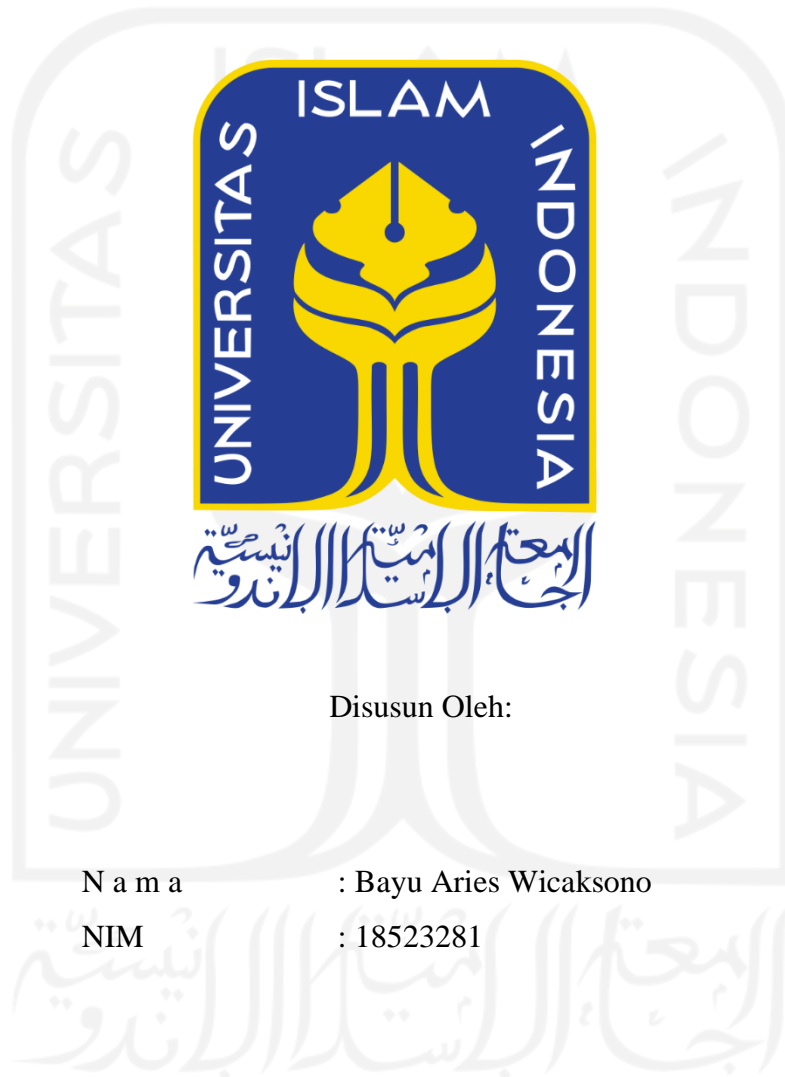


# **IMPLEMENTASI REST API UNTUK PENGEMBANGAN SISTEM PELAPORAN DAN PENGADUAN**



Disusun Oleh:

N a m a : Bayu Aries Wicaksono

NIM : 18523281

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA**

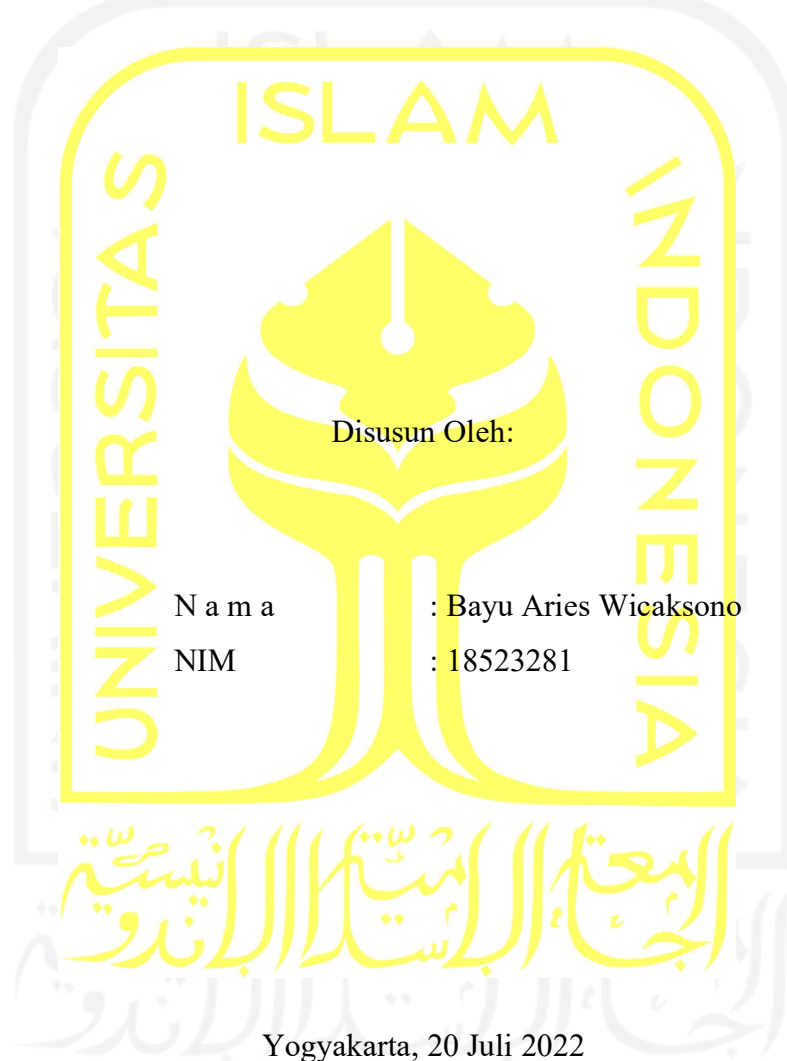
**2022**

**HALAMAN PENGESAHAN DOSEN PEMBIMBING**

**IMPLEMENTASI REST API UNTUK PENGEMBANGAN**

**SISTEM PELAPORAN DAN PENGADUAN**

**TUGAS AKHIR JALUR MAGANG**




Disusun Oleh:

N a m a : Bayu Aries Wicaksono  
NIM : 18523281

Yogyakarta, 20 Juli 2022

Pembimbing,

  
( Irving Vitra Papurungan, S.T., M.Sc., Ph.D. )

HALAMAN PENGESAHAN DOSEN PENGUJI

**IMPLEMENTASI REST API UNTUK PENGEMBANGAN  
SISTEM PELAPORAN DAN PENGADUAN**

**TUGAS AKHIR JALUR MAGANG**

Telah dipertahankan di depan sidang pengujian sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 03 Agustus 2022

Tim Penguji

Irving Vitra Papatungan, S.T., M.Sc., Ph.D.

Anggota 1

Chandra Kusuma Dewa, S.Kom., M.Cs., Ph.D.

Anggota 2

Izzati Muhimmah, S.T., M.Sc., Ph.D.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

**HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR**

Yang bertanda tangan di bawah ini:

Nama : Bayu Aries Wicaksono

NIM : 18523281

Tugas akhir dengan judul:

**IMPLEMENTASI REST API UNTUK PENGEMBANGAN  
SISTEM PELAPORAN DAN PENGADUAN**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 03 Agustus 2022



( Bayu Aries Wicaksono )

## HALAMAN PERSEMBAHAN

Alhamdulillah Segala puji dan syukur penulis panjatkan kepada Allah SWT, yang telah memberikan rahmat dan hidayah-Nya sehingga penulis mampu menyelesaikan Laporan Tugas Akhir. Laporan tugas akhir ini didedikasikan sepenuhnya untuk kedua orang tua, papa dan mama serta kakak tercinta atas segala dukungan dan doa yang selalu dipanjatkan setiap ibadah yang dijalankan. Kepada seluruh keluarga, dosen dan teman-teman yang telah mendukung penulis dalam menyusun laporan tugas akhir baik secara langsung maupun tidak langsung. Terima kasih saya ucapkan kepada semua pihak yang telah mendukung penyelesaian laporan tugas akhir ini semoga Allah SWT selalu memberikan perlindungan dan kesehatan kepada kita semua.



## HALAMAN MOTO

*“A dream is what maked people love life, even when it is painful”*

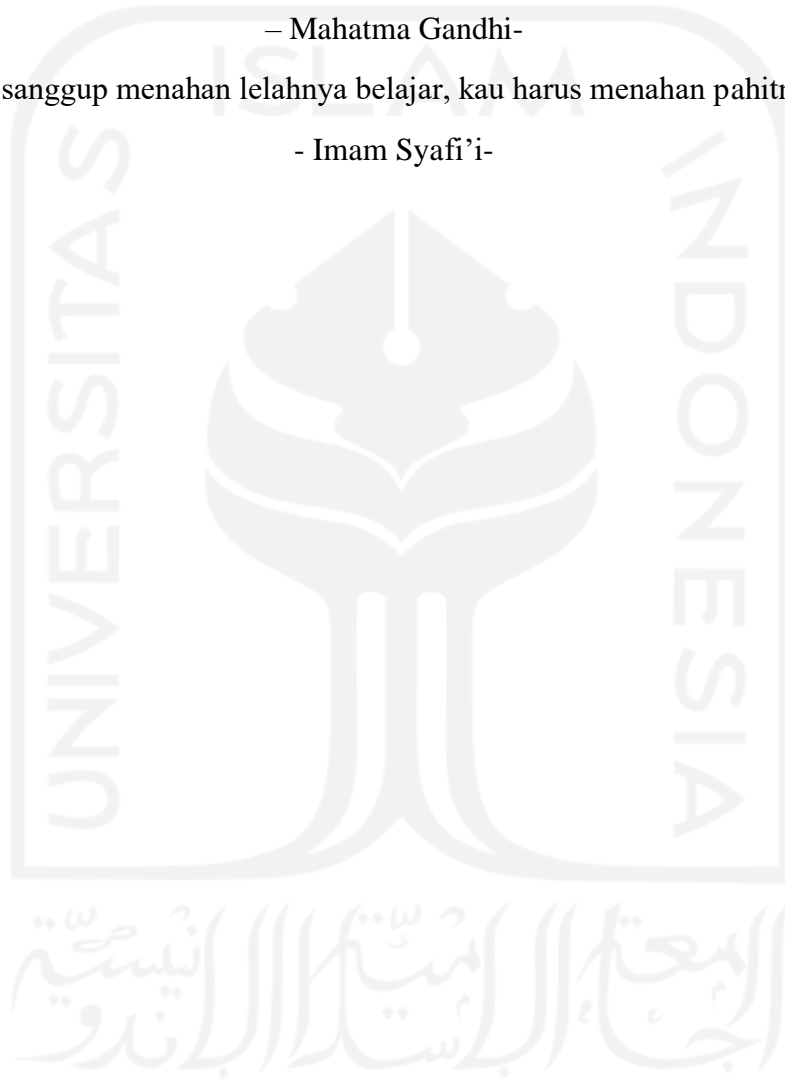
– Theodore Zeldin -

“Hiduplah seakan-akan kau mati besok. Belajarlah seakan-akan kau hidup untuk selamanya”

– Mahatma Gandhi-

“Jika kau tidak sanggup menahan lelahnya belajar, kau harus menahan pahitnya kebodohan”

- Imam Syafi'i-



## KATA PENGANTAR

### **Assalamualaikum Warahmatullahi Wabarakatuh**

Alhamdulillah

Segala puji dan syukur penulis panjatkan kepada Allah SWT, yang telah memberikan rahmat dan hidayah-Nya sehingga penulis mampu menyelesaikan Laporan Tugas Akhir yang berjudul " Implementasi REST API untuk Pengembangan Sistem Pelaporan dan Pengaduan".

Laporan ini disusun sebagai bukti pelaksanaan magang dan syarat untuk menyelesaikan pendidikan pada jenjang strata 1 (S1) pada jurusan Informatika Universitas Islam Indonesia. Laporan ini tidak akan dapat diselesaikan dengan cepat dan maksimal tanpa adanya dukungan serta motivasi dari berbagai pihak. Oleh karena itu, penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Allah SWT, atas limpahan rahmat dan hidayah-Nya yang telah memberikan rezeki, kesehatan, keselamatan, dan kelancaran dalam menyelesaikan Laporan ini.
2. Bapak Sutrisno dan Ibu Titik Sunarti, selaku orang tua penulis yang senantiasa memberikan doa, dukungan, dan selalu bersedia menjadi tempat bercerita bagi penulis.
3. Bapak Irving Vitra Papatungan, S.T., M.Sc., Ph.D., selaku dosen pembimbing Tugas Akhir yang telah memberikan waktu, ilmu, arahan, dan bimbingan kepada penulis dalam menyelesaikan Laporan Tugas Akhir ini.
4. Mas Dhimas Insan, Project Manager GeekGarden selaku pembimbing lapangan serta seluruh staf dan karyawan GeekGarden selama magang di GeekGarden yang selalu membantu penulis dalam mengerjakan tugas-tugas yang diberikan.
5. Bapak dan Ibu dosen Program Studi Informatika, yang telah memberikan ilmu yang bermanfaat kepada penulis selama masa perkuliahan.
6. Seluruh pihak yang tidak bisa penulis sebutkan satu persatu tetapi senantiasa mendukung penulis baik secara langsung maupun tidak langsung.

Semoga segala bantuan, ilmu, dukungan, dan bimbingan yang telah diberikan kepada penulis mendapatkan kebaikan dari Allah SWT. Penulis menyadari Laporan Tugas Akhir ini masih jauh dari kata sempurna, sehingga penulis dengan senang hati akan selalu menerima segala bentuk kritik dan saran yang membangun demi kesempurnaan Laporan Tugas Akhir

ini. Akhir kata, semoga Laporan Tugas Akhir ini dapat memberikan manfaat bagi semua pihak.

**Wassalamu'alaikum Waramahtullahi Wabarakatuh**

Yogyakarta, 20 Juli 2022



( Bayu Aries Wicaksono )





## SARI

Kemampuan beradaptasi pada era digital seperti saat ini sangatlah diperlukan. Pemanfaatan teknologi sangat membantu berbagai aktivitas manusia seperti meningkatkan performa kinerja pegawai. Aplikasi Ayo BerAksi merupakan salah satu produk digital yang membantu Balai Besar Karantina Belawan meningkatkan efisiensi pada pelaporan dan penanganan kasus tindak pidana korupsi. Ayo BerAksi hadir untuk menggantikan proses bisnis konvensional yang sebelumnya telah ada, dengan aplikasi masyarakat, dapat melaporkan kasus tindak pidana korupsi kapanpun dan dimanapun tanpa harus mendatangi kantor Balai Besar Karantina Belawan.

Aplikasi Ayo BarAksi dikembangkan dengan basis *multiplatform*, di mana aplikasi akan tersedia pada *platform website* dan *android*. Pengembangan Ayo BerAksi memanfaatkan REST API bertujuan mempermudah proses pelaporan dan pengaduan kasus penyuaipan dan gratifikasi yang terjadi di lingkungan BBKP-Belawan. Arsitektur REST API yang memiliki beberapa kelebihan diantaranya cocok digunakan pada platform *android*, memiliki *bandwidth* yang ringan, dan punya waktu *respons* yang cepat.

Dalam mengimplementasikan REST API, terdapat beberapa langkah yang harus dilakukan, yaitu pendefinisian, inisialisasi, perancangan, pelaksanaan, pengujian, pemantauan dan pengendalian, serta penutupan proyek. Hasil dari penelitian dan pengujian yang dilakukan, yaitu berupa aplikasi yang dapat mempermudah proses pelaporan dan pengaduan kasus tindak pidana korupsi. Selain itu, sistem yang dikembangkan cukup efisien karena memiliki *size bandwidth* yang kecil dan waktu yang cepat untuk pertukaran data.

Kata kunci: REST API, *multiplatform*, *website*, *android*, *bandwidth*.

## GLOSARIUM

|                          |  |
|--------------------------|--|
| <i>Backend Developer</i> | Bidang kerja atau posisi yang mengharuskan seseorang berinteraksi dengan database, mengelola dan memanipulasi data untuk kemudian ditampilkan. |
| <i>Framework</i>         | Kerangka kerja yang membantu <i>developer</i> menuliskan kode.   |
| Gratifikasi              | Pemberian yang memiliki tujuan atas manfaat yang diperoleh.  |
| Laravel                  | Kerangka kerja berbasis PHP, untuk pengembangan <i>website</i> .   |
| REST API                 | Arsitektur perangkat lunak untuk komunikasi antara <i>server</i> dan <i>client</i> .   |
| <i>Web service</i>       | Desain perangkat lunak yang memungkinkan untuk diakses secara <i>remote</i> .  |

## DAFTAR ISI

|   |      |
|---|------|
| HALAMAN JUDUL .....                                     | i    |
| HALAMAN PENGESAHAN DOSEN PEMBIMBING .....               | ii   |
| HALAMAN PENGESAHAN DOSEN PENGUJI.....                   | iii  |
| HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....            | iv   |
| HALAMAN PERSEMBAHAN .....                               | v    |
| HALAMAN MOTO .....                                      | vi   |
| KATA PENGANTAR.....                                     | vii  |
| SARI .....  | ix   |
| GLOSARIUM .....   | x    |
| DAFTAR ISI .....  | xi   |
| DAFTAR TABEL .....                                      | xiii |
| DAFTAR GAMBAR.....                                      | xiv  |
| BAB I PENDAHULUAN .....                                 | 1    |
| 1.1 Latar Belakang.....                                 | 1    |
| 1.2 Ruang Lingkup .....                                 | 2    |
| 1.3 Tujuan .....  | 3    |
| 1.4 Manfaat .....                                       | 3    |
| 1.5 Sistematika Penulisan .....                         | 4    |
| BAB II LANDASAN TEORI DAN TINJAUAN PUSTAKA.....         | 5    |
| 2.1 <i>Web Service</i> .....                            | 5    |
| 2.2 <i>Representational State Transfer (REST)</i> ..... | 5    |
| 2.3 <i>SOAP (Simple Object Access Protocol)</i> .....   | 6    |
| 2.4 OAuth 2.0 .....                                     | 7    |
| 2.5 Tinjauan Pustaka.....                               | 7    |
| BAB III PELAKSANAAN MAGANG .....                        | 10   |
| 3.1 Manajemen Proyek .....                              | 10   |
| 3.1.1 Pendefinisian Proyek .....                        | 10   |
| 3.1.2 Inisialisasi Proyek.....                          | 10   |
| 3.1.3 Perancangan Proyek.....                           | 11   |
| 3.1.4 Pelaksanaan Proyek .....                          | 12   |
| 3.1.5 Pengujian Proyek .....                            | 62   |
| 3.1.6 Pemantauan dan Pengendalian Proyek .....          | 70   |

|  |    |
|--|----|
| 3.1.7 Penutupan Proyek .....                     | 71 |
| BAB IV REFLEKSI PELAKSANAAN MAGANG.....          | 72 |
| 4.1 Relevansi Akademik.....                      | 72 |
| 4.2 Pembelajaran Magang.....                     | 72 |
| 4.2.1 Manfaat Magang.....                        | 72 |
| 4.2.2 Hambatan dan Tantangan Selama Magang ..... | 73 |
| BAB V PENUTUP .....                              | 75 |
| 5.1 Kesimpulan .....                             | 75 |
| 5.2 Saran .....                                  | 75 |
| 5.2.1 Saran Untuk Pengembangan Selanjutnya ..... | 75 |
| 5.2.2 Saran Untuk Perusahaan .....               | 76 |
| DAFTAR PUSTAKA.....                              | 77 |
| LAMPIRAN .....                                   | 79 |



## DAFTAR TABEL

|   |    |
|---|----|
| Tabel 2.1 Perbandingan REST dan SOAP.....   | 7  |
| Tabel 3.1 Spesifikasi Teknologi .....   | 10 |
| Tabel 3.2 Pembagian Waktu Aktivitas .....   | 11 |
| Tabel 3.3 Kebutuhan Fungsional Sistem .....   | 13 |
| Tabel 3.4 <i>User</i> Aplikasi .....  | 14 |
| Tabel 3.5 Klasifikasi Pengaduan.....  | 44 |
| Tabel 3.6 Kategori Pengaduan.....   | 44 |
| Tabel 3.7 Kesimpulan Hasil Klasifikasi .....  | 44 |
| Tabel 3.8 Rancangan <i>Request</i> dan <i>Response Register Akun</i> .....                | 57 |
| Tabel 3.9 Rancangan <i>Request</i> dan <i>Response Login Akun</i> .....                   | 58 |
| Tabel 3.10 Rancangan <i>Request</i> dan <i>Response</i> Membuat Laporan Penyusunan .....  | 59 |
| Tabel 3.11 Rancangan <i>Request</i> dan <i>Response</i> Membuat Laporan Pengaduan.....    | 60 |
| Tabel 3.12 Rancangan <i>Request</i> dan <i>Response</i> Membuat Laporan Gratifikasi ..... | 60 |
| Tabel 3.13 Rancangan <i>Request</i> dan <i>Response</i> Melihat Seluruh Laporan.....      | 61 |
| Tabel 3.14 Hasil Pengujian <i>Method POST</i> dengan 5 Sampel.....                        | 70 |
| Tabel 3.15 Hasil Pengujian <i>Method GET</i> dengan 5 Sampel.....                         | 70 |

## DAFTAR GAMBAR

|  |    |
|--|----|
| Gambar 3.1 <i>Entity Relationship Diagram</i> .....                        | 15 |
| Gambar 3.2 <i>Use Case Diagram</i> .....                                   | 16 |
| Gambar 3.3 Konfigurasi <i>app/User.php</i> .....                           | 17 |
| Gambar 3.4 <i>User Controller.php</i> .....                                | 18 |
| Gambar 3.5 Konfigurasi <i>config/app.php</i> .....                         | 19 |
| Gambar 3.6 Konfigurasi <i>app/Providers/AuthServiceProvider.php</i> .....  | 19 |
| Gambar 3.7 Konfigurasi <i>config/auth.php</i> .....                        | 20 |
| Gambar 3.8 <i>Model</i> Laporan Penyuapan.....                             | 20 |
| Gambar 3.9 <i>Migration</i> laporan Penyuapan.....                         | 21 |
| Gambar 3.10 <i>Controller</i> Tampilan Laporan Penyuapan.....              | 22 |
| Gambar 3.11 <i>Controller</i> Action Laporan Penyuapan .....               | 22 |
| Gambar 3.12 Daftar Laporan Penyuapan yang Masuk .....                      | 23 |
| Gambar 3.13 <i>Form</i> Laporan Penyuapan pada <i>Android</i> .....        | 24 |
| Gambar 3.14 Detail Laporan Penyuapan .....                                 | 25 |
| Gambar 3.15 Proses Laporan Penyuapan .....                                 | 26 |
| Gambar 3.16 <i>Model</i> Laporan Pengaduan.....                            | 27 |
| Gambar 3.17 <i>Migration</i> Laporan Pengaduan.....                        | 28 |
| Gambar 3.18 <i>Controller View</i> Laporan Pengaduan .....                 | 28 |
| Gambar 3.19 <i>Controller Action</i> Laporan Pengaduan .....               | 29 |
| Gambar 3.20 Daftar Laporan Pengaduan.....                                  | 29 |
| Gambar 3.21 <i>Form</i> Tambah Laporan Pengaduan pada <i>Android</i> ..... | 30 |
| Gambar 3.22 Proses Laporan Pengaduan .....                                 | 31 |
| Gambar 3.23 <i>Models</i> Laporan Gratifikasi .....                        | 32 |
| Gambar 3.24 <i>Migrations</i> Laporan Gratifikasi .....                    | 33 |
| Gambar 3.25 <i>Controllers View</i> Laporan Gratifikasi.....               | 33 |
| Gambar 3.26 <i>Controllers Action</i> Laporan Gratifikasi .....            | 34 |
| Gambar 3.27 Daftar Laporan Gratifikasi yang Masuk .....                    | 34 |
| Gambar 3.28 <i>Form</i> Laporan Gratifikasi pada <i>Android</i> .....      | 35 |
| Gambar 3.29 Proses Laporan Gratifikasi.....                                | 36 |
| Gambar 3.30 <i>Models</i> Analisa Laporan Penyuapan.....                   | 37 |

|  |    |
|--|----|
| Gambar 3.31 <i>Migrations</i> Analisa Laporan Penyuapan.....             | 37 |
| Gambar 3.32 <i>Controllers View</i> Analisa Laporan Penyuapan .....      | 38 |
| Gambar 3.33 <i>Controller Actions</i> Analisa Laporan Penyuapan.....     | 38 |
| Gambar 3.34 Daftar Analisa Laporan Penyuapan .....                       | 39 |
| Gambar 3.35 <i>Form</i> Tambah Analisa Penyuapan.....                    | 40 |
| Gambar 3.36 Detail Analisa Penyuapan .....                               | 41 |
| Gambar 3.37 <i>Models</i> Klasifikasi Laporan Pengaduan.....             | 42 |
| Gambar 3.38 <i>Migrations</i> Klasifikasi Laporan Pengaduan.....         | 42 |
| Gambar 3.39 <i>Controllers View</i> Klasifikasi Laporan Pengaduan .....  | 43 |
| Gambar 3.40 <i>Controller Actions</i> Klasifikasi Laporan Pengaduan..... | 43 |
| Gambar 3.41 Daftar Klasifikasi Pengaduan.....                            | 45 |
| Gambar 3.42 <i>Form</i> Klasifikasi Pengaduan.....                       | 45 |
| Gambar 3.43 <i>Models</i> Analisa laporan Gratifikasi.....               | 46 |
| Gambar 3.44 <i>Migrations</i> Analisa laporan Gratifikasi.....           | 47 |
| Gambar 3.45 <i>Controller View</i> Analisa laporan Gratifikasi .....     | 47 |
| Gambar 3.46 <i>Controller Actions</i> Analisa laporan Gratifikasi .....  | 48 |
| Gambar 3.47 Daftar Analisa Gratifikasi .....                             | 49 |
| Gambar 3.48 <i>Form</i> Analisa Gratifikasi .....                        | 50 |
| Gambar 3.49 <i>Models Logbook</i> Laporan Penyuapan .....                | 51 |
| Gambar 3.50 <i>Migrations Logbook</i> Laporan Penyuapan.....             | 52 |
| Gambar 3.51 <i>Controller View Logbook</i> Laporan Penyuapan.....        | 52 |
| Gambar 3.52 <i>Controller Actions Logbook</i> Laporan Penyuapan.....     | 53 |
| Gambar 3.53 <i>Logbook</i> Pengaduan .....                               | 53 |
| Gambar 3.54 <i>Models Logbook</i> Laporan Gratifikasi .....              | 54 |
| Gambar 3.55 <i>Migrations Logbook</i> Laporan Gratifikasi .....          | 55 |
| Gambar 3.56 <i>Controllers View Logbook</i> Laporan Gratifikasi.....     | 55 |
| Gambar 3.57 <i>Controllers Actions Logbook</i> Laporan Gratifikasi.....  | 56 |
| Gambar 3.58 <i>Logbook</i> Gratifikasi.....                              | 57 |
| Gambar 3.59 Pengujian <i>Register</i> .....                              | 62 |
| Gambar 3.60 <i>Response Endpoint</i> Register.....                       | 63 |
| Gambar 3.61 Pengujian <i>Login</i> .....                                 | 63 |
| Gambar 3.62 <i>Response Endpoint</i> Login .....                         | 64 |
| Gambar 3.63 Percobaan Tambah Laporan Penyuapan .....                     | 64 |
| Gambar 3.64 Mengisikan Token untuk Laporan Penyuapan.....                | 65 |

|   |    |
|---|----|
| Gambar 3.65 <i>Response Endpoint</i> Laporan Percobaan.....   | 65 |
| Gambar 3.66 Percobaan Tambah Laporan Pengaduan.....           | 66 |
| Gambar 3.67 Masukan Token Laporan Pengaduan.....              | 66 |
| Gambar 3.68 <i>Response Endpoint</i> Laporan Pengaduan.....   | 67 |
| Gambar 3.69 Percobaan Tambah Laporan Gratifikasi.....         | 67 |
| Gambar 3.70 Memasukan Token Laporan Gratifikasi.....          | 68 |
| Gambar 3.71 <i>Response Endpoint</i> Laporan Gratifikasi..... | 68 |
| Gambar 3.72 Pengujian Melihat Seluruh Data Laporan.....       | 69 |
| Gambar 3.73 <i>Response Endpoint</i> Seluruh Laporan.....     | 69 |





# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Perkembangan teknologi saat ini sangatlah pesat, bukan hanya berkembang teknologi juga berubah dalam waktu yang sangat cepat. Teknologi baru selalu ditemukan untuk menggantikan atau menyempurnakan teknologi sebelumnya. Perkembangan teknologi membawa manfaat yang luar biasa terutama pada kemajuan peradaban manusia, pekerjaan yang tadinya memerlukan usaha lebih dan waktu yang lama, kini sudah digantikan oleh sistem yang mudah digunakan dan lebih efektif. Salah satunya adalah pada teknologi informasi, yang memiliki banyak peran penting dalam membantu memenuhi kebutuhan manusia. Mulai dari berkomunikasi, belajar, bertransaksi, dan lain-lain (Danuri, 2019).

Pemanfaatan teknologi informasi saat ini sangatlah masif mengikuti perkembangan teknologi yang begitu cepat. Pemanfaatan teknologi informasi bukan hanya dipergunakan untuk kepentingan organisasi, melainkan untuk kepentingan individu. Organisasi dapat memanfaatkan teknologi untuk bersaing dengan kompetitor dan mencapai keunggulan secara kompetitif, sedangkan bagi individu pemanfaatan teknologi dapat dipergunakan untuk mencari pekerjaan atau meningkatkan performa kinerja (Rahmawati, 2012).

Di era digital saat ini teknologi informasi sudah memasuki berbagai bidang, salah satunya pada bidang bisnis. Kemampuan beradaptasi di era digital sangatlah diperlukan dalam bidang bisnis agar tidak tertinggal oleh kompetitor lainnya. Bisnis yang tidak memanfaatkan teknologi akan lebih mudah tertinggal dan bahkan terancam bangkrut (Cholik, 2021).

GeekGarden hadir sebagai perusahaan *software house* yang bergerak pada bidang perencanaan, pengembangan, dan pengolahan sistem aplikasi. GeekGarden membantu usaha bisnis dan perorangan pada level korporasi maupun startup dalam mengembangkan produk digital yang berkualitas. Visi dari GeekGarden yaitu menjadi sebuah perusahaan yang ikut andil dalam memajukan industri IT Indonesia dengan daya saing global. Misi menyempurnakan dan meningkatkan pelayanan dan variasi solusi, serta berkomitmen dalam mengedukasi masyarakat akan pentingnya menggunakan produk dan jasa IT anak bangsa (GeekGarden, n.d.).

Salah satu produk aplikasi yang dikembangkan GeekGarden yaitu mengembangkan aplikasi yang sesuai dengan keinginan *client* seperti aplikasi Ayo Beraksi, dimana aplikasi tersebut merupakan sebuah aplikasi *multiplatform* yang akan tersedia pada *website* dan *android*. Aplikasi ini dikembangkan sebagai media agar membantu proses penanganan kasus korupsi terutama kasus suap, dan gratifikasi yang terjadi di lingkungan Balai Besar Karantina Pertanian Belawan. Karena aplikasi menerapkan basis *multiplatform*, *website* dari Ayo Beraksi akan memanfaatkan REST API untuk mempermudah komunikasi dengan aplikasi *android*. REST API merupakan teknologi yang memungkinkan sebuah fungsi dan layanan yang ada pada *web service* dapat diakses oleh *client* tanpa mengetahui detail yang ada di dalamnya, menggunakan sebuah protokol HTTP (Kurniawan et al., 2013).

## 1.2 Ruang Lingkup

Pelaksanaan magang di GeekGarden berlangsung selama enam bulan dengan periode September 2021 hingga Maret 2022. Magang dilaksanakan di kantor GeekGarden *Software House* yang berlokasi di Yogyakarta. Berposisi sebagai *backend developer*. Selama magang terdapat dua proyek yang dilaksanakan yaitu sebagai berikut:

a. Mengembangkan Modul HRD Aplikasi Mie-Nyinyir

Pengembangan aplikasi Mie-Nyinyir menerapkan konsep *Enterprise Resource Planning* yang bertujuan untuk mengintegrasikan berbagai sumber daya yang dimiliki perusahaan. Dalam proyek ini salah satu modul yang dikembangkan adalah modul HRD yang akan menangani berbagai hal terkait sumber daya manusia. Hasil akhir dari proyek ini berbentuk sebuah *website Enterprise Resource Planning*.

b. Mengembangkan Aplikasi Ayo BerAksi

Tujuan dari pengembangan aplikasi Ayo BerAksi adalah membuat sebuah sistem yang dapat membantu proses penanganan laporan kasus penyuapan, gratifikasi yang terjadi di lingkungan Balai Besar Karantina Pertanian Belawan (BBKP-Belawan). Aplikasi dikembangkan dengan basis *multiplatform*, dimana aplikasi akan tersedia pada *website* dan *android*. Proses pengembangan aplikasi memanfaatkan REST API sebagai protokol komunikasi *website (server)* dan *android (client)*. Pada pengembangan ini lebih difokuskan pada sisi *back end* dari *website* dan pengembangan REST API.

Dalam penulisan tugas akhir ini akan berfokus pada proyek b, yaitu mengembangkan Aplikasi Ayo BerAksi menggunakan REST API. Alasan dipilihnya proyek ini sebagai tugas akhir karena hasil pengerjaan dari proyek ini dapat membantu perusahaan dalam mempercepat penanganan laporan kasus korupsi. Selain itu, belum adanya sebuah sistem yang membantu penanganan laporan korupsi yang terjadi dilingkungan BBKP-Belawan, dan hasil dari tugas akhir ini dapat menjadi sebuah dokumentasi dalam bentuk laporan yang dapat membantu melanjutkan proyek tersebut maupun proyek sejenis lainnya.

### 1.3 Tujuan

Tujuan dari laporan tugas akhir ini yaitu mengimplementasikan REST API dalam mengembangkan aplikasi Ayo BerAksi yang dapat mengolah laporan dan pengaduan tindak pidana korupsi yang ditemui masyarakat dan pegawai BBKP-Belawan.

### 1.4 Manfaat

Manfaat penggunaan REST API dalam pengembangan aplikasi Ayo BerAksi BBKP-Belawan adalah sebagai berikut:

- a. Memudahkan *developer* dalam mengembangkan aplikasi yang berbasis *multiplatform*.
- b. Mempermudah masyarakat dan pegawai BBKP-Belawan melaporkan suap dan gratifikasi.
- c. Membantu memberantas tindak pidana korupsi.

## 1.5 Sistematika Penulisan

Sistematika penulisan berisikan serangkaian urutan dalam penulisan laporan tugas akhir program magang. Adapun sistematika penulisan laporan tugas akhir ini disusun sebagai berikut:

a. BAB I Pendahuluan

Bab ini akan berisikan latar belakang pengembangan sistem pelaporan, ruang lingkup magang, tujuan, manfaat sistem pelaporan, dan sistematika penulisan laporan tugas akhir.

b. BAB II Landasan Teori dan Tinjauan Pustaka

Bab ini akan membahas mengenai teori-teori dan penelitian sebelumnya yang berkaitan dengan proyek pengembangan sistem pelaporan.

c. BAB III Pelaksanaan Magang

Bab ini akan menjelaskan alur pelaksanaan magang dalam mengembangkan proyek, fitur-fitur yang dikembangkan, serta hasil dan pembahasan dari pengembangan proyek.

d. BAB IV Refleksi Pelaksanaan Magang

Bab ini akan membahas mengenai refleksi dari pelaksanaan magang di GeekGarden baik dari secara teknis dan non-teknis.

e. BAB V Penutup

Bab ini berisi tentang kesimpulan yang didapat dari hasil pengerjaan proyek dan saran untuk pihak-pihak terkait.

## BAB II

### LANDASAN TEORI DAN TINJAUAN PUSTAKA

#### 2.1 *Web Service*

Menurut W3C (2001) *web service* merupakan sebuah desain sistem perangkat lunak yang mendukung interaksi antar mesin melalui jaringan. Desain sistem *web service* memungkinkan aplikasi lain untuk menggunakan fungsi (layanan informasi dan data) yang ada pada *web service* tanpa mengetahui detail yang ada didalamnya (Kurniawan et al., 2013).

Terdapat dua jenis arsitektur untuk membangun *web service* yaitu *Simple Object Access Protocol* (SOAP) atau *Representational State Transfer* (REST), selain itu *web service* memanfaatkan protokol HTTP serta format pesan dalam bentuk *text* JSON atau XML (Rizal & Rahmatulloh, 2019). Kedua jenis arsitektur tersebut dijelaskan sebagai berikut:

##### 2.1.1 **Representational State Transfer (REST)**

*Representational State Transfer* atau yang biasa disebut REST diciptakan oleh *cofounder* Apache HTTP Server Project yaitu Roy Fielding. Menurut Fielding (2000), REST adalah sebuah standar arsitektur *web* yang menggunakan protokol *Hypertext Transfer Protocol* dan juga *Uniform Resource Identifier* dalam berkomunikasi. Menurut W3C (2001) REST Web merupakan bagian dari WWW (yang berbasisi HTTP) di mana agent menyediakan antarmuka semantic yang seragam, yang pada dasarnya berupa *create*, *retrieve*, *update* dan *delete*. REST memanipulasi sumber daya hanya dengan melakukan pertukaran representasi.

Menurut Fielding (2000), arsitektur memiliki enam *constraint*/batasan yang dapat dijabarkan sebagai berikut:

##### a. *Client-Server*

Batasan *client-server* menjelaskan sebuah antarmuka yang terpisah antara *client* dan *server*. Pemisahan fungsionalitas tersebut berakibat pada penyederhanaan komponen *server* sehingga meningkatkan skalabilitas. Pemisahan juga memungkinkan kedua komponen untuk berkembang secara independen. Komponen *server* menyediakan sebuah layanan, mendengarkan *request* atas layanan tersebut, dan menolak atau menjalankan *request* serta

mengirim kembali sebuah *response* kepada *client*. Sedangkan komponen *client* menginginkan layanan untuk dijalankan, dan mengirimkan *request* kepada *server* melalui sebuah koneksi.

b. *Stateless*

Batasan ini menjelaskan bahwa tidak ada status *session* yang diperbolehkan pada komponen *server*. Seluruh *request* dari *client* kepada *server* harus berisikan informasi yang cukup untuk dipahami dan konteks yang tersimpan di *server* tidak dapat dimanfaatkan. Oleh karena itu, status sesi sepenuhnya disimpan di *client*.

c. *Cacheable*

Batasan *cacheable* menjelaskan bahwa *response* dari *server* dapat bersifat *cacheable*. Oleh karena itu, *response* harus mendefinisikan secara implisit atau eksplisit dirinya sebagai *cacheable* maupun tidak.

d. *Uniform Interface*

Batasan ini menekankan kepada komponen *interface* yang seragam. Dengan menerapkan prinsip rekayasa perangkat lunak ke komponen *interface*, sumberdaya dapat memanipulasi *create, read, update, delete* (CRUD) dengan memanfaatkan operasi HTTP yaitu POST, GET, PUT, dan DELETE.

e. *Layered System*

Batasan ini memungkinkan arsitektur memiliki lapisan hierarki yang dapat membatasi komponen, sehingga masing-masing komponen tidak dapat “melihat” lapisan lain diluar tempat interaksi mereka.

f. *Code on Demand* (opsional)

Batasan ini memungkinkan fungsionalitas dari *client* diperluas, dengan mengunduh dan mengeksekusi kode berupa skrip. Hal tersebut dapat menyederhanakan kerja *client* dengan mengurangi jumlah fitur yang harus dieksekusi.

### 2.1.2 SOAP (Simple Object Access Protocol)

*Simple Object Access Protocol* merupakan sebuah protokol komunikasi yang dipergunakan untuk bertukar pesan antar aplikasi. SOAP memanfaatkan teknologi XML untuk mendefinisikan sebuah kerangka pesan yang dapat melakukan pertukaran melalui berbagai protokol lainnya. *Framework* SOAP dirancang untuk tidak bergantung pada model

pemrograman tertentu. Pesan yang dikirim oleh SOAP akan berbentuk struktur dasar dari XML dengan *tag-tag* standar (W3C, 2003).

## 2.2 OAuth 2.0

*Open Authentication* (OAuth) 2.0 merupakan sebuah *framework* otorisasi yang memungkinkan aplikasi pihak ketiga untuk mendapatkan akses terbatas pada sebuah HTTP *service* (Hardt, 2012). OAuth 2.0 memanfaatkan mekanisme *scope* untuk membatasi akses informasi. *Laravel Passport* merupakan salah satu *library* Laravel yang mendukung penuh penggunaan OAuth 2 (Laravel, 2019).

Menurut (Hardt, 2012) OAuth 2.0 akan memanfaatkan *access token* yang akan menunjukkan identifikasi dari pengguna, dimana *access token* akan menggantikan otorisasi lain seperti *username*, dan *password*. *Access token* tersebut dapat tergenerate kedalam bentuk *string* ketika pengguna (*client*) melakukan *register* maupun *login*. Token ini akan mewakili cakupan atau *scope* dan durasi akses tertentu yang diberikan *server* kepada *client*.

## 2.3 Tinjauan Pustaka

Dalam mengembangkan sebuah *web service* terdapat dua arsitektur yang populer dan sering digunakan yaitu SOAP (*Simple Object Access Protocol*) dan REST (*Representational State Transfer*). Perbedaan antara REST dan SOAP sebagai standar komunikasi dapat dilihat pada Tabel 2.1.

Tabel 2.1 Perbandingan REST dan SOAP

Sumber: (Wagh & Thool, 2012; Kurniawan et al., 2013; Safitri, R.K., dan Putro, 2021)

| SOAP   | REST  |
|--|---|
| Merupakan teknologi lama.  | Teknologi baru dibandingkan SOAP.   |
| Cocok untuk <i>enterprise</i> dan B2B.   | Belum cocok untuk <i>enterprise</i> , namun bisa diterapkan pada aplikasi perbankan.  |
| Ketat dalam aturan <i>client-server</i> .  | Lebih longgar dalam aturan <i>client-server</i> .                                     |
| Jika pada <i>server</i> terdapat perubahan, maka <i>client</i> juga perlu perubahan. | Jika pada <i>server</i> terdapat perubahan, maka <i>client</i> tidak perlu perubahan. |
| Memiliki muatan data yang berat dibandingkan REST.                                   | <i>Transfer</i> data sangat ringan karena menggunakan URI.                            |
| SOAP bukan infrastruktur nirkabel yang ramah   | REST ramah infrastruktur nirkabel   |
| Menggunakan lebih banyak <i>bandwith</i> .   | Menggunakan lebih sedikit <i>bandwith</i> .   |
| Lebih susah dikembangkan, dan membutuhkan <i>tools</i> .                             | Lebih mudah untuk dikembangkan.   |
| SOAP <i>web service</i> mengembalikan data XML.                                      | REST mendukung berbagai macam format data.  |
| Sulit untuk membuat <i>cache response</i> .  | Mudah untuk membuat <i>cache response</i> .   |

Pada Penelitian ini, akan membahas mengenai pengembangan RESTful API untuk pengaduan dan pelaporan. Sampai saat ini sudah ada beberapa penelitian yang membahas penggunaan REST API dalam menyelesaikan berbagai macam permasalahan. Adapun rangkuman dari beberapa penelitian sebelumnya dapat dilihat pada paragraf dibawah ini.

Penelitian (Kurniawan et al., 2013), membahas mengenai pengembangan portal akademik berbasis *android* dengan mengimplementasikan REST API. Pengembangan aplikasi ini ditujukan untuk mempermudah pengguna dalam mengakses portal akademik melalui aplikasi *mobile* yang sebelumnya telah tersedia pada *website*. Penelitian ini memanfaatkan API dan *web service* dengan arsitektur REST sebagai metode komunikasi aplikasi *android* dengan portal *web server*. Hasil dari penelitian ini yaitu sebuah aplikasi *android* yang terintegrasi dengan portal *website*.

Berdasarkan penelitian oleh Wagh & Thool (2012), mengenai studi perbandingan *framework* SOAP dan REST sebagai penyedia layanan. Tujuan penelitian ini yaitu membandingkan kedua *framework* tersebut untuk sebagai penyedia layanan *web service* dan *framework* mana yang lebih cocok untuk lingkungan nirkabel. Hasil dari penelitian ini menyatakan *framework* REST lebih cocok digunakan pada perangkat *mobile*, selain itu REST memiliki daya konsumsi *resource* yang lebih hemat dibandingkan SOAP.

Pada penelitian lain yang dilakukan oleh Safitri dan Putro (2021), yang membahas mengenai implementasi REST API untuk komunikasi antara ReactJS dan NodeJS. Penelitian ini bertujuan untuk membangun sebuah REST API yang menjadi jembatan komunikasi pada pengembangan sebuah modul yang memanfaatkan dua *framework* yang berbeda. Hasil dari penelitian ini yaitu sebuah REST API yang mampu menghubungkan *framework* ReactJS sebagai *frontend* dan NodeJS sebagai *backend*. Selain itu, penelitian ini menyatakan bahwa penggunaan REST API dalam pengembangan aplikasi mampu bekerja secara efisien karena tidak boros *bandwidth* dan hemat waktu dalam pertukaran data.



Berdasarkan penelitian sebelumnya, telah dilakukan implementasi REST API untuk pengembangan aplikasi dan untuk perbandingan. Pada penelitian yang telah diuraikan, terdapat dua penelitian yang membahas implementasi REST API pada aplikasi *mobile*. Kemudian terdapat satu penelitian yang membahas perbandingan REST dengan SOAP sebagai layanan *web service*. Selain itu, terdapat satu penelitian yang menjelaskan implementasi REST API pada dua *framework* yang berbeda yaitu ReactJS dan NodeJS. Pada penelitian ini dikembangkan sebuah aplikasi berbasis *website* yang menggunakan *framework* Laravel, selain itu sistem akan memanfaatkan arsitektur REST API yang menjadi jembatan komunikasi untuk aplikasi *android*.



## BAB III

### PELAKSANAAN MAGANG

#### 3.1 Manajemen Proyek

Dalam mengembangkan aplikasi Ayo BerAksi, terdapat beberapa tahapan manajemen proyek yang terdiri dari: tahapan pendefinisian proyek, inisialisasi proyek, perancangan, pelaksanaan, pengujian, pemantauan dan pengendalian, serta penutupan proyek. Tahapan tersebut dikerjakan agar dalam pengembangan pengembangan perangkat lunak menjadi terorganisir dan terarah. Detail dari tahapan-tahapan tersebut dijabarkan sebagai berikut:

##### 3.1.1 Pendefinisian Proyek

Aplikasi Ayo BerAksi merupakan sebuah aplikasi berbasis *multiplatform* dimana terdapat aplikasi *website* dan *android* yang akan digunakan untuk melakukan pelaporan dan pengaduan kasus tindak pidana korupsi suap, serta gratifikasi yang terjadi di lingkungan kantor BBKP-Belawan. Aplikasi *android* dari Ayo BerAksi dapat digunakan oleh masyarakat dan juga pegawai yang akan melaporkan kasus tindak pidana korupsi, sedangkan *website* dari Ayo BerAksi akan digunakan oleh pegawai yang berwenang untuk menangani laporan kasus tindak pidana korupsi.

##### 3.1.2 Inisialisasi Proyek

Pada tahap inisialisasi, spesifikasi dari sistem Ayo BerAksi ditentukan agar pengembangan sistem dapat sesuai harapan. Spesifikasi dari beberapa teknologi yang digunakan dalam pengembangan sistem Ayo BerAksi dapat dilihat pada Tabel 3.1.

Tabel 3.1 Spesifikasi Teknologi

| Aspek                 | Spesifikasi |
|-----------------------|-------------|
| Bahas Pemrograman     | PHP 7.3     |
| <i>Framework</i>      | Laravel 8   |
| <i>Database</i>       | MySQL       |
| <i>Source Control</i> | GitLab      |

Selain itu, dalam pengembangan aplikasi Ayo BerAksi ini terdapat sebuah tim yang masing-masing individunya memiliki peran diantaranya sebagai berikut:

- a. *Project Manager* (PM), merupakan pemimpin proyek yang memiliki tanggung jawab atas pengelolaan proyek secara keseluruhan. Selain itu, *project manager* juga berperan sebagai seseorang yang mampu menemukan solusi ketika tim menghadapi masalah.
- b. *UI/UX Designer*, merupakan orang yang bertanggung jawab dalam merancang tampilan muka dari sistem yang akan dibangun.
- c. *Backend Programmer*, merupakan orang yang bertanggung jawab dalam merancang kebutuhan sistem dari sisi server yang berkaitan dengan *database* dan logika, serta mengimplementasikan hasil rancangan ke dalam bentuk kode program.
- d. *Frontend Programmer*, merupakan orang yang bertanggung jawab dalam mengimplementasikan desain tampilan yang sudah dibuat oleh *UI/UX Designer*.
- e. *Quality Assurance* (QA), bertanggung jawab untuk menjamin kualitas dari sistem yang dibangun, dan menguji sistem yang sudah dikembangkan.
- f. *Mobile Developer*, bertanggung jawab dalam mengembangkan sistem dengan basis *android flutter*, dan mengintegrasikan sistem tersebut dengan sistem berbasis *website* memanfaatkan REST API yang telah dibuat.

### 3.1.3 Perancangan Proyek

Ayo BerAksi sebagai aplikasi pelaporan dan pengaduan tindak pidana korupsi yang digunakan oleh masyarakat, dan pegawai BBKP-Belawan. Selama magang, pembagian waktu dalam pengembangan Aplikasi *website* Ayo BerAksi dapat dilihat pada Tabel 3.2. Dalam pembagian waktu belum termasuk beberapa aktivitas lain, seperti perbaikan bug dari hasil uji coba tim *Quality Assurance*.

