

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Sebelumnya**

Dengan pendukung yang diperoleh dari penelitian sebelumnya cukup relevan mencakup pembahasan pada judul topik ini. Sebagai referensi, tabel peneliti dibawah ini menjadi bahan pertimbangan mengenai website pendaftaran penitipan kucing yang akan dibuat dengan sebuah sistem keamanan.

Pertama, penelitian yang berjudul “Pengembangan Aplikasi Evaluasi Dosen Berbasis Android dengan Keamanan Json Web token (JWT)” dilakukan oleh Eko Junirianto pada tahun 2019. Penelitian ini melakukan pengujian dengan *Black Box* dalam aplikasi evaluasi dosen, mulai dari tampilan halaman *login* hingga keluar dari aplikasi. Hasil pengujian tersebut membuktikan bahwa seluruh proses dapat dilaksanakan dengan baik dan mencapai keberhasilan. Berdasarkan pelaksanaan dan uji coba aplikasi, dapat disimpulkan bahwa pembangunan layanan web efektif untuk menyatukan sistem aplikasi evaluasi dosen berbasis *Android* dengan infrastruktur yang telah ada. Penerapan keamanan JWT dalam layanan web juga berkontribusi untuk menjamin keamanan pertukaran data yang lebih terjamin[8].

Kedua, penelitian yang berjudul “Pengamanan Restful API menggunakan JWT untuk Aplikasi Sales Order” dilakukan oleh Edy, Ferdiansyah, Wahyu Pramusinto, Sejati Waluyo pada tahun 2019. Penelitian ini mendapatkan hasil dengan metode *waterfall* dalam penerapan autentikasi JSON Web Token pada RESTful API meningkatkan tingkat keamanan aplikasi, mengingat aplikasi hanya dapat diakses dengan menggunakan token yang valid[4].

Ketiga, penelitian yang berjudul “Sistem Presensi Online Menggunakan Arsitektur Pengembangan Perangkat Lunak Model-View-Viewmodel” dilakukan oleh Adha Setiawan Wiyana, M. Ihsan Alfani

Putera, Sri Rahayu 8 Natasia pada tahun 2021. Penelitian ini mendapatkan hasil dengan metode *Personal Extreme Programming (PXP)*. penggunaan JWT sebagai *implementasi* IMEI, *get location* dan *JWT Authentication* dan berguna sebagai penggunaan sistem kehadiran dibatasi hanya kepada pihak yang memiliki kepentingan, dan menghasilkan data yang akurat[9].

Keempat, penelitian yang berjudul “Perancangan Back-End Server Menggunakan Arsitektur Rest dan Platform Node.JS” dilakukan oleh Ahsan Mubariz, Dahlia Nur, Eddy Tungadi, Muhammad Nur Yasir Utomo pada tahun 2020. Penelitian ini mendapatkan hasil dengan pengujian *Black Box* testing dari hasil perancangan *backend* ini membuat peningkatan respons waktu pada situasi pengujian terjadi ketika diakses oleh pengguna virtual yang jumlahnya meningkat, dengan rata-rata peningkatan sekitar 52% pada skenario yang telah diuji. Ini menunjukkan bahwa sistem yang dikembangkan menunjukkan performa yang lebih unggul dibandingkan dengan sistem yang sudah ada, seperti yang terlihat dari perbedaan waktu *respons* rata-rata dalam skenario pengujian sekitar 3222,5 ms[10].

Kelima, penelitian yang berjudul “Perancangan Web Service Berbasis REST API Untuk Aplikasi Penerimaan Peserta Didik Baru” dilakukan oleh Ryan Afrizal, Fitriyani pada tahun 2020. Penelitian ini mendapatkan hasil dengan metode *waterfall* dari hasil perancangan ini membuat suatu sistem aplikasi yang beroperasi baik melalui *platform* web maupun *mobile*, dengan tujuan membantu calon siswa atau orang tua calon siswa dalam melaksanakan proses pendaftaran peserta didik baru[11].

Keenam, penelitian yang berjudul “IMPLEMENTASI WEB SERVICE DENGAN METODE REST API UNTUK PENDAFTARAN SEKOLAH DI TAMAN KANAK-KANAK ANNIDA TANGERANG SELATAN” dilakukan oleh Abiyyu Naufal Habib, Haris Munandar pada tahun 2022. Penelitian ini mendapatkan hasil dengan metode *REST API* dan *Algoritma* AES dari penerapan ini penelitian dengan penerapan *algoritma* AES akan menjaga keamanan data peserta didik yang mendaftar ketika disimpan dalam basis data yang telah dienkrpsi[12].

Ketujuh, penelitian yang berjudul “*Furriend* Sebagai Pengembangan Usaha Pet Shop and Care Digital Berbasis Website” dilakukan oleh Shandy 9 Ilham Alamsyah, Fisma Meividianugraha Subani, Alvin Febrianto, Refila Dyah Ghizanda Wardoyo, Dodik Arwin Dermawan pada tahun 2024. Penelitian ini mendapatkan hasil dengan menerapkan metode Penelitian dan Pengembangan (R&D), penelitian ini memperlihatkan bahwa situs web telah terbukti menjadi layanan sistem informasi dan alat pemasaran yang banyak digunakan oleh masyarakat. Penggunaan teknologi digital secara umum sangat mendukung kemajuan suatu bisnis, dan implementasi *Furriend* membuktikan memberikan bantuan yang signifikan bagi usaha *pet shop* dalam mengelola layanan dengan cara yang lebih *modern*[13].

Kedelapan, penelitian yang berjudul “Aplikasi Penitipan Hewan Peliharaan di Pet Shop Menggunakan Fitur Pembayaran Virtual Berbasis iOS” dilakukan oleh Rynetta Rizky Nugroho, Silvana Rasio Henim pada tahun 2020. Penelitian ini mendapatkan hasil dengan pengujian *Black Box* dari penelitian ini menunjukkan aplikasi penitipan hewan peliharaan di pet shop yang berbasis iOS telah berfungsi dengan baik sesuai harapan, terbukti dengan pengujian yang dilakukan menunjukkan bahwa semua fitur berjalan dengan baik dan sesuai fungsinya[14].

