yang terkomputerisasi akan lebih banyak menghemat waktu, tidak banyak menyita tenaga, dan menghasilkan keakuratan penyajian data apalagi ditambah dengan sistem database sebagai media penyimpanan datanya (Roihan, et al., 2019). Pada awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel atau smartphone. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance. Android bersama Open Handset Alliance menyatakan mendukung pengembangan open source pada

perangkat mobile. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan open platform perangkat seluler. Android dapat dikembangkan dengan menggunakan software gratis dan resmi seperti Android Studio dan Eclipse. Android Studio adalah IDE (Integrated Development Environment) resmi untuk pengembangan dan pemrograman aplikasi Android dan bersifat open source atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 Mei 2013 pada event Google I/O Conference untuk tahun 2013. Sejak saat itu, Android Studio 7 menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android. Android Studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse disertai dengan ADT plugin (Android Development Tools). Berdasarkan penjelasan tersebut, Android merupakan generasi baru platform mobile yang memberikan banyak fasilitas pengembangan yang free dan open source sehingga saat ini banyak dipakai oleh programmer untuk mengembangkan aplikasinya ke ranah mobile. Jadi Aplikasi android dapat dengan mudah dikembangkan menggunakan IDE Android Studio dari Google karena telah banyak plugin yang mendukung untuk pengembangan aplikasi. Umumnya Android Studio menggunakan bahasa pemrograman berbasis Java untuk pengembangannya.

### 2.3 Kotlin

Dikutip dari situs [codepolitan.com](http://codepolitan.com), Kotlin merupakan bahasa pemrograman pragmatis untuk JVM dan Android yang digabungkan dengan Object Oriented dan fitur fungsional yang berfokus fokus pada interoperabilitas, keamanan, kejelasan dan dukungan integrasi dengan berbagai tools major. Sebagai bahasa pemrograman dengan tujuan umum, Kotlin bisa berjalan dimanapun layaknya Java. Kotlin bisa digunakan untuk mengembangkan berbagai aplikasi server-side, aplikasi mobile (Android), aplikasi desktop (Permana, 2016).

Bahasa pemrograman yang relatif baru ini mengedepankan produktifitas. Kotlin hadir dengan fitur-fitur yang memudahkan dalam pembuatan kode program. Berikut ini adalah fitur-fitur tersebut: (Ardiansyah, 2017).

- a. Aman dari Null, seperti yang sudah disebutkan diatas, tidak perlu lagi menghawatirkan kemungkinan timbulnya Null Pointer Exception. Compiler secara sistematis akan menandai pointer yang berpotensi null.
- b. Pemrograman fungsional, mendukung lambda dan kemampuan mapping, folding, pada Collection java.

- c. Anotasi Data yang otomatis dibuat untuk mengurangi boilerplate seperti equals, hashCode, toString.
- d. Syntax Ramping, yang menawarkan penulisan syntax secara cepat. Membuat fungsi one-liner, membuat Java Beans hanya dengan satu baris, method setter dan getter otomatis.
- e. Ekstensi fungsi yang memungkinkan untuk menambahkan method pada class tanpa harus mengubah source codenya.
- f. Interpolasi String yang memungkinkan untuk memasukkan variabel ke dalam String tanpa penyambungan seperti pada EcmaScript.
- g. Masih banyak lagi, seperti penamaan argumen, parameter opsional, default value untuk parameter.

#### 2.4 Laravel

Laravel dirilis dibawah lisensi MIT dengan kode sumber yang sudah disediakan oleh Github, sama seperti framework-framework yang lain, Laravel dibangun dengan konsep MVC (Model-Controller-View), kemudian Laravel dilengkapi juga command line tool yang bernama “Artisan” yang bisa digunakan untuk packaging bundle dan instalasi bundle melalui command prompt (Aminudin, 2015).

#### 2.5 JSON

JSON adalah suatu format pertukaran data yang memiliki tipe data light-weight keyvalue. JSON sendiri adalah JavaScript Object Notation dimana Formatnya berbentuk teks serta dapat dibaca oleh manusia dan juga digunakan untuk menggambarkan struktur data sederhana dan larik asosiatif (disebut objek). Format JSON acapkali digunakan untuk mentransmisikan data terstruktur melalui suatu koneksi jaringan pada suatu proses yang disebut serialisasi (Roihan, et al., 2019).

#### 2.6 API

API adalah singkatan dari Application Programming Interface, dan memungkinkan developer untuk mengintegrasikan dua bagian dari aplikasi atau dengan aplikasi yang berbeda secara bersamaan. API terdiri dari berbagai elemen seperti function, protocols, dan tools lainnya yang memungkinkan developers untuk membuat aplikasi. Tujuan penggunaan API adalah untuk mempercepat proses development dengan menyediakan function secara terpisah sehingga developer tidak perlu membuat fitur yang serupa. Penerapan API akan sangat terasa jika fitur yang diinginkan sudah sangat kompleks, tentu membutuhkan waktu untuk membuat yang serupa dengannya.

Misalnya: integrasi dengan payment gateway. Terdapat berbagai jenis sistem API yang dapat digunakan, termasuk sistem operasi, library, dan web (Sandi, 2017).