Tabel 3.2 Pembagian Waktu Aktivitas

| No | Aktivitas  | Durasi Waktu |
|----|--|--------------|
| 1  | Analisis Kebutuhan dari Sistem Pelaporan dan Pengaduan         | 4 Hari       |
| 2  | Implementasi <i>Laravel Passport</i> untuk Otentikasi REST API | 1 Hari       |
| 3  | Mengembangkan Fitur Laporan Penyusunan                         | 6 Hari       |
| 4  | Mengembangkan Fitur Laporan Pengaduan                          | 5 Hari       |
| 5  | Mengembangkan Fitur Laporan Gratifikasi                        | 6 Hari       |
| 6  | Mengembangkan Fitur Analisa Laporan Penyusunan                 | 3 Hari       |
| 7  | Mengembangkan Fitur Klasifikasi Laporan Pengaduan              | 3 Hari       |
| 8  | Mengembangkan Fitur Analisa Laporan Gratifikasi                | 3 Hari       |
| 9  | Mengembangkan Fitur Logbook Laporan Penyusunan                 | 4 Hari       |
| 10 | Mengembangkan Fitur Logbook Laporan Gratifikasi                | 3 Hari       |
| 11 | Mengembangkan REST API pada Sistem                             | 2 Pekan      |

Pada bulan pertama pengembangan fitur laporan penyusunan, pengaduan, dan gratifikasi, mendapatkan revisi untuk merubah berbagai input yang ada didalamnya. Karena pada bulan awal, dokumen pengembangan yang diberikan kepada tim pengembang masih belum lengkap, sehingga terjadi perubahan terutama pada tampilan dan juga pada desain database. Sehingga, menambah waktu pengembangan.

#### 3.1.4 Pelaksanaan Proyek

Pada pelaksanaan proyek pengembangan sistem pengaduan dan pelaporan terdapat beberapa aktivitas yang telah dilakukan, diantaranya meliputi:

##### a. Analisis Kebutuhan dari Sistem Pelaporan dan Pengaduan

Pengembangan sistem dimulai dengan menganalisa kebutuhan sistem, dan juga menentukan siapa saja *stakeholder*. Proses ini dilakukan oleh tim yang berperan sebagai *backend programmer* dan *supervisor* yang merupakan *project manager* dari proyek pengembangan. Dalam proses ini, kebutuhan sistem ditentukan berdasarkan dokumen pengembangan dan formulir pengajuan laporan yang digunakan BBKP-Belawan pada pelaporan konvensional. Skema dari kebutuhan sistem yang menggambarkan permasalahan dan juga kebutuhan yang dibutuhkan oleh pengguna, sebagai berikut:

##### 1. Permasalahan

Balai Besar Karantina Pertanian – Belawan membutuhkan sebuah aplikasi yang mampu meningkatkan efisiensi pada pelaporan dan pengaduan kasus penyusunan dan gratifikasi, proses bisnis yang sebelumnya telah ada namun masih menggunakan cara konvensional pada pelaporannya.

##### 2. Kebutuhan Fungsional Sistem

Pada pengembangan aplikasi Ayo BerAksi terdapat beberapa kebutuhan fungsionalitas dan non-fungsionalitas yang akan dikembangkan, yaitu diantaranya seperti pada Tabel 3.3.

Tabel 3.3 Kebutuhan Fungsional Sistem

| No | Kebutuhan                           | Keterangan                   | Pengguna   | Tanggung Jawab                                    |
|----|-------------------------------------|------------------------------|--|---|
| 1  | Melaporkan Kasus Penyipuan          | Kebutuhan Fungsionalitas     | Masyarakat, Pegawai, Tim Kepatuhan, dan Kepala Balai | ✓ (Pengembangan pada bagian website dan REST API) |
| 2  | Melaporkan Kasus Pengaduan          | Kebutuhan Fungsionalitas     | Masyarakat, Pegawai, Tim Kepatuhan, dan Kepala Balai | ✓ (Pengembangan pada bagian website dan REST API) |
| 3  | Melaporkan Kasus Gratifikasi        | Kebutuhan Fungsionalitas     | Masyarakat, Pegawai, Tim Kepatuhan, dan Kepala Balai | ✓ (Pengembangan pada bagian website dan REST API) |
| 4  | Menganalisa Laporan Penyipuan       | Kebutuhan Fungsionalitas     | Tim Kepatuhan  | ✓ (Pengembangan pada bagian website)              |
| 5  | Menganalisa Laporan Pengaduan       | Kebutuhan Fungsionalitas     | Tim Kepatuhan  | ✓ (Pengembangan pada bagian website)              |
| 6  | Menganalisa Laporan Gratifikasi     | Kebutuhan Fungsionalitas     | Tim Kepatuhan  | ✓ (Pengembangan pada bagian website)              |
| 7  | Membuat Logbook Laporan Penyipuan   | Kebutuhan Fungsionalitas     | Tim Kepatuhan  | ✓ (Pengembangan pada bagian website)              |
| 8  | Membuat Logbook Laporan Gratifikasi | Kebutuhan Fungsionalitas     | Tim Kepatuhan  | ✓ (Pengembangan pada bagian website)              |
| 9  | Memantau Analisa                    | Kebutuhan Fungsionalitas     | Kepala Balai   | ✓ (Pengembangan pada bagian website)              |
| 10 | Memantau Logbook                    | Kebutuhan Fungsionalitas     | Kepala Balai   | ✓ (Pengembangan pada bagian website)              |
| 11 | Membuat Feedback                    | Kebutuhan Non-Fungsionalitas | Masyarakat, Pegawai, Tim Kepatuhan, dan Kepala Balai | ✗   |
| 12 | Memantau Feedback                   | Kebutuhan Non-Fungsionalitas | Tim Kepatuhan, dan Kepala Balai                      | ✗   |
| 13 | Membuat Akun                        | Kebutuhan Fungsionalitas     | Masyarakat, dan Pegawai                              | ✓ (Pengembangan pada bagian website dan REST API) |
| 14 | Login Akun                          | Kebutuhan Fungsionalitas     | Masyarakat, Pegawai, Tim Kepatuhan, dan Kepala Balai | ✓ (Pengembangan pada bagian website dan REST API) |
| 15 | Logout Akun                         | Kebutuhan Fungsionalitas     | Masyarakat, Pegawai, Tim Kepatuhan, dan Kepala Balai | ✓ (Pengembangan pada bagian website dan REST API) |
| 16 | Melihat List Seluruh Laporan        | Kebutuhan Non-Fungsionalitas | Masyarakat, Pegawai, Tim Kepatuhan, dan Kepala Balai | ✓ (Pengembangan pada bagian REST API)             |
| 17 | Pembuatan PDF Laporan Penyipuan     | Kebutuhan Fungsionalitas     | Masyarakat, Pegawai, Tim Kepatuhan, dan Kepala Balai | ✗   |
| 18 | Pembuatan PDF Laporan Pengaduan     | Kebutuhan Fungsionalitas     | Tim Kepatuhan, dan Kepala Balai                      | ✗   |
| 19 | Pembuatan PDF Laporan Gratifikasi   | Kebutuhan Fungsionalitas     | Tim Kepatuhan, dan Kepala Balai                      | ✗   |

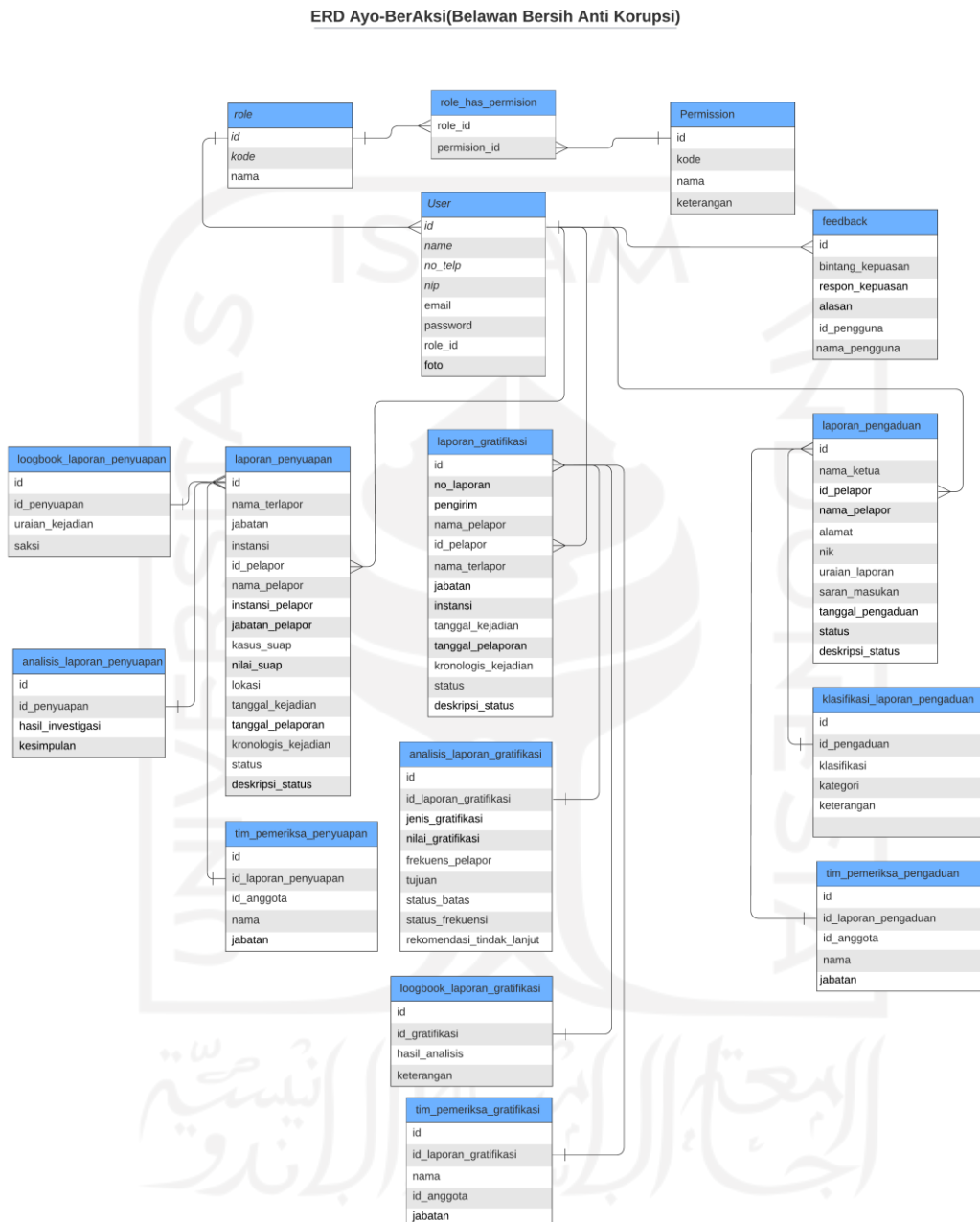
### 3. Pengguna Aplikasi

Terdapat empat *user role* yang menggunakan aplikasi Ayo BerAksi seperti pada Tabel 3.4.

Tabel 3.4 *User* Aplikasi

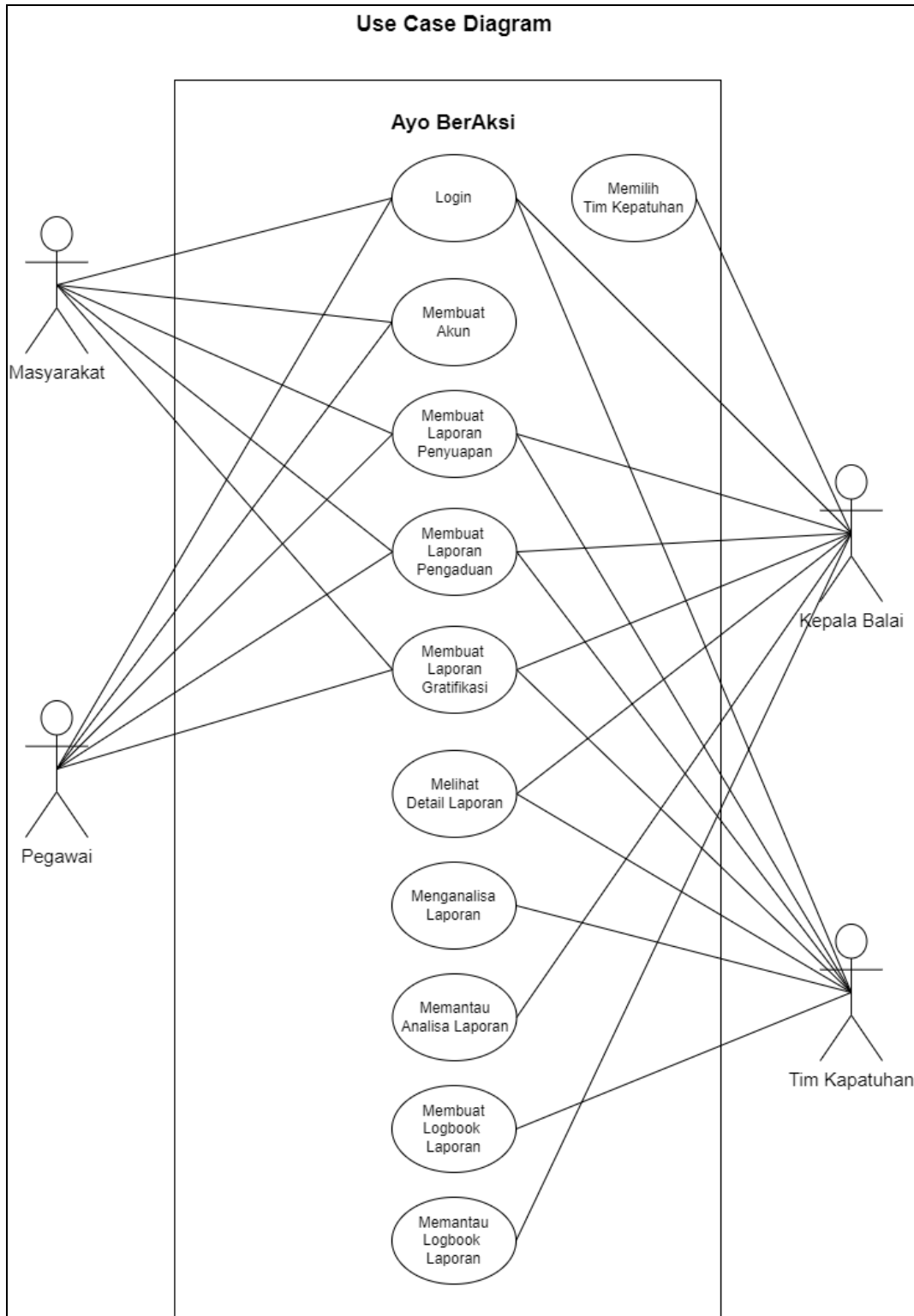
| No | Pengguna      | Keterangan  |
|----|---------------|---|
| 1  | Masyarakat    | Masyarakat yang menggunakan jasa pelayanan Balai Besar karantina Pertanian Belawan.       |
| 2  | Pegawai       | Pegawai yang bekerja di Balai Besar karantina Pertanian Belawan.                          |
| 3  | Tim Kepatuhan | Tim khusus untuk penyelidikan dan analisa kasus penyuapan, pengaduan, dan gratifikasi.    |
| 4  | Kepala Balai  | Seseorang yang bertindak sebagai supervisor pada Balai Besar karantina Pertanian Belawan. |

Hasil dari analisis kebutuhan sistem untuk skema *database* yaitu berupa sebuah diagram ERD (*Entity Relationship Diagram*) yang dapat dilihat pada Gambar 3.1.



Gambar 3.1 *Entity Relationship Diagram*

Selain itu, salah satu hasil analisis kebutuhan sistem adalah *use case diagram* yang memodelkan hubungan interaksi sistem dengan aktor, *use case diagram* dapat dilihat pada Gambar 3.2.



Gambar 3.2 Use Case Diagram

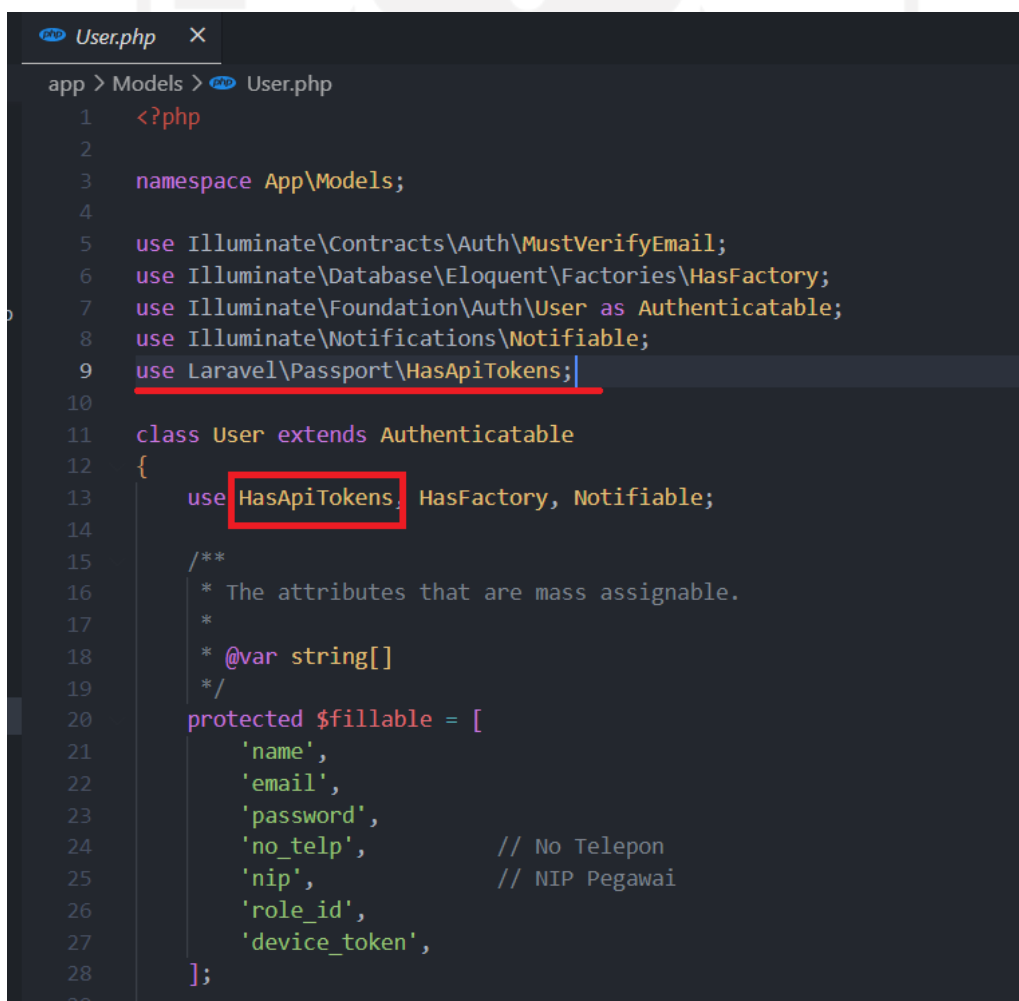
Karena pengembangan ini menggunakan REST API pada aplikasi *android*, untuk itu otentikasi pengguna diperlukan. Oleh karena itu, sistem ini akan memanfaatkan *library* Laravel yaitu *Laravel Passport*. Selain itu, untuk mempermudah proses pengembangan



sistem perlu membuat sebuah *migration* serta *seeder* yang berisikan data *dummy*. Dalam pengembangan *website* sistem pelaporan dan pengaduan, menerapkan konsep *create*, *read*, *update*, dan *delete* (CRUD) serta konsep MVP atau *model*, *view*, dan *controller*. Setelah menyiapkan kebutuhan dari sistem, aktivitas dilanjutkan dengan menginisialisasi pengembangan sistem memanfaatkan *framework* Laravel dan membuat sebuah *repository* baru sebagai media kolaborasi.

#### b. Implementasi Laravel *Passport* untuk Otentikasi REST API

Konfigurasi API pada pengembangan sistem akan memanfaatkan salah satu *library* milik Laravel yaitu *Passport*. Laravel *Passport* digunakan untuk mengautentikasi pengguna yang mengakses *website* dengan memanfaatkan API. *Library Passport* merupakan sebuah *library* yang mendukung penuh penggunaan OAuth 2.0.



```
app > Models > User.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Contracts\Auth\MustVerifyEmail;
6  use Illuminate\Database\Eloquent\Factories\HasFactory;
7  use Illuminate\Foundation\Auth\User as Authenticatable;
8  use Illuminate\Notifications\Notifiable;
9  use Laravel\Passport\HasApiTokens;
10
11 class User extends Authenticatable
12 {
13     use HasApiTokens, HasFactory, Notifiable;
14
15     /**
16      * The attributes that are mass assignable.
17      *
18      * @var string[]
19      */
20     protected $fillable = [
21         'name',
22         'email',
23         'password',
24         'no_telp',          // No Telepon
25         'nip',              // NIP Pegawai
26         'role_id',
27         'device_token',
28     ];
29
```

Gambar 3.3 Konfigurasi *app/User.php*

Seperti pada teori OAuth2 akan memanfaatkan *scope*, pada implementasi OAuth2 yang memanfaatkan Laravel *Passport* akan mengakses *role\_id* dari *user* pada file *User.php* yang akan menjadi *scope*, dimana dari masing-masing *role\_id* akan memiliki batasan dalam pengaksesan data. Selain itu, Laravel *Passport* akan menggunakan akses token yang mempermudah identifikasi pengguna, dimana akses token ini akan menjadi pengganti *email* dan *password*. Akses token akan di generate pada saat user melakukan *login* dan *register* seperti Gambar 3.4.

```

app > Http > Controllers > API > UserController.php
1  <?php
2
3  namespace App\Http\Controllers\API;
4
5  use Illuminate\Http\Request;
6  use App\Http\Controllers\Controller;
7  use App\Models\User;
8  use Exception;
9  use Illuminate\Support\Facades\Auth;
10 use Illuminate\Support\Facades\DB;
11 use Illuminate\Support\Facades\Validator;
12
13 class UserController extends Controller
14 {
15
16     public $successStatus = 200;
17
18     public function login()
19     {
20         if (Auth::attempt(['email' => request('email'), 'password' => request('password')])) {
21             // $user = Auth::user();
22             $user = User::where('email', request('email'))->first();
23             $accessToken = $user->createToken('tokenAyoBeraksi')->accessToken;
24             $success['accessToken'] = $accessToken;
25             $success['name'] = $user->name;
26             $success['token_type'] = 'Bearer';
27             // return response()->json(['success'=>$success], $this->successStatus);
28             return response(['user' => Auth::user(), 'access_token' => $accessToken]);
29         } else {
30             // return response()->json(['error'=>'Unauthorised'], 401);
31             return response()->json([
32                 'success' => false,
33                 'message' => 'Invalid Email or Password',
34             ], 401);
35         }
36     }
37 }

```

Gambar 3.4 User Controller.php

Implementasi REST API dimulai dengan menginstall Laravel *Passport* pada proyek Laravel baru, dengan menggunakan perintah “*composer require laravel/passport*”. Setelah instalasi berhasil, aktivitas dilanjutkan dengan konfigurasi pada file “*config/app.php*” seperti Gambar 3.5, selanjutnya memodifikasi kode pada file “*app/User.php*” dengan menambahkan “*HasApiToken*” seperti pada Gambar 3.3, kemudian mengubah konfigurasi file

“*app/Providers/AuthServiceProvider.php*” seperti Gambar 3.6 dan “*config/auth.php*” seperti Gambar 3.7.

```

config > app.php
136
137     'providers' => [
138
139         //Laravel Passport
140         Laravel\Passport\PassportServiceProvider::class,
141
142         /*
143          * Laravel Framework Service Providers...
144          */
145         Illuminate\Auth\AuthServiceProvider::class,
146         Illuminate\Broadcasting\BroadcastServiceProvider::class,
147         Illuminate\Bus\BusServiceProvider::class,

```

Gambar 3.5 Konfigurasi *config/app.php*

```

app > Providers > AuthServiceProvider.php
1  <?php
2
3  namespace App\Providers;
4
5  use Illuminate\Foundation\Support\Providers\AuthServiceProvider as ServiceProvider;
6  use Illuminate\Support\Facades\Gate;
7  use Laravel\Passport\Passport;
8
9  class AuthServiceProvider extends ServiceProvider
10 {
11     /**
12      * The policy mappings for the application.
13      *
14      * @var array
15      */
16     protected $policies = [
17         // 'App\Models\Model' => 'App\Policies\ModelPolicy',
18         'App\Models' => 'App\Policies\ModelPolicy',
19     ];
20
21     /**
22      * Register any authentication / authorization services.
23      *
24      * @return void
25      */
26     public function boot()
27     {
28         $this->registerPolicies();
29         Passport::routes();
30     }
31 }
32

```

Gambar 3.6 Konfigurasi *app/Providers/AuthServiceProvider.php*

```

config > auth.php
37
38     'guards' => [
39         'web' => [
40             'driver' => 'session',
41             'provider' => 'users',
42         ],
43
44         'api' => [
45             'driver' => 'passport',
46             'provider' => 'users',
47             'hash' => false,
48         ],
49     ],
50

