

4. IMPLEMENTASI SISTEM

Pada Bab ini akan membahas implementasi sistem dan program. Pemrograman menggunakan PHP dengan *framework* Laravel, Javascript, dan menggunakan *database* MySQL. Beberapa tools yang akan digunakan antara lain adalah Browser dan Visual Studio Code.

4.1. Pengaturan Koneksi *Database*

Dalam mengakses dan mengelola data yang masuk dan tersimpan semuanya menggunakan database MySQL yang terhubung dengan program pada file ".env". Source code pengaturan koneksi program dengan *database* dapat terlihat pada segmen program 4.1.

Segmen program 4.1 Pengaturan Koneksi Program dengan *Database*

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=komplain
DB_USERNAME=root
DB_PASSWORD=""
```

4.2. Implementasi Program

Implementasi program merupakan penerapan dari rancangan sistem sesuai dengan desain yang dibuat pada bab sebelumnya. Program dibagi menjadi beberapa segmen untuk lebih mudah dibaca dan dipahami, seperti yang ditunjukkan dalam tabel 4.1.

Tabel 4.1 Form Program

Fitur	Proses	Segmen
Koneksi Database	Pengaturan Koneksi Program dengan <i>Database</i>	Segmen Program 4.1
Menu Pengisian Data Komplain	Proses Pembuatan Data Komplain	Segmen Program 4.2
	Proses Pengiriman Email kepada CSO	Segmen Program 4.3
	Proses Pembuatan Data Live Tracking Baru	Segmen Program 4.4

Menu Live Tracking	Proses Pengambilan Data Komplain Menggunakan Kode Live Tracking	Segmen Program 4.5
	Proses Pengambilan Data Live Tracking Menggunakan Id Komplain	Segmen Program 4.6
Menu Lupa Kode Live Tracking	Proses Pengambilan Data Komplain Menggunakan Nomor Telepon Pasien	Segmen Program 4.7
Menu Riwayat Komplain	Proses Pengambilan Data Riwayat Komplain Menggunakan Nomor Telepon	Segmen Program 4.8
Menu Dashboard Admin	Perhitungan Jumlah Komplain per Hari	Segmen Program 4.9
Menu Daftar Komplain Admin	Perhitungan Jumlah Komplain per Bulan	Segmen Program 4.10
	Proses Pengambilan Beberapa Data Komplain untuk CSO	Segmen Program 4.11
	Proses Pengambilan Beberapa Data Komplain untuk Kepala Ruang	Segmen Program 4.12
	Proses Pengambilan Beberapa Data Komplain untuk Kepala Bidang	Segmen Program 4.13
	Proses Penghapusan Data Komplain	Segmen Program 4.14
Menu Isi Pesan Komplain Admin	Proses Pengambilan Data Komplain CSO Menggunakan Id Komplain	Segmen Program 4.15
	Proses Pengambilan Data Komplain Kepala Ruang Menggunakan Id Komplain	Segmen Program 4.16
	Proses Pengambilan Data Komplain Kepala Bidang Menggunakan Id Komplain	Segmen Program 4.17

	Proses Pengubahan Unit Pada Komplain	Segmen Program 4.18
	Proses Pengubahan Level Pada Komplain	Segmen Program 4.19
	Proses Pengiriman Email kepada Kepala Ruang	Segmen Program 4.20
	Proses Pengiriman Email kepada Kepala Bidang	Segmen Program 4.21
	Proses Pembuatan Laporan	Segmen Program 4.22
	Proses Pengubahan Status Komplain	Segmen Program 4.23
	Proses Pengubahan Penerima	Segmen Program 4.24
	Proses Pengambilan Data Level Menggunakan Nama Level	Segmen Program 4.25
	Proses Memperbarui Data Live Tracking (parameter id_komplain dan tanggal_update)	Segmen Program 4.26
	Proses Menunjukkan Beberapa Proses Data Komplain untuk CSO	Segmen Program 4.27
	Proses Menunjukkan Beberapa Proses Data Komplain untuk Kepala Ruang	Segmen Program 4.28
	Proses Menunjukkan Beberapa Proses Data Komplain untuk Kepala Bidang	Segmen Program 4.29

4.2.1 Menu Pengisian Data Komplain

Proses ini dilakukan pada saat customer mengirimkan komplain baru. Program yang ditampilkan berupa fungsi dari *Controller* Laravel yang telah dibuat. Data yang dimasukkan kedalam *database* yaitu jenis pasien, judul, nama, no_telp, unit, tanggal kejadian, waktu kejadian, tempat kejadian, kronologi, dan gambar.

Segmen program 4.2 Proses Pembuatan Data Komplain