Kesembilan, penelitian berjudul “Task-Technology Fit (TTF) dan Unified Theory of Acceptance and Use of Technology” dilakukan oleh Fitrianto Puja Kesuma, Dedy Syamsuar pada tahun 2021. Penelitian ini menunjukkan (UTAUT) ini menyebarkan kuesioner melalui formulir *Google* yang dapat diakses oleh peserta secara *online*. Data *screening* dilakukan setelah data hasil kuesioner secara menyeluruh dikumpulkan. Periksa data yang tidak lengkap dan tanggapan yang tidak relevan. Pada penelitian ini, penggunaan TTF mengacu pada kemampuan teknologi yang digunakan untuk memenuhi atau cocok dengan tugas yang dilakukan oleh orang. Karena itu, dua variabel ini akan memiliki hubungan langsung dengan kesesuaian tugas dan teknologi. Urutan menunjukkan hubungan yang positif antara atribut tugas dan atribut teknologi dengan kesesuaian

tugas dan teknologi. Apabila seseorang (pengguna) tidak sesuai dengan tugas yang sedang dikerjakannya, mereka tidak akan menggunakan teknologi[15].

Kesepuluh, penelitian berjudul “ANALISIS KEAMANAN APLIKASI WEB OAS POLITEKNIK LP3I JAKARTA MENGGUNAKAN ACUNETIX WEB VULNERABILITY’ dilakukan oleh Hariyanto dan Marini pada tahun 2023. Penelitian ini menunjukkan Penjelasan secara menyeluruh terhadap semua kegiatan yang telah dilakukan sebelumnya mengenai aplikasi web OAS menggambarkan temuan sejumlah kerentanan atau celah keamanan yang signifikan. Dalam konteks ini, pihak yang berkepentingan diharapkan memberikan perhatian khusus terhadap keamanan aplikasi web OAS, terutama jika aplikasi tersebut akan digunakan untuk mengelola seluruh kegiatan administrasi di Politeknik LP3I Jakarta. Peningkatan keamanan aplikasi menjadi suatu keharusan yang mendesak, dan tugas ini dianggap sebagai tantangan yang kompleks. Temuan kerentanan menyoroti pentingnya mengatasi masalah keamanan agar aplikasi dapat beroperasi secara aman dan handal dalam mendukung kegiatan administratif di institusi tersebut[16].

Pada Tabel 2.1 merupakan ringkasan yang dilakukan oleh penulis terhadap penelitian yang pernah dilakukan sebelumnya. Dalam ringkasan terdapat penjelasan mengenai judul penelitian, masalah yang diteliti, metode yang digunakan serta hasil yang dilakukan oleh beberapa peneliti terhadap penelitian yang pernah dilakukan.

Tabel 2. 1 Ringkasan Penelitian Sebelumnya

No	Judul	Penulis	Tahun	Studi kasus	Metode/ Algoritma/Pengujian	Hasil	Perbedaan dengan penelitian yang dilakukan
1	Pengembangan Aplikasi Evaluasi Dosen Berbasis Android dengan Keamanan Json Web token (JWT)	Eko Junirianto	2019	POLITEKNIK PERTANIAN NEGERI SAMARINDA	<i>Black Box</i>	Siswa telah melakukan evaluasi terhadap dosen dengan melakukan pengujian terhadap berbagai tahap dalam aplikasi evaluasi dosen, mulai dari tampilan halaman login hingga keluar dari aplikasi. Hasil pengujian tersebut	Pada penelitian sebelumnya menggunakan JWT pengembangan aplikasi penilaian kinerja pegawai yang dilakukan dengan sistem android. Sedangkan, pada penelitian ini menggunakan JWT sebagai keamanan pada data dan juga pembayaran pendaftaran

						membuktikan bahwa seluruh proses dapat dilakukan dengan baik dan berhasil. Berdasarkan implementasi dan pengujian aplikasi, dapat	penitipankucing dengan sistem website.
--	--	--	--	--	--	---	--

						<p>disimpulkan bahwa pengembangan layanan web dapat digunakan untuk mengintegrasikan sistem aplikasi evaluasi dosen berbasis Android dengan sistem yang sudah ada. Penggunaan keamanan JWT dalam layanan web juga membantu memastikan keamanan pertukaran data yang lebih terjamin.</p>	
--	--	--	--	--	--	---	--

2	Pengamanan Restful API menggunakan JWT untuk Aplikasi Sales Order	Edy, Ferdiansyah, Wahyu Pramusinto, Sejati Waluyo	2019	Aplikasi sales order	Waterfall	Penerapan autentikasi JSON Web Token pada RESTful API meningkatkan tingkat keamanan aplikasi, karena aplikasi tidak dapat diakses tanpa menggunakan token.	Pada penelitian sebelumnya menggunakan JWT untuk aplikasi sales order dengan menggunakan metode waterfall. Sedangkan, pada penelitian ini menggunakan JWT sebagai keamanan pada penitipan kucing dengan metode XP.
3	Sistem Presensi Online Menggunakan Arsitektur	Adha Setiawan Wiyana, M. Ihsan Alfani Putera, Sri	2021	PT. Lintasmaya Network Samarinda	Personal Extreme Programming (PXP)	Sistem kehadiran hanya dapat diakses oleh mereka yang memiliki kepentingan saja dengan hasil data yang akurat.	Pada penelitian sebelumnya menggunakan JWT sebagai implementasi IMEI, get



