

Studi Perbandingan Penerapan Pola Model-View-Controller (MVC) dalam Lima Framework Web Populer: Laravel, Django, Ruby on Rails, Asp.net MVC, Spring MVC

Rengga Anggarah¹, Kenia Nurma Feblia¹, Khalisa Rizgita Amanda¹, Qonita Adzkiatul Mardiyah¹, Filya Chiara Amanda¹, Anis Syarifatul Mursyidah¹, Yovanza Villareal¹, M D Valgiyos Aritonang¹, Yudi Setiawan²

¹Program Studi Informatika, ²Program Studi Sistem Informasi

Fakultas Teknik, Universitas Bengkulu

Corresponding author: keniafebli21@gmail.com

Abstract—Artikel ini menganalisis dan membandingkan implementasi arsitektur Model View Controller (MVC) pada lima framework web populer: Laravel, Django, Ruby on Rails, ASP.NET MVC, dan Spring MVC. Meskipun MVC bertujuan untuk memisahkan logika bisnis, antarmuka pengguna, dan kontrol aplikasi, setiap framework mengadopsi pendekatan dan karakteristik unik dalam menangani komponen Model, View, dan Controller. Analisis ini mengeksplorasi perbedaan dalam kemudahan pengembangan, fleksibilitas, performa, serta aspek-aspek lain seperti konvensi dan fitur bawaan. Hasil perbandingan ini diharapkan dapat menjadi panduan bagi pengembang dalam memilih framework MVC yang paling sesuai dengan bahasa pemrograman, kompleksitas proyek, serta kebutuhan fitur dan skalabilitas aplikasi web yang mereka kembangkan.

Keywords—Model View Controller (MVC), Framework, Laravel, Django, Ruby on Rails, ASP.NET MVC, Spring MVC.

I. PENDAHULUAN

Seiring dengan pesatnya perkembangan teknologi web modern, kebutuhan akan arsitektur perangkat lunak yang efisien dan mudah dikembangkan menjadi semakin mendesak. Salah satu pola arsitektur yang telah terbukti mampu menjawab kebutuhan tersebut adalah *Model View Controller* (MVC). MVC merupakan pendekatan yang membagi aplikasi menjadi data logis yang digunakan oleh pengguna, tampilan antarmuka, dan kontrol alur data. Pemisahan ini memungkinkan pengelolaan kode yang lebih terstruktur serta memudahkan pemeliharaan dan pengembangan sistem dalam jangka panjang.

Banyak *framework* web modern yang menerapkan pola MVC sebagai fondasi utama dalam pengembangan aplikasi, antara lain Laravel, Django, Ruby on Rails, ASP.NET MVC, dan Spring MVC. Meskipun mengadopsi prinsip dasar yang sama, masing-masing *framework* memiliki sintaksis dan fitur yang berbeda dalam mengimplementasikan pola MVC. Bagi seorang pengembang, pemilihan framework tidak dapat dilakukan dengan hanya sekedar mempertimbangkan bahasa pemrograman yang digunakan, faktor lain seperti kemudahan penggunaan serta dukungan ekosistem yang tersedia juga menjadi penentu dalam keberhasilan pengembangan aplikasi. Maka dari itu, penting bagi para pengembang untuk memahami perbandingan implementasi MVC di berbagai

framework agar dapat memilih *framework* yang paling tepat sesuai dengan proyek yang akan dikerjakan.

II. TINJAUAN PUSTKA

A. Peneliti Sebelumnya

Berdasarkan hasil studi [1] mengenai penerapan analisis pada layanan web Ruhe Medistruktur di E-Kampus STT Portal Indonesia Tanjungpinang, dilakukan pengujian terhadap tiga *framework* populer, yakni Laravel, Django, dan Ruby on Rails, guna membandingkan performa masing-masing. Studi tersebut menemukan bahwa Django memiliki waktu respons yang paling singkat serta penggunaan sumber daya CPU dan memori yang jauh lebih hemat dibandingkan dengan dua *framework* lainnya. Hal ini menunjukkan bahwa Django tidak hanya mampu mempercepat waktu muat halaman web, tetapi juga sangat efisien dalam pengelolaan *resource*, sesuatu yang sangat diperlukan untuk lingkungan pendidikan yang menuntut sistem yang stabil tanpa memboroskan sumber daya. Oleh karena itu, peneliti merekomendasikan Django sebagai *framework* unggulan untuk pengembangan layanan web di institusi pendidikan agar didapatkan performa yang optimal dan penggunaan sumber daya yang efisien. Jadi, bagi siapa saja yang ingin membuat sistem kampus yang responsif dan ringan, Django menjadi solusi yang paling sesuai.

