

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Bank BRI, selaku bank terbesar kedua di Indonesia, dihadapkan pada tantangan mengamankan dan memantau mesin CRM yang tersebar di seluruh wilayah Indonesia. Keterbatasan pemantauan CCTV hanya pada ruang kontrol kantor cabang terdekat menyulitkan pemantauan secara terpusat. Masalah lainnya adalah waktu yang dibutuhkan untuk menyimpan video karena proses *backup* manual. Untuk mengatasi permasalahan tersebut, Bank BRI mengusulkan pembuatan *website dashboard monitoring CCTV* guna memfasilitasi pemantauan yang lebih efisien dan terpusat. Selain itu, fitur penyimpanan video yang efisien dan terpusat dalam *website dashboard* akan menghemat waktu dan usaha petugas keamanan dalam menyimpan video, sehingga memungkinkan akses yang cepat dalam situasi darurat.

Laporan ini memiliki batasan masalah yang terfokus pada perancangan sistem pemantauan dan keamanan untuk ATM CRM Bank BRI. Sistem ini akan menampilkan pemantauan CCTV terhubung dengan ATM CRM, dengan akses dan informasi yang dikelola oleh admin melalui *login* menggunakan email dan password. Pengaturan akses oleh superadmin akan memastikan informasi yang ditampilkan pada dashboard sesuai dengan peran dan tanggung jawab masing-masing admin. Namun, laporan ini tidak membahas teknologi *engine* untuk mendapatkan dan menyimpan video CCTV serta tidak termasuk proses deteksi manusia pada video CCTV. Fokus utama laporan ini adalah pada perancangan *website dashboard monitoring CCTV* menggunakan *framework Laravel* untuk pemantauan terpusat dan penyimpanan video yang efisien pada ATM CRM Bank BRI.

Tujuan dari perancangan sistem ini adalah menciptakan suatu sistem yang dapat melakukan pemantauan kamera pengawas secara terpusat pada ATM CRM Bank BRI, dengan fokus pada meningkatkan efisiensi penyimpanan video, menganalisis durasi transaksi, dan memetakan lokasi serta status perangkat.

Dengan demikian, sistem ini bertujuan untuk meningkatkan keamanan dan pengawasan ATM CRM Bank BRI, memudahkan pemantauan dan analisis data, serta mempercepat respons teknisi dalam mengatasi gangguan pada perangkat.

Melalui implementasi *website dashboard monitoring CCTV*, Bank BRI diharapkan dapat meningkatkan keamanan sistem dan perangkat CRM mereka secara menyeluruh. Pemantauan CCTV yang terpusat dan efektif mengurangi penggunaan waktu yang digunakan untuk mengakses video CCTV akan membantu memastikan keamanan area mesin CRM, sementara fitur penyimpanan video yang terorganisir akan menghemat waktu dan usaha petugas keamanan dalam hal *backup* video. Selain itu, fitur perhitungan durasi nasabah akan memberikan data untuk meningkatkan layanan. Dengan demikian, penggunaan *website dashboard* ini diharapkan dapat memberikan manfaat yang signifikan dalam mengatasi tantangan keamanan kompleks dan meningkatkan pengalaman nasabah Bank BRI secara keseluruhan.

## 1.2 Profil

Gambar 1.1 Logo Perusahaan

PT Millenio Amerta Data atau yang sering disebut dengan nama Kecilin merupakan salah satu *startup digital* yang bergerak dibidang teknologi kompresi yang dapat meng-*compress* berbagai tipe *file* seperti teks, dokumen, foto, dan video. *Startup digital* merupakan perusahaan rintisan yang dibangun oleh sekumpulan individu sehingga menghasilkan suatu produk teknologi [1]. Pada tahun 2018, Kecilin berdiri dengan tujuan utama menciptakan kesetaraan dalam akses informasi, bahkan ketika terbatasnya akses data telekomunikasi. Kecilin menghadirkan teknologi kompresi yang efisien dan dapat digunakan dengan

mudah.

Perusahaan yang dipimpin oleh Bisma Manda Samsu ini beralamatkan di Wisma Taranis, Jl. Bendungan Hilir Raya, No. 76, Kel. Bendungan Hilir, Kec. Tanah Abang, Kota Jakarta Pusat, Prov. DKI Jakarta serta memiliki suatu kantor cabang di Daerah Istimewa Yogyakarta. Kecilin menggandeng beberapa investor ternama seperti Alexander Rusli, Monovan Sakti Jaya Kusuma, Toto Sugiri, dan Hadi Wenas.