4	Pengembangan Perangkat Lunak Model-View-Viewmodel	Adha Setiawan Wiyana, M. Ihsan Alfani Putera, Sri,Rahayu Natasia	2020	Sistem Pendaftaran Ujian Masuk Politeknik Negeri Ujung Pandang	Black Box Testing.		location dan JWT Authentication. Sedangkan, pada penelitian ini menggunakan JWT sebagai keamanan pada penitipan kucing dengan metode XP dan m,enjaga sistem keamanan untuk pembayaran.
---	---	---	------	--	--------------------	--	--

5	Perancangan Back-End Server Menggunakan Arsitektur Rest dan Platform Node.JS	Ahsan Mubariz, Dahlia Nur, Eddy Tungadi, Muhammad Nur Yasir Utomo	2020	Pendaftaran peserta didik baru (PPDB)	Model pengembangan software dengan metode waterfall	<p>Hasil perancangan backend membuat peningkatan waktu respons pada kondisi pengujian dengan akses dari virtual users yang meningkat dengan rata- rata peningkatan sebesar 52% pada skenario yang sudah di testing. Adapun sistem yang dibuat memiliki kinerja yang lebih unggul daripada sistem yang sudah ada berdasarkan hasil pengujian, dengan selisih waktu respons rata-rata dari skenario pengujian sebesar</p>	<p>Pada penelitian sebelumnya menggunakan Node.js sebagai perangkat lunak untuk pengembangannya dan juga menjadikan mongodb sebagai sistem database. sedangkan, pada penelitian ini menggunakan perangkat lunak laravel dan mysql sebagai database.</p>
---	--	--	------	---------------------------------------	---	---	---

						3222,5 ms.	
--	--	--	--	--	--	------------	--

6	IMPLEMENTASI WEB SERVICE DENGAN METODE REST API UNTUK PENDAFTARAN SEKOLAH DI TAMAN KANAK-KANAK ANNIDA TANGERANG SELATAN	Abiyyu Naufal Habib, Haris Munandar	2022	Taman Kanak-Kanak Annida	Metode REST API dan Algoritma AES	Penelitian dengan menggunakan algoritmaAES akan membuat datapeserta didik yang mendaftar menjadi aman ketika disimpan dengan database yang sudah di enkripsi.	Pada penelitian sebelumnya peneliti menggunakan implementasi dengan menggunakan bahasa pemrograman java, sedangkan penelitian ini menggunakan bahasa pemrograman PHP.
---	---	-------------------------------------	------	--------------------------	-----------------------------------	---	---

7	<p>Furriend Sebagai Pengembangan Usaha Pet Shop and Care Digital Berbasis Website</p>	<p>Shandy Ilham Alamsyah, Fisma Meividianugraha Subani, Alvin Febrianto, Refila Dyah Ghizanda Wardoyo, Dodik Arwin Dermawan</p>	2024	Studi di Kota Pasuruan	<p>Penelitian memakai metode R&amp;D (Research and Development).</p>	<p>Hasil penelitian menunjukkan bahwa Website terbukti sebagai layanan sistem informasi dan media pemasaran yang sering digunakan oleh masyarakat, usaha digital sangat mendukung kemajuan sebuah usaha, penerapan Furriend sangat membantu usaha pet shop dalam menangani mekanisme pelayanan secara konvensional.</p>	<p>Pada penelitian sebelumnya peneliti menggunakan metode R&amp;D sebagai solusi dalam bentuk ide maupun produk serta mengukur keefektifannya sebagai bukti kesuksesan penelitian, sedangkan pada penelitian ini dengan Extreme Programming membuat kolaborasi kuat antara para pengembang perangkat lunak, pelanggan, dan pemangku kepentingan lainnya.</p>
---	---	---	------	------------------------	--	---	--

8	Aplikasi Penitipan Hewan Peliharaan di Pet Shop Menggunakan Fitur Pembayaran Virtual Berbasis iOS	Rynetta Rizky Nugroho , Silvana Rasio Henim	2020	Pet shop	<i>Black Box testing</i>	Hasil dari penelitian menggunakan <i>Black Box testing</i> karena seluruh fitur sudah berjalan dengan semestinya berdasarkan pengujian yang telah dilakukan.	Penelitian sebelumnya menggunakan bahasa pemrograman Swift dengan IDE Xcode untuk dikembangkan oleh Applesedangkan, penelitian saat ini menggunakan PHP dengan sintaksis yang mirip pada bahasa pemrograman C dan dieksekusi langsung oleh server web tanpa perlu dikompilasi.
---	---	--	------	----------	--------------------------	--	--

9	<i>Task-Technology Fit (TTF) dan Unified Theory of Acceptance and Use of Technology (UTAUT): Analisis Model Penerimaan Teknologi di Perguruan Tinggi</i>	Fitrianto Puja Kesuma, Dedy Syamsuar	2021	Perguruan Tinggi	TTF dan UTAUT	Penelitian ini menyebarkan kuesioner melalui formulir Google yang dapat diisi secara online oleh peserta. Data screening dilakukan setelah semua data dari hasil kuesioner dikumpulkan secara menyeluruh. Periksa kekurangan data dan tanggapan yang tidakterlibat dilakukan.	Penelitian sebelumnya menggunakan penggabungan TTF dan UTAUT, sedangkan pada penelitian ini menggunakan PISS ( <i>Plan, Implement, Survey, dan Summarize</i> ).
---	--	--------------------------------------	------	------------------	---------------	---	---

10	ANALISIS KEAMANAN APLIKASI WEB OAS POLITEKNIK LP3I JAKARTA MENGGUNAKAN ACUNETIX WEB VULNERABILITY	Hariyanto, Marini	2023	POLITEKNIK LP3I JAKARTA	<i>Black box Testing</i>	Penjelasan secara menyeluruh terhadap semua kegiatan yang telah dilakukan sebelumnya mengenai aplikasi web OAS menggambarkan temuan sejumlah kerentanan atau celah keamanan yang signifikan. Dalam konteks ini, pihak yang berkepentingan diharapkan memberikan perhatian khusus terhadap keamanan aplikasi web OAS, terutama jika aplikasi tersebut akan digunakan untuk mengelola seluruh kegiatan	Pada sistem sebelumnya melakukan penerapan analisis keamanan aplikasi web OAS sedangkan dengan penelitian ini menggunakan JWT.
----	---	-------------------	------	-------------------------	--------------------------	--	--