## 2.7 MySQL

Oracle menyediakan server database MySQL dan Perpustakaan Klien MySQL di bawah model lisensi ganda yang dirancang untuk memenuhi kebutuhan pengembangan dan distribusi dari kedua distributor komersial (seperti OEM, ISV dan VAR) dan proyek sumber terbuka., indeks, pengguna, izin, dll) dapat dilakukan melalui antarmuka pengguna (Public, 2010).


## 2.8 Unified Modeling Language (UML)

Menurut (Muhammad & Ariani, 2016) UML berfungsi untuk melakukan standarisasi pemodelan untuk perangkat lunak yang dibuat dan perlu agar mempermudah dalam tahap pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek. kemudian UML yang dapat digunakan dalam membangun sistem diantaranya adalah Use Case Diagram, Activity Diagram, Sequence Diagram dan Class Diagram. Jadi berdasarkan penjelasan diatas dapat disimpulkan bahwa UML muncul dikarenakan adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan dan mendokumentasi dari sistem perangkat lunak. Berikut penjelasan dari masing-masing diagram:

### 2.8.1 Use Case Diagram

Menurut (Muhammad & Ariani, 2016) Use case diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat, secara kasar use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi- fungsi itu

Tabel 1 .Simbol Use Case Diagram

No.	Simbol	Deskripsi
1.	<p><i>Use Case</i></p> <p>Nama use case</p>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal <i>frase</i> nama <i>use case</i> .
2.	<p>Aktor/<i>actor</i></p> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan

No.	Simbol	Deskripsi
		menggunakan kata benda di awal <i>frase</i> nama aktor.
3.	Assosiasi/ <i>association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.	Exstensi/ <i>extend</i>  <<extend>>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.
5.	Menggunakan/ <i>include</i>  <<include>>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.
6.	Generalisasi/ <i>generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

### 2.8.2 Activity Diagram

Activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak, yang perlu di perhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Muhammad & Ariani, 2016)

Tabel 2 .Simbol Activity Diagram

No	Simbol	Deskripsi
1.	Status awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan/decision	Asosiasi percabangan dimana jika adapilihan aktivitas lebih dari satu.
4.	Penggabungan/join	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

### 2.8.3 Class Diagram

(Muhammad & Ariani, 2016) menjelaskan bahwa class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem, kelas memiliki apa yang disebut atribut dan method atau operasi. Diagram Kelas dibuat agar pengguna atau programmer membuat kelas- kelas yang sesuai rancangan didalam kelas agar dokumentasi dan perangkat lunak sinkron.

### 2.9 Waterfall Model

SDLC atau Software Development Life Cycle adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (Muhammad & Ariani, 2016). Air terjun (Waterfall) sering disebut juga model sekuensial linier (sequential linear) atau alur hidup klasik menjelaskan bahwa model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari:

1. Analisis kebutuhan



Proses pengumpulan kebutuhan dilakukan secara intensif untuk mengspesifikasikan kebutuhan perangkat lunak agar dapat mudah dipahami dan menjadi perangkat lunak yang seperti dibutuhkan oleh user.

## 2. Desain Sistem

Proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean.

## 3. Pemrograman

Pemrograman harus ditransletkan kedalam perangkat lunak, hasil dari tahapan ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahapan sebelumnya.

## 4. Pengujian

Pengujian fokus pada perangkat lunak, secara segi logik dan fungsional untuk memastikan bahwa semua bagian sudah diuji, hal ini dilakukan untuk meminimalisir

### 2.10 Black Box Testing

Menurut (Jaya, 2018) Black-Box Testing merupakan Teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak. Blackbox Testing bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya difokuskan pada informasi domain. Blackbox Testing memungkinkan pengembang software untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Keuntungan penggunaan metode Blackbox Testing adalah Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu, Pengujian dilakukan dari sudut pandang pengguna, ini membantu untuk mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan; Programmer dan tester keduanya saling bergantung satu sama lain. Kekurangan dari metode Blackbox Testing adalah Uji kasus sulit disain tanpa spesifikasi yang jelas; Kemungkinan memiliki pengulangan tes yang sudah dilakukan oleh programmer; Beberapa bagian back end tidak diuji sama sekali. Saat ini terdapat banyak metoda atau teknik untuk melaksanakan Black Box Testing, antara lain Equivalence Class Partitioning, Boundary Value Analysis dan Error Guessykesalahan dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

### **3.1 Perancangan Program**

Proses perancangan sistem dari aplikasi E-Redemption menggunakan kotlin android untuk aplikasi mobilyenya serta menggunakan framework laravel untuk aplikasi API dan dikembangkan dengan Waterfall. Metode waterfall ini memiliki beberapa pendekatan terstruktur mulai dari analisis kebutuhan, desain sistem, pemrograman, dan pengujian. Pada bab ini akan menjelaskan tentang deskripsi aplikasi, cara kerja aplikasi, dan penjabaran dari pendekatan metode waterfall.

#### **3.1.1 Deskripsi Aplikasi**

Aplikasi yang dirancang dinamai dengan E-Redemption, merupakan aplikasi yang hadir untuk menangani masalah-masalah yang ada di lingkungan kampus khususnya pada masalah kompensasi mahasiswa dalam mengantri tugas dan mengerjakan tugas, aplikasi ini hanya dikhususkan untuk warga Kampus E-Redemption .Pada aplikasi ini terdapat beberapa fitur,yaitu:

##### **A. Fitur Pengambilan Tugas**

Fitur pengambilan tugas kompen adalah bagian aplikasi yang menampilkan sebuah pekerjaan yang dikirim oleh Admin(tendik) yang berfungsi untuk mengurangi jam kompen dari mahasiswa.