Dalam penelitian lain [2], dilakukan perbandingan kinerja antara tiga *framework* terkemuka dalam pengembangan web *service REST*, yaitu Laravel, Django, dan Node.js. Dari hasil evaluasi, Node.js terbukti memiliki kecepatan eksekusi tertinggi dalam menjalankan tugas-tugas yang diberikan. Namun demikian, Django menunjukkan kelebihan dalam hal kemampuan menerima jumlah permintaan per detik yang lebih besar serta efisiensi pengiriman data HTML yang lebih baik. Laravel sendiri dalam studi ini berada di bawah kedua pesaingnya baik dari sisi kecepatan respons maupun efisiensi *resource*. Dengan demikian, pilihan *framework* yang tepat dapat disesuaikan dengan kebutuhan proyek, antara Node.js yang unggul dalam kecepatan eksekusi dan Django yang kuat dalam kapasitas menangani beban permintaan.

Studi terbaru yang dilakukan oleh [3] mengkaji performa *framework* PHP berbasis MVC, seperti CodeIgniter, Laravel, dan Phalcon, khususnya dalam menjalankan operasi basis data

MySQL dan PostgreSQL. Hasil pengujian menunjukkan bahwa Phalcon menjadi yang tercepat dalam melakukan operasi penyisipan dan pembaruan data, berkat desain arsitektur yang ringan dan dioptimalkan untuk performa tinggi. Laravel juga memberikan hasil yang memadai, meskipun sedikit kalah cepat dibanding Phalcon. Sementara itu, CodeIgniter memiliki performa yang relatif lebih rendah dibanding kedua *framework* tersebut. Studi ini menyimpulkan bahwa jika kecepatan dalam pengelolaan *database* menjadi prioritas utama, Phalcon adalah pilihan terbaik, tetapi Laravel tetap menjadi opsi yang fleksibel dan dapat diandalkan untuk pengembangan aplikasi secara umum.

Penelitian yang dilakukan oleh [4] menjelaskan kemudahan dalam mengelola konten web menggunakan Ruby on Rails. Mereka menguraikan bahwa fitur *Active Record* memudahkan interaksi dengan database layaknya mengedit sebuah file teks, serta pola arsitektur MVC (*Model-View-Controller*) membantu strukturisasi dan percepatan proses pengembangan aplikasi web. *Active Record* memungkinkan pengembang untuk menghindari penulisan SQL manual karena secara otomatis menghubungkan tabel *database* dengan kode program. MVC memisahkan tanggung jawab antara logika aplikasi, tampilan, dan kontrol, sehingga tim pengembang dapat bekerja secara paralel tanpa mengganggu satu sama lain. Studi ini menggambarkan Ruby on Rails sebagai alat yang menyediakan kerangka kerja yang konsisten dan terorganisir, yang membantu pengembang fokus pada pengembangan fitur tanpa terbebani pengaturan teknis yang berulang, sehingga mempercepat proses pembuatan aplikasi web.

Sebuah studi komparatif [5] meneliti dua *framework* pengembangan web populer, yaitu CodeIgniter dan Laravel. Hasil penelitian menunjukkan keunggulan Laravel dalam hal struktur yang lebih modern dan fitur yang lebih komprehensif. Selain itu, tim peneliti melakukan uji kinerja dengan membuat aplikasi sederhana sebagai patokan (*benchmark*). Hasil pengujian tersebut mengungkapkan bahwa Laravel lebih efisien dan fleksibel, sehingga cocok untuk proyek yang memerlukan kemampuan skalabilitas tinggi. Sebaliknya, CodeIgniter unggul sebagai *framework* yang ringan dan mudah digunakan, namun kurang memadai untuk aplikasi yang membutuhkan fitur-fitur kompleks seperti yang ditawarkan oleh Laravel. Berdasarkan temuan ini, Laravel direkomendasikan bagi pengembang yang membutuhkan *framework* modern dengan fitur lengkap, sementara CodeIgniter lebih sesuai untuk proyek berskala kecil dan ringan sesuai dengan kebutuhannya.

B. Laravel

Sebuah *framework* PHP bernama Laravel, yang berlisensi MIT dan mengimplementasikan konsep MVC (*model view controller*), menjadi dasar untuk pengembangan web berbasis MVP. Ditulis dalam bahasa PHP, Laravel memiliki tujuan untuk meningkatkan kualitas perangkat lunak dengan mengurangi pengeluaran biaya di tahap awal pengembangan dan selama pemeliharaan. Lebih jauh lagi, Laravel diciptakan untuk membuat pengalaman bekerja dengan aplikasi menjadi lebih baik bagi pengembang melalui sintaks yang ekspresif, gamblang, dan efisien dalam penggunaan waktu [6].

C. Django

Django adalah sebuah *framework* web yang menggunakan bahasa pemrograman Python dan mendukung pengembangan

situs web dengan pendekatan *rapid development*. Tujuan utama Django adalah mempermudah pembuatan situs web dan basis data yang kompleks. Salah satu kelebihan Django adalah implementasi ORM (*Object Relational Mapper*) yang terintegrasi, sehingga pengembang tidak perlu menyesuaikan *query* ketika terjadi perubahan pada basis data yang digunakan [7].