Kecilin memiliki beberapa bidang produk yaitu Kecilin Data Optimizer, Cloud Compression, Livestream Compression, Internet Accelerator, dan Kecilin Drive. Salah satu produk yang akan dibahas pada laporan ini adalah dashboard monitoring BRI-CRM. Produk yang termasuk dalam bidang Livestream Compression ini merupakan kerja sama dengan Bank BRI. Produk ini memiliki fitur utama yaitu *compressing data* dan *backup video CCTV* serta *streaming video CCTV* pada ATM CRM Bank BRI. **Estimasi biaya dalam perancangan sistem ini adalah 500-800 juta rupiah.**

Pada bulan Maret 2021, penulis bergabung sebagai karyawan di Kecilin.id sebagai anggota divisi *Livestream*. Peran utama penulis adalah sebagai *frontend developer*, bertanggung jawab untuk mengimplementasikan desain UI/UX menjadi sebuah *website* dan *menghubungkan data* yang dibuat oleh tim *backend* agar berfungsi dengan baik pada antarmuka pengguna. Selain itu, dalam beberapa proyek, penulis juga membantu dalam pembuatan algoritma dan fungsionalitas pada tim *backend*.

### 1.3 Landasan Teori

#### 1.3.1 Tinjauan Pustaka

Penelitian yang berjudul Sistem Keamanan Pemantauan CCTV Online Berbasis Android Pada Rumah Cantik Syifa menguraikan bahwa pemantauan kinerja karyawan merupakan salah satu aspek penting. Pemantauan kinerja karyawan memungkinkan karyawan untuk bekerja sesuai standar yang ditargetkan. Dengan meningkatnya mobilitas masyarakat yang tinggi, pemantauan CCTV *online* menggunakan android menjadi salah satu solusi karena dapat diakses selama



24 jam nonstop dimanapun berada. Penelitian ini menggunakan koneksi internet dari telkom, *router mikrotik RB951Ui-2HnD* dan *DVR CCTV TDV-3308-H1 Series*. Kesimpulan dari penelitian ini adalah sistem keamanan pemantauan *CCTV online* berbasis android dapat bermanfaat bagi pemilik perusahaan karena dapat memantau pelanggan yang keluar masuk klinik secara *realtime* menggunakan *smartphone* [2].

Referensi kedua merupakan penelitian yang dilakukan oleh Khaidir Nalakhudin, dkk. *Ketersediaan CCTV pada suatu rumah sakit dapat memantau area krusial seperti antrean pasien, ruang tunggu, area parkir, ruang kerja, dll.* Oleh karena itu, untuk menjaga stabilitas perangkat *CCTV* diperlukan pemantauan secara *realtime*. Perpaduan *The Duddle* sebagai media *monitoring* dan *Telegram Bot* sebagai media notifikasi dapat menjadi solusi. Dengan adanya notifikasi secara *realtime* dapat memudahkan teknisi dalam mencari titik yang terjadi gangguan pada perangkat *CCTV* [3].

Jurnal berjudul "Rancang Bangun Sistem Monitoring Kamera *CCTV Online* Dengan Penerapan *Hik-Connet*" membahas penggunaan monitoring *CCTV online* sebagai solusi untuk memantau keadaan kantor selama *WFH* dan *PSBB* selama pandemi *COVID-19*. Penelitian ini bertujuan meningkatkan keamanan dan menjawab pertanyaan seputar penggunaan aplikasi monitoring *CCTV*. Hasil analisis menunjukkan bahwa sebelumnya pemantauan *CCTV* hanya terbatas pada jaringan lokal. Namun, dengan penerapan aplikasi *Hik-Connet*, pengguna dapat memantau *CCTV* dari lokasi yang berbeda tanpa perlu datang ke lokasi *DVR*. Penggunaan internet sebagai penghubung memungkinkan akses jarak jauh. Aplikasi ini juga memiliki fitur monitoring dan backup file, menghilangkan kebutuhan untuk melakukan backup di lokasi *DVR* [4].