##### **B. Fitur Pembuatan Tugas**

Fitur pembuatan tugas kompen adalah bagian aplikasi yang memberikan pekerjaan kepada mahasiswa untuk mengurangi kompen dari mahasiswa yang mengambil pekerjaan tersebut dengan point-point jam yang sudah di tentukan oleh Admin.

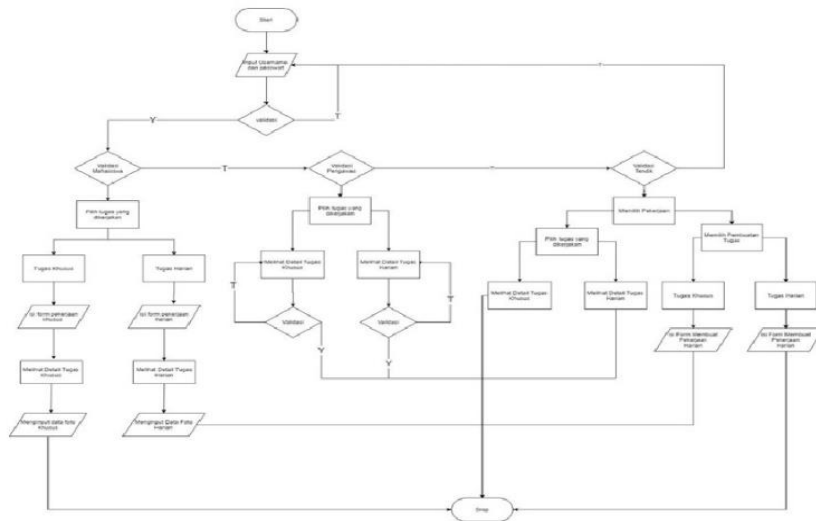
##### **C. Fitur Pengawasan Tugas**

Fitur Pengawasan Tugas Kompen adalah fitur aplikasi dimana pengawas dapat melihat tugas apa saja yang mahasiswa kerjakan dan memverifikasi tugas yang sudah terlaksanakan oleh mahasiswa.

#### **3.1.2 Cara Kerja Program**

Gambar 3.1 menunjukkan Mahasiswa yang akan melakukan kompensasi hanya butuh membuka aplikasi E-Redemption lalu mahasiswa tersebut diharuskan login terlebih dahulu lalu memilih tugas yang akan dikerjakan lalu mahasiswa harus mengambil gambar untuk bukti bahwa mahasiswa tersebut melakukan tugas nya, admin(tendik) akan memberikan tugas kepada mahasiswa lalu

pengawas akan memverifikasi data mahasiswa yang telah selesai melakukan tugas kompen.



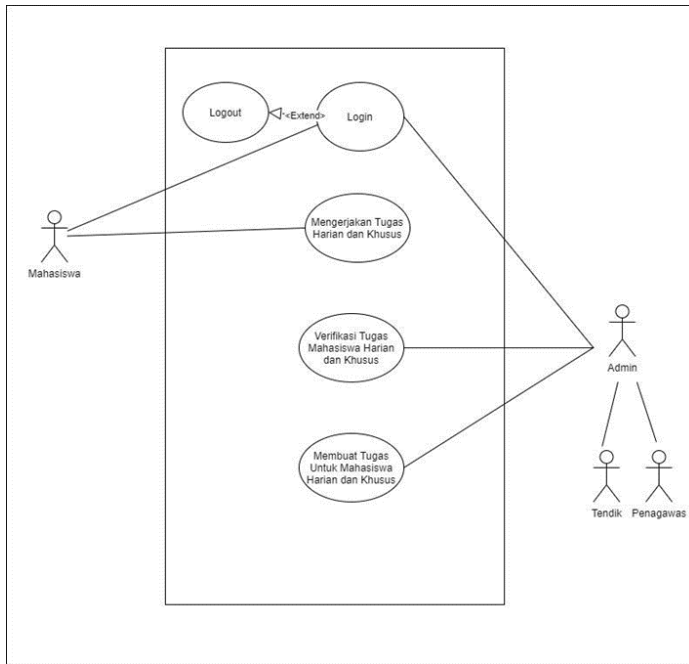
Gambar 3.1 Flowchat Mahasiswa

### 3.1.3 Desain Sistem

Tahap ini adalah proses perancangan sistem yang akan dibangun sebelum masuk ke tahap realisasi atau pemrograman. Desain sistem dapat dilakukan setelah mendapatkan hasil dari analisis kebutuhan user dan analisis sistem yang sedang berjalan, pada tahap ini akan dijelaskan dengan bahasa pemodelan berorientasi object atau UML, UML yang akan di jabarkan adalah Use Case Diagram, Activity Diagram dan Class Diagram.

#### a. Use Case Diagram

Pada use case diagram ini memiliki 3 aktor dengan 3 role berbeda. Gambar 3.2 merupakan use case diagram untuk melakukan banyak aktifitas lainnya, Mahasiswa, pengawas dan admin(tendik) harus melakukan login terlebih dahulu, setelah itu admin(tendik) memberikan tugas kepada mahasiswa, mahasiswa tersebut mengambil tugas yang telah diberikan tendi setelah itu pengawas akan memverifikasi tugas yang sudah dikerjakan mahasiswa, setelah itu admin(tendik) melihat verifikasi tugas yang di berikan oleh pengawas.