```
public function store(Request $request)
{
    try {
        // Mendapatkan tanggal kejadian dari request
        $tanggal_kejadian =
$request->input('tanggal_kejadian');

        // Mengambil tahun, bulan, dan tanggal dari
tanggal kejadian
        $tahun = date('y', strtotime($tanggal_kejadian));
        $bulan = date('m', strtotime($tanggal_kejadian));
        $tanggal = date('d',
strtotime($tanggal_kejadian));

        // Membuat kode acak sepanjang 4 digit
        $kode_acak = str_pad(rand(0, 9999), 4, '0',
STR_PAD_LEFT);

        // Menggabungkan kode default 'GR' dengan tahun,
bulan, tanggal, dan kode acak
        $kode = 'GR' . $tahun . $bulan . $tanggal .
$kode_acak;

        $komplain = new Komplain();
        $jenis_pasien = $request->input('jenis_pasien');

        // Menambahkan kata-kata tambahan berdasarkan
jenis pasien
        if ($jenis_pasien === 'BPJS') {
            $jenis_pasien = 'pasien BPJS';
        } else if ($jenis_pasien === 'Umum') {
            $jenis_pasien = 'pasien umum';
        }

        $komplain->jenis_pasien = $jenis_pasien;
        $komplain->judul = $request->input('judul');
        $komplain->nama = $request->input('nama');

        // Ganti awalan nomor telepon jika dimulai dengan
"+62" menjadi "0"
        $no_telp = $request->input('no_telp');
        if (substr($no_telp, 0, 3) === '+62') {
            $no_telp = '0' . substr($no_telp, 3);
        }
        $komplain->no_telp = $no_telp;

        $komplain->unit = $request->input('unit');
        $komplain->tanggal_kejadian =
$request->input('tanggal_kejadian');
        $komplain->waktu_kejadian =
$request->input('waktu_kejadian');
```

```

        $komplain->tempat_kejadian =
$request->input('tempat_kejadian');
        $komplain->kronologi =
$request->input('kronologi');
        $komplain->id_status = 1;
        $komplain->id_level = 1;

        // Simpan gambar di server
        if ($request->hasFile('gambar')) {
            $file = $request->file('gambar');
            $fileName = time() . '_' .
$file->getClientOriginalName(); //mengirim gambar dengan
format file (time + nama file)
            $file->move(public_path('uploads'),
$fileName);
            $komplain->gambar = $fileName;
        } else {
            // Jika input gambar kosong, set gambar
menjadi "Tidak ada gambar"
            $komplain->gambar = "Tidak ada gambar";
        }

        $komplain->kode = $kode;
        $komplain->penerima = null;
        $komplain->laporan = null;
        $komplain->keterangan =
$request->input('keterangan');

        // Set created_at dan updated_at ke waktu
sekarang
        $now = \Carbon\Carbon::now();
        $komplain->created_at = $now;
        $komplain->updated_at = $now;

        $komplain->save();
        return response()->json(['message' => 'Komplain
created successfully', 'kode' => $kode], 201);
        // return redirect()->route('get-kode');
    } catch (\Exception $e) {
        // Tangkap kesalahan dan kembalikan respons yang
sesuai
        return response()->json(['error' =>
$e->getMessage()], 500);
    }
}

```

Proses ini akan dimulai setelah pasien telah mengisi komplain, lalu admin akan mendapatkan notifikasi email. Notifikasi email tersebut digunakan agar admin dapat mengetahui ada data komplain baru masuk.

Segmen program 4.3 Proses Pengiriman Email kepada CSO

```
public function index(){
    $komplains = Komplain::latest()->first();
    $data_email = [
        'jenis_pasien' => $komplains->jenis_pasien,
        'nama' => $komplains->nama,
        'judul' => $komplains->judul,
        'kronologi' => $komplains->kronologi,
    ];

    // Catat waktu mulai pengiriman email
    $start_time = now();
    Log::info('NotifikasiKomplainMasuk run');

    Mail::to(["email@gmail.com"])->send(new
    NotifikasiKomplainMasuk($data_email));

    // Catat waktu selesai pengiriman email
    $end_time = now();
    Log::info('NotifikasiKomplainMasuk done');

    // Hitung waktu eksekusi
    $execution_time =
    $end_time->diffInSeconds($start_time);
    Log::info("Execution time: {$execution_time}
    seconds");
    return 'Sukses mengirimkan email';
}
```

Proses ini akan dilakukan bersamaan dengan proses mengirimkan email. Dimana data status komplain (live tracking) akan mulai ditambahkan dalam *database*. Proses ini mengambil beberapa data dari database komplain seperti *id_komplain*, *id_status* dan *tanggal_sebelum_update*, sedangkan untuk data *tanggal_update* akan diambil dari frontend.

Segmen program 4.4 Proses Pembuatan Data Live Tracking Baru