Referensi [5] membahas tentang implementasi teknologi *CCTV* berbasis kamera *IP* untuk meningkatkan keamanan lingkungan di lingkungan masyarakat *RT 04 RW 13* di Komplek *Griya Prima Asri Baleendah*. Saat ini, pengamanan dilakukan oleh 9 satuan pengamanan yang terbagi dalam 2 shift, namun cakupan pemantauan menjadi sulit. Solusi yang diusulkan adalah menggunakan *CCTV*

murah, mudah digunakan, dan memiliki fungsi pengawasan yang lebih baik. Kamera IP mengirim gambar melalui internet ke NVR yang merekam dan menyimpan video. Pengguna dapat mengakses CCTV melalui smartphone atau laptop menggunakan aplikasi CMSclient atau Yoosee. Implementasi CCTV telah dilakukan di beberapa titik pengawasan dan dapat diperluas di masa mendatang. Hasil pengabdian masyarakat menunjukkan keberhasilan pengawasan keamanan lingkungan dengan CCTV, dan pengembangan lebih lanjut dapat dilakukan dengan menambahkan CCTV di titik pengawasan lainnya.

Penelitian yang dilakukan oleh [6] membahas tentang pengembangan sistem informasi QC pada PT Victory Chingluh Indonesia, perusahaan manufaktur sepatu olahraga. Saat ini, bagian QC menggunakan cara konvensional dalam pembuatan laporan QC, yang menyebabkan perbedaan data antara bagian QC, perencanaan, dan produksi. Oleh karena itu, peneliti merancang dan mengembangkan sistem dashboard monitoring dan reporting QC proses laminating berbasis web. Penelitian ini menggunakan database MySQL dan bahasa pemrograman PHP dengan framework Laravel. Sistem ini juga dilengkapi dengan fitur laporan dalam bentuk dashboard untuk mempermudah pemantauan dan analisis laporan.

### 1.3.2 *Bank Republik Indonesia*



Gambar 1. 2 Logo Bank BRI

Bank Rakyat Indonesia merupakan bank pemerintah pertama setelah kemerdekaan Indonesia didirikan pada tanggal 16 Desember 1895 oleh R. Wiriadarmaja diberi nama "*De Poerwokerto Hulp-en Spaarbank der Inlandasche*

*Hoof den*". BRI dikenal dengan layanan untuk segmen *micro-banking* dan juga kredit usaha kecil.

Segala layanan dan produk perbankan didukung oleh jaringan yang luas dari BRI yang menjangkau seluruh wilayah Indonesia, bahkan daerah-daerah terpencil yang sulit diakses. Saat ini, BRI memiliki lebih dari 4.500 lokasi yang tersebar di seluruh Indonesia, termasuk kantor cabang, kantor cabang pembantu, kantor cabang khusus, kantor perwakilan luar negeri, kantor kas bayar, kantor mobil bank, *payment point*, dan pos pelayanan desa. Sebagai tambahan, BRI juga telah meluncurkan layanan *E-Tax* sebagai bagian dari layanan teknologi *cash management* yang ditawarkan kepada nasabah korporasi [7].

### 1.3.3 *Cash Recycling Machine*

*Cash Recycling Machine* (CRM) merupakan evolusi dari mesin *Cash Deposit Machine* (CDM). Mesin CDM hanya berfungsi untuk melakukan penarikan tunai, sedangkan mesin CRM mampu melakukan penyetoran dan penarikan tunai. Dengan adanya mesin CRM, konsumen yang memiliki mobilitas tinggi dapat diuntungkan, karena mereka dapat langsung melakukan setoran tunai ke bank tanpa perlu mengisi formulir setoran. Selain itu, penggunaan mesin CRM juga dapat mengurangi waktu tunggu di bank, karena konsumen dapat menggunakan mesin tersebut di luar jam kerja atau bahkan pada hari libur [8].

### 1.3.4 *Monitoring*

*Monitoring* merupakan salah satu aktivitas yang bertujuan untuk memastikan pencapaian semua tujuan organisasi dan manajemen. *Monitoring* merupakan langkah yang dilakukan untuk mengevaluasi apakah kegiatan yang telah dilakukan sesuai dengan rencana yang telah ditetapkan. Selain itu, tujuan dari *monitoring* juga meliputi mengidentifikasi masalah yang muncul agar dapat segera ditangani, melakukan penilaian terhadap efektivitas pola kerja dan manajemen yang digunakan untuk mencapai tujuan, serta memahami hubungan antara kegiatan yang dilakukan dengan tujuan yang ingin dicapai sehingga dapat memperoleh ukuran kemajuan yang jelas [9].