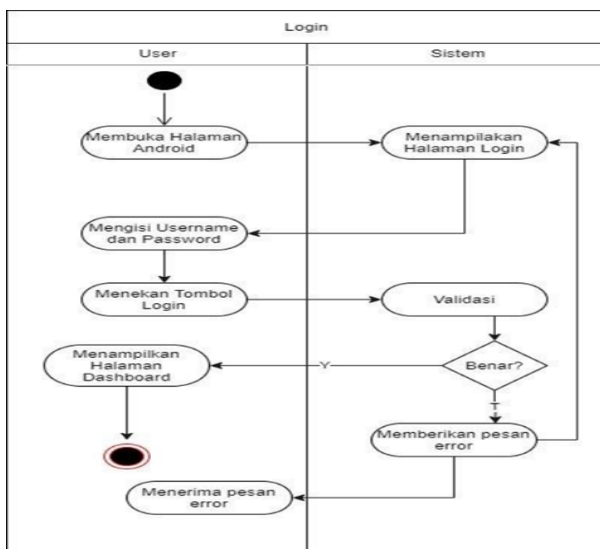


Gambar 3.2 Use Case Diagram

b. Activity Diagram

1. Login

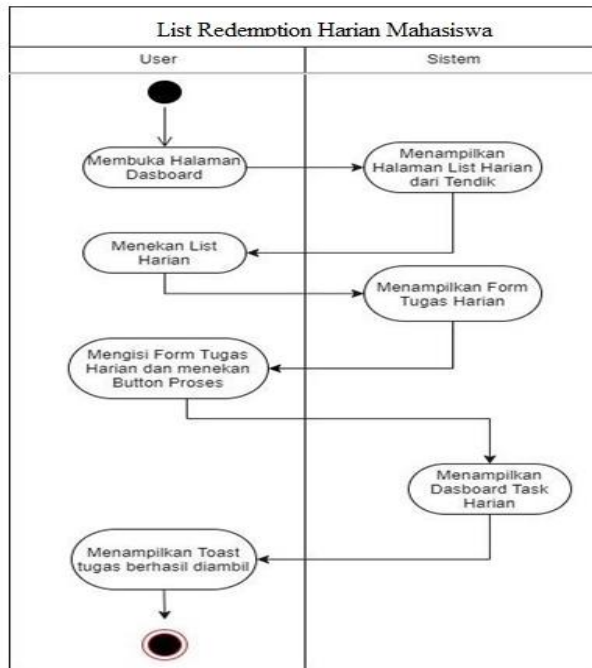
Aktifitas ini dilakukan oleh mahasiswa, pengawas dan tendik untuk melanjutkan aktifitasnya ke halaman berikutnya. Gambar 3.3 merupakan activity diagram dari proses login. User diharuskan melakukan login lalu menginput Username dan password menjadi syarat untuk proses validasi pada login, jika login berhasil maka sistem akan mengantarkan user ke halaman dashboard dari aplikasi dan jika gagal maka sistem akan mengembalikan user ke halaman login dengan pesan jika login gagal.



Gambar 3.3 Activity Diagram

## 2. Pengambilan Tugas Redemption Harian

Aktifitas ini dilakukan oleh mahasiswa untuk pengambilan tugas harian. Gambar 3.4 bertujuan untuk mahasiswa mengambil tugas harian diharuskan melihat list tugas harian yang sudah diberikan oleh Admin (tendik), setelah selesai memilih tugas harian yang ingin dikerjakan mahasiswa diharuskan menginput form tugas harian, setelah selesai mahasiswa harus menekan tombol proses setelah itu akan ada toast jika mahasiswa tersebut telah mengambil tugas tersebut.



Gambar 3.4 Activity diagram list Harian

## 3. Pengambilan Tugas Redemption Khusus

Aktifitas ini dilakukan oleh mahasiswa untuk pengambilan tugas khusus. Gambar 3.5 bertujuan untuk mahasiswa mengambil tugas khusus diharuskan melihat list tugas khusus yang sudah diberikan oleh Admin (tendik), setelah selesai memilih tugas khusus yang ingin dikerjakan mahasiswa diharuskan menginput form tugas khusus, setelah selesai mahasiswa harus menekan tombol proses setelah itu akan ada toast jika mahasiswa tersebut telah mengambil tugas tersebut.



Gambar 3.5 Activity diagram list Khusus

#### 4. Pengawas Verifikasi Tugas Khusus

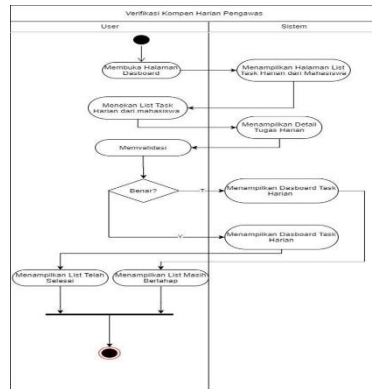
Aktifitas ini dilakukan oleh pengawas untuk verifikasi mahasiswa yang telah selesai melakukan tugas khusus. Gambar 3.6 bertujuan untuk pengawas melihat list tugas khusus, pengawas akan melihat detail tugas khusus mahasiswa, setelah itu pengawas melihat tugas mahasiswa tersebut apakah sudah memenuhi syarat untuk memverifikasi tugas tersebut jika ya pengawas akan kembali kehalaman dashboard jika tidak, tugas khusus mahasiswa tersebut akan berstatus bertahap.