						<p>administrasi di Politeknik LP3I Jakarta.</p> <p>Peningkatan keamanan aplikasi menjadi suatu keharusan yang mendesak, dan tugas ini dianggap sebagai tantangan yang kompleks.</p> <p>Temuan kerentanan menyoroti pentingnya mengatasi masalah keamanan agar aplikasi dapat beroperasi secara aman dan handal dalam mendukung kegiatan administratif di institusi tersebut.</p>	
--	--	--	--	--	--	--	--

## 2.2 Landasan Teori

### 2.2.1 Web Browser

*Web browser* merupakan perangkat lunak yang berperan sebagai perantara antara pengguna dan *Web browser*, berfungsi sebagai akses pintu untuk menampilkan dokumen atau informasi web yang diperoleh dari *Web browser*. *Browser* juga sebagai perangkat lunak yang dijalankan pada komputer pemakai (*user*) yang menampilkan dokumen atau informasi web yang diambil dari web *server*. *Browser* merupakan jenis perantara pengguna dengan *server* web yang paling sering digunakan. Bisa menyebut *Web Browser* sebagai pintu ruang informasi yang kita kenal sebagai *world wide web*, bukan internet. Semua hal yang user lakukan, apakah itu streaming film, browsing, dan beberapa hal lainnya, kamu melakukannya di web, bukan di internet. Internet adalah jaringan perangkat komputer, router, dan *server* di seluruh dunia. Sementara web atau *World Wide Web* seperti perangkat lunak yang berjalan di internet, ia tidak memiliki eksistensi fisik. Web digunakan untuk mengakses semua situs web dan layanan dengan bantuan berbagai protokol. Yang harus kamu lakukan adalah mengetikkan URL (*Unique Resource Locators*) di *address bar*, dan *browser* Anda akan melakukan sisanya untuk menampilkan situs web yang menyertakan hal-hal teknis seperti menghubungi *server* DNS untuk mendapatkan alamat IP situs web[17].

### 2.2.2 REST API (Representational State Transfer Application Programming Interface)

REST API, singkatan dari *Representational State Transfer Application Programming Interface*, merupakan suatu arsitektur dalam dunia *Interface* Pemrograman Aplikasi (API) yang umumnya diterapkan dalam pengembangan layanan web. Melalui pemanfaatan REST API, berbagai sistem dapat berkomunikasi secara efektif menggunakan pendekatan komunikasi yang disebut REST. Proses komunikasi ini memungkinkan sistem-sistem tersebut untuk

menyampaikan dan mengakses informasi dengan lebih efisien. Cara kerja REST API melibatkan pemetaan sumber daya yang disimpan dalam database sistem melalui penggunaan *endpoint* API. Ketika akses terhadap data diperlukan, REST API menggunakan perintah permintaan HTTP, seperti *GET* untuk mendapatkan sumber daya, *POST* untuk menciptakan sumber daya baru, *PUT* untuk memperbarui sumber daya, atau *DELETE* untuk menghapus sumber daya. Format data yang umum digunakan untuk merepresentasikan informasi dalam REST API adalah JSON, memberikan kemudahan dalam proses pembacaan dan pemahaman data. Secara keseluruhan, REST API dapat dianggap sebagai pendekatan yang memfasilitasi interaksi yang efisien antar sistem menggunakan protokol HTTP, dengan menerapkan representasi sumber daya yang jelas, seperti JSON, untuk mendukung komunikasi terstandarisasi dan efektif dalam pengembangan layanan web[18].

### 2.2.3 JWT (JSON Web Token)

JSON Web Token (JWT) merupakan sebuah string token yang digunakan untuk keperluan otentikasi dan pertukaran data. JWT terdiri dari tiga komponen utama, yakni *header*, *payload*, dan *signature*, serta terdapat dua jenis token, yaitu token pembawa (*bearer* token) dan token pemegang kunci (*key* token), dengan dua skema berdasarkan tujuan, yaitu token identitas dan token akses. Prinsip kerja JWT mirip dengan penggunaan *password* pada proses otentikasi, dimana *server* menghasilkan token JWT setelah pengguna berhasil login, kemudian token tersebut disimpan di *local storage* atau *cookies* di *browser* pengguna. Digunakan untuk memberikan izin akses ke halaman tertentu, pengguna harus mengirim token tersebut sebagai bukti bahwa mereka telah berhasil *login*, menjadikan JWT sebagai elemen kunci dalam proses otentikasi dan pengaturan izin pada aplikasi web. JWT, juga dikenal sebagai Token Web JSON, memberikan solusi potensial untuk menggantikan penggunaan nama pengguna dan kata sandi

konvensional dengan token *string* acak panjang yang dirancang untuk mengautentikasi sistem serta memfasilitasi pertukaran informasi. Keunggulan *implementasi* JWT terletak pada efisiensi pengaksesan data tanpa memerlukan koneksi langsung ke *database* atau layanan, menjadikannya pilihan umum dalam pengembangan layanan API. Sebagai alat keamanan, JWT terdiri dari tiga bagian, yaitu *header*, *payload*, dan *signature*, di mana *header* berisi metadata, *payload* berisi klaim atau informasi terkait, dan *signature* digunakan untuk memverifikasi integritas dan keabsahan token. Melalui cara kerjanya, klien memulai dengan permintaan masuk atau pendaftaran menggunakan metode *POST*, *server* merespons dengan membuat rahasia JWT yang diterima oleh *browser*, dan klien mengirimkan *Header* Otorisasi JWT kepada *server* untuk memverifikasi *signature* JWT dan mendapatkan informasi pengguna[19].