### 1.3.5 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah alat yang digunakan untuk membuat model sistem dengan cara yang profesional. DFD menggambarkan sistem tersebut sebagai suatu jaringan proses fungsional yang saling terhubung melalui aliran data, baik secara manual maupun komputerisasi [10]. DFD adalah teknik grafis yang digunakan untuk menggambarkan desain informasi saat data bergerak dari input menjadi output. DFD dapat digunakan untuk mempresentasikan sistem atau perangkat lunak pada berbagai tingkat abstraksi [11].




### 1.3.6 Simbol pada Data Flow Diagram

Tabel 1.1 memuat informasi tentang simbol-simbol yang digunakan dalam DFD versi E. Yourdon dan De Marco serta versi Chris Gane dan Trish Sarson. Simbol-simbol tersebut memberikan representasi visual yang konsisten untuk entitas, proses, aliran data, penyimpanan data, serta aliran kendali dalam DFD.

Tabel 1.1 Simbol Data Flow Diagram

	Yourdon	C. Gane dan T. Sarson	Keterangan
Aliran data/ <i>Data Flow</i>			Aliran data mengalir diantara proses, simpanan data, dan kesatuan luar.
Proses/ <i>Process</i>			Suatu proses terjadi karena adanya aliran data yang masuk dan hasil dari proses juga berupa aliran data yang mengalir keluar.
Simpanan data/ <i>data store</i>			<i>Data store</i> merupakan tempat penyimpanan data yang berupa <i>file</i> atau tabel di <i>database</i> .



	Yourdan	C. Gane dan T. Sarson	Keterangan
Kesatuan luar, batas sistem/ <i>External entity, boundary</i>			Kesatuan luar atau <i>external entity</i> merupakan kesatuan dilingkungan luar sistem, yang akan memberikan <i>input</i> ataupun menerima <i>output</i> dari sistem.
Aliran phisik/ <i>Material Flow</i>			

### 1.3.7 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah suatu diagram yang digunakan untuk menggambarkan desain konseptual dari model basis data relasional. ERD menggambarkan hubungan antara entitas-entitas dalam basis data. [12] ERD adalah representasi visual yang memodelkan struktur data dan hubungan antara data dengan menggunakan notasi grafis. ERD membantu dalam memahami dan merancang struktur data dalam basis data serta menggambarkan bagaimana entitas-entitas tersebut saling terhubung satu sama lain [13].

### 1.3.8 Komponen Utama Entity Relationship Diagram [12]

#### 1. Entitas (*Entity*)

Entitas adalah objek yang ada dalam dunia nyata dan dapat dibedakan dari objek lainnya. Entitas dapat berupa orang, benda, lingkungan, atau transaksi. Terdapat dua jenis entitas berdasarkan sifatnya:

##### a. Entitas Kuat (Strong Entity):

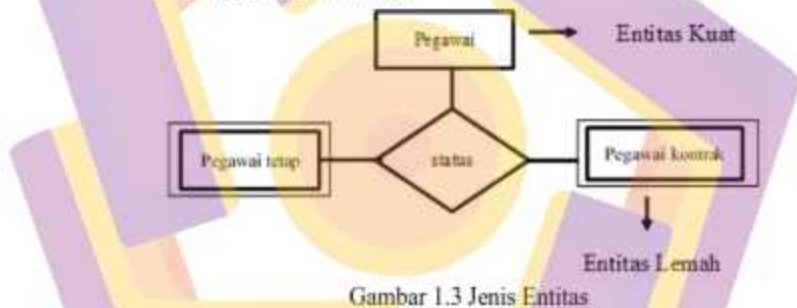
Entitas kuat digambarkan dengan kotak persegi panjang bergaris tunggal. Entitas ini memiliki identitas yang mandiri dan tidak bergantung pada entitas lain dalam basis data. Misalnya, dalam basis



data penjualan, entitas "Pelanggan" dapat dianggap sebagai entitas kuat karena memiliki atribut yang unik dan tidak bergantung pada entitas lain.

b. Entitas Lemah (Weak Entity):

Entitas lemah adalah entitas yang tidak dapat berdiri sendiri dan bergantung pada entitas kuat atau hubungan dengan entitas lain untuk mendapatkan identitasnya. Entitas lemah digambarkan dengan kotak persegi panjang bergaris ganda. Entitas ini membutuhkan kunci (key) dari entitas kuat sebagai identitasnya. Misalnya, dalam basis data pemesanan, entitas "Item Pemesanan" dapat dianggap sebagai entitas lemah karena membutuhkan entitas "Pemesanan" sebagai kunci utamanya.