Gambar 3.6 Activity diagram Verifikasi Khusus Pengawas

#### 5. Pengawas Verifikasi Tugas Harian

Aktifitas ini dilakukan oleh pengawas untuk verifikasi mahasiswa yang telah selesai melakukan tugas harian. Gambar 3.7 bertujuan untuk pengawas melihat

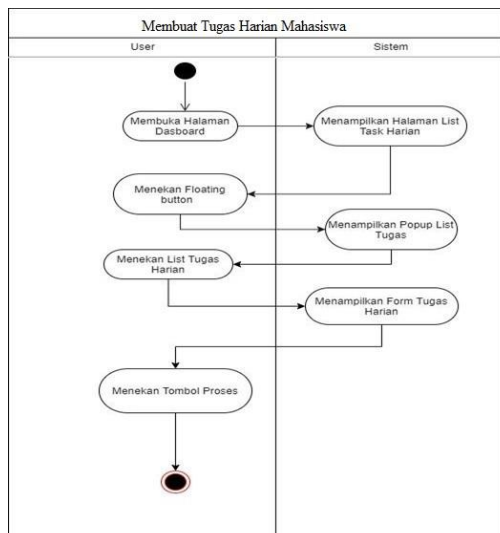
list tugas harian, pengawas akan melihat detail tugas khusus mahasiswa, setelah itu pengawas melihat tugas mahasiswa tersebut apakah sudah memenuhi syarat untuk memverifikasi tugas tersebut jika ya pengawas akan kembali kehalaman dashboard jika tidak, tugas khusus mahasiswa tersebut akan berstatus bertahap.



Gambar 3.8 Activity diagram Membuat Tugas Harian Mahasiswa

#### 6. Admin Membuat Tugas Harian

Aktifitas ini dilakukan oleh Admin (tendik) untuk membuat tugas Harian mahasiswa. Gambar 3.8 bertujuan untuk admin membuat tugas dengan proses, dari dashboard pilih kehalaman list lalu menekan tombol floating button untuk memilih tugas harian setelah itu isi form tugas harian lalu menekan tombol proses untuk memberikan tugas harian kepada mahasiswa.

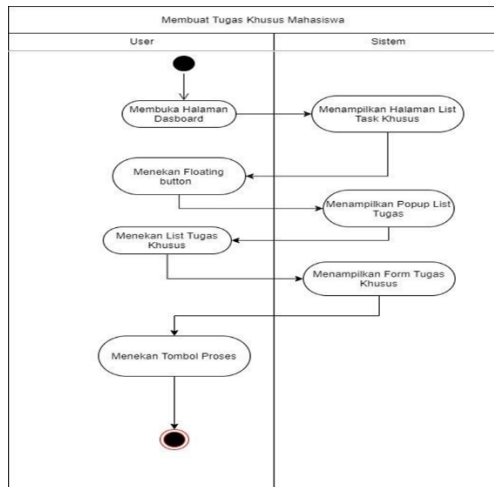


Gambar 3.8 Activity diagram Membuat Tugas Harian Mahasiswa

#### 7. Admin Membuat Tugas Khusus

Aktifitas ini dilakukan oleh Admin (tendik) untuk membuat tugas khusus mahasiswa. Gambar 3.9 bertujuan untuk admin membuat tugas dengan proses, dari dashboard lalu pilih kehalaman list lalu menekan tombol floating button untuk memilih tugas khusus

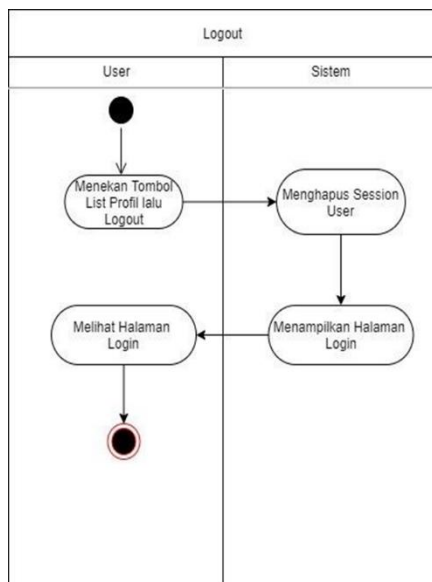
setelah itu isi form tugas khusus lalu menekan tombol proses untuk memberikan tugas khusus kepada mahasiswa.



Gambar 3.9 Activity diagram Membuat Tugas Khusus Mahasiswa

## 8. Logout

Aktifitas ini dilakukan oleh mahasiswa, pengawas dan tendik untuk melanjutkan aktifitasnya ke halaman berikutnya. Gambar 3.10 dimana user saat berada di dashboard diharuskan klik tombol button logout lalu system akan menghapus session user lalu system akan menampilkan halaman login.