D. Ruby on Rails

Ruby on Rails, atau sering disebut Rails, adalah sebuah kerangka kerja aplikasi web sumber terbuka yang berjalan menggunakan bahasa pemrograman Ruby. Sebagai *framework full-stack*, Rails memungkinkan pembuatan halaman dan aplikasi web yang mengumpulkan informasi dari server web, berinteraksi atau mengirimkan *query* ke server web, dan mengolah template dari sumber yang telah ditentukan. Hasilnya, Rails menyediakan fitur sistem *routing* yang independen dari server web [8].

E. ASP.NET MVC

ASP.NET MVC adalah sebuah arsitektur yang digunakan untuk mengembangkan situs web dinamis. Pengembangan situs web berbasis arsitektur ini memerlukan Microsoft .NET *Framework* versi 3.5 atau yang lebih tinggi. Salah satu keunggulan situs web yang dikembangkan menggunakan ASP.NET MVC adalah kemampuannya yang ramah terhadap SEO (*Search Engine Optimizer*). Ini berarti arsitektur ini memiliki keunggulan dasar, yaitu kemudahan untuk diindeks oleh mesin pencari seperti Google, Yahoo, dan Bing. ASP.NET dikembangkan oleh Microsoft dan pertama kali dirilis pada Januari 2002 dengan lisensi sumber terbuka. *Framework* ini dibangun menggunakan CLR (*Common Language Runtime*) dan memungkinkan penulisan kode untuk ASP.NET menggunakan bahasa pemrograman berbasis .NET seperti C# dan Visual Basic [9].

F. Spring MVC

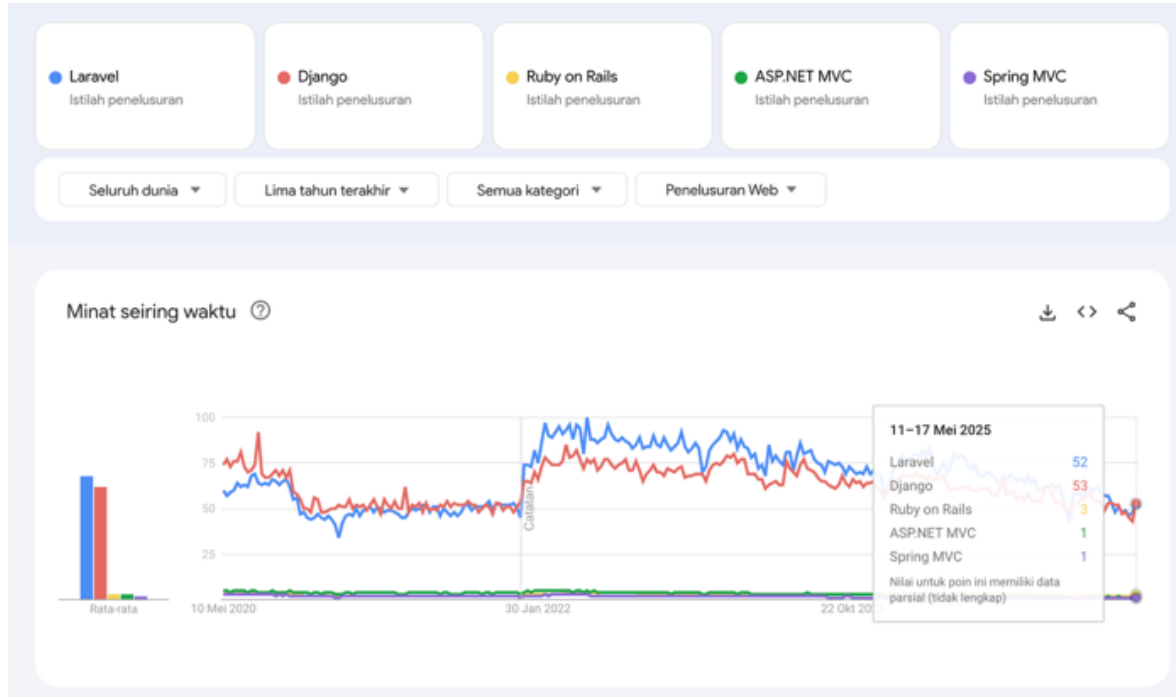
Spring, yang mencakup Spring MVC, adalah sebuah *framework* yang terbukti keandalannya dan memiliki basis pengguna yang besar. Dalam konteks pengembangan aplikasi Java EE, Spring sering dianggap sebagai *standard stack development*. Arsitektur fundamental Spring adalah *container Inversion of Control* (IoC) yang beroperasi berdasarkan prinsip *JavaBean* [4]. Ini memungkinkan Spring untuk menerapkan teknik *Dependency Injection* pada objek. *Dependency Injection* merupakan wujud implementasi IoC yang memungkinkan container untuk memasok dependensi yang dibutuhkan oleh suatu objek melalui objek lain. Selain itu, Spring menyediakan berbagai tingkat abstraksi untuk manipulasi data dan dapat dengan mudah bekerja sama dengan berbagai ORM dan *library* seperti Hibernate, iBatis, dan JDBC untuk keperluan akses data [10].