Gambar 1.3 Jenis Entitas

2. Atribut (*attribute*)

Atribut adalah informasi yang terkait dengan entitas. Atribut juga dapat dianggap sebagai properti dari suatu objek. Dalam notasi ERD, atribut digambarkan sebagai lingkaran elips. Terdapat beberapa jenis atribut, antara lain:

- a. Atribut Sederhana: Atribut yang nilainya tunggal dan tidak dapat dipecah menjadi bagian-bagian yang lebih kecil.
- b. Atribut Komposit: Atribut yang terdiri dari beberapa bagian yang lebih kecil. Atribut komposit dapat dipecah menjadi atribut-atribut sederhana.

- c. Atribut Bernilai Tunggal: Atribut yang memiliki satu nilai pada setiap entitas.
- d. Atribut Bernilai Banyak: Atribut yang memiliki banyak nilai pada setiap entitas.
- e. Atribut Turunan: Atribut yang nilainya dihasilkan dari perhitungan atau operasi pada atribut-atribut lain dalam entitas.
- f. Atribut Identitas atau Atribut Kunci: Atribut yang digunakan untuk mengidentifikasi secara unik setiap entitas dalam basis data. Atribut ini penting dalam membentuk kunci primer (primary key) suatu entitas.



Gambar 1.4 Macam-macam atribut ERD

### 3. Relasi

Relasi merupakan hubungan antara entitas yang digambarkan dalam bentuk belah ketupat dalam notasi ERD. Terdapat tiga jenis relasi berdasarkan derajatnya:

- a. Relasi Unary: Relasi yang menghubungkan satu entitas dengan dirinya sendiri. Relasi ini menunjukkan hubungan atau ketergantungan antara atribut dalam satu entitas. Contoh dari relasi unary adalah hubungan antara entitas "Karyawan" dengan atribut "Atasan" yang juga merupakan entitas "Karyawan".
- b. Relasi Binary: Relasi yang menghubungkan dua entitas. Relasi ini

menggambarkan hubungan antara dua entitas dalam basis data. Contohnya adalah hubungan antara entitas "Pemesanan" dengan entitas "Produk" dalam basis data pemesanan.

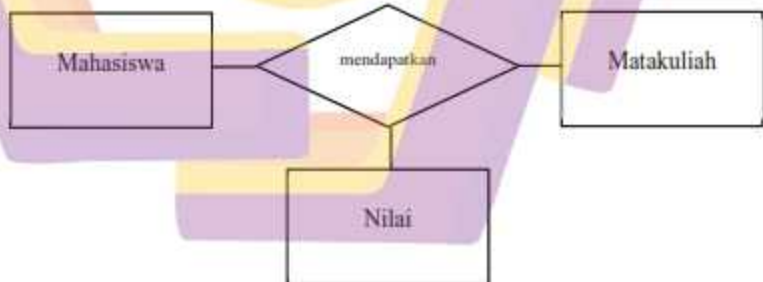
- c. Relasi Ternary: Relasi yang menghubungkan lebih dari dua entitas yang tidak sejenis. Relasi ini digunakan untuk menggambarkan hubungan kompleks antara tiga entitas atau lebih dalam basis data. Contohnya adalah hubungan antara entitas "Pemesanan", "Produk", dan "Pelanggan" dalam basis data pemesanan.



Gambar 1. 5 Entity Unary



Gambar 1. 6 Entity Binary



Gambar 1.7 Entity Ternary

### 1.3.9 PHP



Gambar 1. 8 Logo PHP

PHP (*PHP: Hypertext Preprocessor*) adalah bahasa pemrograman sisi server yang terintegrasi dengan HTML untuk membuat halaman web yang dinamis. Sintak dan perintah PHP akan dieksekusi di server, dan hasilnya akan dikirimkan ke browser dalam format HTML [14].