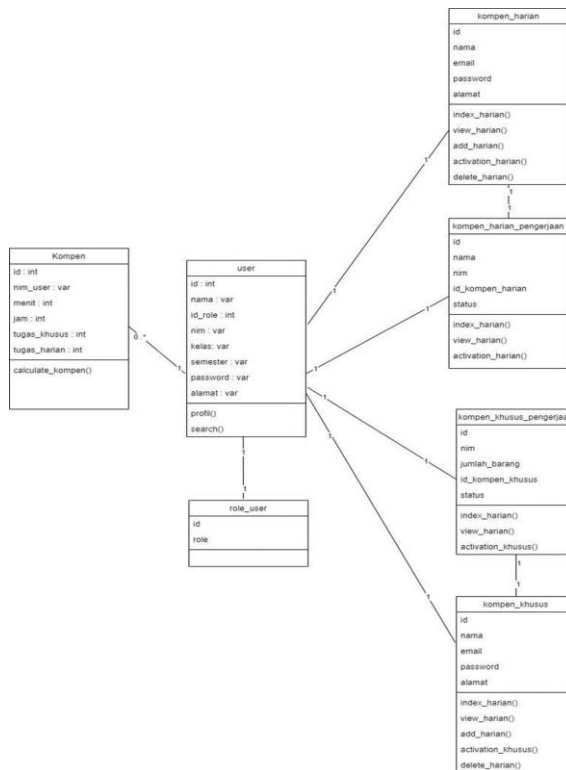
Gambar 3.10 Activity diagram Logout

### c. Class Diagram

Gambar 3.11 merupakan class diagram dari modul admin. Pada bagian ini, diagram yang dijelaskan berkaitan dengan model class dan class controller yang digunakan dalam pengkodean pada tahap berikutnya, terdapat class user dimana ada



3 role dimana class user tersebut memiliki function profil dan search, class komponen dimana memiliki terkaitan dengan class user dimana tidak semua user memiliki komponen karena memiliki 3 role yang berbeda. Class redemption harian dimana memiliki function index, view, add, activation dan delete. Class redemption harian pengerjaan dimana disetiap task nya memiliki 1 task redemption dan sama seperti class redemption khusus dan pengerjaan khusus redemption.



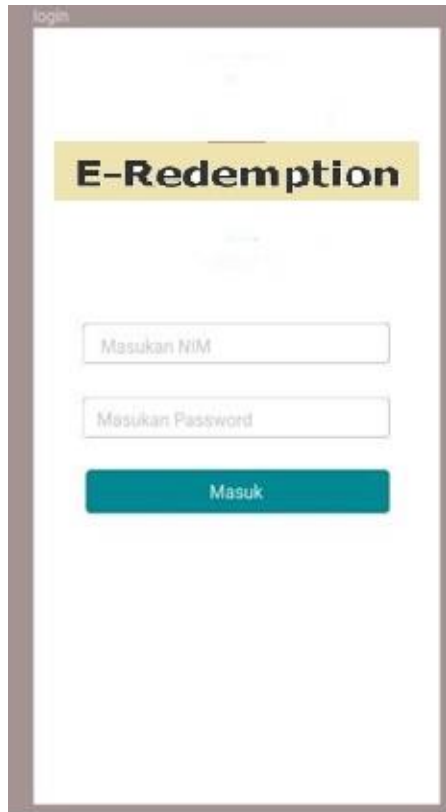
Gambar 3.11 Class Diagram

### 3.2 Realisasi Program Aplikasi

Bagian ini akan dijelaskan realisasi dari perancangan sistem yang dibangun. Pembahasan pada bagian ini dikhususkan pada implementasi modul admin aplikasi autentik, karena source code yang digunakan untuk membangun API memiliki banyak kesamaan dengan source code yang digunakan untuk implementasi modul admin.

#### a. Halaman Login

Gambar 3.12 adalah halaman login pada Aplikasi. Halaman ini merupakan halaman awal sebelum user melakukan aktifitas lainnya yang tersedia di dalam fitur aplikasi. Terdapat isian berupa username dan password sebagai salah satu syarat untuk dapat melakukan login.



Gambar 3.12 Halaman login

b. Halaman List Redemption Khusus

Halaman ini merupakan halaman untuk membuat pekerjaan yang sudah di berikan oleh admin. Gambar 3.13 menjelaskan bahwa terdapat tugas – tugas yang harus dikerjakan dimana ada jumlah barang yang ada, point yang didapat dan nama barang yang harus dibelikan.

Mhs List pekerjaan

List Redemption Khusus Mahasiswa

Task Khusus	Task harian	List Khusus	List harian
Jumlah	Point	Nama Barang	
20	20	AC	
Jumlah	Point	Nama Barang	
20	20	Lem	

Gambar 3.13 Halaman List Redemption Khusus Mahasiswa

c. Halaman Form Redemption Khusus

Halaman ini dimaksudkan untuk mahasiswa yang ingin membuat tugas kompen khusus, pada gambar 3.14 mahasiswa diharuskan mengisi form pengugasan yang telah mahasiswa pilih mahasiswa bisa memilih berkelompok atau tidak.