```

Gambar 3.7 Konfigurasi *config/auth.php*

### c. Mengembangkan Fitur Laporan Penyuaan

Setelah selesai melakukan konfigurasi API, aktivitas selanjutnya yaitu melanjutkan pengembangan salah satu fitur yaitu fitur laporan pengaduan. Pengembangan dimulai dengan membuat *model* dari fitur yang akan menjadi komponen yang berhubungan dan mengelola secara langsung *database*, seperti Gambar 3.8.

```

app > Models > LaporanPenyuapan.php
1 namespace App\Models;
2
3 use Illuminate\Database\Eloquent\Factories\HasFactory;
4 use Illuminate\Database\Eloquent\Model;
5
6 class LaporanPenyuapan extends Model
7 {
8     use HasFactory;
9
10     protected $table = 'laporan_penyuapans';
11     protected $fillable = [
12         'nama_terlapor', // nama yang dilaporkan
13         'jabatan', // jabatan yang dilaporkan
14         'instansi', // instansi
15         'id_pelapor', // id pelapor
16         'nama_pelapor',
17         'jabatan_pelapor',
18         'instansi_pelapor',
19         'kasus_suap', // nama suap
20         'nilai_suap',
21         'lokasi',
22         'tanggal_kejadian',
23         'tanggal_pelaporan',
24         'kronologis_kejadian',
25         'status', // status laporan
26         'deskripsi_status', // deskripsi/alasan di tolak apa di terima
27     ];
28
29 // Relationship
30 public function user()
31 {
32     return $this->belongsTo('App\Models\User', 'id_pelapor', 'id');
33 }
34 public function timPemeriksa()
35 {
36     return $this->hasMany('App\Models\TimPemeriksaPenyuapan');
37 }
38 }
39

```

Gambar 3.8 *Model* Laporan Penyuaan

Kemudian dilanjutkan dengan membuat *migrations* seperti Gambar 3.9 untuk membuat skema basis data, dan membuat *controllers* untuk menghubungkan *models* dan *view* seperti pada Gambar 3.10 serta *controllers actions* yang mengatur manipulasi data seperti penyimpanan, perubahan, dan penghapusan data seperti Gambar 3.11.

```
database > migrations > 2021_11_15_080714_create_laporan_penyuapans_table.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  class CreateLaporanPenyuapansTable extends Migration
8  {
9
10     /**
11      * Run the migrations.
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('laporan_penyuapans', function (Blueprint $table) {
17             $table->id();
18             $table->string('nama_terlapor');
19             $table->string('jabatan');
20             $table->string('instansi');
21             $table->unsignedBigInteger('id_pelapor')->nullable();
22             $table->string('nama_pelapor');
23             $table->string('jabatan_pelapor')->nullable();
24             $table->string('instansi_pelapor')->nullable();
25             $table->string('kasus_suap');
26             $table->string('nilai_suap');
27             $table->string('lokasi');
28             $table->date('tanggal_kejadian')->nullable();
29             $table->date('tanggal_pelaporan')->nullable();
30             $table->text('kronologis_kejadian');
31             $table->string('status')->nullable();
32             $table->string('deskripsi_status')->nullable();
33             $table->timestamps();
34         });
35     }
36 }
```

Gambar 3.9 Migration laporan Penyuapan

```

app > Http > Controllers > Web > LaporanPenyuapanController.php
1  <?php
2
3  namespace App\Http\Controllers\Web;
4
5  use App\Http\Controllers\Controller;
6  use App\Helpers\LaporanHelper as LprHelper;
7  use Illuminate\Http\Request;
8
9  class LaporanPenyuapanController extends Controller
10 {
11     public function index()
12     {
13         return view('penyuapan.view');
14     }
15
16     public function create()
17     {
18         return view('penyuapan.form');
19     }
20
21     public function show($id)
22     {
23         $data = array(
24             'showPenyuapan' => LprHelper::showPenyuapan($id),
25             'id' => $id,
26         );
27         return view('penyuapan.show', $data);
28     }
29
30     public function edit($id)
31     {
32         $data = array(
33             'showPenyuapan' => LprHelper::showPenyuapan($id),
34             'tim_kepatuhan' => LprHelper::timKepatuhan(),
35             'detailTim' => LprHelper::detailTimPenyuapan($id),
36             'id' => $id,
37         );
38         return view('penyuapan.form', $data);

```

Gambar 3.10 Controller Tampilan Laporan Penyuapan

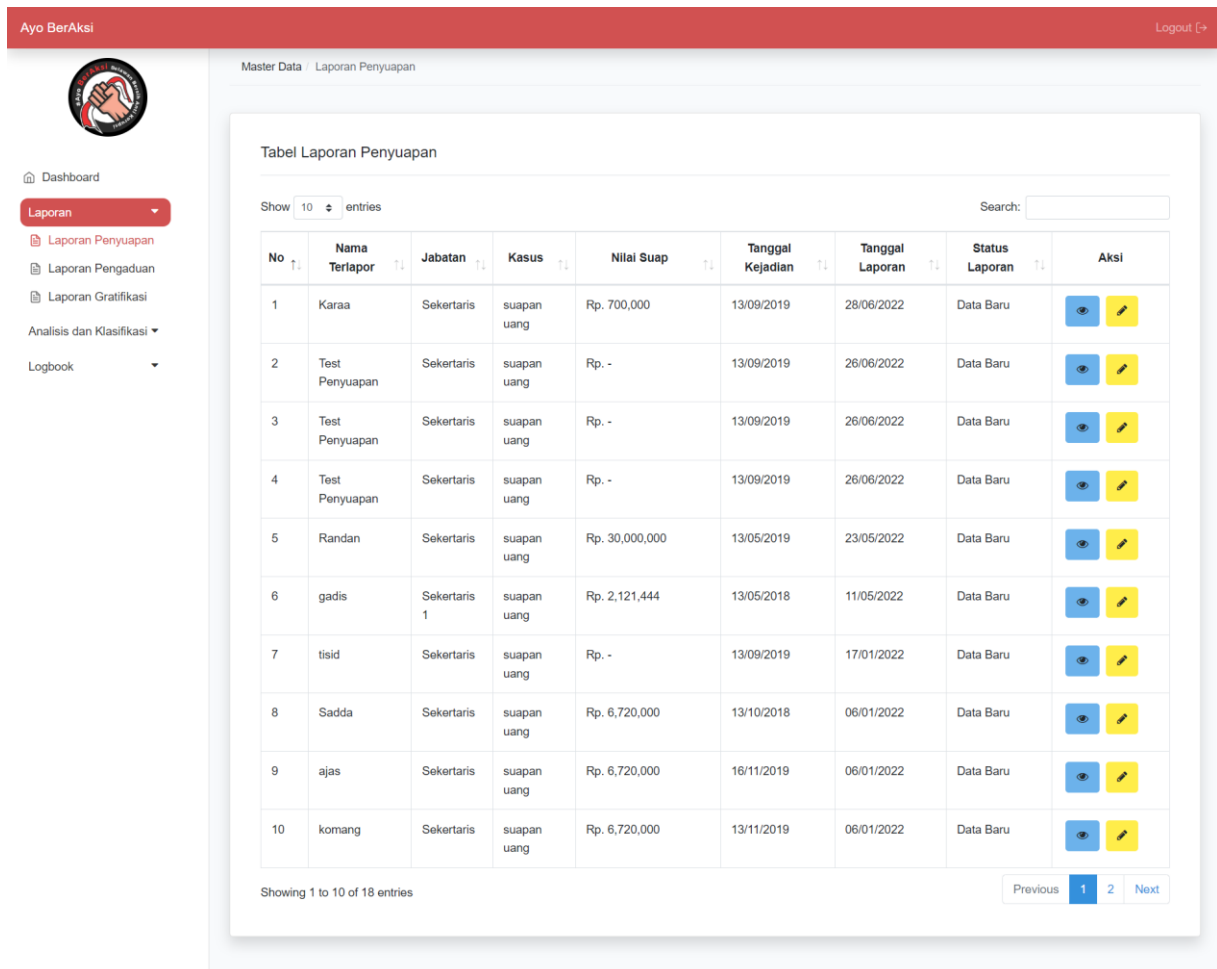
```

app > Http > Controllers > Actions > LaporanPenyuapanController.php
53     public function store(Request $request)
54     {
55         DB::beginTransaction();
56         try {
57             $validator = Validator::make($request->all(), [
58                 'nama_terlapor' => 'required',
59                 'jabatan' => 'required',
60                 'instansi' => 'required',
61                 'kasus_suap' => 'required',
62                 'lokasi' => 'required',
63                 'tanggal_kejadian' => 'required',
64                 'kronologis_kejadian' => 'required',
65             ]);
66
67             if ($validator->fails()) {
68                 return response()->json([
69                     'success' => false,
70                     'message' => $validator->errors(),
71                 ], 401);
72             }
73
74             //save data
75             $penyuapan = new LaporanPenyuapan();
76             $penyuapan->nama_terlapor = $request->nama_terlapor;
77             $penyuapan->jabatan = $request->jabatan;
78             $penyuapan->instansi = $request->instansi;
79             $penyuapan->id_pelapor = Auth::user()->id;
80             $penyuapan->nama_pelapor = Auth::user()->name;
81             $penyuapan->jabatan_pelapor = $request->jabatan_pelapor;
82             $penyuapan->instansi_pelapor = $request->instansi_pelapor;
83             $penyuapan->kasus_suap = $request->kasus_suap;
84             $penyuapan->nilai_suap = $request->nilai_suap;
85             $penyuapan->lokasi = $request->lokasi;
86             $penyuapan->tanggal_kejadian = $request->tanggal_kejadian;
87             $penyuapan->tanggal_pelaporan = Carbon::now()->format('Y-m-d');
88             $penyuapan->kronologis_kejadian = $request->kronologis_kejadian;
89             $penyuapan->status = $request->status;
90             $penyuapan->deskripsi_status = $request->deskripsi_status;

```

Gambar 3.11 Controller Action Laporan Penyuapan





















Fitur ini memanfaatkan operasi CRUD dimana pengguna dapat membuat laporan, membaca laporan, dan mengedit laporan. Pengguna dari fitur ini adalah seluruh *stakeholder* dari sistem yaitu: kepala balai, pegawai, tim kepatuhan, dan masyarakat. Gambar 3.12 merupakan tampilan daftar laporan yang masuk.



Master Data | Laporan Penyuapan

Tabel Laporan Penyuapan

Show 10 entries Search:

| No | Nama Terlapor  | Jabatan      | Kasus       | Nilai Suap     | Tanggal Kejadian | Tanggal Laporan | Status Laporan | Aksi  |
|----|----------------|--------------|-------------|----------------|------------------|-----------------|----------------|---|
| 1  | Karaa          | Sekretaris   | suapan uang | Rp. 700,000    | 13/09/2019       | 28/06/2022      | Data Baru      |       |
| 2  | Test Penyuapan | Sekretaris   | suapan uang | Rp. -          | 13/09/2019       | 26/06/2022      | Data Baru      |       |
| 3  | Test Penyuapan | Sekretaris   | suapan uang | Rp. -          | 13/09/2019       | 26/06/2022      | Data Baru      |       |
| 4  | Test Penyuapan | Sekretaris   | suapan uang | Rp. -          | 13/09/2019       | 26/06/2022      | Data Baru      |       |
| 5  | Randan         | Sekretaris   | suapan uang | Rp. 30,000,000 | 13/05/2019       | 23/05/2022      | Data Baru      |   |
| 6  | gadis          | Sekretaris 1 | suapan uang | Rp. 2,121,444  | 13/05/2018       | 11/05/2022      | Data Baru      |   |
| 7  | tisid          | Sekretaris   | suapan uang | Rp. -          | 13/09/2019       | 17/01/2022      | Data Baru      |   |
| 8  | Sadda          | Sekretaris   | suapan uang | Rp. 6,720,000  | 13/10/2018       | 06/01/2022      | Data Baru      |   |
| 9  | ajas           | Sekretaris   | suapan uang | Rp. 6,720,000  | 16/11/2019       | 06/01/2022      | Data Baru      |   |
| 10 | komang         | Sekretaris   | suapan uang | Rp. 6,720,000  | 13/11/2019       | 06/01/2022      | Data Baru      |   |

Showing 1 to 10 of 18 entries Previous 1 2 Next

Gambar 3.12 Daftar Laporan Penyuapan yang Masuk

Untuk membuat laporan penyuapan baru, hanya dapat dilakukan melalui aplikasi *android*, dimana pengguna perlu melakukan *login* akun terlebih dahulu sebelum mengisi *form*. Pengguna perlu mengisi *field* dari *form* seperti nama terlapor, jabatan, instansi, kasus suap yang dilakukan, nilai suap, lokasi kejadian, tanggal\_kejadian, dan kronologis kejadian. *Form* untuk membuat laporan dapat dilihat pada Gambar 3.13.

Gambar 3.13 *Form* Laporan Penyuapan pada *Android*

Laporan penyuapan yang masuk dapat dilihat oleh kepala balai dan tim kepatuhan melalui *website*. Kepala balai dapat membaca laporan yang masuk, kemudian kepala balai akan memilih tiga orang tim kepatuhan untuk menganalisa dan menyelidiki kasus suap yang terjadi. Tim kepatuhan hanya dapat melihat laporan suap yang ditugaskan kepadanya. Tampilan detail laporan penyuapan seperti pada Gambar 3.14.



The screenshot shows a web application interface with a red header bar containing 'Ayo BerAksi' and 'Logout'. The main content area is titled 'Detail Laporan Penyuapan' and contains a form with the following fields:

- Nama Terlapor: Karas
- Jabatan: Sekertaris
- Asal Perusahaan: Jak\*\*\* Sport
- Kasus Penyuapan: suapan uang
- Nilai Suap: 700000
- Lokasi Kejadian: Kantor JKS
- Tanggal Kejadian: 13/09/2019
- Tanggal Pelaporan: 28/06/2022
- Kronologis: Pelaku memberikan sejumlah uang kepada tersangka menggunakan .....
- Nama Pelapor: Seraya
- Status Laporan: Diterima
- Alasan: Alasan

A blue button labeled 'Kembali' is located at the bottom right of the form.

Gambar 3.14 Detail Laporan Penyuapan

Untuk mengubah laporan dapat dilakukan oleh kepala balai, dimana kepala balai dapat menentukan apakah laporan penyuapan tersebut diterima atau tidak. Jika diterima maka kepala balai akan memilih tiga orang tim kepatuhan, sedangkan jika ditolak maka kepala balai wajib mengisi alasan penolakan. Tampilan laporan penyuapan seperti pada Gambar 3.15.

The screenshot shows a web application interface for reporting a bribery case. The interface is in Indonesian and features a sidebar on the left with navigation options: Dashboard, Laporan (selected), Laporan Penyuapan, Laporan Pengaduan, Laporan Gratifikasi, Analisis dan Klasifikasi, and Logbook. The main content area is titled 'Master Data | Laporan Penyuapan | Tambah Laporan Penyuapan' and contains two forms.

**Detail Laporan Penyuapan**

Form fields include:

- Nama Terlapor: Karaa
- Jabatan: Sekertaris
- Asal Perusahaan: Jak\*\*\* Sport
- Kasus Penyuapan: suapan uang
- Nilai Suap: Rp. 700,000
- Lokasi Kejadian: Kantor JKS
- Tanggal Kejadian: 13/09/2019
- Tanggal Pelaporan: 28/06/2022
- Kronologis: Pelaku memberikan sejumlah uang kepada tersangka menggunakan .....
- Nama Pelapor: Seraya
- Jabatan Pelapor: Jabatan Pelapor
- Asal Perusahaan Pelapor: Instansi Pelapor

**Tambah Laporan Penyuapan**

Form fields include:

- Status Laporan:  Diterima  Ditolak
- Ketua: Lala
- Anggota 1: Bagus
- Anggota 2: Indra

Buttons: Kembali (blue), Simpan (green)

Gambar 3.15 Proses Laporan Penyuapan

#### d. Mengembangkan Fitur Laporan Pengaduan

Setelah selesai membuat fitur laporan pengaduan, tahapan pengembangan dilanjutkan dengan membuat fitur laporan pengaduan. Pengguna dari fitur ini yaitu seluruh *stakeholder* yaitu: kepala balai, tim kepatuhan, pegawai, dan masyarakat. Pengembangan dimulai dengan membuat *model* dari fitur yang akan menjadi komponen yang berhubungan dan mengelola secara langsung *database*, seperti Gambar 3.16.

```

app > Models > LaporanPengaduan.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class LaporanPengaduan extends Model
9  {
10     use HasFactory;
11
12     protected $table = 'laporan_pengaduans';
13     protected $fillable = [
14         'nama_ketua',          // nama yang dilaporkan
15         'id_pelapor',         // id pelapor
16         'nama_pelapor',
17         'alamat',
18         'nik',
19         'uraian_laporan',
20         'saran_masukan',
21         'tanggal_pengaduan',
22         'status',             // status laporan
23         'deskripsi_status',
24     ];
25
26     // Relationship
27     public function user()
28     {
29         return $this->belongsTo('App\Models\User', 'id_pelapor', 'id');
30     }
31
32     public function timPemeriksa()
33     {
34         return $this->hasMany('App\Models\TimPemeriksaPengaduan');
35     }
36

```

Gambar 3.16 Model Laporan Pengaduan

Kemudian dilanjutkan dengan membuat *migrations* seperti Gambar 3.17 untuk membuat skema basis data, dan membuat *controllers* untuk menghubungkan *models* dan *view* seperti pada Gambar 3.18 serta *controllers actions* yang mengatur manipulasi data seperti penyimpanan, perubahan, dan penghapusan data seperti Gambar 3.19.

```

database > migrations > 2021_11_16_030348_create_laporan_pengaduans_table.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  class CreateLaporanPengaduansTable extends Migration
8  {
9      /**
10     * Run the migrations.
11     *
12     * @return void
13     */
14     public function up()
15     {
16         Schema::create('laporan_pengaduans', function (Blueprint $table) {
17             $table->id();
18             $table->string('nama_ketua');
19             $table->unsignedBigInteger('id_pelapor')->nullable();
20             $table->string('nama_pelapor');
21             $table->string('alamat');
22             $table->string('nik');
23             $table->text('uraian_laporan');
24             $table->text('saran_masukan');
25             $table->date('tanggal_pengaduan')->nullable();
26             $table->string('status')->nullable();
27             $table->string('deskripsi_status')->nullable();
28             $table->timestamps();
29         });
30     }
31 }

```

Gambar 3.17 Migration Laporan Pengaduan

```

app > Http > Controllers > Web > LaporanPengaduanController.php
1  <?php
2
3  namespace App\Http\Controllers\Web;
4
5  use App\Http\Controllers\Controller;
6  use App\Helpers\LaporanHelper as LprHelper;
7  use Illuminate\Http\Request;
8
9  class LaporanPengaduanController extends Controller
10 {
11     public function index()
12     {
13         return view('pengaduan.view');
14     }
15
16     public function create()
17     {
18         return view('pengaduan.form');
19     }
20
21     public function show($id)
22     {
23         $data = array(
24             'showPengaduan' => LprHelper::showPengaduan($id),
25             'id' => $id,
26         );
27         return view('pengaduan.show', $data);
28     }
29
30     public function edit($id)
31     {
32         $data = array(
33             'showPengaduan' => LprHelper::showPengaduan($id),
34             'tim_kepatuhan' => LprHelper::timKepatuhan(),
35             'detailTim' => LprHelper::detailTimPengaduan($id),
36             'id' => $id,
37         );
38         return view('pengaduan.form', $data);

```

Gambar 3.18 Controller View Laporan Pengaduan

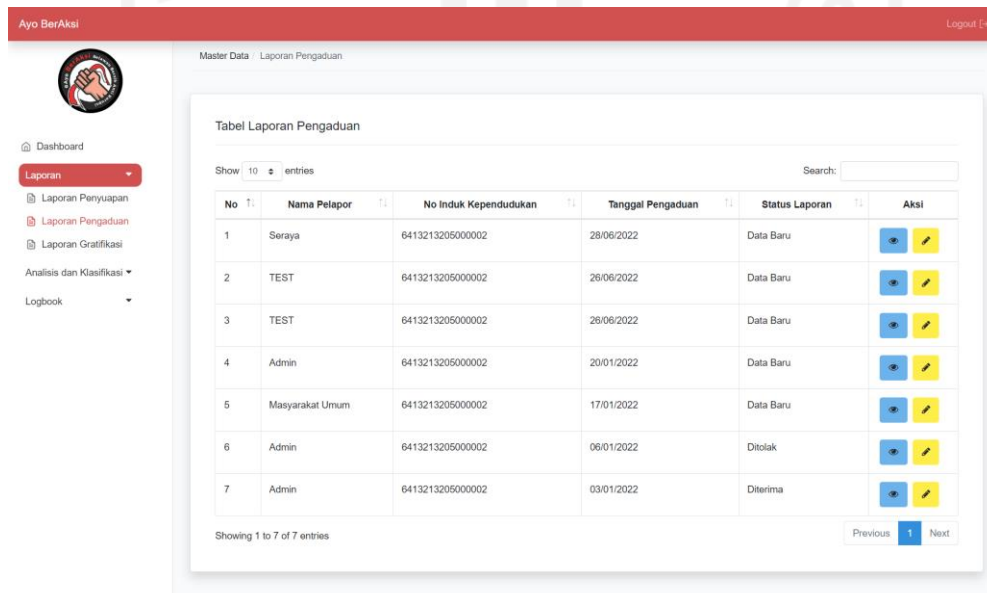
```

app > Http > Controllers > Actions > LaporanPengaduanController.php
139 public function update(Request $request, $id)
140 {
141     DB::beginTransaction();
142     try {
143         $validator = Validator::make($request->all(), [
144             'alamat' => 'required',
145             'nik' => 'required',
146             'uraian_laporan' => 'required',
147             'saran_masukan' => 'required',
148         ]);
149
150         if ($validator->fails()) {
151             return response()->json([
152                 'success' => false,
153                 'message' => $validator->errors(),
154             ], 401);
155         }
156
157         // save data
158         $laporan = LaporanPengaduan::findOrFail($id);
159         $laporan->nama_ketua = $request->nama_ketua;
160         $laporan->id_pelapor = Auth::user()->id;
161         $laporan->nama_pelapor = Auth::user()->name;
162         $laporan->alamat = $request->alamat;
163         $laporan->nik = $request->nik;
164         $laporan->uraian_laporan = $request->uraian_laporan;
165         $laporan->saran_masukan = $request->saran_masukan;
166         $laporan->tanggal_pengaduan = $request->tanggal_pengaduan;
167         $laporan->save();
168
169         $request->id_pengguna = Auth::user()->id;
170         $request->title = "Laporan Gratifikasi";
171         $request->body = "Laporkan Gratifikasi anda berhasil diupdate";
172
173         $notif = $this->PushNotificationController->sendPushNotification($request);
174         // $notification = PushNotificationController::sendPushNotification($request);
175
176         $data = new LaporanPengaduanResource($laporan);















```

Gambar 3.19 Controller Action Laporan Pengaduan

Fitur ini kembali memanfaatkan CRUD dimana pengguna dapat membuat, mengedit, dan membaca laporan pengaduan. Gambar 3.20 merupakan tampilan daftar laporan pengaduan.



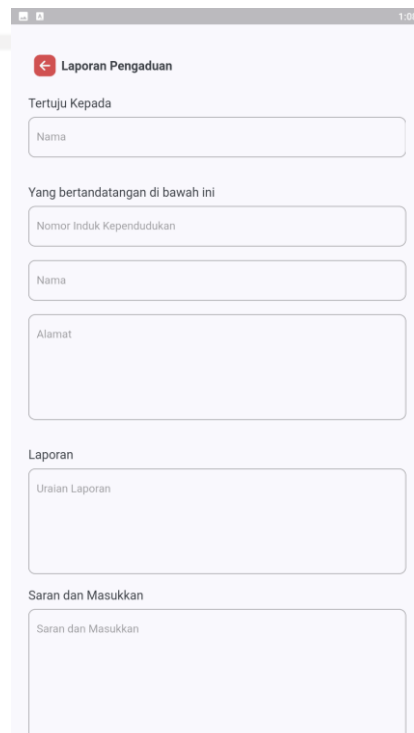
The screenshot shows a web application interface for managing complaint reports. The sidebar on the left contains navigation options: Dashboard, Laporan (selected), Laporan Penyuaian, Laporan Pengaduan, Laporan Gratifikasi, Analisis dan Klasifikasi, and Logbook. The main content area displays a table titled "Tabel Laporan Pengaduan" with 7 entries. The table columns are No, Nama Pelapor, No Induk Kependudukan, Tanggal Pengaduan, Status Laporan, and Aksi. The data rows are as follows:

| No | Nama Pelapor    | No Induk Kependudukan | Tanggal Pengaduan | Status Laporan | Aksi  |
|----|-----------------|-----------------------|-------------------|----------------|---|
| 1  | Seraya          | 6413213205000002      | 28/06/2022        | Data Baru      |   |
| 2  | TEST            | 6413213205000002      | 26/06/2022        | Data Baru      |   |
| 3  | TEST            | 6413213205000002      | 26/06/2022        | Data Baru      |   |
| 4  | Admin           | 6413213205000002      | 20/01/2022        | Data Baru      |   |
| 5  | Masyarakat Umum | 6413213205000002      | 17/01/2022        | Data Baru      |   |
| 6  | Admin           | 6413213205000002      | 06/01/2022        | Ditolak        |   |
| 7  | Admin           | 6413213205000002      | 03/01/2022        | Diterima       |   |

At the bottom of the table, it shows "Showing 1 to 7 of 7 entries" and navigation buttons for "Previous", "1", and "Next".

Gambar 3.20 Daftar Laporan Pengaduan

Seperti fitur sebelumnya, untuk membuat laporan pengaduan baru pengguna hanya dapat melakukannya melalui *android*, serta pengguna perlu melakukan *login* terlebih dahulu. Selain itu, pengguna juga perlu mengisi beberapa *field* pada *form* seperti nama orang yang dituju, NIK, nama pengguna, alamat, uraian dari laporan, dan saran serta masukan. Tampilan *form* pada *android* dapat dilihat pada Gambar 3.21.