PHP merupakan bahasa pemrograman sisi server yang sangat kuat untuk membuat halaman web yang dinamis dan interaktif. PHP juga merupakan perangkat lunak *open source* yang dapat digunakan secara gratis, menjadi alternatif yang baik selain bahasa pemrograman sisi server lainnya [15].

PHP adalah salah satu bahasa pemrograman yang digunakan untuk mengembangkan aplikasi berbasis web. PHP adalah bahasa *scripting* tingkat tinggi yang terintegrasi dengan dokumen HTML [16].

#### 1.3.10 Javascript

JavaScript adalah bahasa *scripting* yang populer di internet dan dapat berfungsi pada sebagian besar browser terkenal seperti Internet Explorer, Mozilla Firefox, Netscape, dan Opera. JavaScript dapat dimasukkan ke dalam file HTML menggunakan tag `<script>` [15].

JavaScript merupakan bahasa pemrograman yang digunakan untuk membuat program-program yang membuat dokumen HTML menjadi lebih interaktif di dalam browser. Dengan JavaScript, dapat ditambahkan fungsionalitas



ke dalam halaman web sehingga halaman tersebut dapat berperilaku seperti sebuah program yang berjalan di antarmuka web [17].

### 1.3.11 *Framework Laravel*



Gambar 1. 9 Logo Laravel

Laravel adalah sebuah *framework open source* yang didasarkan pada bahasa pemrograman PHP dan dibangun dengan menggunakan konsep model-view-controller (MVC). *Framework* ini dikembangkan oleh Taylor Otwell pada tahun 2011 dan dilisensikan di bawah MIT *License*. Laravel juga menggunakan platform GitHub sebagai tempat berbagi kode. Tujuan utama pembuatan *framework* ini adalah meningkatkan kualitas perangkat lunak dengan biaya pengembangan yang rendah dan meningkatkan pengalaman kerja dengan sintak yang ekspresif, jelas, dan membantu menghemat waktu [18]

### 1.3.12 *Fitur Laravel*

Laravel menawarkan berbagai fitur yang tidak tersedia pada semua *framework*. Dengan adanya fitur-fitur ini, kita dapat melakukan berbagai hal. Berikut ini adalah beberapa fitur yang ada dalam *framework* Laravel [19].

#### 1. *Blade Template Engine*

Laravel memanfaatkan Blade sebagai *template engine*. Blade memungkinkan *layout* yang telah dirancang dapat digunakan dalam tampilan lain, sehingga memberikan konsistensi dalam desain dan struktur selama proses pengembangan. Blade memiliki beberapa keunggulan, yaitu

tidak membatasi pengembang untuk menggunakan kode PHP biasa di dalam tampilan dan desain tampilan Blade akan tetap di-cache sampai ada modifikasi.

## 2. *Routing*

Dalam Laravel, semua permintaan (*request*) diarahkan dengan bantuan rute (*routes*). *Routing* mengarahkan permintaan ke *controller* yang terkait. Adopsi *routing* ini dianggap dapat memudahkan pengembangan situs web dan meningkatkan kinerjanya. Terdapat tiga kategori *routing* yang umum digunakan dalam Laravel, yaitu *basic routing*, *route parameters*, dan *named routing*.

## 3. *Modularity*

Laravel menyediakan berbagai modul dan perpustakaan (*library*) yang terkait dengan Composer. Fitur ini membantu Anda untuk memperkaya dan meningkatkan fungsionalitas website yang dibangun, serta menyederhanakan proses pembaruan (*update*).

## 4. *Testability*

Laravel memiliki fitur pengujian (*testing*) yang komprehensif. *Framework* ini mendukung pengujian menggunakan PHPUnit dan *file phpunit.xml* yang dapat disesuaikan sesuai dengan kebutuhan aplikasi web yang sedang dikembangkan. Selain itu, Laravel juga menyediakan metode bantu yang nyaman dalam proses pengujian. Metode ini memungkinkan Anda untuk menguji website dengan cara yang ekspresif.

## 5. *Query Builder and ORM*

Laravel *database query builder* menawarkan antarmuka yang mudah digunakan untuk membuat dan menjalankan *query database*. Fitur ini memungkinkan Anda untuk melakukan berbagai operasi database di dalam website dengan lancar. Selain itu, Laravel *database query builder* juga mendukung berbagai sistem *database* yang berbeda.