```
public function store(Request $request)
{
    try {
        // Mengambil ID dari entri terbaru dalam tabel
        'komplain'
        $latestKomplain = Komplain::latest()->first();

        $live_Tracking = new Live_Tracking();
        $live_Tracking->id_komplain =
        $latestKomplain->id;
        $live_Tracking->id_status =
        $latestKomplain->id_status;
        $live_Tracking->tanggal_sebelum_update =
```

```

$latestKomplain->created_at;
    $live_Tracking->tanggal_update =
$request->input('tanggal_update');

    $live_Tracking->save();
    return response()->json(['message' =>
'LiveTracking successfully save'], 201);
    // return redirect()->route('get-kode');
    } catch (\Exception $e) {
    // Tangkap kesalahan dan kembalikan respons yang
sesuai
    return response()->json(['error' =>
$e->getMessage()], 500);
    }
}

```

4.2.2 Menu Live Tracking

Proses ini dilakukan pada saat customer ingin mencari tau status komplain sampai dimana menggunakan kode live tracking. Proses ini membutuhkan data kode live tracking untuk mengambil data komplain dari *database*. Jika proses telah selesai dilakukan maka outputnya adalah berupa data komplain.

Segmen program 4.5 Proses Pengambilan Data Komplain Menggunakan Kode Live Tracking

```

public function getLiveTracking($kode)
{
    $komplain = Komplain::where('kode', $kode)->first();

    if (!$komplain) {
        return response()->json(['message' => 'Komplain not
found'], 404);
    }

    return response()->json(['kode' => $komplain]);
}

```

Setelah data komplain telah diambil menggunakan kode, maka proses ini akan mengambil id komplain dari data komplain lalu digunakan untuk mengecek status komplain. Proses ini akan juga akan mengambil tabel status_komplain untuk ditampilkan dalam status komplain. Kemudian hasil dari proses tersebut adalah berupa array yang menyimpan beberapa data seperti id_status, nama_status, tanggal_sebelum_update dan tanggal_update.

Segmen program 4.6 Proses Pengambilan Data Live Tracking Menggunakan Id Komplain

```

public function getDataLiveTracking($idKomplain)
{

```

```

        // Menggunakan where() untuk mendapatkan semua entri yang
        cocok dengan idKomplain tertentu
        $komplainData = Live_Tracking::join('status_komplain',
        'live_tracking.id_status', '=', 'status_komplain.id')
        ->where('id_komplain', $idKomplain)
        ->select('live_tracking.id', 'live_tracking.id_status',
        'live_tracking.tanggal_sebelum_update',
        'live_tracking.tanggal_update',
        'status_komplain.nama_status')
        ->get();

        // Memeriksa apakah ada data yang ditemukan
        if ($komplainData->isEmpty()) {
            return response()->json(['message' => 'Komplain not
found'], 404);
        }

        // Menyiapkan array untuk menyimpan data
        $response = [];

        foreach ($komplainData as $komplain) {
            $response[] = [
                'id_status' => $komplain->id_status,
                'nama_status' => $komplain->nama_status,
                'tanggal_sebelum_update' =>
                $komplain->tanggal_sebelum_update,
                'tanggal_update' => $komplain->tanggal_update
            ];
        }

        return response()->json($response);
    }

```

4.2.3 Menu Lupa Kode Live Tracking

Proses ini dilakukan pada saat pasien yang lupa dengan kode live tracking atau kode live tracking hilang karena tidak disimpan. Proses ini akan mengambil kode live tracking dengan menggunakan nomor telepon dari pasien. Untuk mempermudah dalam memasukkan nomor telepon, kode akan terbaca dua pilihan jika di awal nomor telepon +62 maka kode akan mengganti menjadi angka 0 agar proses dapat berjalan sedangkan jika di awal nomor telepon angka 0 maka proses akan tetap berjalan.

Segmen program 4.7 Proses Pengambilan Data Komplain Menggunakan Nomor Telepon Pasien

```

public function getCodeLiveTracking($no_telp)
{
    // Menghilangkan karakter "+" dari nomor telepon
    $no_telp = str_replace('+', '', $no_telp);

```