Form Kompen Khusus

## Form Redemption Khusus

Nama barang  
Lemari

Jumlah point  
20

Jumlah Barang  
20

Kelompok
  Tidak Kelompok

Masukan Nim

Tambah

Kelompok

4616010033	Abdul wahid	✘
<u>4616010033</u>	Abdul wahid	✘
4616010033	Abdul wahid	✘

Proses

Gambar 3.14 Halaman Form Redemption Khusus

d. Halaman Form Redemption Harian

Halaman ini adalah untuk mahasiswa yang ingin membuat tugas kompen Harian, pada gambar 3.15 mahasiswa diharuskan mengisi form pengugasan yang telah mahasiswa pilih mahasiswa bisa memilih berkelompok atau tidak.

Gambar 3.15 Halaman Form Redemption Harian

e. Halaman List Pekerjaan Redemption Khusus Pengawas

Halaman ini adalah halaman untuk list kompen mahasiswa, pada gambar 3.16 terdapat kompen dari mahasiswa yang sudah menyelesaikan tugas kompen nya dan terdapat nim,nama,nama barang yang sudah di beli dan status masih bertahap.

Task Khusus	Task harian	Verifikasi Khusus	Verifikasi Harian
4616010033 Abdul wahid Lem 20			Tahap
4616010000 Yono yan Lem 20			Tahap

Gambar 3.16 Halaman List Khusus Pengawas

f. Halaman Detail Verifikasi Redemption Khusus Pengawas

Halaman ini merupakan halaman dari pengawas untuk melihat tugas mahasiswa, seperti yang terlihat pada gambar 3.17 dibawah ada detail dari setiap pekerjaan yang mahasiswa kerjakan lalu ada tombol untuk verifikasi apakah tugas mahasiswa tersebut layak untuk di verifikasi atau tidak.



Detail Form Kompen Pengawas	
Verifikasi Redemption Khusus Pengawas	
Nama Mahasiswa	Abdul Wahid
NIM	4616010033
Kelas	TI-8B
Jumlah Kelompok	443435,45543543,4543543,3423443
Nama barang	Lem
Jumlah Barang	20
Tugas Yang dikerjakan	Tugas Khusus
Jumlah Point	20
Photo	
<input type="button" value="YA"/> <input type="button" value="Tidak"/>	

Gambar 3.17 Halaman Verifikasi Redemption Khusus Pengawas

g. Halaman Create redemption khusus

Halaman ini adalah halaman untuk create kompen mahasiswa, pada gambar 3.18 terdapat form untuk membuat barang yang akan ditugaskan kepada mahasiswa yang melakukan tugas redemption khusus.

Form Kompen Khusus and edit

### Task Redemption Khusus

Nama Barang:

Jumlah Barang:

20 — +

Masukan Point:

Masukan harga satuan:

Masukan Nota:

Date

Proses

Gambar 3.17 Halaman Task Redemption Khusus.

## Source Code JSP

```
public function login(Request $request){ try{
    $loginData = $request->validate([ 'nim' => 'required', 'password' => 'required'
    ]);
    if( !auth()->attempt($loginData) ){
        return SendResponse::fail('Invalid Login Data!',403);
    };
    $accessToken = auth()->user()->createToken('authToken')-
    >accessToken;
    $data['user'] = User::find(Auth::user()->id);
    $data['access_token'] = $accessToken; return SendResponse::success($data,200);
}catch(\Exception $e){
    return SendResponse::fail($e->getMessage(),500);
}
public function kompenKhususList(){ try{
    $data=RedemptionKhususPengerjaanTable::with(['kompen','mahasiswa','kelompok'])-
    >orderBy('created_at','DESC');
    if(Auth::user()->role == 'mahasiswa'){
        $data = $data->where('nim_user',Auth::user()->nim);
    }else{
        $data = $data->with(['mahasiswa','kelompok'])-
        >whereHas('mahasiswa', function($query) {
            $query-
            >where('kelas','like','%'.Auth::user()->kelas.'%');
        });
    }
    $data = $data->get(); if($data->isEmpty()){
        return SendResponse::message('Tidak ada data',200);
    }
    return SendResponse::success($data,200);
}catch(\Exception $e){
    return SendResponse::fail($e->getMessage(),500);
}
}
public function startRedemptionKhusus(Request $request,$id_task){ try{
    $kh = KompenKhususTable::find($id_task); if(!$kh || $kh->status == 0){
```



```

return SendResponse::fail('Tugas redemption tidak ditemukan',404);
}

if($request->jumlah_barang > $kh->jumlah_barang){
return SendResponse::fail('Jumlah barang melebihi kapasitas',403);
}

if($request->kelompok){
$skels = explode(",",$request->kelompok);
$notfound = 0; foreach($skels as $skel){
$test[] = $skel;
$ms = User::where('nim',$skel)->first();
$notfound += !$ms? 1 : 0; if(Auth::user()->nim == $skel){
$notfound += 1;
}
}
}

if($notfound != 0){
return SendResponse::fail('Terdapat NIM user yang tidak tersedia',400);
}
}

$task['id_redemption_Khusus'] = $id_task;
$task['jumlah_barang'] = $request->jumlah_barang;
$task['nim_user'] = Auth::user()->nim;
$task['tanggal_diambil'] = Carbon::now();
$task['status'] = 1;

}

//do komponen, create data komponen_Khusus_pengerjaan
$skhp_table = ItemManager::insert($task,'App\Models\Komponen_KhususPengerjaanTable');

if($request->kelompok){
$skels = explode(",",$request->kelompok); foreach($skels as $skel){