Berikut adalah karakteristik dari JSON Web Token (JWT):

### 1. *Header*

*Header* umumnya terdiri dari dua elemen: jenis token, JWT, dan algoritma *penghashing* yang digunakan, seperti HMAC SHA256 atau RSA.

### 2. *Payload*

Bagian kedua dari token adalah *payload*, yang berisi klaim. Klaim adalah pernyataan tentang suatu entitas (biasanya, pengguna) dan metadata tambahan. Ada tiga jenis klaim: *reserved*, *public* dan *private*.

### 3. *Signature*

Untuk *menciptakan* bagian *signature*, kita perlu mengambil bagian *header* yang telah dienkripsi, *payload* yang telah dienkripsi, kunci rahasia, algoritma yang ditetapkan dalam *header*, dan kemudian menandatangani. Tanda tangan digunakan untuk memverifikasi bahwa pengirim adalah pengirim yang sah dan

untuk memastikan bahwa pesan yang dikirim tidak mengalami perubahan.

#### 2.2.4 Two Factor Authenticatin (2FA)

Metode *autentikasi* menggunakan faktor tambahan yang terpisah dari faktor utama, yaitu *password*. Faktor tambahan ini digunakan untuk membuktikan keaslian identitas pengguna yang masuk. Teknologi *Two-Factor Authentication* (2FA) bertujuan mengurangi risiko akses oleh pihak yang bukan pengguna asli, terutama jika mereka berhasil mendapatkan password dengan cara apapun[6]. Implementasi 2FA meminta pengguna untuk menyediakan beberapa faktor tambahan seperti:

1. Informasi yang dikenal oleh pengguna (kata sandi),
2. Benda yang dimiliki oleh pengguna (seperti ponsel atau perangkat keras lainnya).

Cara yang paling sering digunakan untuk menerapkan 2FA termasuk:

1. SMS (*Short Message Service*) merupakan metode 2FA yang paling umum digunakan, di mana sistem mengirimkan kode unik berupa enam digit melalui SMS. Pendekatan ini muncul karena sebagian besar penduduk Amerika telah menggunakan ponsel, mencapai 99%. Dalam metode ini, server terhubung dengan modul GSM untuk mengirimkan kode unik ke nomor pengguna yang sedang mencoba untuk login ke sistem.
2. TOTP (*TimedOne Time Password*) Metode ini biasanya dimulai dengan *synchronize generator* kode unik dari *provider* tertentu, contoh *Google authentication* dan *Microsoft authentication*. Aplikasi *authentication* ini menggunakan kombinasi potongan *timestamp*, nilai hashing, dan potongan kode dari *verification*. Keuntungan menggunakan metode ini pengguna tidak perlu jaringan selular lagi untuk *authenticating*.

### 2.2.5 MySQL

MySQL adalah sebuah perangkat lunak *database* sumber terbuka yang umumnya dipakai untuk mengelola basis data menggunakan bahasa SQL. MySQL merupakan jenis *database relational*, di mana data disimpan membuat suatu struktur tabel yang saling terkait. Kelebihan penyimpanan data dalam *database* adalah data dapat disusun secara tabular, memudahkan dalam penyimpanan dan penampilan data. MySQL sering dikombinasikan dengan *web server* untuk mempermudah proses instalasi. Secara keseluruhan, perangkat lunak MySQL sering digunakan untuk mengelola basis data dalam pengembangan aplikasi yang membutuhkan sistem basis data. SQL adalah kependekan dari *Structured Query Language*, sebuah bahasa terstruktur yang dirancang khusus untuk mengelola *database*. Bahasa ini tidak hanya memungkinkan pengguna untuk mengelola data dalam *database*, tetapi juga memberikan kemampuan untuk mengetahui lokasi atau susunan informasi yang diinginkan[20].

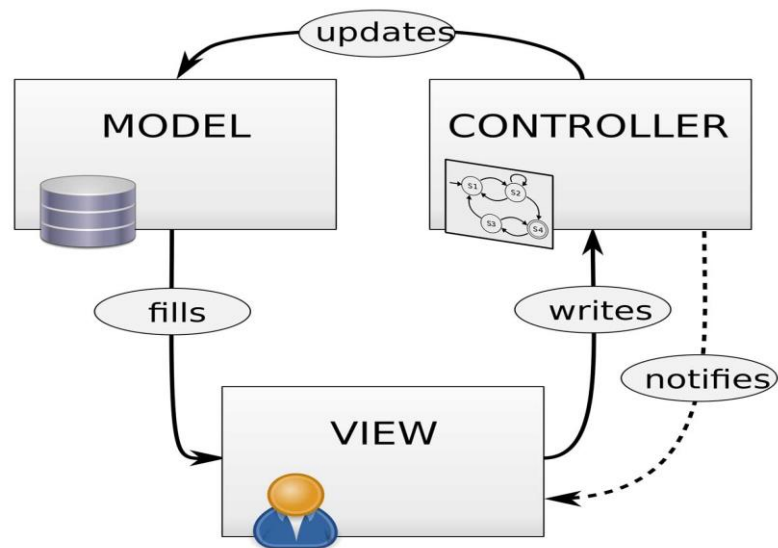
### 2.2.6 XAMPP

XAMPP merupakan pilihan yang populer bagi mereka yang ingin belajar tentang pengembangan web dan melakukan pengujian sebelum mengimplementasikannya di *server* produksi. XAMPP memberikan kemudahan dalam menjalankan lingkungan web secara lokal dengan menyediakan solusi *all-in-one* untuk mengelola *server* MySQL dan komponen-komponen lainnya[21].