```

        $komplain = Komplain::where('no_telp',
$no_telp)->get();

        if ($komplain->isEmpty()) {
            return response()->json(['message' => 'Komplain
not found'], 404);
        }
        return response()->json($komplain);
    }

```

4.2.4 Menu Riwayat Komplain

Proses ini dilakukan pada saat pasien ingin melihat riwayat komplain yang pernah ditulis oleh pasien. Proses ini menggunakan nomor telepon untuk mempermudah pasien dalam mencari.

Segmen program 4.8 Proses Pengambilan Data Riwayat Komplain Menggunakan Nomor Telepon

```

public function getIsiPesanKomplainByTelepon($no_telp)
{
    // Menghilangkan karakter "+" dari nomor telepon
    $no_telp = str_replace('+', '', $no_telp);

    $komplain = Komplain::where('no_telp',
$no_telp)->get();

    if ($komplain->isEmpty()) {
        return response()->json(['message' => 'Komplain
not found'], 404);
    }
    return response()->json($komplain);
}

```

4.2.5 Menu Dashboard Admin

Proses ini dilakukan pada saat pihak rumah sakit ingin melihat laporan dari komplain yang sudah berjalan baik itu sehari ataupun sebulan. Data setiap komplain yang masuk akan bertambah setiap hari atau setiap bulan. Proses ini digunakan untuk analisis dari data statistik yang didapatkan dari komplain yang masuk. Proses ini akan dimasukkan dalam dashboard admin untuk mempermudah admin.

Segmen program 4.9 Perhitungan Jumlah Komplain per Hari

```

public function getCountKomplainPerDay($tanggal)

```

```

    {
        // Ubah format tanggal dari URL ke dalam format yang
        sesuai
        $tanggalFormatted = date('Y-m-d',
        strtotime($tanggal));

        // Debugging: Cetak format tanggal yang telah
        diformat
        // echo $tanggalFormatted;

        // Gunakan Carbon untuk mengonversi tanggal ke dalam
        zona waktu yang benar
        $tanggalDateTime =
        \Carbon\Carbon::createFromFormat('Y-m-d', $tanggalFormatted);

        // Debugging: Cetak tanggal yang telah dikonversi
        // echo $tanggalDateTime;

        // Gunakan whereDate untuk mencocokkan tanggal dengan
        format yang benar
        $countKomplain = Komplain::whereDate('created_at',
        $tanggalFormatted)->count();

        // Debugging: Cetak jumlah komplain yang ditemukan
        // echo $countKomplain;

        return response()->json(['count' => $countKomplain]);
    }

```

Segmen program 4.10 Perhitungan Jumlah Komplain per Bulan

```

public function getCountKomplainPerMonth($tahun, $bulan)
{
    // Buat tanggal awal dan akhir dari bulan yang
    diberikan
    $startDate = \Carbon\Carbon::create($tahun, $bulan,
    1)->startOfMonth();
    $endDate = $startDate->copy()->endOfMonth();

    // Hitung jumlah komplain dalam rentang tanggal
    tersebut
    $countKomplain = Komplain::whereBetween('created_at',
    [$startDate, $endDate])->count();

    return response()->json(['count' => $countKomplain]);
}

```

4.2.6 Menu Daftar Komplain Admin

Proses ini dilakukan pada saat admin CSO ingin melihat daftar isi komplain yang baru masuk atau belum dikerjakan oleh admin CSO. Hasil dari proses ini akan berisikan beberapa data komplain dalam array.

Segmen program 4.11 Proses Pengambilan Beberapa Data Komplain untuk CSO

```
public function getDataKomplain()
{
    $komplains = Komplain::where('id_status', 1)
        ->where('id_level', 1)
        ->whereNull('keterangan')
        ->select('id', 'nama', 'judul', 'unit',
'created_at')->get();
    return response()->json($komplains);
}
```

Proses ini dilakukan pada saat admin Kepala Ruang ingin melihat daftar isi komplain yang baru masuk atau belum dikerjakan oleh admin Kepala Ruang. Hasil dari proses ini akan sama seperti proses diatas yang berisikan beberapa data komplain dalam array. Namun perbedaannya adalah proses ini akan mengambil id_level 2 dan 3 agar berbeda dengan admin CSO maupun admin Kepala Bidang dan data dari *database* akan diambil berdasarkan penerima yang memiliki kata-kata “Kepala Ruang”.

Segmen program 4.12 Proses Pengambilan Beberapa Data Komplain untuk Kepala Ruang