```

```
$kelh = new KelompokKhusus;
$kelh->id_kompen_Khusus_pengerjaan = $khp_table->id;
$kelh->nim = $kel;
$kelh->save();
}
}

//update status kompen_Khusus
$kh->jumlah_barang = $kh->jumlah_barang - $khp_table->jumlah_barang;
$kh->save();

$shp_table['kelompok'] = KelompokKhusus::where('id_kompen_Khusus_pengerjaan',$khp_table->id)->get();

return SendResponse::success($shp_table,200);
}catch(Exception $e){
return SendResponse::fail($e->getMessage(),500);
}
}

public function startKompenHarian(Request $request,$id_task){ try{
$kh = KompenHarianTable::find($id_task); if(!$kh || $kh->status == 0){
return SendResponse::fail('Tugas kompen tidak ditemukan',404);
}

if($request->kelompok){
$kels = explode(",",$request->kelompok);
$notfound = 0; foreach($kels as $kel){
$test[] = $kel;
$ms = User::where('nim',$kel)->first();
$notfound += !$ms? 1 : 0; if(Auth::user()->nim == $kel){
$notfound += 1;
}
}
}
```

```

if($notfound != 0){
return SendResponse::fail('Terdapat NIM user yang tidak tersedia',400);
}
}

$task['id_kompen_harian'] = $id_task;
$task['nim_user'] = Auth::user()->nim;
$task['tanggal_diambil'] = Carbon::now();
$task['status'] = 1;
//do kompen, create data kompen_harian_pengerjaan
$khp_table = ItemManager::insert($task,'App\Models\Kompen HarianPengerjaanTable');

if($request->kelompok){
$skels = explode(",",$request->kelompok); foreach($skels as $skel){
$kelh = new KelompokHarian;
$kelh->id_kompen_harian_pengerjaan = $khp_table->id;
$kelh->nim = $skel;
$kelh->save();
}
}

//update status kompen_harian
$kh->status = 0;
$kh->save();

$khp_table['kelompok'] = KelompokHarian::where('id_kompen_harian_pengerjaan',$khp_table->id)->get();

return SendResponse::success($khp_table,200);
}catch(\Exception $e){
return SendResponse::fail($e->getMessage(),500);
}
}

public function kompenKhususList(){ try{

```

```

$data = KompenKhususPengerjaanTable::with(['kompen','maha siswa','kelompok'])-
>orderBy('created_at','DESC');
if(Auth::user()->role == 'mahasiswa'){
$data = $data->where('nim_user',Auth::user()->nim);
}else{
$data = $data->with(['mahasiswa','kelompok'])-
>whereHas('mahasiswa', function($query) {
$query-
>where('kelas','like','%'.Auth::user()->kelas.'%');
});
}
$data = $data->get(); if($data->isEmpty()){
return SendResponse::message('Tidak ada data',200);
}
return SendResponse::success($data,200);
}catch(\Exception $e){
return SendResponse::fail($e->getMessage(),500);
}
}
public function kompenKhususDetail($id){

try{
$data = KompenKhususPengerjaanTable::with(['kompen','maha siswa','kelompok'])-
>where('id',$id)->first();
if(!$data){
return SendResponse::fail('Data tidak ditemukan',404)
;
}
if(Auth::user()->role == 'mahasiswa'){
$nim = Auth::user()->nim; if($data->nim_user != $nim){
$indexKel = 0; foreach($data['kelompok'] as $skel){
if($skel['nim'] == Auth::user()->nim){
$indexKel +=1;
}
}
if($indexKel != 1){

```

```

return SendResponse::fail('Data tidak ditemuk
an',404);
}
}
}
return SendResponse::success($data,200);
}catch(\Exception $e){
return SendResponse::fail($e->getMessage(),500);
}
}
public function kompenKhususConfirm(Request $request,$id){

try{
if(Auth::user()->role != 'pengawas'){
return SendResponse::fail('Anda tidak dapat melakukan
ini',403);
}
$kompen = KompenKhususPengerjaanTable::with('kompen')-
>find($id);
if(!$kompen){
return SendResponse::fail('Data tidak ditemukan',404)
;
}

$status = $request->status;
if(!$status || $status != 2 && $status != 3 ){
return SendResponse::fail('Status tidak valid',400);
}else{
$kompen->status = $status;
$kompen->save();
}
if($status == 2){
$task = KompenKhususTable::find($kompen-
>id_kompen_khusus);
$this->calculateKompen($task->id,$kompen->id);
}
}

```

```

return SendResponse::success($kompen,200);
}catch(\Exception $e){
return SendResponse::fail($e->getMessage(),500);
}
}

Public function updateKhusus(Request $request,$id){

    try{

        if(Auth::user()->role == 'mahasiswa'){

            return SendResponse::fail('Role anda tidak dapat melakukan
ini',403);

        }

        $task = KompenKhususTable::find($id);if(!$task){

            return SendResponse::fail('Tidak ditemukan',404);

        }

        $req = $request->all();

        $req['harga_total'] = $req['jumlah_barang'] * $req['harga
_satuan'];

        $update = ItemManager::update($req,$id,'App\Models\Kompen
KhususTable');

        $data = KompenKhususTable::find($id); return
SendResponse::success($data,200);

    }catch(\Exception $e){

        return SendResponse::fail($e->getMessage(),500);

    }

}
}

```