Gambar 3.21 *Form* Tambah Laporan Pengaduan pada *Android*

Setelah laporan pengaduan berhasil dibuat, laporan akan diterima oleh kepala balai untuk dilihat seperti pada Gambar 3.22. Selanjutnya kepala balai dapat melakukan perubahan dengan memilih tiga tim kepatuhan jika laporan diterima dan mengisi alasan jika laporan tidak diterima.

The image shows a web application interface for reporting a complaint. The interface is titled "Ayo Beraksi" and has a sidebar with navigation options: Dashboard, Laporan, Laporan Penyusunan, Laporan Pengaduan, Laporan Gratifikasi, Analisis dan Klasifikasi, and Logbook. The main content area is divided into two sections: "Detail Laporan Pengaduan" and "Tambah Laporan Pengaduan".

The "Detail Laporan Pengaduan" section contains the following fields:

- Kepada: Payan
- Nama Pelapor: Seraya
- Alamat Lengkap: Kerayan
- No Induk Kependudukan: 6413213205000002
- Uraian Laporan: Laporana yang saya kirimkan kemaren tidak diproses secara baik .....
- Saran dan Masukan: sebaiknya segera dikaji ulang pengambilan keputusan .....
- Tanggal Pengaduan: 28/06/2022

The "Tambah Laporan Pengaduan" section contains the following fields:

- Status Laporan:  Diterima  Ditolak
- Ketua: Bagus
- Anggota 1: Lala
- Anggota 2: Indra

At the bottom right of the form, there are two buttons: "Kembali" (blue) and "Simpan" (green).

Gambar 3.22 Proses Laporan Pengaduan

e. Mengembangkan Fitur Laporan Gratifikasi

Kemudian setelah fitur laporan pengaduan berhasil dikembangkan, proses pengembangan aplikasi dilanjutkan untuk membuat fitur laporan gratifikasi. Pengembangan dimulai dengan membuat *model* dari fitur yang akan menjadi komponen yang berhubungan dan mengelola secara langsung *database*, seperti Gambar 3.23.

```

app > Models > LaporanGratifikasi.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class LaporanGratifikasi extends Model
9  {
10     use HasFactory;
11
12     protected $table = 'laporan_gratifikasis';
13     protected $fillable = [
14         'no_laporan',
15         'pengirim',
16         'id_pelapor',          // id pelapor
17         'nama_pelapor',
18         'nama_terlapor',      // nama yang dilaporkan
19         'jabatan',
20         'instansi',
21         'tanggal_kejadian',
22         'tanggal_pelaporan',
23         'kronologis_kejadian',
24         'status',             // status laporan
25         'deskripsi_status',   // alasan diambil
26         'tindaklanjut',
27     ];
28
29     // Relationship
30     public function user()
31     {
32         return $this->belongsTo('App\Models\User', 'id_pelapor', 'id');
33     }
34
35     public function timPemeriksa()
36     {
37         return $this->hasMany('App\Models\TimPemeriksaGratifikasi');
38     }

```

Gambar 3.23 *Models* Laporan Gratifikasi

Kemudian dilanjutkan dengan membuat *migrations* seperti Gambar 3.24 untuk membuat skema basis data, dan membuat *controllers* untuk menghubungkan *models* dan *view* seperti pada Gambar 3.25 serta *controllers actions* yang mengatur manipulasi data seperti penyimpanan, perubahan, dan penghapusan data seperti Gambar 3.26.



```

database > migrations > 2021_11_16_032749_create_laporan_gratifikasi_table.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  class CreateLaporanGratifikasiTable extends Migration
8  {
9      /**
10     * Run the migrations.
11     *
12     * @return void
13     */
14     public function up()
15     {
16         Schema::create('laporan_gratifikasis', function (Blueprint $table) {
17             $table->id();
18             $table->string('no_laporan')->nullable();
19             $table->string('pengirim')->nullable();
20             $table->unsignedBigInteger('id_pelapor')->nullable();
21             $table->string('nama_pelapor');
22             $table->string('nama_terlapor');
23             $table->string('jabatan');
24             $table->string('instansi');
25             $table->date('tanggal_kejadian')->nullable();
26             $table->date('tanggal_pelaporan')->nullable();
27             $table->text('kronologis_kejadian');
28             $table->string('status')->nullable();
29             $table->string('deskripsi_status')->nullable();
30             $table->string('tindaklanjut')->nullable();
31             $table->timestamps();
32         });
33     }
34

```

Gambar 3.24 Migrations Laporan Gratifikasi

```

app > Http > Controllers > Web > LaporanGratifikasiController.php
4
5  use App\Http\Controllers\Controller;
6  use App\Helpers\LaporanHelper as LprHelper;
7  use Illuminate\Http\Request;
8
9  class LaporanGratifikasiController extends Controller
10 {
11     public function index()
12     {
13         return view('gratifikasi.view');
14     }
15
16     public function create()
17     {
18         return view('gratifikasi.form');
19     }
20
21     public function show($id)
22     {
23         $data = array(
24             'showGratifikasi' => LprHelper::showGratifikasi($id),
25             'id' => $id,
26         );
27         return view('gratifikasi.show', $data);
28     }
29
30     public function edit($id)
31     {
32         $data = array(
33             'showGratifikasi' => LprHelper::showGratifikasi($id),
34             'tim_kepatuhan' => LprHelper::timKepatuhan(),
35             'detailTim' => LprHelper::detailTimGratifikasi($id),
36             'id' => $id,
37         );
38         return view('gratifikasi.form', $data);
39     }
40 }

```

Gambar 3.25 Controllers View Laporan Gratifikasi













```

app > Http > Controllers > Actions > LaporanGratifikasiController.php
52 public function store(Request $request)
53 {
54     DB::beginTransaction();
55     try {
56         $validator = Validator::make($request->all(), [
57             'nama_terlapor' => 'required',
58             'tanggal_kejadian' => 'required',
59             'kronologis_kejadian' => 'required',
60         ]);
61
62         if ($validator->fails()) {
63             return response()->json([
64                 'success' => false,
65                 'message' => $validator->errors(),
66             ], 401);
67         }
68
69         // save data
70         $laporan = new LaporanGratifikasi();
71         $laporan->no_laporan = $request->no_laporan;
72         $laporan->pengirim = $request->pengirim;
73         $laporan->id_pelapor = Auth::user()->id;
74         $laporan->nama_pelapor = Auth::user()->name;
75         $laporan->nama_terlapor = $request->nama_terlapor;
76         $laporan->jabatan = $request->jabatan;
77         $laporan->instansi = $request->instansi;
78         $laporan->tanggal_kejadian = $request->tanggal_kejadian;
79         $laporan->tanggal_pelaporan = Carbon::now()->format('Y-m-d');
80         $laporan->kronologis_kejadian = $request->kronologis_kejadian;
81         $laporan->status = $request->status;
82         $laporan->deskripsi_status = $request->deskripsi_status;
83         $laporan->tindaklanjuti = $request->tindaklanjuti;
84         $laporan->save();
85
86         $request->id_pengguna = Auth::user()->id;
87         $request->title = "Laporan Gratifikasi";

```

Gambar 3.26 *Controllers Action* Laporan Gratifikasi

Seperti pada fitur sebelumnya laporan gratifikasi akan memanfaatkan CRUD dimana pengguna dapat membuat, membaca, dan mengubah laporan gratifikasi. Pengguna dari fitur ini yaitu: kepala balai, pegawai, tim kepatuhan dan masyarakat. Tampilan daftar laporan gratifikasi yang masuk seperti Gambar 3.27.

| No | Nama Pelapor    | Nama Tertapor    | Instansi/Perusahaan | Jabatan                 | Tanggal Kejadian | Status Laporan | Aksi  |
|----|-----------------|------------------|---------------------|-------------------------|------------------|----------------|---|
| 1  | Seraya          | Walan            | Bright To***        | Manager Ketenagakerjaan | 15/08/2020       | Data Baru      |   |
| 2  | TEST            | Test gratifikasi | Bright To***        | Manager Ketenagakerjaan | 15/08/2020       | Data Baru      |   |
| 3  | TEST            | Test gratifikasi | Bright To***        | Manager Ketenagakerjaan | 15/08/2020       | Data Baru      |   |
| 4  | Admin           | koja             | Bright Tower        | Manager Ketenagakerjaan | 15/08/2020       | Data Baru      |   |
| 5  | Masyarakat Umum | Madam bak        | Bright Tower        | Manager Ketenagakerjaan | 21/11/2012       | Diterima       |   |
| 6  | Admin           | Fadian           | Hotel Bintang       | Manager keuangan        | 21/08/2020       | Diterima       |   |

Gambar 3.27 Daftar Laporan Gratifikasi yang Masuk

Pembuatan laporan gratifikasi, hanya dapat dilakukan pada aplikasi *android*. Dimana semua pengguna dapat melaporkan kasus gratifikasi yang terjadi. Pengguna harus login terlebih dahulu sebelum bisa mengirim laporan. Selain itu, pengguna mengisi beberapa *field* seperti nama terlapor, jabatan, instansi, tanggal kejadian, serta kronologis kejadian dari gratifikasi. Tampilan *form* pada *android* dapat dilihat pada Gambar 3.28.

Gambar 3.28 *Form* Laporan Gratifikasi pada *Android*

Laporan yang berhasil dibuat akan diterima dan dibaca oleh kepala balai, tampilan laporan pada *website* dapat dilihat pada Gambar 3.29. Seperti fitur sebelumnya, kepala balai akan memilih apakah laporan ditolak atau diterima. Jika laporan ditolak maka akan ada *field* alasan dan masukan tindak lanjut, sedangkan jika diterima juga terdapat *field* alasan dan tindak lanjut, serta memilih tim kepatuhan yang menindak lanjuti laporan gratifikasi.

The screenshot displays the 'Ayo BerAksi' web application interface. The top navigation bar includes 'Ayo BerAksi' on the left and 'Logout [+]' on the right. The breadcrumb trail shows 'Master Data > Laporan Gratifikasi > Pengajuan Laporan Gratifikasi'. The left sidebar contains a menu with 'Dashboard', 'Laporan' (highlighted), 'Laporan Penyusunan', 'Laporan Pengaduan', 'Laporan Gratifikasi', 'Analisis dan Klasifikasi', and 'Logbook'. The main content area is divided into two sections:

**Detail Laporan Pengaduan**

- Nomor Laporan: Belum ada
- Nama Pelapor: Seraya
- Nama Pengirim: Seraya
- Nama Terlapor: Walan
- Jabatan: Manager Ketenagakerjaan
- Instansi/Perusahaan: Bright To\*\*\*
- Kronologis Kejadian: Tersangka memberikan kendaraan mobil sejumlah..... pada saat .....
- Tanggal Kejadian: 28/06/2022
- Tanggal Pelaporan: 15/08/2020

**Tambah Laporan Pengaduan**

Hasil Analisa Tim Kepatuhan :

Diterima  Ditolak

Alasan: Benar Tersangka memberikan mobil kepada .....

Tindaklanjut: Lanjutkan Proses Penyelidikan .....

Ketua: Indra

Anggota 1: Bagus

Anggota 2: Lala

Buttons: Kembali (blue), Simpan (green)

Gambar 3.29 Proses Laporan Gratifikasi

f. Mengembangkan Fitur Analisa Laporan Penyusunan

Pengembangan dimulai dengan membuat *model* dari fitur yang akan menjadi komponen yang berhubungan dan mengelola secara langsung *database*, seperti Gambar 3.30.

```

app > Models > AnalisisLaporanPenyuapan.php
1  |<?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class AnalisisLaporanPenyuapan extends Model
9  {
10     use HasFactory;
11
12     protected $table = 'analisis_laporan_penyuapans';
13     protected $fillable = [
14         'id_penyuapan',
15         'hasil_investigasi',
16         'kesimpulan',
17     ];
18
19     // Relationship
20     public function laporanPenyuapan()
21     {
22         return $this->belongsTo('App\Models\LaporanPenyuapan', 'id_pengaduan', 'id');
23     }
24 }
25

```

Gambar 3.30 *Models* Analisa Laporan Penyuapan

Kemudian dilanjutkan dengan membuat *migrations* seperti Gambar 3.31 untuk membuat skema basis data, dan membuat *controllers* untuk menghubungkan *models* dan *view* seperti pada Gambar 3.32 serta *controllers actions* yang mengatur manipulasi data seperti penyimpanan, perubahan, dan penghapusan data seperti Gambar 3.33.

```

database > migrations > 2021_12_24_034749_create_analisis_laporan_penyuapans_table.php
1  |<?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  class CreateAnalisisLaporanPenyuapansTable extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::create('analisis_laporan_penyuapans', function (Blueprint $table) {
17             $table->id();
18             $table->unsignedBigInteger('id_penyuapan')->nullable();
19             $table->text('hasil_investigasi');
20             $table->text('kesimpulan');
21             $table->timestamps();
22         });
23     }
24 }

```

Gambar 3.31 *Migrations* Analisa Laporan Penyuapan

```

app > Http > Controllers > Web > AnalisisLaporanPenyuapanController.php
1  k?php
2
3  namespace App\Http\Controllers\Web;
4
5  use App\Http\Controllers\Controller;
6  use App\Helpers\LaporanHelper as LprHelper;
7  use Illuminate\Http\Request;
8
9  class AnalisisLaporanPenyuapanController extends Controller
10 {
11     /**
12      * Display a listing of the resource.
13      *
14      * @return \Illuminate\Http\Response
15      */
16     public function index()
17     {
18         return view('analisisLaporan.penyuapan.view');
19     }
20
21     /**
22      * Show the form for creating a new resource.
23      *
24      * @return \Illuminate\Http\Response
25      */
26     public function create()
27     {
28         return view('analisisLaporan.penyuapan.form');
29     }
30

```

Gambar 3.32 *Controllers View* Analisa Laporan Penyuapan

```





app > Http > Controllers > Actions > AnalisisLaporanPenyuapanController.php
46     public function store(Request $request)
47     {
48         DB::beginTransaction();
49         try {
50             $validator = Validator::make($request->all(), [
51                 'hasil_investigasi' => 'required',
52                 'kesimpulan' => 'required',
53             ]);
54
55             if ($validator->fails()) {
56                 return response()->json([
57                     'success' => false,
58                     'message' => $validator->errors(),
59                 ], 401);
60             }
61
62             $analisis = new AnalisisLaporanPenyuapan();
63             $analisis->id_penyuapan = $request->id_penyuapan;
64             $analisis->hasil_investigasi = $request->hasil_investigasi;
65             $analisis->kesimpulan = $request->kesimpulan;
66             $analisis->save();
67
68             $logbook = new LogbookLaporanPenyuapan();
69             $logbook->id_penyuapan = $request->id_penyuapan;
70             $logbook->save();
71
72             $request->id_pengguna = $request->id_pengguna;
73             $request->title = "Laporan Penyuapan";
74             $request->body = "Laporan Penyuapan Anda sudah di analisis";
75             $notif = $this->PushNotificationController->sendPushNotification($request);
76
77             DB::commit();
78             $message = 'Data analisis Berhasil disimpan';
79             return response()->json([
80                 'data' => $analisis,
81                 'message' => $message,
82             ]);
83         } catch (\Throwable $th) {

```

Gambar 3.33 *Controller Actions* Analisa Laporan Penyuapan

Fitur analisa laporan merupakan fitur yang dapat digunakan tim kepatuhan, dan kepala balai. Pada fitur ini tim kepatuhan dapat membuat hasil analisa, mengubah hasil analisa. Sedangkan kepala balai hanya dapat memantau jalanya proses penyelidikan dan hasil analisa yang dibuat tim kepatuhan. Gambar 3.34 merupakan tampilan daftar analisa dari laporan yang diterima untuk tindak lanjut.

The screenshot shows a web application interface with a red header bar containing 'Ayo BerAksi' and 'Logout (->'. Below the header is a sidebar with a logo and navigation menu items: Dashboard, Penugasan (highlighted), Analisis Penyusunan, Klasifikasi Pengaduan, Analisis Gratifikasi, and Logbook. The main content area is titled 'Master Data Analisis Laporan Penyusunan' and contains a table titled 'Tabel Analisis Laporan Penyusunan'. The table has columns for No, Nama Terlapor, Jabatan, Kasus, Tanggal Kejadian, Tanggal Laporan, and Aksi. There are two rows of data. Below the table, it says 'Showing 1 to 2 of 2 entries' and has 'Previous', '1', and 'Next' navigation buttons.

| No | Nama Terlapor | Jabatan      | Kasus       | Tanggal Kejadian | Tanggal Laporan | Aksi  |
|----|---------------|--------------|-------------|------------------|-----------------|---|
| 1  | Gill          | Sekretaris 3 | suapan uang | 13/09/2018       | 03/01/2022      |   |
| 2  | Karaa         | Sekretaris   | suapan uang | 13/09/2019       | 28/06/2022      |   |

Gambar 3.34 Daftar Analisa Laporan Penyusunan

Tim kepatuhan dapat menambahkan hasil analisa dan penyelidikan pada *website*. Dimana hanya anggota tim yang telah dipilih yang dapat mengisi form analisa. Dalam *form* analisa terdapat data lengkap dari laporan penyusunan, dan tim kepatuhan dapat mengisi *field* hasil investigasi serta Kesimpulan yang didapat. Gambar *form* membuat analisa laporan penyusunan dapat dilihat pada Gambar 3.35.

The screenshot shows a web application interface for managing bribery reports. The top navigation bar includes 'Ayo BerAksi' and 'Logout (-)'. The sidebar on the left contains a logo and navigation items: Dashboard, Penugasan (selected), Analisis Penyuaipan, Klasifikasi Pengaduan, Analisis Gratifikasi, and Logbook. The main content area is titled 'Detail Laporan Penyuaipan' and contains the following fields:

- Nama Terlapor: Karaa
- Jabatan: Sekertaris
- Asal Perusahaan: Jak\*\*\* Sport
- Kasus Penyuaipan: suapan uang
- Nilai Suap: 700000
- Lokasi Kejadian: Kantor JKS
- Tanggal Kejadian: 13/09/2019
- Tanggal Pelaporan: 28/06/2022
- Kronologi: Pelaku memberikan sejumlah uang kepada tersangka menggunakan .....
- Nama Pelapor: Seraya
- Jabatan Pelapor: Jabatan Pelapor
- Asal Perusahaan Pelapor: Instansi Pelapor

Below the detail form is a section titled 'Tambah Analisis Laporan Penyuaipan' with two text areas:

- Hasil Investigasi: Benar telah terjadi penyuaipan pada anggota yang bernama .....
- Kesimpulan Tim Investigasi: Ajukan pelaporan kepada KPK .....

At the bottom right of this section are two buttons: 'Kembali' (blue) and 'Simpan' (green).

Gambar 3.35 *Form* Tambah Analisa Penyuaipan

Untuk mengubah data hasil analisa tim kepatuhan dapat masuk kembali ke halaman *edit*, kemudian mengubah *field* yang sudah diisi. Selain itu, kepala balai dapat memantau jalan proses penyelidikan dan analisa dengan fitur melihat analisa penyuaipan. Tampilan fitur lihat analisa dapat dilihat pada Gambar 3.36.



The screenshot shows a web application interface with a red header bar containing 'Ayo BerAksi' and a 'Logout' link. The main content area is titled 'Detail Laporan Penyuapan' and contains a form with the following fields:

- Nama Teraapor: Karaa
- Jabatan: Sekertaris
- Asal Perusahaan: Jak\*\*\* Sport
- Kasus Penyuapan: suapan uang
- Nilai Suap: 700000
- Lokasi Kejadian: Kantor JKS
- Tanggal Kejadian: 13/09/2019
- Tanggal Pelaporan: 28/06/2022
- Kronologis: Pelaku memberikan sejumlah uang kepada tersangka menggunakan .....
- Nama Pelapor: Seraya

A blue 'Kembali' button is located at the bottom right of the form.

Gambar 3.36 Detail Analisa Penyuapan

g. Mengembangkan Fitur Klasifikasi Laporan Pengaduan

Fitur Klasifikasi laporan pengaduan akan digunakan untuk mengklasifikasi laporan pengaduan yang masuk dari masyarakat. Fitur klasifikasi akan digunakan oleh tim kepatuhan dan diawasi oleh kepala balai. Pengembangan dimulai dengan membuat *model* dari fitur yang akan menjadi komponen yang berhubungan dan mengelola secara langsung *database*, seperti Gambar 3.37.

```

app > Models > KlasifikasiLaporanPengaduan.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class KlasifikasiLaporanPengaduan extends Model
9  {
10     use HasFactory;
11
12     protected $table = 'klasifikasi_laporan_pengaduans';
13     protected $fillable = [
14         'id_pengaduan',
15         'klasifikasi',
16         'kategori',
17         'keterangan',
18     ];
19
20     // Relationship
21     public function laporanPengaduan()
22     {
23         return $this->belongsTo('App\Models\LaporanPengaduan', 'id_pengaduan', 'id');
24     }

```

Gambar 3.37 *Models* Klasifikasi Laporan Pengaduan

Kemudian dilanjutkan dengan membuat *migrations* seperti Gambar 3.38 untuk membuat skema basis data, dan membuat *controllers* untuk menghubungkan *models* dan *view* seperti pada Gambar 3.39 serta *controllers actions* yang mengatur manipulasi data seperti penyimpanan, pengubahan, dan penghapusan data seperti Gambar 3.40.

```

database > migrations > 2021_12_23_014257_create_klasifikasi_laporan_pengaduans_table.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  class CreateKlasifikasiLaporanPengaduansTable extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::create('klasifikasi_laporan_pengaduans', function (Blueprint $table) {
17             $table->id();
18             $table->unsignedBigInteger('id_pengaduan')->nullable();
19             $table->string('klasifikasi');
20             $table->string('kategori')->nullable();
21             $table->text('keterangan');
22             $table->timestamps();
23         });
24     }
25

```

Gambar 3.38 *Migrations* Klasifikasi Laporan Pengaduan

```

app > Http > Controllers > Web > KlasifikasiLaporanPengaduanController.php
1  k?php
2
3  namespace App\Http\Controllers\Web;
4
5  use App\Http\Controllers\Controller;
6  use App\Helpers\LaporanHelper as LprHelper;
7  use Illuminate\Http\Request;
8
9  class KlasifikasiLaporanPengaduanController extends Controller
10 {
11     /**
12      * Display a listing of the resource.
13      *
14      * @return \Illuminate\Http\Response
15      */
16     public function index()
17     {
18         return view('klasifikasiLaporanPengaduan.view');
19     }
20
21     /**
22      * Show the form for creating a new resource.
23      *
24      * @return \Illuminate\Http\Response
25      */
26     public function create()
27     {
28         return view('klasifikasiLaporanPengaduan.form');
29     }
30

```

Gambar 3.39 *Controllers View* Klasifikasi Laporan Pengaduan

```

app > Http > Controllers > Actions > KlasifikasiLaporanPengaduanController.php
50 public function store(Request $request)
51 {
52     DB::beginTransaction();
53     try {
54         $validator = Validator::make($request->all(), [
55             'klasifikasi' => 'required',
56             // 'kategori' => 'required',
57         ]);
58
59         if ($validator->fails()) {
60             return response()->json([
61                 'success' => false,
62                 'message' => $validator->errors(),
63             ], 401);
64         }
65
66         $analsis = new KlasifikasiLaporanPengaduan();
67         $analsis->id_pengaduan = $request->id_pengaduan;
68         $analsis->klasifikasi = $request->klasifikasi;
69         $analsis->kategori = $request->kategori;
70         if ($analsis->klasifikasi == 'A') {
71             $analsis->keterangan = 'diselesaikan BBKP Belawan';
72         }
73         if ($analsis->klasifikasi == 'B') {
74             if ($analsis->kategori == '1' || $analsis->kategori == '2' || $analsis->kategori == '3') {
75                 $analsis->keterangan = 'diselesaikan BBKP Belawan';
76             } else if ($analsis->kategori == '4' || $analsis->kategori == '5') {
77                 $analsis->keterangan = 'disampaikan kepada Inspektorat Investigasi';
78             } elseif ($analsis->kategori == '6') {
79                 $analsis->keterangan = 'disampaikan kepada Sekertariat Badan Karantina Pertanian';
80             }
81         }
82     }
83     $analsis->save();
84
85     $request->id_pengguna = $request->id_pengguna;
86     $request->title = "Laporan Pengaduan";

```

Gambar 3.40 *Controller Actions* Klasifikasi Laporan Pengaduan

Dalam mengklasifikasi laporan pengaduan, terdapat dua jenis klasifikasi dan lima jenis kategori yang masing-masing harus dipilih, untuk kemudian mendapatkan keterangan kesimpulan hasil penyelesaian laporan. Tabel 3.5 merupakan jenis klasifikasi laporan pengaduan, Tabel 3.6 merupakan tabel jenis kategori laporan pengaduan, dan

Tabel 3.7 merupakan kesimpulan hasil yang diambil dalam menyelesaikan laporan pengaduan.