III. HASIL DAN PEMBAHASAN

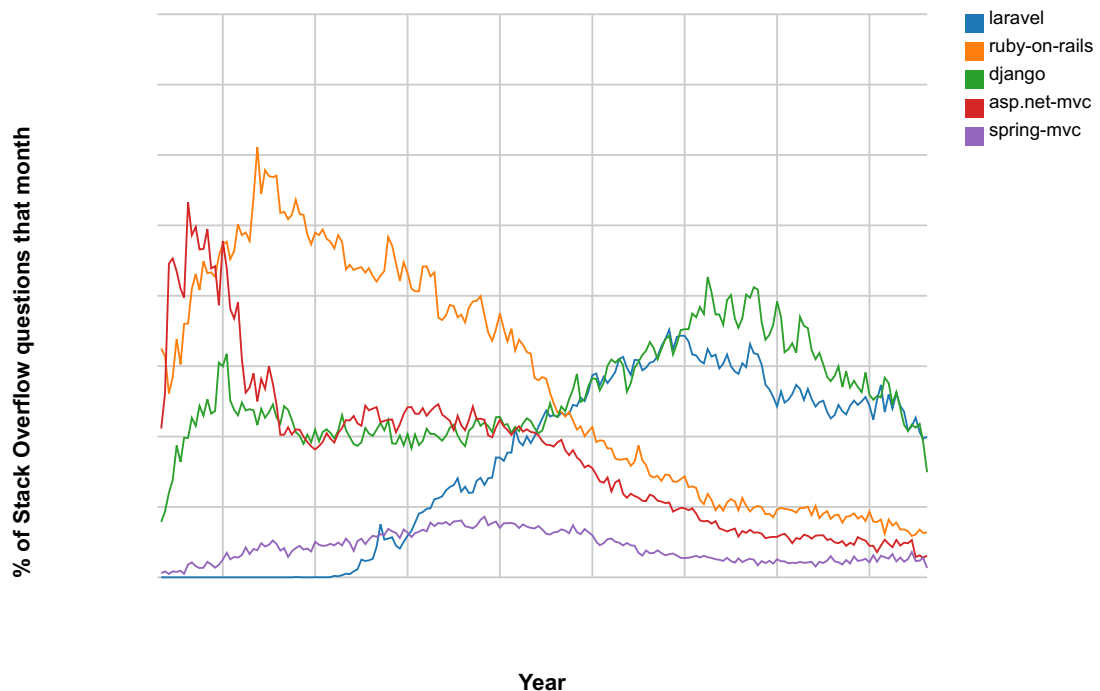
A. Komparasi Struktur

1. Popularity dan Current Trends: Untuk mengkomparasikan kelima framework web ini, digunakan data tren dan popularitas yang diperoleh dari berbagai sumber kredibel.

ASP.NET MVC, dan Spring MVC, memiliki dukungan yang luas terhadap sistem manajemen basis data (DBMS), namun terdapat beberapa perbedaan yang menonjol di antara mereka. Dengan basis bahasa PHP, Laravel secara resmi mendukung MySQL, MariaDB, PostgreSQL, SQLite, dan SQL Server. Dukungan untuk MongoDB juga tersedia melalui paket eksternal. Django,



Gambar 1. Trend dan popularity Laravel, Django, Ruby on Rails, ASP.NET MVC, dan Spring MVC berdasarkan Google Trend



Gambar 2. Popularity Laravel, Django, Ruby on Rails, ASP.NET MVC, dan Spring MVC berdasarkan Stack Overflow Trend

2. Database: Dikutip dari Referensi [3], kelima framework MVC populer, yaitu Laravel, Django, Ruby on Rails, yang ditulis dalam Python, secara bawaan mendukung MySQL, PostgreSQL, SQLite, dan Oracle, serta

menawarkan kontrol yang baik terhadap isolasi transaksi dan manajemen skema database. Sementara itu, Ruby on Rails secara langsung mendukung MySQL, PostgreSQL, dan SQLite, namun untuk integrasi dengan DBMS lain seperti SQL Server atau MongoDB, diperlukan adapter tambahan. Berbeda dengan ketiganya, ASP.NET MVC yang dikembangkan oleh Microsoft, sangat optimal digunakan dengan SQL Server, tetapi juga mendukung MySQL, PostgreSQL, SQLite, dan Oracle melalui provider pihak ketiga, serta menawarkan integrasi kuat dengan layanan cloud dan enterprise.

Adapun Spring MVC yang berbasis Java, memiliki fleksibilitas paling tinggi karena mendukung hampir semua DBMS utama, baik SQL maupun NoSQL, melalui Hibernate atau Spring Data, sehingga memudahkan pengembang dalam memilih basis data sesuai kebutuhan aplikasi. Dengan demikian, meskipun kelima framework ini sama-sama mendukung DBMS relasional populer, Spring MVC dan ASP.NET MVC cenderung lebih unggul dalam hal fleksibilitas dan integrasi enterprise, sedangkan Laravel, Django, dan Rails lebih mengutamakan kemudahan dan kecepatan pengembangan aplikasi berbasis DBMS populer.