### 2.2.7 Laravel

*Laravel* merupakan sebuah kerangka kerja yang populer untuk mengembangkan aplikasi web yang stabil, aman, dan dapat dengan mudah diperbesar. Dengan fitur-fitur lengkap dan pendekatan yang kuat, *Laravel* meningkatkan produktivitas pengembang dan memungkinkan penggunaan optimal bahasa pemrograman PHP yang cukup terbilang dinamis[22]. *Laravel* merupakan sebuah kerangka

kerja web yang dibangun dengan menggunakan bahasa pemrograman PHP dan bersifat sumber terbuka (*open source*). Kerangka kerja ini pertama kali diperkenalkan pada tahun 2011, mengadopsi pola pengembangan yang dikenal sebagai MVC (*Model, View, dan Controller*)[23].



Gambar 2. 1 MVC *Laravel*

(Sumber:<https://i0.wp.com/informatika.uc.ac.id/wpcontent/uploads/2016/12/MVC-Diagram.jpg> )

MVC adalah sebuah pola arsitektur yang diterapkan dalam pengembangan aplikasi. Pola arsitektur ini terbagi menjadi beberapa komponen dengan masing-masing memiliki nama yang disebut:

#### 2.2.7.1 Model

*Model digunakan* untuk memanipulasi data yang terkait dengan basis data, melibatkan berbagai proses seperti penyisipan (*insert*), pembaruan (*update*), penghapusan (*delete*), dan tindakan lainnya.

#### 2.2.7.2 View

*View digunakan* untuk menampilkan data di layar *browser*, umumnya dalam bentuk *file-file* HTML, CSS, dan *JavaScript*.

### 2.2.7.3 Controller

*Controller* berfungsi *sebagai* tempat untuk menuliskan logika aplikasi dan umumnya digunakan sebagai penghubung antara *model* dan *view*.

Dengan menerapkan prinsip-prinsip MVC pada *Laravel*, pengembangan aplikasi menjadi lebih mudah bagi para pengembang. Hal ini karena para pengembang dapat fokus pada tugas-tugas tertentu sesuai dengan tanggung jawab pekerjaan mereka masing – masing[19]. Dalam *laravel*, terdapat file bawaan seperti *vendor*, dan untuk menjalankannya, diperlukan koneksi internet untuk mengunduh pustaka yang diperlukan selama proses instalasi. Berikut adalah elemen dasar dari *Laravel*:

#### a) *Artisan*

*Artisan* merupakan antarmuka baris perintah atau *command line* yang beroperasi melalui terminal, menyediakan berbagai perintah yang berguna dalam proses pengembangan dan pembuatan aplikasi. Salah satu fitur yang disediakan oleh PHP *Artisan* adalah *php artisan serve*. Fungsi dari perintah ini adalah untuk menjalankan *website* yang telah dibuat tanpa memerlukan *server* web lokal tambahan.

#### b) *Routing*

*Routing* adalah suatu proses yang dirancang untuk memastikan bahwa suatu elemen yang diinginkan dapat mencapai tujuannya. Melalui penggunaan *routing*, dapat ditetapkan halaman atau destinasi yang akan ditampilkan ketika diakses oleh pengguna.

#### c) *Controller*



*Controller* merupakan suatu proses yang berfungsi untuk menangkap permintaan, melakukan inisialisasi, dan memanggil model agar dapat dikirimkan ke *view*.

#### 1. *View (Blade Templating)*

*Blade* adalah mesin templat bawaan dari *Laravel* yang memudahkan penggunaan kode untuk menghasilkan output dalam *framework Laravel*. *Blade* menyediakan sintaksis yang lebih sederhana dan efisien untuk membangun struktur dalam aplikasi *Laravel*.

#### 2. *Middleware*

*Middleware* adalah perantara antara permintaan yang diterima dan *controller* yang dituju.

#### 3. *Session*

*Session* adalah metode yang digunakan untuk menyimpan data di *server* dan informasi tersebut dapat digunakan di beberapa halaman, termasuk halaman itu sendiri.

#### 4. *Migration*

*Migration* adalah sebuah fitur dalam *Laravel* yang berfungsi sebagai Sistem kontrol versi untuk *database*. Penggunaan *migration* mempermudah penulis dalam pembuatan tabel data. *Migration* menciptakan atau menghasilkan *file-file* dalam sistem kontrol migrasi. *File-file* ini dapat digunakan untuk berbagi antara para pengembang yang sedang bekerja pada aplikasi.

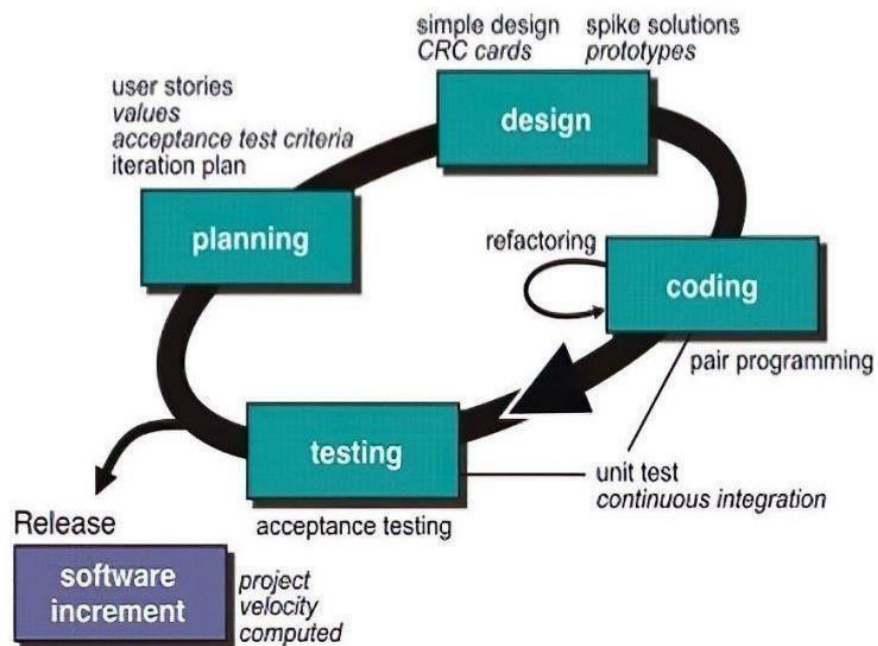
#### 5. *Model*

Model merupakan salah satu komponen MVC yang secara langsung berinteraksi dengan *database*. Dapat dikatakan bahwa model berperan sebagai perantara dalam setiap jalur program yang terkait dengan data. Setelah

terhubung dengan *database*, model akan dipanggil melalui *controller*.