```
public function getDataKomplainLevelHijau()
{
    $komplains = Komplain::join('level_komplain',
'komplain.id_level', '=', 'level_komplain.id')
        ->where(function($query) {
            $query->where('komplain.id_level', 2)
                ->orWhere('komplain.id_level', 3)
                ->where('penerima', 'LIKE', '%Kepala
Ruang%');
        })
        ->where('komplain.id_status', 2)
        ->select('komplain.id', 'komplain.nama',
'komplain.judul', 'komplain.unit', 'komplain.created_at',
'level_komplain.namaLevel')
        ->get();

    return response()->json($komplains);
}
```

Proses ini dilakukan pada saat admin Kepala Bidang ingin melihat daftar isi komplain yang baru masuk atau belum dikerjakan oleh admin Kepala Bidang. Hasil dari proses ini akan sama seperti proses diatas yang berisikan beberapa data komplain dalam array. Namun perbedaannya adalah proses ini akan mengambil id_level 3 dan 4 agar berbeda dengan admin CSO maupun admin Kepala Ruang dan data dari *database* akan diambil berdasarkan penerima yang memiliki kata-kata “Kepala Bidang”.

Segmen program 4.13 Proses Pengambilan Beberapa Data Komplain untuk Kepala Bidang

```
public function getDataKomplainLevelMerah()
{
    $komplains = Komplain::join('level_komplain',
    'komplain.id_level', '=', 'level_komplain.id')
        ->where(function($query) {
            $query->where('komplain.id_level', 3)
                ->orWhere('komplain.id_level', 4);
        })
        ->where('penerima', 'LIKE', '%Kepala Bidang%')
        ->where('komplain.id_status', 2)
        ->select('komplain.id', 'komplain.nama',
        'komplain.judul', 'komplain.unit', 'komplain.created_at',
        'level_komplain.namaLevel')
        ->get();

    return response()->json($komplains);
}
```

Proses ini dilakukan pada saat admin ingin menghapus komplain karena bisa jadi komplain telah diselesaikan langsung di tempat atau komplain yang sama sedang dalam proses.

Segmen program 4.14 Proses Penghapusan Data Komplain

```
public function destroy(Request $request)
{
    try {
        $komplain = Komplain::findOrFail($request->id);

        // Hapus terlebih dahulu entri terkait dalam
        tabel live_tracking
        $komplain->liveTrackings()->delete();

        // Hapus entri dari tabel komplain
        $komplain->delete();
        return response()->json(['message' => 'Data
        berhasil dihapus'], 201);
    } catch (\Exception $e) {
        // Tangani kesalahan jika terjadi
    }
}
```

```

        return response()->json(['error' =>
$e->getMessage()], 500);
    }
}

```

4.2.7 Menu Isi Pesan Komplain Admin

Proses ini dilakukan pada saat admin CSO ingin melihat isi dari komplain yang baru masuk atau belum dikerjakan oleh admin CSO. Hasil dari proses ini akan berisikan data komplain berdasarkan id_komplain.

Segmen program 4.15 Proses Pengambilan Data Komplain CSO Menggunakan Id Komplain

```

public function getIsiPesanKomplainById($id)
{
    $komplain = Komplain::find($id);

    if (!$komplain) {
        return response()->json(['message' => 'Komplain
not found'], 404);
    }

    return response()->json($komplain);
}

```

Proses ini dilakukan pada saat admin Kepala Ruang ingin melihat isi dari komplain yang baru masuk atau belum dikerjakan oleh admin Kepala Ruang. Hasil dari proses ini akan berisikan data komplain berdasarkan id_komplain dan data level untuk mengetahui jenis level apa yang didapatkan dalam komplain tersebut.

Segmen program 4.16 Proses Pengambilan Data Komplain Kepala Ruang Menggunakan Id Komplain

```

public function getIsiPesanKomplainKepalaRuangById($id)
{
    $komplain = Komplain::leftJoin('level_komplain',
'komplain.id_level', '=', 'level_komplain.id')
->where('komplain.id', $id)
->select('komplain.*',
'level_komplain.namaLevel')
->first();

    if (!$komplain) {
        return response()->json(['message' => 'Komplain
not found'], 404);
    }
}

```

```

        return response()->json($komplain);
    }

```

Proses ini dilakukan pada saat admin Kepala Bidang ingin melihat isi dari komplain yang baru masuk atau belum dikerjakan oleh admin Kepala Bidang. Hasil dari proses ini akan berisikan data komplain berdasarkan id_komplain dan data level untuk mengetahui jenis level apa yang didapatkan dalam komplain tersebut.

Segmen program 4.17 Proses Pengambilan Data Komplain Kepala Bidang Menggunakan Id Komplain