3. **Programming Paradigm:** Dikutip berdasarkan Referensi Kelima framework ini mengadopsi paradigma Model-View-Controller (MVC) sebagai dasar arsitektur mereka, yang memisahkan aplikasi menjadi tiga komponen utama: model (data dan logika bisnis), view (antarmuka pengguna), dan controller (pengatur alur data dan interaksi). Baik Laravel dengan Eloquent maupun Ruby on Rails dengan ActiveRecord, keduanya menggunakan active record pattern dalam ORM mereka. Pola ini memungkinkan model untuk terhubung secara langsung dengan tabel database, menyederhanakan tugas-tugas CRUD. Django dan Spring MVC menggunakan data mapper pattern dengan ORM masing-masing (Django ORM dan Hibernate), yang memisahkan logika bisnis dari struktur database secara lebih eksplisit. ASP.NET MVC mengadopsi multi-paradigma dengan Entity Framework yang mendukung code-first, database-first, dan model-first, serta integrasi kuat dengan arsitektur enterprise. Secara bahasa pemrograman, Laravel berbasis PHP, Django Python, Rails Ruby, ASP.NET C#, dan Spring Java, yang juga memengaruhi gaya dan kompleksitas pengembangan [5].
4. **Routing:** Dikutip berdasarkan Referensi [2] Laravel, Django, Ruby on Rails, ASP.NET MVC, dan Spring MVC memiliki cara yang berbeda namun efektif dalam menangani routing aplikasi web. Routing di Laravel sangat intuitif dengan definisi rute yang dikelompokkan dalam file `routes/web.php` dan `routes/api.php`, serta dukungan middleware yang mudah diaplikasikan. Django menggunakan sistem modular dengan file `urls.py` di setiap aplikasi, yang kemudian di-include ke dalam file routing utama, memudahkan pengelolaan rute pada proyek besar. Ruby on Rails mengelola routing secara terpusat di file `config/routes.rb` dengan prinsip convention over configuration sehingga rute otomatis mengarah ke controller dan action yang sesuai. Sistem routing pada ASP.NET MVC yang mendukung pendekatan konvensional dan attribute routing memberikan fleksibilitas dalam menangani rute dinamis dan parameter yang kompleks. Berbeda dengan itu, Spring MVC menggunakan anotasi, seperti `@RequestMapping`, pada controller dan method untuk mencapai fleksibilitas yang tinggi dalam mendefinisikan rute serta kemampuannya untuk mendukung berbagai metode HTTP.
5. **Built-in Modules:** Dikutip dari Referensi [2] Laravel, Django, Ruby on Rails, ASP.NET MVC, dan Spring MVC masing-masing menyediakan sistem modularitas bawaan dengan karakteristik unik yang memengaruhi cara pengembang membangun aplikasi yang terstruktur dan mudah dikembangkan. Laravel memiliki sistem modular yang kuat dengan dukungan paket seperti `laravel-modules` yang memungkinkan pemisahan aplikasi menjadi modul-modul independen lengkap dengan controller, model, view, dan konfigurasi sendiri, sehingga memudahkan pengelolaan aplikasi besar. Django secara natural modular karena setiap aplikasi (app) dalam proyek memiliki struktur sendiri yang berfungsi sebagai modul terpisah namun terintegrasi. Ruby on Rails menggunakan konsep engine dan gem sebagai modul yang dapat dipasang secara plug-and-play, meskipun struktur aplikasi Rails cenderung lebih terpusat. ASP.NET MVC mendukung modularitas melalui area dan paket NuGet, serta integrasi middleware yang memudahkan pengembangan fitur modular dan reusable. Dengan sistem dependency injection dan konfigurasi berbasis anotasi, Spring MVC menyediakan modularitas tingkat tinggi. Dukungan modul melalui Spring Boot starter semakin mempermudah pengelolaan komponen aplikasi secara terpisah dan memungkinkan skalabilitas yang baik.
6. **HTTPS Support:** Semua framework mendukung HTTPS secara penuh, namun implementasinya bergantung pada konfigurasi server dan middleware. Laravel menyediakan middleware `\App\Http\Middleware\RequireHttps` untuk mengarahkan trafik ke HTTPS secara otomatis. Django menggunakan `SecurityMiddleware` yang mengaktifkan HTTPS dan fitur keamanan terkait. Ruby on Rails juga mendukung HTTPS dengan konfigurasi di environment dan `middleware.force_ssl`. ASP.NET MVC mendukung HTTPS lewat konfigurasi di IIS dan atribut `[RequireHttps]` pada controller atau action. Spring MVC mengandalkan konfigurasi server (Tomcat, Jetty) dan Spring Security untuk enforce HTTPS. Dukungan HTTPS di kelima framework ini sudah sangat matang dan menjadi standar keamanan aplikasi web modern [3].
7. **Support REST API:** Dikutip dari Referensi [5], Dalam hal dukungan REST API, Django menonjol dengan Django REST Framework (DRF) yang menawarkan fitur-fitur komprehensif seperti serialisasi data, autentikasi, routing API otomatis, dan pembuatan dokumentasi. Laravel juga mendukung REST API dengan routing resource dan Eloquent ORM yang memudahkan manipulasi data JSON. Ruby on Rails mendukung RESTful routing secara konvensional, namun tidak memiliki framework API sekomprehensif DRF sehingga pengembangan API bisa lebih kompleks. ASP.NET MVC dengan Web API dan Entity