Tabel 3.5 Klasifikasi Pengaduan

| Klasifikasi |                                   |
|-------------|-----------------------------------|
| No          | Keterangan                        |
| A           | Laporan tidak berkadar pengawasan |
| B           | Laporan berkadar pengawasan       |

Tabel 3.6 Kategori Pengaduan

| Kategori |   |
|----------|---|
| No       | Keterangan                                      |
| 1        | Kejadian force major                            |
| 2        | Pelanggaran kode etik/kinerja pelaku/pelaksana  |
| 3        | Pelanggaran/penyimpangan mekanisme dan prosedur |
| 4        | Penyimpangan/penyelewengan/penyalahgunaan dana  |
| 5        | Intervensi mengakibatkan kerugian masyarakat    |
| 6        | Pelanggaran hukum terhadap kebijakan/ketetapan  |

Tabel 3.7 Kesimpulan Hasil Klasifikasi

| Kesimpulan Hasil Laporan |          |  |
|--------------------------|----------|--|
| Klasifikasi              | Kategori | Keterangan   |
| A                        |          | Langsung diselesaikan BBKP Belawan                       |
| B                        | 1, 2, 3  | Diselesaikan BBKP Belawan                                |
| B                        | 4, 5     | Disampaikan Kepada Inspektorat Investigasi               |
| B                        | 6        | Disampaikan Kepada Sekretariat Badan Karantina Pertanian |





Untuk menambahkan klasifikasi laporan tim kepatuhan yang telah dipilih dapat menggunakan fitur tambah klasifikasi, dimana pada fitur ini tim kepatuhan akan memilih menggunakan *radio button* salah satu jenis klasifikasi dan kategori dari laporan yang masuk setelah dianalisa. Tampilan menu daftar klasifikasi pada Gambar 3.41, dan tambah klasifikasi dapat dilihat pada Gambar 3.42.

Ayo BerAksi Logout

Master Data | Klasifikasi Laporan Pengaduan

Tabel Klasifikasi Laporan Pengaduan

Show 10 entries Search:

| No | Nama Pelapor | No Induk Kependudukan | Tanggal Pengaduan | Aksi  |
|----|--------------|-----------------------|-------------------|---|
| 1  | Admin        | 6413213205000002      | 03/01/2022        |   |
| 2  | Seraya       | 6413213205000002      | 28/06/2022        |   |

Showing 1 to 2 of 2 entries Previous **1** Next

Gambar 3.41 Daftar Klasifikasi Pengaduan

Ayo BerAksi Logout

Master Data | Klasifikasi Laporan Pengaduan | [Tambah Klasifikasi Laporan Pengaduan](#)

Dashboard

Penugasan

Analisis Penyuapan

Klasifikasi Pengaduan

Analisis Gratifikasi

Logbook

### Detail Laporan Pengaduan

Kepada  
Payan

Nama Pelapor  
Seraya

Alamat Lengkap  
Kerayan

No Induk Kependudukan  
6413213205000002

Uraian Laporan  
Laporana yang saya kirimkan kemaren tidak diproses secara baik ....

Saran dan Masukan  
sebaiknya segera dikaji ulang pengambilan keputusan .....

Tanggal Pengaduan  
28/06/2022

### Tambah Klasifikasi Laporan Pengaduan

Klasifikasi  
 A  B

Kategori  
 1  2  3  4  5  6

Keterangan Kategori:  
Lingkai yang dipilih, ditulis dalam kolom yang tersedia

1. kejadian force majeure
2. Pelanggaran kode etik/kinerja pelaku/pelaksana
3. Pelanggaran/penyimpangan mekanisme dan prosedur
4. Penyimpangan/penyelewengan/penyalahgunaan dana
5. Intervensi mengakibatkan kerugian masyarakat
6. Pelanggaran hukum terhadap kebijakan/ketetapan

[Kembali](#) [Simpan](#)

Gambar 3.42 Form Klasifikasi Pengaduan

#### h. Mengembangkan Fitur Analisa Laporan Gratifikasi

Setelah mengembangkan fitur klasifikasi laporan pengaduan, pengembangan dilanjutkan untuk membuat fitur analisa laporan gratifikasi. Pengembangan dimulai dengan membuat *model* dari fitur yang akan menjadi komponen yang berhubungan dan mengelola secara langsung *database*, seperti Gambar 3.43.

```

app > Models > AnalisisLaporanGratifikasi.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class AnalisisLaporanGratifikasi extends Model
9  {
10 {
11     use HasFactory;
12     protected $table = 'analisis_laporan_gratifikasis';
13     protected $fillable = [
14         'id_gratifikasi',
15         'jenis_gratifikasi',
16         'nilai_gratifikasi',
17         'frekuensi_pelapor',
18         'tujuan',
19         'status_batas',
20         'status_frekuensi',
21         'rekomendasi_tindak_lanjut',
22     ];
23
24     // Relationship
25     public function laporanGratifikasi()
26     {
27         return $this->belongsTo('App\Models\LaporanGratifikasi', 'id_gratifikasi', 'id');
28     }
29 }

```

Gambar 3.43 *Models* Analisa laporan Gratifikasi

Kemudian dilanjutkan dengan membuat *migrations* seperti Gambar 3.44 untuk membuat skema basis data, dan membuat *controllers* untuk menghubungkan *models* dan *view* seperti pada Gambar 3.45 serta *controllers actions* yang mengatur manipulasi data seperti penyimpanan, perubahan, dan penghapusan data seperti Gambar 3.46.

```

database > migrations > 2021_12_24_080723_create_analisis_laporan_gratifikasis_table.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  class CreateAnalisisLaporanGratifikasisTable extends Migration
8  {
9      /**
10     * Run the migrations.
11     *
12     * @return void
13     */
14     public function up()
15     {
16         Schema::create('analisis_laporan_gratifikasis', function (Blueprint $table) {
17             $table->id();
18             $table->unsignedBigInteger('id_gratifikasi')->nullable();
19             $table->string('jenis_gratifikasi');
20             $table->string('nilai_gratifikasi');
21             $table->string('frekuensi_pelapor');
22             $table->text('tujuan');
23             $table->string('status_batas');
24             $table->string('status_frekuensi');
25             $table->string('rekomendasi_tindak_lanjut');
26             $table->timestamps();
27         });
28     }
29

```

Gambar 3.44 *Migrations* Analisa laporan Gratifikasi

```

app > Http > Controllers > Web > AnalisisLaporanGratifikasiController.php
1  <?php
2
3  namespace App\Http\Controllers\Web;
4
5  use App\Http\Controllers\Controller;
6  use App\Helpers\LaporanHelper as LprHelper;
7  use Illuminate\Http\Request;
8
9  class AnalisisLaporanGratifikasiController extends Controller
10 {
11     /**
12     * Display a listing of the resource.
13     *
14     * @return \Illuminate\Http\Response
15     */
16     public function index()
17     {
18         return view('analisisLaporan.gratifikasi.view');
19     }
20
21     /**
22     * Show the form for creating a new resource.
23     *
24     * @return \Illuminate\Http\Response
25     */
26     public function create()
27     {
28         return view('analisisLaporan.gratifikasi.form');
29     }
30

```

Gambar 3.45 *Controller View* Analisa laporan Gratifikasi

```

app > Http > Controllers > Actions > AnalisisLaporanGratifikasiController.php
137 public function update(Request $request, $id)
138 {
139     DB::beginTransaction();
140     try {
141         $validator = Validator::make($request->all(), [
142             'jenis_gratifikasi' => 'required',
143             'nilai_gratifikasi' => 'required',
144             'frekuensi_pelapor' => 'required',
145             'tujuan' => 'required',
146             'status_batas' => 'required',
147             'status_frekuensi' => 'required',
148             'rekomendasi_tindak_lanjut' => 'required',
149         ]);
150
151         if ($validator->fails()) {
152             return response()->json([
153                 'success' => false,
154                 'message' => $validator->errors(),
155             ], 401);
156         }
157
158         $analisis = AnalisisLaporanGratifikasi::findOrFail($id);
159         // $analisis->id_gratifikasi = $request->id_gratifikasi;
160         $analisis->jenis_gratifikasi = $request->jenis_gratifikasi;
161         $analisis->nilai_gratifikasi = $request->nilai_gratifikasi;
162         $analisis->frekuensi_pelapor = $request->frekuensi_pelapor;
163         $analisis->tujuan = $request->tujuan;
164         $analisis->status_batas = $request->status_batas;
165         $analisis->status_frekuensi = $request->status_frekuensi;
166         $analisis->rekomendasi_tindak_lanjut = $request->rekomendasi_tindak_lanjut;
167         $analisis->save();
168
169         $request->id_pengguna = $request->id_pengguna;
170         $request->title = "Laporan Gratifikasi";
171         $request->body = "Laporan Gratifikasi Anda sudah di Analisi";
172         $notif = $this->PushNotificationController->sendPushNotification($request);
173
174         DB::commit();

```

Gambar 3.46 *Controller Actions* Analisa laporan Gratifikasi

Fitur ini sama seperti fitur analisa laporan penyusunan, dimana akan digunakan oleh tim kepatuhan yang sudah ditunjuk dan kepala balai yang akan memonitoring. Fitur ini digunakan untuk menganalisa detail dari gratifikasi yang terjadi dan membuat kesimpulan dari hasil analisa. Gambar daftar analisa gratifikasi seperti pada Gambar 3.47.



Master Data / Analisis Laporan Gratifikasi

Tabel Analisis Laporan Gratifikasi

Show 10 entries Search:

| No ↑ | Nama Pelapor ↑ | Tanggal Pelaporan ↑ | Aksi |
|------|----------------|---------------------|------|
| 1    | Fadlan         | 03/01/2022          |      |
| 2    | Madam bak      | 01/01/1970          |      |
| 3    | Walan          | 28/06/2022          |      |


Showing 1 to 3 of 3 entries Previous 1 Next

Gambar 3.47 Daftar Analisa Gratifikasi

Untuk membuat analisa laporan gratifikasi, tim kepatuhan dapat membuat pada *website* sistem, dan seperti fitur analisa laporan penyusunan hanya tim yang sudah ditunjuk yang bisa membuat hasil analisa. Tim kepatuhan perlu mengisi hasil analisis berdasarkan temuan pada investigasi, beberapa *field* yang harus diisi diantaranya jenis hadiah sumbangan, nilai hadiah yang diterima, frekuensi pelapor menerima hadiah pada tahun ini, daan tujuan dari pemberian hadiah. Selain itu, tim kepatuhan juga perlu mengisikan kesimpulan yang menjadi rekomendasi bagi pelapor. Gambar *form* analisa gratifikasi seperti pada Gambar 3.48.

Ayo BerAksi Logout

Master Data / Analisis Laporan Gratifikasi / Tambah Analisis Laporan Gratifikasi



Dashboard

**Penugasan**

- Analisis Penyuapan
- Klasifikasi Pengaduan
- Analisis Gratifikasi**

Logbook

### Detail Laporan Gratifikasi

Nomor Laporan: Belum ada

Nama Pelapor: Seraya

Nama Pengirim: Seraya

Nama Terlapor: Walan

Jabatan: Manager Ketenagakerjaan

Instansi/Perusahaan: Bright To\*\*\*

Kronologis Kejadian: Tersangka memberikan kendaraan mobil sejumlah..... pada saat .....

Tanggal Kejadian: 28/06/2022

Tanggal Pelaporan: 15/08/2020

Alasan: Benar Tersangka memberikan mobil kepada.....

Tindaklanjut: Lanjutkan Proses Penyelidikan .....

---

### Tambah Analisis Laporan Gratifikasi

Jenis Hadiah & Sumbangan: Kendaraan bermotor...

Nilai Hadiah & Sumbangan yang diterima: Rp. 150.000.000

Frekuensi Pelapor menerima Hadiah & Sumbangan pada tahun ini: 3 x dalam setahun terakhir....

Tujuan Pemberian hadiah & sumbangan oleh pemberi: Hadiah atas ulang tahun....

---

### Kesimpulan

Apakah Hadiah & Sumbangan dibawah batas yang ditetapkan  
 YA  TIDAK

Apakah frekuensi pemberian hadiah & sumbangan dibawah batas yang ditetapkan bagi pelapor  
 YA  TIDAK

Rekomendasi tindaklanjut bagi pelapor  
 DITERIMA  DIKEMBALIKAN

[Kembali](#) [Simpan](#)

Gambar 3.48 Form Analisa Gratifikasi

### i. Mengembangkan Fitur Logbook Laporan Penyuaan

Setelah mengembangkan fitur analisa laporan gratifikasi, pengembangan dilanjutkan dengan membuat fitur *logbook* laporan penyuaan. Pengembangan dimulai dengan membuat *model* dari fitur yang akan menjadi komponen yang berhubungan dan mengelola secara langsung *database*, seperti Gambar 3.49.

```

app > Models > LogbookLaporanPenyuapan.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class LogbookLaporanPenyuapan extends Model
9  {
10     use HasFactory;
11
12     protected $table = 'logbook_laporan_penyuapans';
13     protected $fillable = [
14         'id_penyuapan',
15         'uraian_kejadian',
16         'saksi',
17     ];
18
19     // Relationship
20     public function laporanPenyuapan()
21     {
22         return $this->belongsTo('App\Models\LaporanPenyuapan', 'id_penyuapan', 'id');
23     }
24 }

```

Gambar 3.49 Models Logbook Laporan Penyuaan

Kemudian dilanjutkan dengan membuat *migrations* seperti Gambar 3.50 untuk membuat skema basis data, dan membuat *controllers* untuk menghubungkan *models* dan *view* seperti pada Gambar 3.51 serta *controllers actions* yang mengatur manipulasi data seperti penyimpanan, pengubahan, dan penghapusan data seperti Gambar 3.52.

```

database > migrations > 2022_01_03_035612_create_logbook_laporan_penyuapans_table.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  class CreateLogbookLaporanPenyuapansTable extends Migration
8  {
9      /**
10     * Run the migrations.
11     *
12     * @return void
13     */
14     public function up()
15     {
16         Schema::create('logbook_laporan_penyuapans', function (Blueprint $table) {
17             $table->id();
18             $table->unsignedBigInteger('id_penyuapan')->nullable();
19             $table->text('uraian_kejadian')->nullable();
20             $table->text('saksi')->nullable();
21             $table->timestamps();
22         });
23     }
24

```

Gambar 3.50 *Migrations Logbook Laporan Penyuaan*

```

app > Http > Controllers > Web > LogbookLaporanPenyuapanController.php
1  <?php
2
3  namespace App\Http\Controllers\Web;
4
5  use App\Http\Controllers\Controller;
6  use App\Helpers\LaporanHelper as LprHelper;
7  use Illuminate\Http\Request;
8
9  class LogbookLaporanPenyuapanController extends Controller
10 {
11     /**
12     * Display a listing of the resource.
13     *
14     * @return \Illuminate\Http\Response
15     */
16     public function index()
17     {
18         return view('logbook.penyuapan.view');
19     }
20
21     /**
22     * Show the form for creating a new resource.
23     *
24     * @return \Illuminate\Http\Response
25     */
26     public function create()
27     {
28         return view('logbook.penyuapan.form');
29     }
30

```

Gambar 3.51 *Controller View Logbook Laporan Penyuaan*

```

app > Http > Controllers > Actions > LogbookLaporanPenyuapanController.php
175
176 public function getList()
177 {
178     $query = LogbookLaporanPenyuapan::orderBy('id', 'ASC')->get();
179     $data = new LogbookLaporanPenyuapanCollection($query);
180     $total = $query->count();
181     return response()->json([
182         'data' => $data,
183         'total' => $total,
184         'message' => 'Data List Logbook Penyuapan'
185     ]);
186 }
187
188 public function getListLogbookTugas()
189 {
190     $laporan_id = Auth::user()->timPemeriksaPenyuapan->pluck('id_laporan_penyuapan')->toArray();
191     $query = LogbookLaporanPenyuapan::where('id_penyuapan', $laporan_id);
192     $data = new LogbookLaporanPenyuapanCollection($query);
193     $total = $query->count();
194     return response()->json([
195         'data' => $data,
196         'total' => $total,
197         'message' => 'Data List Laporan'
198     ]);
199 }

```

Gambar 3.52 Controller Actions Logbook Laporan Penyuapan

Fitur ini berfungsi untuk membuat catatan hasil analisa laporan penyuapan dari tiga orang tim kepatuhan yang telah ditunjuk. Tim kepatuhan wajib mengisikan *field* uraian kejadian, dan daftar saksi serta kepala balai dapat memonitoring logbook laporan penyuapan. Tampilan dari fitur *logbook* dapat dilihat pada Gambar 3.53.

The screenshot shows a web application interface for reporting a complaint. The top navigation bar is red and contains the text 'Ayo BerAksi' on the left and 'Logout (-)' on the right. Below the navigation bar is a sidebar with a logo and a list of menu items: 'Dashboard', 'Penugasan', 'Logbook' (highlighted in red), 'Logbook Penyuapan', and 'Logbook Gratifikasi'. The main content area is titled 'Master Data' and 'Tambah Analisis Laporan Penyuapan'. It contains a form with the following fields:

- Nama Pelapor: Seraya
- Nama Terlapor: Karaa
- Tanggal Kejadian: 2019-09-13
- Tempat Kejadian: Kantor JKS
- Uraian Kejadian: Terlapor memberikan uang suap melalui .....
- Saksi: Jaja, Seraya, Handi

At the bottom right of the form are two buttons: 'Kembali' (blue) and 'Simpan' (green).

Gambar 3.53 Logbook Pengaduan

#### j. Mengembangkan Fitur Logbook Laporan Gratifikasi

Selanjutnya pengembangan fitur *logbook* laporan gratifikasi. Sama seperti *logbook* laporan penyusunan, fungsi fitur *logbook* laporan gratifikasi yaitu membuat catatan dari hasil analisa laporan gratifikasi. Pengembangan dimulai dengan membuat *model* dari fitur yang akan menjadi komponen yang berhubungan dan mengelola secara langsung *database*, seperti Gambar 3.54.

```

app > Models > LogbookLaporanGratifikasi.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class LogbookLaporanGratifikasi extends Model
9  {
10     use HasFactory;
11
12     protected $table = 'logbook_laporan_gratifikasis';
13     protected $fillable = [
14         'id_gratifikasi',
15         'hasil_analisis',
16         'keterangan',
17     ];

```

Gambar 3.54 Models Logbook Laporan Gratifikasi

Kemudian dilanjutkan dengan membuat *migrations* seperti Gambar 3.55 untuk membuat skema basis data, dan membuat *controllers* untuk menghubungkan *model* dan *view* seperti pada Gambar 3.56 serta *controllers actions* yang mengatur manipulasi data seperti penyimpanan, perubahan, dan penghapusan data seperti Gambar 3.57.

```

database > migrations > 2021_12_29_075651_create_logbook_laporan_gratifikasis_table.php
1 | <?php
2
3 | use Illuminate\Database\Migrations\Migration;
4 | use Illuminate\Database\Schema\Blueprint;
5 | use Illuminate\Support\Facades\Schema;
6
7 | class CreateLogbookLaporanGratifikasisTable extends Migration
8 | {
9 |     /**
10 |      * Run the migrations.
11 |      *
12 |      * @return void
13 |      */
14 |     public function up()
15 |     {
16 |         Schema::create('logbook_laporan_gratifikasis', function (Blueprint $table) {
17 |             $table->id();
18 |             $table->unsignedBigInteger('id_gratifikasi')->nullable();
19 |             $table->unsignedBigInteger('id_analisis_gratifikasi')->nullable();
20 |             $table->string('hasil_analisis')->nullable();
21 |             $table->text('keterangan')->nullable();
22 |             $table->timestamps();
23 |         });
24 |     }

```

Gambar 3.55 Migrations Logbook Laporan Gratifikasi

```

app > Http > Controllers > Web > LogbookLaporanGratifikasiController.php
8
9 | class LogbookLaporanGratifikasiController extends Controller
10 | {
11 |     /**
12 |      * Display a listing of the resource.
13 |      *
14 |      * @return \Illuminate\Http\Response
15 |      */
16 |     public function index()
17 |     {
18 |         return view('logbook.gratifikasi.view');
19 |     }
20
21 |     /**
22 |      * Show the form for creating a new resource.
23 |      *
24 |      * @return \Illuminate\Http\Response
25 |      */
26 |     public function create()
27 |     {
28 |         return view('logbook.gratifikasi.form');
29 |     }
30
31 |     /**
32 |      * Show the form for editing the specified resource.
33 |      *
34 |      * @param int $id
35 |      * @return \Illuminate\Http\Response
36 |      */
37 |     public function edit($id)
38 |     {
39 |         $data = array(
40 |             'showGratifikasi' => LprHelper::showGratifikasi($id),
41 |             'detailAnalisis' => LprHelper::detailAnalisisGratifikasi($id),
42 |             'id' => $id,
43 |             'detailLogbook' => LprHelper::detaillogbookGratifikasi($id),
44 |         );
45 |         return view('logbook.gratifikasi.form', $data);

```

Gambar 3.56 Controllers View Logbook Laporan Gratifikasi

```

app > Http > Controllers > Actions > LogbookLaporanGratifikasiController.php
110
111 public function update(Request $request, $id)
112 {
113     DB::beginTransaction();
114     try {
115         $validator = Validator::make($request->all(), [
116             'hasil_analisis' => 'required',
117             'keterangan' => 'required',
118         ]);
119
120         if ($validator->fails()) {
121             return response()->json([
122                 'success' => false,
123                 'message' => $validator->errors(),
124             ], 401);
125         }
126
127         $logbook = LogbookLaporanGratifikasi::findOrFail($id);
128         $logbook->id_gratifikasi = $request->id_gratifikasi;
129         $logbook->hasil_analisis = $request->hasil_analisis;
130         $logbook->keterangan = $request->keterangan;
131         $logbook->save();
132
133         DB::commit();
134         $message = 'Data logbook Berhasil disimpan';
135         return response()->json([
136             'data' => $logbook,
137             'message' => $message,
138         ]);
139     } catch (\Throwable $th) {
140         DB::rollback();
141         return response()->json([
142             'succses' => false,
143             'message' => $th->getMessage(),
144         ], 500);
145     }
146 }

```

Gambar 3.57 *Controlllers Actions Logbook Laporan Gratifikasi*

Kepala balai dapat memonitor pembuatan *logbook* laporan gratifikasi, sedangkan tim kepatuhan dapat membuat *logbook* baru dengan mengisi *field* hasil analisis apakah gratifikasi harus diterima atau ditolak pelapor, dan keterangan dari tim kepatuhan pada *form logbook* laporan gratifikasi. Gambar 3.58 merupakan tampilan *logbook* gratifikasi.



Gambar 3.58 *Logbook Gratifikasi*

#### k. Mengembangkan REST API pada Sistem

*Endpoint* dari REST API yang dikembangkan pada sistem ini akan digunakan pada aplikasi *android* yang dikembangkan oleh rekan *android developer*. Adapun REST API yang dibuat pada sistem ini sebagai berikut:

##### 1. *Request* dan *Response Register Akun*

API ini digunakan untuk membuat akun baru, masyarakat dapat membuat akun melalui aplikasi *android*. Adapun *request* dan *response* dari API ini dapat dilihat pada Tabel 3.8.

Tabel 3.8 Rancangan *Request* dan *Response Register Akun*

|                          |   |
|--------------------------|---|
| <b><i>Method</i></b>     | POST  |
| <b><i>URI</i></b>        | http://localhost:8000/api/register  |
| <b><i>Header</i></b>     | Accept: application/json  |
| <b><i>Body</i></b>       | name, email, password, no_telp, nip, dan role_id  |
| <b><i>Keterangan</i></b> | Terdapat beberapa kolom body yang perlu diisi sebelum melakukan request, yaitu name, email, password, no_telp, nip, role_id. Untuk nip jika pengguna masyarakat maka dapat dikosongkan. |

|                        |  |
|------------------------|--|
| <b>Contoh Response</b> | <pre> 1  { 2    "user": { 3      "name": "TES", 4      "email": "tes@gmail.com", 5      "no_telp": "081234567890", 6      "nip": "1", 7      "role_id": "7", 8      "updated_at": "2022-06-26T05:18:18.000000Z", 9      "created_at": "2022-06-26T05:18:18.000000Z", 10     "id": 48 11   }, 12   "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9. eyJhdWQiOiIzIiwianRpIjoizjc3ZjRmZmZjNiZjEyNGI4OTYz CjpxYXQiOiJlZmYyMjA2OTgsIm5iZiI6MTY1NjIyMDY5ODcwiXhwI pQ8w8pFVjG0dFG3qIs-Gcd1JRNijjwTYF9kwOCYNGB7yqXXHwW2W 13  }</pre> |
|------------------------|--|

## 2. Request dan Response Login Akun

Api ini digunakan untuk *login* ke dalam sistem dengan akun yang sudah didaftarkan sebelumnya. Adapun *request* dan *response* dari API ini dapat dilihat pada Tabel 3.9.