```

public function getIsiPesanKomplainKepalaBidangById($id)
{
    $komplain = Komplain::leftJoin('level_komplain',
    'komplain.id_level', '=', 'level_komplain.id')
        ->where('komplain.id', $id)
        ->select('komplain.*',
    'level_komplain.namaLevel')
        ->first();

    if (!$komplain) {
        return response()->json(['message' => 'Komplain
not found'], 404);
    }

    return response()->json($komplain);
}

```

Proses ini dilakukan pada saat admin merasa bahwa unit yang ditulis oleh pasien belum tentu sesuai dengan kronologi yang terjadi.

Segmen program 4.18 Proses Pengubahan Unit Pada Komplain

```

public function editUnit(Request $request){
    try {
        $komplain = Komplain::findOrFail($request->input('id'));

        // Ubah nilai kolom 'unit' sesuai input dari request
        $komplain->unit = $request->input('unit');

        $komplain->save();
        return response()->json(['message' => 'unit komplain
berhasil diubah'], 201);
        // return redirect()->route('get-kode');
    } catch (\Exception $e) {
        // Tangkap kesalahan dan kembalikan respons yang sesuai
        return response()->json(['error' => $e->getMessage()],
    500);
    }
}

```

```
}
}
```

Proses ini dilakukan pada saat admin CSO akan memilih level yang sesuai dengan komplain yang diterima kemudian penerima komplain merupakan pihak yang bertanggung jawab terhadap proses dan penyelesaian komplain.

Segmen program 4.19 Proses Pengubahan Level Pada Komplain

```
public function edit_level(Request $request){
    try {
        $komplain = Komplain::findOrFail($request->input('id'));

        $komplain->penerima = $request->input('penerima');

        // Ubah nilai kolom 'id_level & id_status' sesuai input
        dari request
        $komplain->id_level = $request->input('id_level');
        $komplain->id_status = $request->input('id_status');

        $komplain->save();
        return response()->json(['message' => 'komplain
        berhasil terkirim'], 201);
    } catch (\Exception $e) {
        return response()->json(['error' =>
        $e->getMessage()], 500);
    }
}
```

Proses ini dilakukan pada saat admin CSO selesai memilih level. Setelah itu admin Kepala Ruang akan mendapatkan notifikasi email masuk.

Segmen program 4.20 Proses Pengiriman Email kepada Kepala Ruang

```
public function indexKepalaRuang(Request $request){
    // Ambil data input dari permintaan
    $jenis_pasien = $request->input('jenis_pasien');
    $nama = $request->input('nama');
    $judul = $request->input('judul');
    $kronologi = $request->input('kronologi');
    $level = $request->input('level');

    $data_email = [
        'jenis_pasien' => $jenis_pasien,
        'nama' => $nama,
        'judul' => $judul,
        'kronologi' => $kronologi,
        'level' => $level,
    ];
}
```

```

        // Catat waktu mulai pengiriman email
        $start_time = now();
        Log::info('NotifikasiKomplainMasuk run');

        Mail::to(["email@gmail.com"])->send(new
NotifikasiRuangKomplainMasuk($data_email));

        // Catat waktu selesai pengiriman email
        $end_time = now();
        Log::info('NotifikasiKomplainMasuk done');

        // Hitung waktu eksekusi
        $execution_time =
$end_time->diffInSeconds($start_time);
        Log::info("Execution time: {$execution_time}
seconds");
        return 'Sukses mengirimkan email>';
    }

```

Proses ini dilakukan pada saat admin CSO selesai memilih level. Setelah itu admin Kepala Bidang akan mendapatkan notifikasi email masuk.

Segmen program 4.21 Proses Pengiriman Email kepada Kepala Bidang

```

public function indexKepalaBidang(Request $request){
    // Ambil data input dari permintaan
    $jenis_pasien = $request->input('jenis_pasien');
    $nama = $request->input('nama');
    $judul = $request->input('judul');
    $kronologi = $request->input('kronologi');
    $level = $request->input('level');

    $data_email = [
        'jenis_pasien' => $jenis_pasien,
        'nama' => $nama,
        'judul' => $judul,
        'kronologi' => $kronologi,
        'level' => $level,
    ];

    // Catat waktu mulai pengiriman email
    $start_time = now();
    Log::info('NotifikasiKomplainMasuk run');

    Mail::to(["email@gmail.com"])->send(new
NotifikasiBidangKomplainMasuk($data_email));

    // Catat waktu selesai pengiriman email
    $end_time = now();
    Log::info('Notifikasi Komplain Masuk done');
}

```