Framework sangat kuat dalam membangun REST API dengan fitur lengkap seperti versioning, autentikasi, dan dokumentasi Swagger. Spring MVC dengan Spring Boot dan Spring Data REST menawarkan solusi REST API yang sangat fleksibel dan scalable, mendukung berbagai format data dan protokol. Secara umum, Django, ASP.NET MVC, dan Spring MVC lebih unggul dalam fitur REST API dibanding Laravel dan Rails.

8. Template Engine: Dikutip berdasarkan Referensi [7], Laravel menggunakan Blade, template engine yang intuitif dan powerful dengan fitur inheritance, komponen, dan directive khusus. Dengan template engine bawaan yang sederhana namun mumpuni, Django memungkinkan penggunaan inheritance dan filter dalam proses templating. Ruby on Rails memakai ERB (Embedded Ruby) yang memungkinkan penulisan kode Ruby dalam HTML secara langsung. ASP.NET MVC menggunakan Razor, template engine yang kuat dengan sintaks mirip C# yang mudah dibaca dan mendukung layout serta partial views. Spring MVC mendukung berbagai template engine seperti JSP, Thymeleaf, dan FreeMarker, dengan Thymeleaf menjadi pilihan populer karena kemudahan integrasi dan kemampuan template natural. Blade dan Razor dianggap lebih modern dan expressive dibanding ERB dan Django template, sementara Spring menawarkan fleksibilitas memilih engine sesuai kebutuhan.
9. Authentication: Dikutip dari Referensi [2], Django menyediakan sistem autentikasi built-in yang lengkap dengan user management, permission, dan integrasi mudah dengan Django REST Framework untuk API. Laravel memiliki sistem autentikasi bawaan yang mudah dikonfigurasi dengan fitur seperti login, registrasi, reset password, dan middleware auth. Ruby on Rails menggunakan Devise sebagai gem populer untuk autentikasi yang lengkap dan modular. ASP.NET MVC mengandalkan ASP.NET Identity yang sangat lengkap, mendukung OAuth, OpenID Connect, dan integrasi enterprise. Spring Security, yang digunakan oleh Spring MVC, menyediakan fitur keamanan yang sangat mumpuni dan dapat disesuaikan dengan berbagai kebutuhan. Fitur ini mendukung beragam metode autentikasi dan otorisasi, serta memiliki kemampuan integrasi dengan LDAP dan OAuth. Dari sisi fitur dan fleksibilitas, Spring Security dan ASP.NET Identity berada di level enterprise, sementara Django, Laravel, dan Rails menawarkan solusi yang lebih cepat dan mudah digunakan.
10. Learning Curve dan Community: Dikutip dari Referensi [7], Django dan Ruby on Rails seringkali dipandang lebih mudah dipelajari dibandingkan framework lain. Faktor-faktor yang berkontribusi antara lain adalah kemudahan bahasa Python dan Ruby, serta filosofi framework yang fokus pada konvensi dan kemudahan pengembangan. Laravel juga cukup mudah dipelajari terutama bagi yang sudah familiar dengan PHP. Dibandingkan framework lain, ASP.NET MVC dan Spring MVC cenderung lebih sulit dipelajari karena kerumitan bahasa C# dan Java, serta adanya konsep-konsep enterprise yang lebih mendalam. Dari sisi komunitas, Django, Rails, dan Laravel memiliki komunitas besar dan aktif dengan banyak paket dan

tutorial, sedangkan Spring dan ASP.NET memiliki komunitas enterprise yang kuat dan dokumentasi sangat lengkap. Pilihan framework seringkali dipengaruhi oleh kebutuhan proyek dan latar belakang pengembang.

11. Package: Dikutip dari Referensi [8], Ruby on Rails memiliki ekosistem besar dengan ribuan gem yang memudahkan penambahan fitur. Django juga kaya dengan paket (PyPI) yang mature dan teruji. Dengan Composer sebagai package manager, Laravel menawarkan sejumlah besar paket siap pakai, meskipun ekosistemnya masih tergolong lebih muda jika dibandingkan dengan Rails dan Django. ASP.NET MVC menggunakan NuGet sebagai package manager dengan banyak paket enterprise dan open source. Spring MVC memanfaatkan Maven/Gradle untuk manajemen dependensi dengan ekosistem Java yang luas dan banyak library untuk berbagai kebutuhan. Semua framework ini menyediakan cara mudah untuk mengelola paket dan dependensi, namun Rails dan Django unggul dalam jumlah paket komunitas yang siap pakai.