## 6. *Authentication*

Laravel menyederhanakan implementasi *otentikasi* secara signifikan. Seluruh proses konfigurasi *otentikasi* dilakukan secara otomatis. *File*

konfigurasi *otentikasi* dapat ditemukan di 'config/auth.php'. Di dalam *file* ini, terdapat opsi-opsi *otentikasi* yang terdokumentasi dengan baik dan dapat disesuaikan sesuai kebutuhan Anda.

#### 7. *Schema Builder*

*Class Laravel Schema* menyediakan pengelolaan tabel yang bersifat independen terhadap jenis *database* yang digunakan. Skema ini kompatibel dengan berbagai jenis *database* yang didukung oleh Laravel dan memiliki antarmuka pemrograman aplikasi (API) yang konsisten di seluruh sistem.

#### 8. *Configuration Management Features*

Semua *file* konfigurasi Laravel disimpan di dalam direktori "config". Setiap opsi konfigurasi didokumentasikan dengan baik.

#### 9. *Email Class*

Laravel menyediakan API untuk beberapa perpustakaan email populer seperti SwiftMailer dengan koneksi ke SMTP, Postmark, Mailgun, SparkPost, Amazon SES, dan sendmail. Fitur ini memungkinkan untuk mengirimkan email melalui aplikasi lokal atau layanan cloud dengan mudah.

#### 10. Redis

Laravel menggunakan Redis sebagai penghubung antara sesi yang ada dengan *cache* umum (*general-purpose*). Redis terhubung langsung dengan sesi, dan Redis sendiri adalah aplikasi *open source* yang digunakan untuk menyimpan data dalam format *key-value*. Redis juga dikenal sebagai server struktur data yang mendukung penyimpanan key dengan tipe data seperti strings, hashes, lists, sets, dan sorted sets.

#### 11. Event and Command Bus

Laravel Command Bus menyediakan cara yang sederhana dan mudah dimengerti untuk mengelompokkan tugas-tugas yang diperlukan oleh aplikasi agar dapat dijalankan dengan efisien.

### 1.3.13 *Laravel Add-On Package*

Laravel menyediakan berbagai fitur tambahan yang dapat digunakan. Fitur-fitur ini meliputi routing, migration, test, views, dan beberapa fitur lainnya yang sangat berguna. Selain itu, terdapat banyak paket (*packages*) untuk Laravel yang dapat meningkatkan kecepatan, keamanan, dan kinerja aplikasi. Beberapa fitur yang sering digunakan antara lain[19]:

1. Spatie

Aturan dan izin (*permissions*) memainkan peran penting dalam banyak aplikasi web. Laravel menyediakan berbagai paket yang mendukung pembentukan aturan dan izin, yang dapat meningkatkan efektivitas kode program. Salah satu paket yang direkomendasikan adalah Spatie Roles & Permission. Beberapa keunggulan dari Spatie antara lain: manajemen peran (*roles*), izin (*permissions*), middleware, izin langsung (*direct permissions*), dukungan untuk instruksi Blade, dan perintah Artisan.

2. Laravel User Verification

*Package User Verification* memfasilitasi penanganan verifikasi pengguna dan validasi email. Fitur ini mencakup pembuatan dan penyimpanan *token* verifikasi untuk pengguna yang terdaftar, pengiriman dan pengaturan antrean email dengan tautan *token* verifikasi, penanganan *token* verifikasi, serta penandaan pengguna yang telah diverifikasi. *Package User Verification* juga menyediakan fungsionalitas lain, seperti pemeriksaan *middleware* pada rute.

3. Migration Generator

*Migration Generator* adalah sebuah paket Laravel yang memungkinkan untuk melakukan migrasi dari *database* yang sudah ada. Paket ini juga mencakup pembuatan indeks dan *foreign keys*. Dengan menggunakan paket ini dalam aplikasi Laravel, dapat melakukan proses migrasi untuk semua tabel yang ada dalam *database* dengan mudah.



### 1.3.14 MVC

MVC adalah suatu pendekatan pemrograman yang memisahkan tiga komponen utama dalam membangun aplikasi, yaitu manipulasi data, antarmuka pengguna (*user interface*), dan pengendalian aplikasi. Struktur MVC terdiri dari tiga komponen penting, yaitu Model, View, dan Controller [20].