```

        // Hitung waktu eksekusi
        $execution_time =
$send_time->diffInSeconds($start_time);
        Log::info("Execution time: {$execution_time}
seconds");
        return 'Sukses mengirimkan email';
    }

```

Proses ini dilakukan pada saat admin CSO, Kepala Ruang, dan Kepala Bidang ingin menuliskan laporan yang telah diselesaikan. Kemudian dalam laporan komplain akan terdapat beberapa data seperti penerima, laporan, jenis_pasien, nama, judul, dan kronologi.

Segmen program 4.22 Proses Pembuatan Laporan

```

public function reply(Request $request){
    try {
        $komplain =
Komplain::findOrFail($request->input('id'));
        $komplain->penerima =
$request->input('penerima');
        $komplain->laporan = $request->input('laporan');
        $komplain->jenis_pasien =
$request->input('jenis_pasien');
        $komplain->nama = $request->input('nama');
        $komplain->judul = $request->input('judul');
        $komplain->kronologi =
$request->input('kronologi');

        // Simpan PDF di server
        $fileName = time() . '_laporan.pdf';

        // Menyimpan konten HTML dari view blade ke
variabel
        $html = view('laporan',
compact('komplain'))->render();

        // Buat instance TCPDF dengan kertas A4 (210x297
mm)
        $pdf = new TCPDF('P', 'mm', 'A4', true, 'UTF-8',
false);

        $pdf->SetTitle('Laporan Komplain');

        // Tambahkan halaman baru
        $pdf->AddPage();

        // Tulis konten ke dokumen
        $pdf->writeHTML($html, true, false, true, false,
'');
    }
}

```

```

        // Tambahkan gambar ke PDF jika ada
        if ($request->hasFile('gambar')) {
            $imgFile = $request->file('gambar');
            $imgPath = $imgFile->getPathName();
            $pdf->Image($imgPath, 15, 140, 75, 113,
            'JPG', '', 'T', false, 300, '', false, false, 1, false,
            false, false);
        }

        $pdf->Output(public_path('uploads/' . $fileName),
        'F');

        $komplain->laporan = $fileName; // Simpan nama
file PDF
        $komplain->id_status =
$request->input('id_status');

        // Simpan data komplain
        $komplain->save();

        return response()->json(['message' => 'Komplain
berhasil terkirim dan PDF disimpan'], 201);
    } catch (\Exception $e) {
        return response()->json(['error' =>
$e->getMessage()], 500);
    }
}

```

Proses ini dilakukan pada saat admin CSO, Kepala Ruang, dan Kepala Bidang selesai mengirimkan laporan dan data komplain akan di update dalam proses ini agar status komplain pada pasien dapat berubah.

Segmen program 4.23 Proses Pengubahan Status Komplain

```

public function edit_status(Request $request){
    try {
        $komplain =
        Komplain::findOrFail($request->input('id'));
        $komplain->id_status = $request->input('id_status');
        $komplain->keterangan =
        $request->input('keterangan');

        $komplain->save();
        return response()->json(['message' => 'komplain
berhasil terupdate'], 201);
    } catch (\Exception $e) {
        return response()->json(['error' =>
$e->getMessage()], 500);
    }
}

```

Proses ini dilakukan pada saat admin ingin memindahkan komplain ke penerima lain sehingga pada saat penerimaan komplain dapat diterima oleh penerima yang bersangkutan.

Segmen program 4.24 Proses Pengubahan Penerima

```
public function editPenerima(Request $request){
    try {
        $komplain = Komplain::findOrFail($request->input('id'));

        // Ubah nilai kolom 'penerima' sesuai input dari request
        $komplain->penerima = $request->input('penerima');

        $komplain->save();
        return response()->json(['message' => 'penerima komplain
berhasil diupdate'], 201);
        // return redirect()->route('get-kode');
    } catch (\Exception $e) {
        // Tangkap kesalahan dan kembalikan respons yang sesuai
        return response()->json(['error' => $e->getMessage()],
500);
    }
}
```

Proses ini dilakukan pada saat admin menerima komplain dan countdown pengerjaan diambil dari proses ini. Proses ini mengambil nama level lalu data yang diambil adalah durasi pengerjaan.

Segmen program 4.25 Proses Pengambilan Data Level Menggunakan Nama Level

```
public function show($namaLevel)
{
    $level = Level_komplain::where('namaLevel',
$namaLevel)->get();

    if ($level->isEmpty()) {
        return response()->json(['message' => 'Komplain
not found'], 404);
    }
    return response()->json($level);
}
```

Proses ini dilakukan pada saat admin selesai menerima komplain, menyelesaikan komplain dan mengirimkan laporan komplain.

Segmen program 4.26 Proses Memperbarui Data Live Tracking (parameter id_komplain dan tanggal_update)