B. Komparasi Langsung

Analisis komparatif yang detail mengenai lima *framework* MVC terkemuka—Laravel, Django, Ruby on Rails, ASP.NET MVC, dan Spring MVC—terangkum dalam tabel di atas. Setiap framework memiliki atribut unik yang ditentukan oleh aspek-aspek seperti bahasa pemrograman, tata letak folder MVC, dukungan ORM, ukuran komunitas, dan tingkat penggunaan dalam industri. Laravel menunjukkan keunggulan dalam sintaks yang intuitif dan ekosistem PHP yang komprehensif, sementara Django menonjol dalam hal keamanan dan integrasi fitur. Ruby on Rails terkenal dengan pendekatannya yang mengedepankan "konvensi daripada konfigurasi" untuk pengembangan prototipe yang cepat. ASP.NET MVC menjadi pilihan yang tepat untuk sistem yang beroperasi di lingkungan Microsoft dan memerlukan kinerja tinggi, dan Spring MVC sangat diminati oleh perusahaan (*enterprise*) dan dukungan Java yang handal. Meskipun setiap *framework* memiliki kelebihan dan kekurangan, penentuan *framework* yang paling sesuai sangat bergantung pada tuntutan proyek dan keahlian tim pengembangan.

KESIMPULAN

Artikel ini menyajikan analisis komprehensif mengenai implementasi pola arsitektur *Model-View-Controller* (MVC) dalam lima *framework* web populer, yakni Laravel, Django, Ruby on Rails, ASP.NET MVC, dan Spring MVC. Kelima *framework* ini mengadopsi konsep MVC untuk memisahkan logika bisnis, tampilan antarmuka, dan pengontrol aplikasi demi menciptakan sistem yang terstruktur, mudah dikembangkan, dan dipelihara. Namun, masing-masing *framework* memiliki pendekatan, *sintaks*, dan fitur yang berbeda dalam implementasinya.

Laravel, yang berbasis PHP, unggul dalam *sintaks* yang elegan dan kemudahan pengembangan dengan banyak fitur bawaan dan ekosistem yang terus berkembang. Django, berbasis Python, menawarkan keamanan tinggi dan efisiensi penggunaan *resource*, sangat cocok untuk sistem berskala besar dan pendidikan. Ruby on Rails mempermudah pengembangan cepat dengan pendekatan *convention over configuration*, meskipun memiliki performa yang cenderung lebih lambat. ASP.NET MVC yang dikembangkan oleh Microsoft memberikan performa tinggi dan integrasi

enterprise yang kuat, namun memiliki kurva belajar yang curam karena kompleksitas bahasa dan konsep *enterprise*. Sementara itu, Spring MVC menawarkan fleksibilitas dan skalabilitas tinggi, didukung oleh arsitektur modular berbasis Java dan ekosistem *enterprise* yang luas, meskipun setup awalnya cukup kompleks.

Tabel 1. Tabel Perbandingan Langsung MCV Berbagai Framework

Aspek	Laravel	Django	Ruby on Rails	ASP.NET MVC	Spring MVC
Bahasa Pemrograman	PHP	Python	Ruby	C#	Java
Struktur Folder MVC	app/Http/Controllers/, resources/views/, app/Models	project/views.py, models.py, templates/	app/controllers/, app/models/, app/views/	Controllers/, Models/, Views/	src/main/java/..., src/main/resources/templates/
Routing	routes/web.php, Route::get()	urls.py, path(), re_path()	config/routes.rb dengan DSL routing	Startup.cs, atau [Route] attribute	Anotasi @RequestMapping di controller
ORM / DB Support	Eloquent ORM (MySQL, PostgreSQL, SQLite, SQL Server)	Django ORM (PostgreSQL, MySQL, SQLite, Oracle)	ActiveRecord (PostgreSQL, MySQL, SQLite)	Entity Framework (SQL Server, PostgreSQL, dll.)	Hibernate (MySQL, PostgreSQL, Oracle, dll.)
Dokumentasi & Komunitas	Dokumentasi lengkap, komunitas aktif	Dokumentasi komprehensif, banyak tutorial	Komunitas besar, banyak plugin (gem)	Komunitas besar, didukung Microsoft	Komunitas besar, banyak digunakan di enterprise
Kelebihan	Sintaks elegan, banyak fitur bawaan, ekosistem luas	Keamanan tinggi, banyak fitur built-in	Konvensi over konfigurasi, cepat dalam prototyping	Performa tinggi, integrasi kuat dengan Microsoft	Scalable, cocok untuk enterprise, banyak tools Spring
Kekurangan	Kadang membingungkan bagi pemula, performa menengah	Kurang fleksibel, kurva belajar tinggi	Performa bisa lambat, learning curve tinggi	Kompleks untuk pemula, learning curve curam	Kompleks, setup awal rumit
Penggunaan Industri	BBC, Pfizer, startup dan UKM	Instagram, Spotify, Mozilla	GitHub, Shopify, Airbnb	Microsoft, Stack Overflow, enterprise lain	Netflix, Amazon, eBay, banyak perusahaan besar

Dalam aspek *routing*, *modularitas*, dukungan REST API, *engine templating*, sistem autentikasi, dan integrasi basis data, masing-masing *framework* memiliki kelebihan dan kekurangannya. Django, ASP.NET MVC, dan Spring MVC lebih unggul dalam membangun REST API yang kompleks dan *scalable*. Laravel dan Ruby on Rails lebih menonjol

dalam kemudahan penggunaan dan *prototyping*. Dalam hal *community support* dan dokumentasi, Laravel, Django, dan Rails memiliki komunitas besar dan aktif, sedangkan ASP.NET MVC dan Spring MVC didukung oleh komunitas *enterprise* dan dokumentasi teknis yang sangat lengkap.