### 2.2.8 Extreme Programming

*Extreme Programming*, atau yang dikenal sebagai XP, merupakan suatu pendekatan pengembangan perangkat lunak yang simpel dan melibatkan praktik-praktik tangkas yang dirancang oleh Kent Beck, Ron Jeffries, dan Ward Cunningham. XP termasuk salah satu metode tangkas yang paling terkenal dan digunakan secara luas di bidang pengembangan perangkat lunak. Tujuan utama XP adalah membentuk tim yang relatif kecil hingga menengah, tanpa memerlukan tim yang besar[24]. Pendekatan ini bertujuan untuk mengatasi ketidakjelasan persyaratan dan perubahan persyaratan dengan respons yang cepat. Berikut adalah skema tentang metode XP:



Gambar 2. 2 Skema *Extreme Programming*

(Sumber: <https://vitolavecchia.altervista.org/> )

Ada beberapa tahapan yang ada pada *Extreme Programming* yaitu terdiri dari:

#### 2.2.8.1 *Planning* (Perencanaan)

Proses ini dimulai dengan mendengarkan kelompok

untuk memenuhi aktivitas sistem yang memungkinkan pengguna memahami proses bisnis dan mendapatkan pemahaman yang jelas tentang fitur utama, fungsionalitas, dan keluaran yang diinginkan. Dalam pengembangan aplikasi web, langkah awalnya adalah mengidentifikasi masalah dalam sistem yang sudah berjalan, diikuti oleh analisis kebutuhan pengguna terhadap sistem yang akan dikembangkan.

#### 2.2.8.2 Design (Perancangan)

Pada tahap perancangan, dilakukan pembuatan model sistem berdasarkan hasil analisis kebutuhan yang telah diperoleh. Selain itu, juga dibuat pemodelan basis data untuk menggambarkan hubungan antar data. Dalam perancangan sistem, salah satu model yang digunakan adalah *Unified Modelling Language (UML)* meliputi beberapa jenis diagram, seperti *Use-Case Diagram*, *Activity Diagram*, *Component Diagram*, dan *Deployment Diagram*. Sementara itu, untuk pemodelan basis data, digunakan *Entity Relationship Diagram (ERD)* dan *Logical Record Structure (LRS)*

#### 2.2.8.3 Coding (Pengkodean)

Tahap pengkodean merupakan proses *implementasi* dari perancangan model sistem ke dalam kode program yang menghasilkan prototipe perangkat lunak. Dalam konteks pembangunan aplikasi *web*, digunakan bahasa HTML, CSS, dan *Javascript* digunakan untuk membangun dan mengatur tampilan serta interaksi pada halaman *web*. HTML berfungsi untuk mengonsep struktur konten dalam halaman web, CSS digunakan untuk mengelola tampilan dan susunan halaman, sementara *Javascript* digunakan untuk memberikan interaktivitas dan fungsionalitas pada halaman tersebut. Dalam konteks implementasi basis data, MySQL diadopsi sebagai Sistem Manajemen Basis Data.

#### 2.2.8.4 *Testing* (Pengujian)

Tahap pengujian adalah saat di mana aplikasi yang telah dibangun diuji untuk memastikan kualitasnya. Pada tahap ini, pengguna sistem menentukan kelayakan fitur dan fungsionalitas dari sistem secara keseluruhan, serta melakukan tinjauan terhadap aplikasi. Jenis pengujian yang diterapkan dalam pengujian aplikasi web untuk pemilihan peserta pelatihan kerja dapat dikategorikan sebagai *Black Box Testing*, yang melibatkan evaluasi masukan dan keluaran sistem tanpa memperhatikan struktur internal aplikasi.

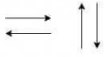











#### 2.2.9 VSC (Visual Studio Code)

*VSC* adalah editor kode yang efektif dengan fitur pemecahan masalah yang kuat. Dengan menggunakan *VSC*, dapat mengidentifikasi dan memperbaiki masalah mengenai JWT. Dimana, dapat mengatur titik henti dan melihat jejak panggilan fungsi untuk memperbaiki masalah[12]. Microsoft mengembangkan aplikasi ini, yang dapat diakses pada berbagai sistem operasi seperti *Linux*, *Mac*, dan *Windows*. *Visual Studio Code* secara langsung mendukung bahasa pemrograman seperti Javascript, Typescript, dan Node.js. Selain itu, pengguna dapat mengakses berbagai bahasa pemrograman lainnya melalui instalasi plugin yang dapat diunduh dari *marketplace Visual Studio Code*. Beberapa contoh bahasa pemrograman tersebut meliputi C++, C#, Python, Go, Java, PHP, dan beberapa bahasa lainnya[25].

#### 2.2.10 Flowchart

*Flowchart* atau diagram alir adalah suatu bentuk diagram yang menggambarkan *urutan* langkah-langkah atau algoritma dalam sebuah sistem. *Flowchart* digunakan oleh analis sistem untuk mendokumentasikan dan menjelaskan logika sistem yang akan dibangun kepada *programmer*. Dengan menggunakan *flowchart*, analis sistem dapat membantu dalam memberikan solusi terhadap potensi masalah yang mungkin timbul saat membangun sistem.