```
public function update(Request $request, Live_Tracking
$live_Tracking)
{
    try {
        // Mengambil ID dari entri terbaru dalam tabel
        'komplain'
        $idKomplain = $request->input('idKomplain');
        $latestKomplain = Komplain::where('id',
        $idKomplain)->first();

        if (!$latestKomplain) {
            return response()->json(['error' => 'Komplain
not found'], 404);
        }

        // Mengambil ID dari entri terbaru dalam tabel
        'komplain'
        $live_Tracking = new Live_Tracking();
        $live_Tracking->id_komplain =
        $latestKomplain->id;
        $live_Tracking->id_status =
        $latestKomplain->id_status;
        $live_Tracking->tanggal_sebelum_update =
        $latestKomplain->created_at;
        $live_Tracking->tanggal_update =
        $request->input('tanggal_update');

        $live_Tracking->save();
        return response()->json(['message' =>
        'LiveTracking successfully save'], 201);
        // return redirect()->route('get-kode');
    } catch (\Exception $e) {
        // Tangkap kesalahan dan kembalikan respons yang
        sesuai
        return response()->json(['error' =>
        $e->getMessage()], 500);
    }
}
```

4.2.8 Menu Daftar Proses Komplain Admin

Proses ini dilakukan pada saat admin CSO mendapatkan data baru untuk dikonfirmasi laporan komplain yang ditulis oleh kepala ruang atau kepala bidang sudah benar atau tidak.

Segmen program 4.27 Proses Menunjukkan Beberapa Proses Data Komplain untuk CSO

```
public function getDataKomplainWaiting()
```

```

{
    $komplains = Komplain::join('level_komplain',
    'komplain.id_level', '=', 'level_komplain.id')
    ->where(function($query) {
        $query->where('komplain.id_level', 2)
        ->orWhere('komplain.id_level', 3)
        ->orWhere('komplain.id_level', 4);
    })
    ->where(function($query) {
        $query->where('komplain.keterangan', 'menunggu
konfirmasi');
    })
    ->where(function($query) {
        $query->where('komplain.id_status', 5);
    })
    ->where(function($query) {
        $query->where('penerima', 'LIKE', '%Kepala Ruang%')
        ->orWhere('penerima', 'LIKE', '%Kepala Bidang%');
    })
    ->select('komplain.id', 'komplain.nama',
    'komplain.judul', 'komplain.unit', 'komplain.created_at',
    'level_komplain.namaLevel', 'komplain.id_level',
    'komplain.id_status', 'komplain.penerima',
    'komplain.keterangan')
    ->get();

    return response()->json($komplains);
}

```

Proses ini dilakukan pada saat admin Kepala Ruang ingin melihat daftar komplain yang sudah diterima dan sedang dalam “proses” atau “menunggu laporan”.

Segmen program 4.28 Proses Menunjukkan Beberapa Proses Data Komplain untuk Kepala Ruang

```

public function getDataProsesKomplainLevelHijau()
{
    $komplains = Komplain::join('level_komplain',
    'komplain.id_level', '=', 'level_komplain.id')
    ->where(function($query) {
        $query->where('komplain.id_level', 2)
        ->orWhere('komplain.id_level', 3);
    })
    ->where('penerima', 'LIKE', '%Kepala Ruang%')
    ->where(function($query) {
        $query->where('komplain.id_status', 3)
        ->orWhere('komplain.id_status', 4);
    })
    ->select('komplain.id', 'komplain.nama',
    'komplain.judul', 'komplain.unit', 'komplain.created_at',

```

```

'level_komplain.namaLevel', 'komplain.keterangan')
->get();

return response()->json($komplains);
}

```

Proses ini dilakukan pada saat admin Kepala Bidang ingin melihat daftar komplain yang sudah diterima dan sedang dalam “proses” atau “menunggu laporan”.

Segmen program 4.29 Proses Menunjukkan Beberapa Proses Data Komplain untuk Kepala Bidang

```

public function getDataProsesKomplainLevelMerah()
{
    $komplains = Komplain::join('level_komplain',
    'komplain.id_level', '=', 'level_komplain.id')
    ->where(function($query) {
        $query->where('komplain.id_level', 3)
        ->orWhere('komplain.id_level', 4);
    })
    ->where('penerima', 'LIKE', '%Kepala Bidang%')
    ->where(function($query) {
        $query->where('komplain.id_status', 3)
        ->orWhere('komplain.id_status', 4);
    })
    ->select('komplain.id', 'komplain.nama',
    'komplain.judul', 'komplain.unit', 'komplain.created_at',
    'level_komplain.namaLevel', 'komplain.keterangan')
    ->get();

    return response()->json($komplains);
}

```