#### 1. Model

Model merupakan bagian yang secara langsung berinteraksi dengan basis data untuk melakukan manipulasi data, menangani validasi dari *controller*, namun tidak memiliki keterkaitan langsung dengan bagian view.

#### 2. View

View merupakan bagian yang menerima dan mempresentasikan data kepada *user*.

#### 3. Controller

Controller berperan dalam mengatur hubungan antara model dan view. Tugas controller adalah menerima permintaan (*request*) dan data, kemudian menentukan proses apa yang akan dilakukan selanjutnya.

### 1.3.15 Scrum



Gambar 1. 10 Alur Scrum [21]

Scrum adalah suatu framework kerja di mana individu dapat mengatasi permasalahan yang kompleks dan selalu berubah, sambil menghasilkan produk dengan nilai maksimal secara kreatif dan efisien. Scrum memiliki tiga pilar penting yaitu transparansi, inspeksi, dan adaptasi. Scrum menyediakan empat acara yang dibungkus dalam sebuah sprint yang dilakukan untuk inspeksi dan adaptasi. Scrum terdiri atas beberapa tim yang terdiri dari Product Owner, Tim Pengembang, dan Scrum Master [22].

1. Product Owner

Product Owner memiliki tanggung jawab untuk memaksimalkan nilai produk dan hasil yang dihasilkan oleh tim pengembang. Mereka bertanggung jawab dalam mengelola Product Backlog. Sebagai individu tunggal, Product Owner dapat mewakili aspirasi dari komite dan mengatur prioritas dalam Product Backlog. Namun, jika ada perubahan prioritas, hal tersebut harus melalui Product Owner [22].

2. Tim Pengembang

Tim pengembang terdiri dari berbagai peran yang diperlukan dalam proses pembangunan produk perangkat lunak. Beberapa peran umumnya meliputi *software developer*, *ui designer*, *interaction designer*, dan *quality assurance*. Tim ini bekerja sama secara kolaboratif untuk membangun produk sesuai dengan prioritas pekerjaan yang telah disepakati bersama Product Owner [23].

3. Scrum Master

Scrum Master adalah peran yang difokuskan pada kesuksesan implementasi proses Scrum dalam pengembangan perangkat lunak. Tugas Scrum Master adalah membantu Product Owner dan tim pengembangan bekerja secara kolaboratif dalam kerangka kerja Scrum [23].

### 1.3.16 Pengujian Black Box

Black box testing adalah metode pengujian yang dilakukan hanya dengan mengamati hasil eksekusi melalui data uji dan memeriksa fungsionalitas dari perangkat lunak tanpa mengetahui secara detail bagaimana proses pengujian tersebut berlangsung. Dalam konteks pengujian perangkat lunak, hal ini berarti

penguji hanya melakukan evaluasi berdasarkan tampilan luar (interface) dan fungsionalitas perangkat lunak tanpa mengetahui secara detail bagaimana proses tersebut berjalan atau berinteraksi dengan komponen internal. Dengan kata lain, penguji hanya mengetahui input dan output yang dihasilkan oleh perangkat lunak tanpa mengetahui implementasi internalnya.

Pengujian black box pada perangkat lunak dirancang untuk mendeteksi kesalahan yang mungkin terjadi pada persyaratan fungsional tanpa mempertimbangkan kerja internal program tersebut. Teknik pengujian black box berfokus pada informasi yang terkait dengan domain perangkat lunak itu sendiri. Hal ini dilakukan dengan menguji test case dan mempartisi domain input dan output program serta memberikan cakupan pengujian yang menyeluruh.

Dalam pengujian black box, fokus utama adalah pada spesifikasi fungsional dari perangkat lunak. Tester dapat menentukan serangkaian kondisi input dan melakukan pengujian pada spesifikasi fungsional program untuk memastikan bahwa masukan dan keluaran perangkat lunak sesuai dengan persyaratan yang dibutuhkan [24].

Dengan demikian, pengujian black box berfokus pada pengujian perangkat lunak dari segi spesifikasi fungsional tanpa melakukan pengujian terhadap desain dan kode program untuk mengetahui apakah fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang telah ditentukan.