Pemilihan *framework* MVC terbaik sangat bergantung pada kebutuhan spesifik proyek, bahasa pemrograman yang dikuasai pengembang, kompleksitas sistem yang dibangun, serta kebutuhan fitur dan skalabilitas. Tidak ada satu *framework* yang paling unggul secara mutlak, tetapi melalui studi perbandingan ini, pengembang dapat memiliki gambaran menyeluruh dalam menentukan pilihan *framework* yang paling sesuai untuk proyek mereka.

REFERENCES

- [1] D. Saputra, "Analisis Perbandingan Performa Web Service Rest Menggunakan Framework Laravel, Django Dan Ruby On Rails Untuk Akses Data Dengan," *J. Bangkit Indones.*, vol. 7, no. 2, p. 17, 2018, doi: 10.52771/bangkitindonesia.v7i2.90.
- [2] A. Amarulloh, K. Kurniasih, and M. Muchlis, "ANALISIS PERBANDINGAN PERFORMA WEB SERVICE REST MENGGUNAKAN FRAMEWORK LARAVEL, DJANGO, DAN Node JS PADA APLIKASI BERBASIS WEBSITE," *J. Tek. Inform.*, vol. 9, no. 1, pp. 14–19, 2023.
- [3] M. H. Rahman, M. Naderuzzaman, M. A. Kashem, B. M. Salahuddin, and M. Z. Mahmud, "Comparative Study: Performance of MVC Frameworks on RDBMS," *Int. J. Inf. Technol. Comput. Sci.*, vol. 16, no. 1, pp. 26–34, 2024, doi: 10.5815/ijitcs.2024.01.03.
- [4] A. Tapiador and J. Salvachúa, "Content management in ruby on rails," *Proc. IADIS Int. Conf. - Web Based Communities Soc. Media 2011, Soc. Media 2011, Internet Appl. Res. 2011, Part IADIS, MCCSIS 2011*, pp. 263–265, 2011.
- [5] W. Muthia Kansa, Saherih, and Muchlis, "Analisis Perbandingan Struktur dan Performa Framework Codeigniter dan Laravel dalam Pengembangan Web Application," *J. Tek. Inform. STMIK Antar Bangsa*, vol. 9, no. 1, pp. 25–31, 2023.
- [6] B. Hermanto, M. Yusman, and N. Nagara, "Sistem Informasi Manajemen Keuangan pada PT. Hulu Balang Mandiri Menggunakan Framework Laravel," *J. Komputasi*, vol. 7, no. 1, pp. 17–26, 2019, doi: 10.23960/komputasi.v7i1.2051.
- [7] Bilya Putra Aji and A. Hernawan, "Sistem Informasi Surat Elektronik Untuk Akademik UIN Mataram (Dengan Python Django Framework)," *J. Begawe Teknol. Inf.*, vol. 3, no. 2, pp. 252–262, 2022, doi: 10.29303/jbegati.v3i2.777.
- [8] A. Hani and Y. Supendi, "Implementasi Custom Algorithm Berbasis Web Menggunakan Ruby on Rails pada Analisis Pengelolaan Resiko Pinjaman oleh Perusahaan Asuransi," vol. 5, no. 2, pp. 29–34, 2024.
- [9] F. A. Juyuspan and P. Oktivasari, "Pengembangan Website Dinamis Menggunakan Asp.Net Mvc Dan Sql Server Dengan Metode Rad (Studi Kasus: Pt X)," *InfoTekJar (Jurnal Nas. Inform. dan Teknol. Jaringan)*, vol. 2, no. 1, pp. 16–21, 2017, doi: 10.30743/infotekjar.v2i1.141.
- [10] M. Glend Steven, "Pengembangan Aplikasi Manajemen Proyek Perangkat Lunak Berbasis Spring: Modul Core System dan

- [11] Manajemen Source Code,” 2011.
- [12] “Google Trends,” [Online]. Available: <https://trends.google.co.id/trends/explore?date=today%205-y&q=Laravel,Django,Ruby%20on%20Rails,ASP.NET%20MVC,Spring%20MVC&hl=id>.
- [12] “Stack Overflow Trend”. [Online]. Available: <https://trends.stackoverflow.co/?tags=laravel%2Cruby-on-rails%2Cdjangon%2Casp.net-mvc%2Cspring->