*Flowchart* biasanya dijelaskan dengan menggunakan simbol-simbol yang telah ditetapkan[26]. Berikut merupakan simbol flowchart:

	<b>Flow</b> Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga dengan Connecting Line.		<b>Input/output</b> Simbol yang menyatakan proses input atau output tanpa tergantung peralatan.
	<b>On-Page Reference</b> Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang sama.		<b>Manual Operation</b> Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer.
	<b>Off-Page Reference</b> Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang berbeda.		<b>Document</b> Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk fisik, atau output yang perlu dicetak.
	<b>Terminator</b> Simbol yang menyatakan awal atau akhir suatu program.		<b>Predefine Proses</b> Simbol untuk pelaksanaan suatu bagian (sub-program) atau prosedur.
	<b>Process</b> Simbol yang menyatakan suatu proses yang dilakukan komputer.		<b>Display</b> Simbol yang menyatakan peralatan output yang digunakan.
	<b>Decision</b> Simbol yang menunjukan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak.		<b>Preparation</b> Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal.

Gambar 2. 3 Simbol *Flowchart*  
(Sumber: <https://www.dicoding.com> )

### 2.2.11 Postman

*Postman* adalah *platform* kerja sama yang digunakan untuk menguji, mengelola, dan mendokumentasikan API. Dengan *Postman*, pengguna dapat dengan mudah melakukan pengujian dan *debug* API menggunakan lingkungan pengembangan yang kuat. Pada *Postman* mempunyai sebuah *endpoint* pada API meliputi pengambilan informasi (*GET*), penambahan informasi (*POST*), pembaruan informasi (*PUT/PATCH*), atau penghapusan informasi (*DELETE*). *Endpoint* juga dapat memiliki parameter tambahan untuk memfilter, mengurutkan, atau mengatur data yang diperoleh melalui API. Dengan menggunakan *endpoint* yang sesuai, pengguna atau pengembang dapat berkomunikasi dengan API dan memanipulasi data sesuai kebutuhan[27].

### 2.2.12 UML (Unified Modeling Language)

*Unified Modeling Language* (UML) adalah sebuah alat atau model yang digunakan untuk merancang pengembangan perangkat lunak yang berbasis pada pemrograman berorientasi objek. UML juga











memberikan pedoman standar untuk membuat gambaran yang rinci tentang suatu sistem, mencakup aspek-aspek seperti konsep bisnis, spesifikasi kelas dalam bahasa pemrograman tertentu, desain skema database, dan komponen yang diperlukan dalam pengembangan perangkat lunak tersebut[28]. Jenis-jenis UML meliputi:

### 2.2.12.1 Use Case Diagram

Merupakan representasi visual yang mengilustrasikan interaksi umum antara pengguna dan sistem tertentu. Diagram ini membentuk gambaran tentang bagaimana sistem tersebut digunakan melalui penjelasan naratif yang menggambarkan interaksi antara pengguna dan sistem. Komponen-komponen dalam diagram kasus pengguna mencakup aktor dan interaksi yang dilakukan oleh mereka. Aktor dapat berupa manusia, perangkat keras, sistem lain, atau entitas lain yang terlibat dalam interaksi dengan sistem.

Tabel 2 1 Simbol *use case* (sumber:

<https://.dumetschool.com%2fblog%2fMenenal-Use-Case-Diagram> )







SIMBOL	NAMA	KETERANGAN
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemennya ( <i>sinergi</i> ).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

### 2.2.12.2 Activity Diagram

Merupakan membentuk representasi *visual* dari sejumlah alur aktivitas yang ada dalam sistem yang sedang direncanakan. Diagram ini memberikan gambaran mengenai cara setiap alur aktivitas dimulai, keputusan-keputusan yang mungkin terjadi, dan cara mereka mencapai akhirnya. Selain itu, diagram aktivitas mampu menggambarkan proses paralel yang mungkin terjadi selama beberapa eksekusi.

Tabel 2 2 Simbol *Activity Diagram*[26]







(sumber: <https://www.slideserve.com/> )

Simbol	Keterangan
	Start Point
	End Point
	Activities
	Fork (Percabangan)
	Join (Penggabungan)
	Decision
Swimlane	Sebuah cara untuk mengelompokkan activity berdasarkan Actor (mengelompokkan activity dalam sebuah urutan yang sama)

### 2.2.12.3 Sequence Diagram

Digunakan *untuk* menggambarkan interaksi antara objek-objek di dalam dan sekitar sistem dengan cara memvisualisasikan pesan-pesan yang dikomunikasikan sepanjang waktu. Diagram ini terdiri dari dimensi vertikal yang mencakup aspek waktu dan dimensi *horizontal* yang memuat objek-objek yang terlibat dalam interaksi tersebut.

Tabel 2 3 Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menggambar orang yang sedang berinteraksi dengan sistem.
2		<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan
3		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari foem
4		<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel
5		<i>A focus of Control &amp; A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya message
6		<i>A message</i>	Menggambarkan Pengiriman Pesan

### 2.2.13 Black Box Testing

*Pengujian Blackbox*, atau yang dikenal sebagai *Behavioral Testing*, merupakan proses pengujian yang bertujuan untuk mengamati hasil *input* dan *output* dari perangkat lunak tanpa memerlukan pengetahuan terhadap struktur kode perangkat lunak tersebut. Proses pengujian ini umumnya dilakukan pada tahap akhir pengembangan perangkat lunak untuk mengevaluasi fungsionalitasnya. Keistimewaan dari pengujian ini terletak pada kemampuannya untuk memastikan kinerja yang baik tanpa memerlukan keahlian dalam menulis kode program. Siapa pun dapat melakukan pengujian ini, dan ini menjadi sebuah metode yang sangat berguna dalam penelitian, di mana pilihan metode penelitian yang tepat sangat penting untuk mendapatkan data yang valid. Metode penelitian, sebagai suatu pendekatan ilmiah, menjadi kunci untuk memperoleh data yang valid, membuktikan, dan mengembangkan pemahaman yang mendalam, sehingga dapat digunakan untuk mengatasi, memecahkan, dan mengantisipasi masalah yang terkait[29].