Tabel 3.9 Rancangan *Request* dan *Response Login Akun*

|                        |   |
|------------------------|---|
| <b>Method</b>          | POST  |
| <b>URI</b>             | http://localhost:8000/api/login   |
| <b>Header</b>          | Accept: application/json  |
| <b>Body</b>            | email, dan password   |
| <b>Keterangan</b>      | Terdapat beberapa kolom body yang perlu diisi sebelum melakukan request, yaitu email, dan password.   |
| <b>Contoh Response</b> | <pre> 1  { 2    "user": { 3      "id": 48, 4      "name": "TEST", 5      "email": "tes@gmail.com", 6      "no_telp": "081234567890", 7      "nip": "1", 8      "email_verified_at": null, 9      "role_id": 7, 10     "device_token": null, 11     "created_at": "2022-06-26T05:18:18.000000Z", 12     "updated_at": "2022-06-26T05:18:18.000000Z" 13   }, 14   "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9. eyJhdWQiOiIzIiwianRpIjoimDE3MTM2ZmE3NjJmYmMxMTRiMTQz CjpxYXQiOiJlZmYyMjA2OTgsIm5iZiI6MTY1NjIyMDY5ODcwiXhwI YNFfRFJFwi2r9uKaQKfZhhjpcXUHxZppsRiwQ0ZD05IxxIolnej0t 15  }</pre> |

### 3. *Request* dan *Response* Membuat Laporan Penyuapan

API ini digunakan untuk membuat laporan penyuapan baru, dimana akan ada beberapa *field* yang perlu diisi pengguna. Adapun *request* dan *response* dari API ini dapat dilihat pada Tabel 3.10.

Tabel 3.10 Rancangan *Request* dan *Response* Membuat Laporan Penyuapan

|                        |  |
|------------------------|--|
| <b>Method</b>          | POST   |
| <b>URI</b>             | http://localhost:8000/api/actions/laporan-penyuapan/add  |
| <b>Header</b>          | Accept: application/json   |
| <b>Body</b>            | nama_terlapor, jabatan, instansi, kasus_suap, nilai_suap, lokasi, tanggal_kejadian, dan kronologis_kejadian.   |
| <b>Authorization</b>   | Bearer Token: mengisi token yang didapat saat login atau register.   |
| <b>Keterangan</b>      | Terdapat beberapa kolom body yang perlu diisi sebelum melakukan request, yaitu nama_terlapor, jabatan, instansi, kasus_suap, nilai_suap, lokasi, tanggal_kejadian, dan kronologis_kejadian.  |
| <b>Contoh Response</b> | <pre> 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23       "data": {         "nama_terlapor": "Test Penyuapan",         "jabatan": "Sekertaris",         "instansi": "Jak*** Sport",         "id_pelapor": 51,         "nama_pelapor": "TEST",         "jabatan_pelapor": null,         "instansi_pelapor": null,         "kasus_suap": "suapan uang",         "nilai_suap": "-",         "lokasi": "Kantor JKS",         "tanggal_kejadian": "2019-09-13",         "tanggal_pelaporan": "2022-06-26",         "kronologis_kejadian": "Pelaku memberikan sejumlah uang",         "status": null,         "deskripsi_status": null,         "updated_at": "2022-06-26T06:06:00.000000Z",         "created_at": "2022-06-26T06:06:00.000000Z",         "id": 20       },       "message": "Data Berhasil disimpan" </pre> |

### 4. *Request* dan *Response* Membuat Laporan Pengaduan

API ini digunakan untuk membuat laporan pengaduan baru. Adapun *request* dan *response* dari API ini dapat dilihat pada Tabel 3.11.

Tabel 3.11 Rancangan *Request* dan *Response* Membuat Laporan Pengaduan

|                        |   |
|------------------------|---|
| <b>Method</b>          | POST  |
| <b>URI</b>             | http://localhost:8000/api/actions/laporan-pengaduan/add   |
| <b>Header</b>          | Accept: application/json  |
| <b>Body</b>            | nama_ketua, alamat, nik, uraian_laporan, dan saran_masukan.   |
| <b>Authorization</b>   | Bearer Token: mengisi token yang didapat saat login atau register.  |
| <b>Keterangan</b>      | Terdapat beberapa kolom body yang perlu diisi sebelum melakukan request, yaitu nama_ketua, alamat, nik, uraian_laporan, dan saran_masukan.  |
| <b>Contoh Response</b> | <pre> 1  { 2    "data": { 3      "nama_ketua": "Test Pengaduan", 4      "id_pelapor": 51, 5      "nama_pelapor": "TEST", 6      "alamat": "Yogyakarta", 7      "nik": "6413213205000002", 8      "uraian_laporan": "Laporana yang saya kirimkan kemaren tidak diproses", 9      "saran_masukan": "sebaiknya segera dikaji ulang pengambilan keputusan .", 10     "tanggal_pengaduan": "2022-06-26", 11     "status": null, 12     "deskripsi_status": null, 13     "updated_at": "2022-06-26T06:13:55.000000Z", 14     "created_at": "2022-06-26T06:13:55.000000Z", 15     "id": 6 16   }, 17   "message": "Data Berhasil disimpan" 18 } </pre> |

#### 5. *Request* dan *Response* Membuat Laporan Gratifikasi

API ini digunakan untuk membuat laporan gratifikasi baru. Adapun *request* dan *response* dari API ini dapat dilihat pada Tabel 3.12.

Tabel 3.12 Rancangan *Request* dan *Response* Membuat Laporan Gratifikasi

|                      |   |
|----------------------|---|
| <b>Method</b>        | POST  |
| <b>URI</b>           | http://localhost:8000/api/actions/laporan-gratifikasi/add   |
| <b>Header</b>        | Accept: application/json  |
| <b>Body</b>          | nama_terlapor, jabatan, instansi, tanggal_kejadian, dan kronologis_kejadian.  |
| <b>Authorization</b> | Bearer Token: mengisi token yang didapat saat login atau register.  |
| <b>Keterangan</b>    | Terdapat beberapa kolom body yang perlu diisi sebelum melakukan request, yaitu nama_terlapor, jabatan, instansi, tanggal_kejadian, dan kronologis_kejadian. |

|                        |   |
|------------------------|---|
| <b>Contoh Response</b> | <pre> 1  { 2    "data": { 3      "no_laporan": null, 4      "pengirim": null, 5      "id_pelapor": 51, 6      "nama_pelapor": "TEST", 7      "nama_terlapor": "Test gratifikasi", 8      "jabatan": "Manager Ketenagakerjaan", 9      "instansi": "Bright To***", 10     "tanggal_kejadian": "2020-08-15", 11     "tanggal_pelaporan": "2022-06-26", 12     "kronologis_kejadian": "Tersangka memberikan kendaraan mobil 13     "status": null, 14     "deskripsi_status": null, 15     "tindaklanjut": null, 16     "updated_at": "2022-06-26T06:22:20.000000Z", 17     "created_at": "2022-06-26T06:22:20.000000Z", 18     "id": 7 19   }, 20   "message": "Data Berhasil disimpan" 21 } </pre> |
|------------------------|---|

#### 6. Request dan Response Melihat Seluruh Laporan

API ini digunakan untuk menampilkan seluruh laporan yang pengguna buat. Adapun *request* dan *response* dari API ini dapat dilihat pada Tabel 3.13.

Tabel 3.13 Rancangan *Request* dan *Response* Melihat Seluruh Laporan

|                        |   |
|------------------------|---|
| <b>Method</b>          | GET   |
| <b>URI</b>             | http://localhost:8000/api/actions/laporan/get-list/all  |
| <b>Header</b>          | Accept: application/json  |
| <b>Body</b>            | -   |
| <b>Authorization</b>   | Bearer Token: mengisi token yang didapat saat login atau register.  |
| <b>Keterangan</b>      | Pada <i>request</i> ini hanya perlu memasukan token pengguna pada authorization.  |
| <b>Contoh Response</b> | <pre> 1  { 2    "total query": 3, 3    "message": "Data List Seluruh Laporan", 4    "data": [ 5      { 6        "id": 7, 7        "nama_ketua": "Payan", 8        "id_pelapor": { 9          "id": 64, 10         "name": "Seraya", 11         "email": "seraya@gmail.com", 12         "no_telp": "081234567890", 13         "nip": null, 14         "email_verified_at": null, 15         "role_id": 7, 16         "device_token": null, 17         "created_at": "2022-06-28T14:39:23.000000Z", 18         "updated_at": "2022-06-28T14:39:23.000000Z" 19       }, 20       "nama_pelapor": "Seraya", 21       "alamat": "Kerayan", 22       "nik": "6413213205000002", 23       "uraian_laporan": "Laporan yang saya kirimkan kemaren t: 24       "saran_masukan": "sebaiknya segera dikaji ulang pengambil 25       "tanggal_pengaduan": "2022-06-28", </pre> |

### 3.1.5 Pengujian Proyek

Pengujian aplikasi secara keseluruhan dilakukan oleh tim *Quality Assurance* (QA), dimana tim QA akan mengecek keseluruhan fitur dengan berbagai skenario. Setiap *bug* yang ditemukan oleh tim QA akan dituliskan pada dokumen bug. QA akan mendeskripsikan *bug* secara detail dengan skenario tertentu dan bukti tangkapan layar yang bertujuan memudahkan tim *developer*, serta jika diperlukan tim QA dan *developer* dapat mendiskusikan *bug* tersebut.

Setelah menuliskan dokumen *bug*, QA akan memberikan dokumen tersebut kepada *developer* sebagai sebuah *task* pekerjaan. *Developer* kemudian memperbaiki *bug* tersebut dan melakukan *commit* ke repositori git. Selanjutnya tim *developer* menghubungi tim QA untuk mengkonfirmasi setiap bug yang sudah diperbaiki.

Selain QA, tim *backend developer* juga menguji beberapa fitur yang ada pada sistem. Salah satunya yaitu pengujian fitur REST API, digunakan untuk memastikan seluruh *endpoint* dari REST API berjalan dengan baik. Pengujian dilakukan dengan memanfaatkan sebuah aplikasi yang akan bekerja sebagai sebuah *client* yaitu Postman. Selain itu, Postman sendiri juga digunakan sebagai sebuah dokumentasi untuk membantu *mobile developer* untuk mengembangkan aplikasi *android*. Beberapa pengujian REST API yang tim *backend developer* lakukan, diantaranya:

#### a. Pengujian *Response* dan *Request Register*

Pengujian ini digunakan untuk mencoba membuat akun baru menggunakan REST API, dimana pada Postman perlu memasukan *endpoint* yang diperlukan, memilih *method* POST, dan memasukan beberapa *field* pada *body* seperti pada Gambar 3.59.

The screenshot shows the Postman interface for a POST request. The URL is `http://localhost:8000/api/register`. The request body is set to form-data and contains the following fields:

| KEY  | VALUE            | DESCRIPTION |
|--|------------------|-------------|
| <input checked="" type="checkbox"/> name         | Seraya           |             |
| <input checked="" type="checkbox"/> email        | seraya@gmail.com |             |
| <input checked="" type="checkbox"/> password     | password         |             |
| <input checked="" type="checkbox"/> no_telp      | 081234567890     |             |
| <input checked="" type="checkbox"/> nip          |                  |             |
| <input checked="" type="checkbox"/> role_id      | 7                |             |
| <input checked="" type="checkbox"/> device_token |                  |             |

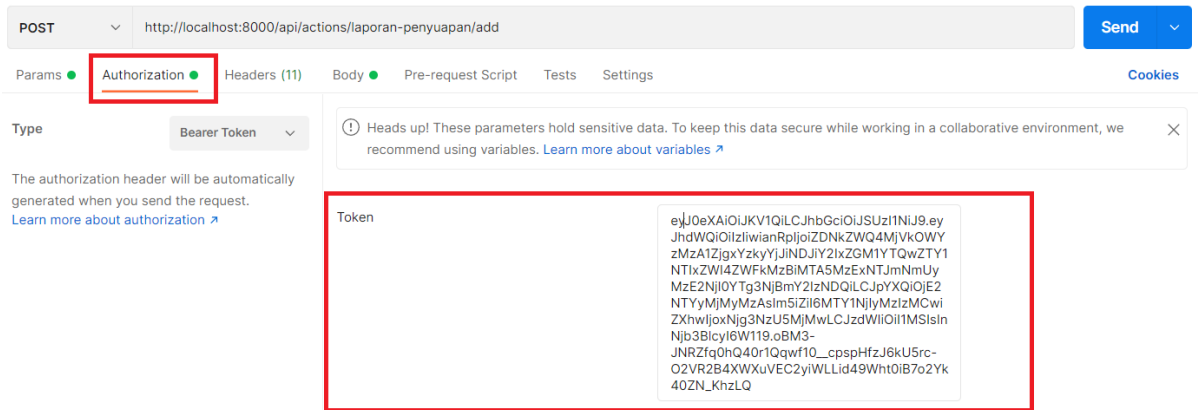
Gambar 3.59 Pengujian *Register*

Setelah *request* berhasil dijalankan, akan ada *output* atau keluaran berupa data yang berformat JSON seperti pada Gambar 3.60. Selain itu, akan ada *output* berupa *status code* 200 yang berarti *request* berhasil dijalankan, dengan *time* atau waktu eksekusi 400 ms, dan



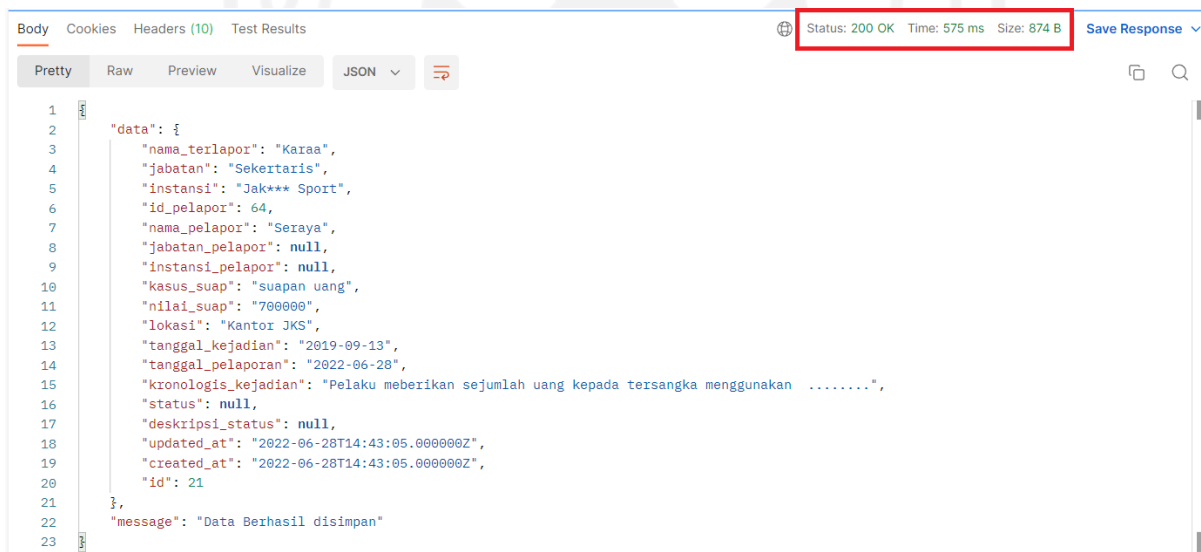






Gambar 3.64 Mengisikan Token untuk Laporan Penyuaan

Hasil dari *request* yang berhasil dijalankan, akan terdapat *output* atau keluaran berupa data yang berformat JSON seperti pada Gambar 3.65. Selain itu, akan ada *output* berupa *status code* 200 yang berarti *request* berhasil dijalankan, dengan *time* atau waktu eksekusi 575 ms, dan 874 byte.



Gambar 3.65 Response Endpoint Laporan Percobaan

#### d. Pengujian *Response* dan *Request* Membuat Laporan Pengaduan

Pengujian ini digunakan untuk memastikan fitur buat laporan pengaduan dapat berjalan dengan baik pada REST API. Dimulai dari memasukkan *endpoint API*, memilih *method* POST, dan *field* yang diperlukan, serta pada bagian *authorization* diperlukan *token* yang didapat saat *login* atau *register* sebelumnya. Gambar 3.66 merupakan tampilan dari *field* yang perlu dimasukkan pada *body*, dan Gambar 3.67 merupakan *token* yang perlu dimasukkan pada *authorization*.



```

1  "data": {
2    "nama_ketua": "Payan",
3    "id_pelapor": 64,
4    "nama_pelapor": "Seraya",
5    "alamat": "Kerayan",
6    "nik": "6413213205000002",
7    "uraian_laporan": "Laporana yang saya kirimkan kemaren tidak diproses secara baik .....",
8    "saran_masukan": "sebaiknya segera dikaji ulang pengambilan keputusan .....",
9    "tanggal_pengaduan": "2022-06-28",
10   "status": null,
11   "deskripsi_status": null,
12   "updated_at": "2022-06-28T14:45:09.000000Z",
13   "created_at": "2022-06-28T14:45:09.000000Z",
14   "id": 7
15 },
16 },
17 "message": "Data Berhasil disimpan"
18

```

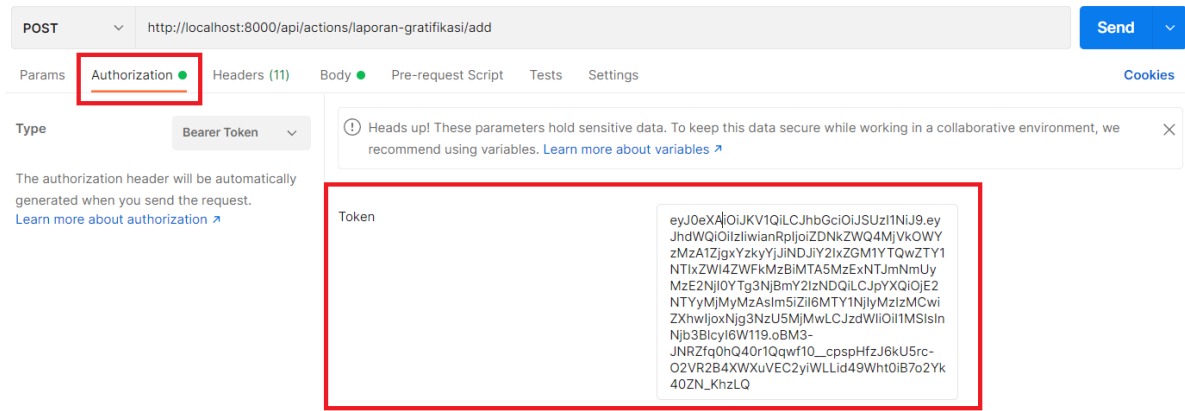
Gambar 3.68 *Response Endpoint* Laporan Pengaduan

#### e. Pengujian *Response* dan *Request* Membuat Laporan Gratifikasi

Pengujian ini digunakan untuk memastikan fitur buat laporan gratifikasi dapat berjalan dengan baik pada REST API. Dimulai dari memasukkan *endpoint API*, memilih *method* POST, dan *field* yang diperlukan, serta pada bagian *authorization* diperlukan *token* yang didapat saat *login* atau *register* sebelumnya. Gambar 3.69 merupakan tampilan dari *field* yang perlu dimasukkan pada *body*, dan Gambar 3.70 merupakan *token* yang perlu dimasukkan pada *authorization*.

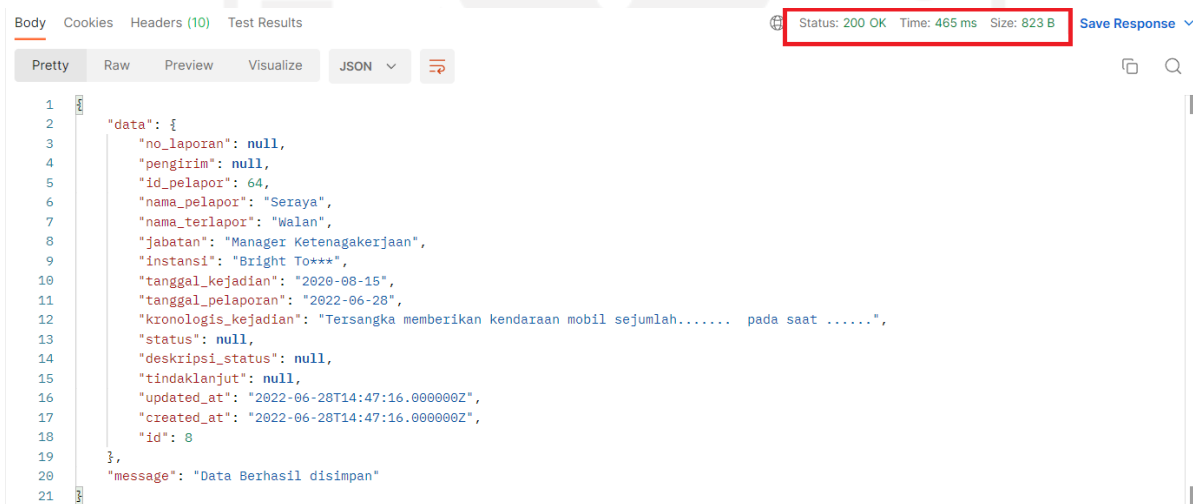
| Field   | Value   | Description                |
|---|---|----------------------------|
| <input checked="" type="checkbox"/> nama_terlapor       | Walan   | Nama orang yang dilaporkan |
| <input checked="" type="checkbox"/> jabatan             | Manager Ketenagakerjaan                                 |                            |
| <input checked="" type="checkbox"/> instansi            | Bright To***  |                            |
| <input checked="" type="checkbox"/> tanggal_kejadian    | 2020-08-15  |                            |
| <input checked="" type="checkbox"/> kronologis_kejadian | Tersangka memberikan kendaraan mobil sejumlah..... p... |                            |

Gambar 3.69 Percobaan Tambah Laporan Gratifikasi



Gambar 3.70 Memasukan Token Laporan Gratifikasi

Hasil dari *request* yang berhasil dijalankan, akan terdapat *output* atau keluaran berupa data yang berformat JSON seperti pada Gambar 3.71. Selain itu, akan ada *output* berupa *status code* 200 yang berarti *request* berhasil dijalankan, dengan *time* atau waktu eksekusi 465 ms, dan 823 byte.



Gambar 3.71 Response Endpoint Laporan Gratifikasi

#### f. Pengujian *Response* dan *Request* Melihat Seluruh Laporan

Pengujian ini akan digunakan untuk memastikan fitur lihat seluruh laporan yang telah dibuat pengguna berhasil berjalan dengan baik. Pada pengujian ini, hanya perlu memasukan *endpoint*, memilih *method* GET, dan *token* pada bagian *authorization*. Gambar 3.72 merupakan *token* yang perlu dimasukan pada *authorization*.



Tabel 3.14 Hasil Pengujian *Method POST* dengan 5 Sampel

| No        | <i>Request Method</i> | Keterangan   | Waktu (ms) | <i>Bandwidth</i> (byte) |
|-----------|-----------------------|--------------|------------|-------------------------|
| 1         | POST                  | Membuat akun | 412        | 925                     |
| 2         |                       |              | 378        | 925                     |
| 3         |                       |              | 501        | 925                     |
| 4         |                       |              | 501        | 925                     |
| 5         |                       |              | 454        | 925                     |
| Rata-rata |                       |              | 449,2      | 925                     |

Tabel 3.15 Hasil Pengujian *Method GET* dengan 5 Sampel

| No        | <i>Request Method</i> | Keterangan              | Waktu (ms) | <i>Bandwidth</i> (byte) |
|-----------|-----------------------|-------------------------|------------|-------------------------|
| 1         | GET                   | Melihat Seluruh Laporan | 275        | 2.41                    |
| 2         |                       |                         | 248        | 2.41                    |
| 3         |                       |                         | 228        | 2.41                    |
| 4         |                       |                         | 260        | 2.41                    |
| 5         |                       |                         | 266        | 2.41                    |
| Rata-rata |                       |                         | 255,4      | 2.41                    |

Berdasarkan hasil pengujian yang telah dilakukan *server* REST API dapat menerima *request* dengan baik dari *client*. Selain itu, hasil pengujian menggunakan *method* POST, dan GET yang digunakan pada *client* (*android*) menghasilkan *size bandwidth* yang tergolong kecil, dan cepat dalam menangani *request* pertukaran data, sehingga dapat dikatakan efisien karena tidak boros dalam penggunaan *bandwidth* dan hemat waktu.

### 3.1.6 Pemantauan dan Pengendalian Proyek

Tahapan pemantauan dan pengendalian merupakan tahap yang dilakukan untuk memonitoring proyek yang sedang berjalan, sehingga pengembangan dapat berjalan dengan baik dan jika terdapat kendala akan lebih mudah untuk diselesaikan. Dalam pemantauan dan pengendalian proyek ini terdapat beberapa *tools* yang digunakan, diantaranya:

- a. Telegram, merupakan aplikasi yang digunakan untuk komunikasi dan bertukar pesan. GeekGarden memanfaatkan aplikasi ini sebagai wadah komunikasi tim pengembang. Selain itu, telegram juga digunakan sebagai media untuk melaporkan progres dari sistem yang dikembangkan.
- b. WhatsApp, sama seperti telegram WhatsApp juga digunakan sebagai media komunikasi antar tim. Namun, terkadang WhatsApp lebih digunakan untuk *personal message* dengan rekan tim, dan *project manager*.

- c. GitLab, merupakan sebuah aplikasi repositori, dan kolaborasi. GeekGarden menggunakan GitLab sebagai media kolaborasi tim, *version control* pengembangan proyek, dan pemantauan pengembangan.

Selain itu, karena proses magang yang dijalani berlangsung secara WFH dan WFO bergantian. Terkadang proses pelaporan progres pengembangan dan kendala yang ditemui dapat secara langsung dilaporkan kepada *project manager*.

### 3.1.7 Penutupan Proyek

Penutupan aktivitas proyek pengembangan aplikasi Ayo BerAksi masih belum dilakukan. Karena masa magang yang telah berakhir dan sebelum menyelesaikan beberapa fitur yang tersisa seperti *download* laporan, penulis diminta untuk pindah ke salah satu proyek untuk menyelesaikan *bug*. Saat ini, pengembangan aplikasi Ayo BerAksi dilanjutkan oleh rekan kerja yang lain yang masih bekerja di GeekGarden.

## BAB IV

### REFLEKSI PELAKSANAAN MAGANG

#### 4.1 Relevansi Akademik

Aplikasi Ayo BerAksi merupakan sebuah aplikasi *multiplatform* dimana terdapat aplikasi berbasis *website* dan *android* yang menerapkan arsitektur *Representational State Transfer* (REST) dan *framework* OAuth 2. Pada pelaksanaan pengembangan aplikasi ini tidak terdapat perbedaan antara teori *Representational State Transfer* dan penerapannya dalam pengembangan aplikasi Ayo BerAksi. Aplikasi Ayo BerAksi menerapkan antarmuka yang terpisah antara *server* (*website*) dan juga *client* (*android*). Selain itu, aplikasi ini juga menerapkan komponen antarmuka yang seragam, dimana dengan memanfaatkan protokol HTTP. REST memanipulasi sumber daya berupa *create, retrieve, update dan delete* (CRUD), dengan operasi yang dimiliki HTTP memanfaatkan *method* POST, GET, PUT, dan DELETE. Penerapan REST juga lebih cocok untuk digunakan pada aplikasi berbasis *android*, karena REST lebih ramah infrastruktur nirkabel.

Penerapan OAuth 2.0 pada pengembangan aplikasi Ayo BerAksi juga tidak terdapat perbedaan dengan teori yang dijelaskan, dimana OAuth 2.0 memanfaatkan mekanisme *scope* untuk membatasi akses informasi. OAuth 2.0 dikembangkan pada *framework* Laravel dengan memanfaatkan *library* Laravel *Passport* yang mendukung penuh penggunaan OAuth 2.0. Pada penerapannya OAuth 2.0 akan memanfaatkan *Access Token* yang akan tergenerate saat pengguna melakukan *login* dan *register*. Token ini nantinya akan menjadi sebuah tanda pengenal yang dipergunakan untuk mendapatkan akses ke server.

#### 4.2 Pembelajaran Magang

##### 4.2.1 Manfaat Magang

Selama enam bulan melaksanakan magang di GeekGarden, banyak sekali manfaat yang didapat dari kegiatan magang diantaranya sebagai berikut:

- a. Mempelajari ilmu baru dan mempelajari *tools* baru

Selama melaksanakan magang mendapatkan kesempatan untuk mempelajari berbagai ilmu baru terutama yang berkaitan dengan pengembangan aplikasi seperti apa itu REST API, bagaimana cara membuat REST API, dan bagaimana cara menguji REST API yang telah



dibuat. Selain itu, mendapatkan pelajaran baru mengenai penggunaan *source code* seperti GitLab, bagaimana cara berkolaborasi menggunakan GitLab, dan bagaimana cara melakukan *commit* pada GitLab. Serta mempelajari berbagai macam *tools* dan teknologi baru yang mampu membantu dalam pengembangan aplikasi seperti Postman, dan juga GitLab.

b. Manajemen diri

Terlibat dalam perusahaan pengembangan aplikasi sebagai karyawan magang sangat berbeda dengan dunia perkuliahan. Terutama karena adanya tuntutan dan tanggung jawab yang diberikan oleh perusahaan. Oleh karena itu, perlu menerapkan manajemen diri agar dapat memenuhi berbagai tanggung jawab dan tuntutan tersebut. Salah satu aspek manajemen diri yang diterapkan yaitu manajemen waktu.

Untuk menjalani magang sekaligus kuliah, tentunya perlu untuk mengatur waktu sebaik mungkin dalam mengerjakan tugas kantor maupun kuliah. Selain itu, juga perlu membiasakan diri untuk tidak menunda pekerjaan terutama saat menjalani *work from home*.

c. Bekerja dalam sebuah tim profesional

Dalam mengembangkan aplikasi, akan terdapat sebuah tim yang terdiri dari beberapa individu. Masing-masing individu tersebut memiliki peranan yang penting dalam pengembangan aplikasi. Peranan atau *role* ini membuat pekerjaan menjadi lebih efisien karena tugas dan tanggung jawab akan dibagikan ke masing-masing anggota tim. Selain itu, *role* ini membantu bagaimana menyelesaikan sebuah masalah dengan baik, dan berkonsultasi ketika tidak mengerti mengenai tugas yang diberikan.

#### 4.2.2 Hambatan dan Tantangan Selama Magang

Selama melaksanakan magang terdapat beberapa hambatan yang ditemui, seperti kesulitan membagi waktu karena pelaksanaan magang bersamaan dengan jadwal kuliah, kesulitan untuk berkomunikasi dengan *supervisor* terutama saat mendapatkan jadwal untuk *work from home* karena *supervisor* sering melakukan meeting dengan *client* dan sibuk dengan pekerjaan pengembangan aplikasi lainnya.

Selain itu, juga terdapat kesulitan menggunakan berbagai *tools* dan konsep baru yang belum pernah digunakan sebelumnya. Sehingga perlu mencari tau dan mempelajari terlebih dahulu *tools* dan konsep tersebut, mengenai bagaimana cara menggunakannya, dan bagaimana cara mengimplementasikannya.

Selain hambatan, juga terdapat beberapa tantangan dalam menjalankan magang. Seperti saat diminta untuk masuk ke dalam pengembangan aplikasi *enterprise resource planning* Mie-Nyinyir yang sudah berjalan beberapa bulan, pada pengembangan tersebut terdapat tantangan untuk mempelajari alur pengembangan aplikasi, dan ikut menyesuaikan kode yang dibuat seperti pada pengembangan sebelumnya. Kemudian juga tertantang untuk mencoba berbagai *tools* baru, hingga konsep baru yang digunakan untuk menyelesaikan proyek.



## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Selama menjalani magang di perusahaan GeekGarden dalam waktu enam bulan, dimana dalam kurun waktu tersebut ikut andil dalam mengembangkan dua proyek, yaitu: Modul HRD Mie-Nyinyir, dan Ayo BerAksi. Pengembangan aplikasi Ayo BerAksi menjadi fokus pada laporan ini, Ayo BerAksi merupakan aplikasi yang berfungsi sebagai sebuah media pelaporan dan penanganan kasus penyuapan dan gratifikasi yang terjadi di lingkungan BBKP-Belawan. Dengan Ayo BerAksi masyarakat dan pegawai BBKP-Belawan dapat melaporkan kasus penyuapan dan gratifikasi yang terjadi, tanpa perlu mendatangi kantor BBKP-Belawan, dan tanpa perlu waktu yang lama untuk melaporkan sehingga proses pelaporan dan penanganan lebih efisien.

Pada pengerjaan proyek Ayo BerAksi tugas yang didapatkan telah dikerjakan dengan baik sehingga tujuan dan manfaat dari laporan ini terpenuhi. Tujuan pada laporan ini yaitu mengimplementasikan REST API dalam pengembangan aplikasi Ayo BerAksi sedangkan manfaat dari laporan ini yaitu mempermudah *developer* mengembangkan aplikasi berbasis *multiplatform* dengan memanfaatkan REST API, memberikan kemudahan masyarakat dan pegawai BBKP-Belawan dalam membuat laporan kasus penyuapan dan gratifikasi, serta membantu memberantas tindak pidana korupsi. Selain itu, hasil dari pengembangan aplikasi *website* Ayo BerAksi dapat dilihat pada **Bab 3.1.4** dan **Bab 3.1.5**.

#### **5.2 Saran**

##### **5.2.1 Saran Untuk Pengembangan Selanjutnya**

Adapun beberapa saran yang dapat disampaikan untuk pengembangan selanjutnya, sebagai berikut:

- a. Perlu disediakan *cache* pada sistem agar proses transaksi pertukaran data menjadi lebih cepat, dan penggunaan *cache* berfungsi untuk menghindari akses data yang sama berulang kali.
- b. Perlu pengujian keamanan lebih lanjut, seperti *penetration testing* untuk memastikan keamanan dari sistem yang telah dikembangkan.

- c. Sebaiknya perlu disediakan metode *register* dan *login* akun lain seperti menggunakan Google Authentication pada aplikasi Ayo BerAksi, sehingga memberikan kemudahan untuk pengguna melakukan *register* dan *login* akun.

### 5.2.2 Saran Untuk Perusahaan

Adapun beberapa saran yang dapat disampaikan untuk pengembangan selanjutnya, sebagai berikut:

- a. Sebaiknya terdapat sebuah tim khusus yang menangani pengujian keamanan sistem, sehingga setiap sistem yang dikembangkan teruji keamanannya.
- b. Sebaiknya anggota tim terutama bagian *backend* dan *frontend developer* yang bertugas pada sebuah proyek tidak dipindah ke proyek lain sebelum aplikasi selesai.
- c. Selain itu, sebaiknya tim *developer* perlu dilibatkan pada saat *meeting* dengan *client*, terutama saat menganalisa kebutuhan sistem, sehingga ikut membantu menganalisa kebutuhan sistem dari sisi *developer*.

## DAFTAR PUSTAKA


- Cholik, C. A. (2021). Perkembangan Teknologi Informasi Komunikasi / ICT dalam Berbagai Bidang. *Jurnal Fakultas Teknik*, 2(1), 39–46.
- Danuri, M. (2019). Development and Transformation of Digital Technology. *Infokam*, XV(II), 116–123.
- Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures. In *Journal of Chemical Information and Modeling* (Vol. 53, Issue 9).
- GeekGarden. (n.d.). *Geekgarden | Company Profile*. Retrieved April 5, 2022, from <https://geekgarden.id/id/company-profile>
- Hardt, D. (2012). *The OAuth 2.0 Authorization Framework*. <https://doi.org/10.17487/RFC6749>
- Kurniawan, Y. K., Oslan, Y., & Kristanto, H. (2013). Implementasi Rest - Api Untuk Portal Akademik UKDW Berbasis Android. *Jurnal EKSIS*, 6, 29–40.
- Laravel. (2019). Laravel - The PHP Framework For Web Artisans. In *Laravel* (p. 1). <https://laravel.com/docs/9.x/passport>
- Rahmawati, D. (2012). Analisis Faktor Faktor yang Berpengaruh Terhadap Pemanfaatan Teknologi Informasi. *Jurnal Ekonomi Dan Pendidikan*, 5(1), 107–118. <https://doi.org/10.21831/jep.v5i1.606>
- Rizal, R., & Rahmatulloh, A. (2019). Restful Web Service untuk Integrasi Sistem Akademik dan Perpustakaan Universitas Perjuangan. *Jurnal Ilmiah Informatika*, 7(01), 54. <https://doi.org/10.33884/jif.v7i01.1004>
- Safitri, R.K., dan Putro, H. P. (2021). Implementasi REST API untuk Komunikasi Antara ReactJS dan NodeJS (Studi Kasus : Modul Manajemen User Solusi247). *Automata*, 2(1), 0–4. <https://journal.uui.ac.id/AUTOMATA/article/view/17381>
- W3C. (2001). Web Services Architecture. In *Architecting Web Services* (pp. 25–65). [https://doi.org/10.1007/978-1-4302-1140-2\\_2](https://doi.org/10.1007/978-1-4302-1140-2_2)
- W3C. (2003). SOAP Version 1.2 Part 1: Messaging Framework. In *W3C REC REC-soap12-part1-20030624*, June (Vol. 2011, Issue April 2007). <https://www.w3.org/TR/soap12-part1/>
- Wagh, K., & Thool, R. (2012). A Comparative Study of SOAP Vs REST Web Services

Provisioning Techniques for Mobile Host. *Journal of Information Engineering and Applications*, 2(5), 12–16.

<http://www.iiste.org/Journals/index.php/JIEA/article/view/2063>



## LAMPIRAN

|  |  |                                 |
|--|--|---------------------------------|
|   | <b>UNIT PELAKSANA TEKNIS</b><br><b>BALAI BESAR KARANTINA PERTANIAN BELAWAN</b><br>JL. Sulawesi II Ujung Baru Belawan |                                 |
|  | <b>FORMULIR</b>  | Kode Dokumen F.BBKP.BLW.7.3.1.1 |
|  |  | Revisi ke / Cetakan 01 / 01     |
|  | Tanggal 1 Februari 2019  |                                 |
| <b>FORMULIR LAPORAN BENTURAN<br/>KEPENTINGAN, PENGADUAN,<br/>SARAN DAN MASUKAN</b> | Paraf Manajer Mutu   |                                 |

**FORMULIR LAPORAN BENTURAN KEPENTINGAN, PENGADUAN, SARAN DAN  
MASUKAN  
BALAI BESAR KARANTINA PERTANIAN BELAWAN**

Kepada:

.....

Yang bertandatangan di bawah ini:

Nama :

Alamat lengkap :

No Induk Kependudukan :

Uraian laporan :

Saran dan Masukan :

Belawan, ..... / ..... / 20 .....

Pelapor,

( ..... )

Nama dan Tanda Tangan


Catatan:

Laporan harus disertai dengan bukti-bukti sebagai pendukung laporan

F.BBKP.BLW.7.3.1.1

*Dokumen ini bersifat rahasia, dilarang memperbanyak atau menyalin tanpa seijin  
Balai Besar Karantina Pertanian Belawan*

Lampiran A Formulir Laporan Pengaduan

|   |   |  |
|---|---|--|
|  | <b>UNIT PELAKSANA TEKNIS</b><br><b>BALAI BESAR KARANTINA PERTANIAN BELAWAN</b><br><b>JL. Sulawesi II Ujung Baru Belawan</b> |  |
|   | <b>FORMULIR</b>   | Kode Dokumen: F.BBKP.BLW.7.3.1.4<br>Revisi ke / Cetakan: 01 / 01<br>Tanggal: 1 Februari 2019 |
| <b>KLASIFIKASI LAPORAN</b>  | Paraf Manajer Mutu  |  |

**KLASIFIKASI LAPORAN**

| Uraian Laporan | Klasifikasi* |   | Kategori^ |   |   |   |   |   | Keterangan |
|----------------|--------------|---|-----------|---|---|---|---|---|------------|
|                | A            | B | 1         | 2 | 3 | 4 | 5 | 6 |            |
|                |              |   |           |   |   |   |   |   |            |

Mengetahui,  
Kepala BBKP Belawan

Belawan,.....  
Ketua Tim Fungsi Kepatuhan

(.....)  
Nama dan Tandatangan

(.....)  
Nama dan Tandatangan

**Keterangan :**

\*Lingkari yang dipilih, ditulis dalam kolom yang tersedia

A: Laporan tidak berkadar pengawasan

B: Laporan berkadar pengawasan

^Beri tanda silang yang dipilih ditulis dalam kolom yang tersedia

- 1: kejadian force major
- 2: Pelanggaran kode etik/kinerja pelaku/pelaksana
- 3: Pelanggaran/penyimpangan mekanisme dan prosedur
- 4: Penyimpangan/penyelewengan/penyalahgunaan dana
- 5: Intervensi mengakibatkan kerugian masyarakat
- 6: Pelanggaran hukum terhadap kebijakan/ketetapan

**Keterangan**

Diisi kesimpulan hasil penyelesaian laporan

Jika A langsung diselesaikan BBKP Belawan

Jika B dan kategori 1, 2, 3 diselesaikan BBKP Belawan

Jika B dan kategori 4, 5 disampaikan kepada Inspektorat Investigasi

Jika B dan kategori 6 disampaikan kepada Sekertariat Badan Karantina Pertanian

F.BBKP.BLW.7.3.1.4

*Dokumen ini bersifat rahasia, dilarang memperbanyak atau menyalin tanpa seijin  
Balai Besar Karantina Pertanian Belawan*



## LAPORAN PENYUAPAN

### IDENTITAS PELAPOR

Nama Pelapor :  
Jabatan :  
Asal Perusahaan :

### IDENTITAS TERLAPOR

Nama Terlapor :  
Jabatan :  
Asal Perusahaan :

### DETAIL LAPORAN

Kasus penyipuan yang terjadi :

Nilai Suap yang diterima atau diberi :

Lokasi Kejadian :

Tanggal Kejadian :

Deskripsi singkat kejadian :

Dengan ini saya ..... sebagai pelapor menyatakan bahwa laporan yang saya sampaikan kepada tim kepatuhan adalah benar dan bukan merupakan fitnah, apabila berdasarkan hasil investigasi menyatakan laporan yang sampaikan tidak benar, maka saya akan bertanggungjawab terhadap laporan yang saya sampaikan

.....  
Pelapor

(Nama)

F.BBKP.BLW.8.10.0.1

## PENUNJUKAN TIM INVESTIGASI

Berdasarkan laporan penyuaapan yang disampaikan dengan rincian :

|                          |                           |
|--------------------------|---------------------------|
|                          | <b>IDENTITAS PELAPOR</b>  |
| <b>Nama Pelapor</b> :    |                           |
| <b>Jabatan</b> :         |                           |
| <b>Asal Perusahaan</b> : |                           |
|                          | <b>IDENTITAS TERLAPOR</b> |
| <b>Nama Terlapor</b> :   |                           |
| <b>Jabatan</b> :         |                           |
| <b>Asal Perusahaan</b> : |                           |

Deskripsi singkat kejadian :

Maka dengan ini Kepala Balai dan Tim Kepatuhan menunjuk nama yang tersebut dibawah ini sebagai Tim Investigasi untuk melakukan investigasi terhadap laporan diatas, yaitu :

| Nama | Bidang | Jabatan Dalam Tim |
|------|--------|-------------------|
|      |        | Ketua             |
|      |        | Anggota           |
|      |        | Anggota           |

Belawan,.....

Manajer Puncak

(.....)

F.BBKP.BLW.8.10.0.3

Lampiran D Formulir Penunjukan Tim Investigasi Laporan Penyuaapan

## LAPORAN HASIL INVESTIGASI

**IDENTITAS PELAPOR**

Nama Pelapor :  
Jabatan :  
Asal Perusahaan :

**IDENTITAS TERLAPOR**

Nama Terlapor :  
Jabatan :  
Asal Perusahaan :

Hasil Investigasi :

Kesimpulan Tim Investigasi :

Belawan.....  
Ketua Tim Investigasi

(.....)

F.BBKP.BLW.8.10.0.4


**LOGBOOK LAPORAN PENYUAPAN**

| No | Nama Pelapor | Nama Terlapor | Tgl/Tempat Kejadian | Uraian Kejadian | Saksi (bila ada) |
|----|--------------|---------------|---------------------|-----------------|------------------|
|    |              |               |                     |                 |                  |
|    |              |               |                     |                 |                  |
|    |              |               |                     |                 |                  |
|    |              |               |                     |                 |                  |
|    |              |               |                     |                 |                  |
|    |              |               |                     |                 |                  |
|    |              |               |                     |                 |                  |
|    |              |               |                     |                 |                  |
|    |              |               |                     |                 |                  |

Belawan,.....  
Tim Fungsi Kepatuhan

(.....)

F.BBKP.BLW.8.10.0.2

|   |   |   |
|---|---|---|
|  | <b>UNIT PELAKSANA TEKNIS</b><br><b>BALAI BESAR KARANTINA PERTANIAN BELAWAN</b><br><b>JL. Sulawesi II Ujung Baru Belawan</b> |   |
|   | <b>FORMULIR</b>   | Kode Dokumen : F.BBKP.BLW.8.7.0.1<br>Revisi ke / Cetakan : 01 / 01<br>Tanggal : 1 Februari 2019 |
| <b>LAPORAN HADIAH, SUMBANGAN DAN KEUNTUNGAN SERUPA TAWARAN PENYUAPAN</b>          |   | Paraf Manajer Mutu :  |

**LAPORAN HADIAH, SUMBANGAN DAN KEUNTUNGAN SERUPA SERTA TAWARAN PENYUAPAN**

Nomor laporan : (diisi Tim Kepatuhan)  
 Pelapor :  
 Pengirim :  
 Nama :  
 Jabatan :  
 Instansi/Perusahaan :  
 Waktu Kejadian :  
 Kronologis :

.....

.....

Penerima Laporan :

Dilaporkan Oleh :

(.....)

(.....)

**Hasil Analisa Tim Kepatuhan :**

Diterima

Ditolak


|                       |                 |
|-----------------------|-----------------|
| <b>Alasan :</b>       | <b>Alasan :</b> |
| <b>Tindaklanjut :</b> |                 |

Tim Kepatuhan

(.....)

F.BBKP.BLW.8.7.0.1

*Dokumen ini bersifat rahasia, dilarang memperbanyak atau menyalin tanpa seijin Balai Besar Karantina Pertanian Belawan*

|  |   |   |
|--|---|---|
|       | <b>UNIT PELAKSANA TEKNIS</b><br><b>BALAI BESAR KARANTINA PERTANIAN BELAWAN</b><br><b>JL. Sulawesi II Ujung Baru Belawan</b> |   |
|  | <b>FORMULIR</b>   | Kode Dokumen : F.BBKP.BLW.8.7.0.2<br>Revisi ke / Cetakan : 01 / 01<br>Tanggal : 1 Februari 2019 |
| <b>ANALISA LAPORAN HADIAH, SUMBANGAN DAN KEUNTUNGAN SERUPA SERTA TAWARAN PENYUAPAN</b> |   | Paraf Manajer Mutu :  |

**ANALISA LAPORAN HADIAH, SUMBANGAN DAN KEUNTUNGAN SERUPA SERTA TAWARAN PENYUAPAN**

Nama Pelapor :

Tanggal Pelaporan :

Nomor Pelaporan :

**ANALISA**

Jenis Hadiah & Sumbangan :

Nilai Hadiah & Sumbangan yang diterima :

Frekuensi Pelapor menerima Hadiah & Sumbangan pada tahun ini :

Tujuan pemberian hadiah & sumbangan oleh pemberi :

**KESIMPULAN**

Apakah Hadiah & Sumbangan dibawah batas yang ditetapkan  YA  TIDAK

Apakah frekuensi pemberian hadiah & sumbangan dibawah batas yang ditetapkan bagi pelapor  YA  TIDAK

F.BBKP.BLW.8.7.0.2

*Dokumen ini bersifat rahasia, dilarang memperbanyak atau menyalin tanpa seijin Balai Besar Karantina Pertanian Belawan*



**UNIT PELAKSANA TEKNIS**  
**BALAI BESAR KARANTINA PERTANIAN BELAWAN**  
JL. Sulawesi II Ujung Baru Belawan

|  |                     |                    |
|--|---------------------|--------------------|
| <b>FORMULIR</b>  | Kode Dokumen        | F.BBKP.BLW.8.7.0.2 |
|  | Revisi ke / Cetakan | 01 / 01            |
|  | Tanggal             | 1 Februari 2019    |
| <b>ANALISA LAPORAN HADIAH, SUMBANGAN DAN KEUNTUNGAN SERUPA SERTA TAWARAN PENYUAPAN</b> | Paraf Manajer Mutu  |                    |


Rekomendasi tindakan lanjut bagi pelapor  DITERIMA  
 DIKEMBALIKAN

|                          |   |
|--------------------------|---|
| Tim Kepatuhan yang Hadir | 1 |
|                          | 2 |
|                          | 3 |

Belawan, .....  
Tim Fungsi Kepatuhan

F.BBKP.BLW.8.7.0.2

*Dokumen ini bersifat rahasia, dilarang memperbanyak atau menyalin tanpa seijin  
Balai Besar Karantina Pertanian Belawan*

|  |  |  |
|--|--|--|
|       | <b>UNIT PELAKSANA TEKNIS</b><br><b>BALAI BESAR KARANTINA PERTANIAN BELAWAN</b><br>JL. Sulawesi II Ujung Baru Belawan |  |
|  | <b>FORMULIR</b>  | Kode Dokumen: F.BBKP.BLW.8.7.0.3<br>Revisi ke / Cetakan: 01 / 01<br>Tanggal: 1 Februari 2019 |
| <b>LOGBOOK LAPORAN HADIAH, SUMBANGAN DAN KEUNTUNGAN SERUPA SERTA TAWARAN PENYUAPAN</b> |  | Paraf Manajer Mutu   |

**LOGBOOK LAPORAN HADIAH, SUMBANGAN DAN KEUNTUNGAN SERUPA SERTA TAWARAN PENYUAPAN**

| NO | NOMOR LAPORAN | PELAPOR | TANGGAL PELAPORAN | JENIS HADIAH, SUMBANGAN & KEUNTUNGAN SERUPA | PEMBERI HADIAH, SUMBANGAN & KEUNTUNGAN SERUPA | HASIL ANALISA TIM FUNGSI KEPATUHAN |              | KETERANGAN |
|----|---------------|---------|-------------------|---|---|------------------------------------|--------------|------------|
|    |               |         |                   |   |   | DITERIMA                           | DIKEMBALIKAN |            |
|    |               |         |                   |   |   |                                    |              |            |
|    |               |         |                   |   |   |                                    |              |            |
|    |               |         |                   |   |   |                                    |              |            |
|    |               |         |                   |   |   |                                    |              |            |
|    |               |         |                   |   |   |                                    |              |            |

F.BBKP.BLW.8.7.0.3

*Dokumen ini bersifat rahasia, dilarang memperbanyak atau menyalin tanpa seijin Balai Besar Karantina Pertanian Belawan*

Lampiran J Formulir Logbook Laporan Gratifikasi

