

Pembuatan REST API Manajemen Data Karyawan Berbasis Website Menggunakan *Spring Boot*

Leonard Surya Adji ¹, Evangs Mailoa ^{2*}

^{1,2*} Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Kota Salatiga, Provinsi Jawa Tengah, Indonesia.

Email: 672020008@student.uksw.edu ¹, evangs.mailoa@uksw.edu ^{2*}

Histori Artikel:

Dikirim 7 Maret 2024; Diterima dalam bentuk revisi 5 April 2024; Diterima 15 April 2024; Diterbitkan 10 Mei 2024. Semua hak dilindungi oleh Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) STMIIK Indonesia Banda Aceh.

Abstrak

Penelitian ini membahas penerapan REST API dengan Spring Boot dalam pengelolaan data karyawan di perusahaan. Dengan fokus pada efisiensi, ketepatan, dan keamanan data, aplikasi ini menggunakan Java, Spring Boot, PostgreSQL dan MyBatis sebagai fondasi utama. Pengujian sistem menunjukkan keberhasilan dalam pencarian, pembuatan, pengubahan, dan penghapusan data karyawan. Melalui struktur Model View Control (MVC), aplikasi memastikan pemisahan tugas yang jelas. Keberhasilan ini menegaskan aplikasi sebagai solusi efektif untuk manajemen data karyawan, terutama dalam industri E-commerce yang dinamis.

Kata Kunci: REST API; Spring Boot; Pengelolaan Data Karyawan; Efisiensi; Keamanan; Model View Control (MVC).

Abstract

This study explores the implementation of REST API with Spring Boot in managing employee data in an E-commerce company. With a focus on efficiency, accuracy, and data security, the application utilizes Java, Spring Boot, and MyBatis as the main foundations. System testing demonstrates success in searching, creating, updating, and deleting employee data. Through the Model View Control (MVC) structure, the application ensures clear task separation. This success affirms the application as an effective solution for employee data management, particularly in the dynamic E-commerce industry.

Keyword: REST API; Spring Boot; Employee Data Management; Efficiency; Security; Model View Control (MVC).

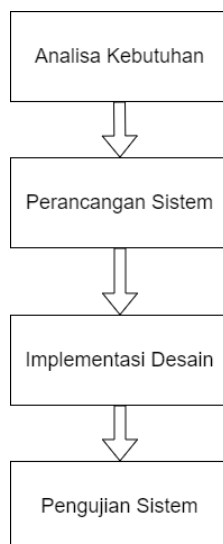
1. Pendahuluan

Dalam perkembangan bisnis yang semakin pesat, kebutuhan akan tenaga kerja juga mengalami peningkatan signifikan. Pengelolaan karyawan menjadi aspek yang tidak dapat diabaikan, mengingat dampaknya terhadap hubungan antara perusahaan dan karyawan (Aidhawani & Nasution, 2023). Keberhasilan suatu perusahaan dalam mencapai tujuan bisnisnya sangat dipengaruhi oleh efektivitas manajemen karyawan. Peran sentral karyawan sebagai aset utama perusahaan tercermin dalam kemampuannya untuk menggerakkan roda bisnis dan memberikan kontribusi pada pencapaian kualitas dan kuantitas yang diinginkan (Saputra *et al.*, 2019). Oleh karena itu, perlu diakui peran strategis karyawan dalam mencapai kesuksesan perusahaan serta menjaga harmoni dalam lingkungan kerja. Dengan demikian, manajemen karyawan yang baik dapat dianggap sebagai kunci utama untuk memastikan pertumbuhan dan keberlanjutan bisnis dalam era dinamika pasar yang terus berkembang (Saputra *et al.*, 2019). Dalam struktur organisasi suatu perusahaan berskala besar seperti perusahaan *E-commerce*, keberadaan sejumlah besar karyawan dengan berbagai divisi dan data-data pribadi yang berbeda telah menjadi aspek yang umum. Oleh karena itu, manajemen data karyawan menjadi kebutuhan yang krusial. Hal ini bertujuan untuk mencegah terjadinya kekacauan dan memastikan ketertiban dalam penyimpanan informasi karyawan (Dhika *et al.*, 2019);(Choirinisa, 2022) Pengelolaan data dapat menggunakan aplikasi konvensional seperti *Microsoft Word*, *Microsoft Excel* sebagai media untuk pengolahan data. Namun, penggunaan metode ini dapat memunculkan potensi penyimpangan data yang kurang efisien dan meningkatkan risiko kehilangan data, yang tentu saja dapat berdampak merugikan bagi karyawan. Oleh karena itu, penting untuk mengevaluasi dan mengadopsi metode pengelolaan data karyawan yang lebih canggih dan terintegrasi guna meningkatkan efisiensi, ketepatan, dan keamanan data, seiring dengan perkembangan teknologi informasi saat ini (Triandi Putra, 2022);(Darmi *et al.*, 2023);(Murni Wijaya & Mutaqin, 2019). Dalam era kemajuan teknologi yang terasa semakin nyata, penggunaan *database* sebagai solusi penyimpanan data telah diakui sebagai kebutuhan penting oleh perusahaan-perusahaan modern. Risiko kehilangan data dapat diminimalkan melalui penerapan sistem informasi *REST API*, di mana *Spring Boot*, sebuah *framework* yang mendapat popularitas, dipilih sebagai alat pendukung (Umar, 2023);(Rumetna, 2018).

Metode *Representational State Transfer (REST) Application Programming Interface (API)* diaplikasikan dalam pengembangan antarmuka berbasis *web*, memungkinkan pengelolaan informasi global melalui *website* (Adi Pranata *et al.*, 2018);(Pangestu *et al.*, 2022);(Hassanudin; Asgar, Hari; Hartono, 2022). *Website*, sebagai sistem operasi informasi, menjadi wadah penampungan informasi dari berbagai sumber. Akses luas bagi pengguna (*client*) dilakukan melalui *server* menggunakan protokol *Hyper Text Transfer Protocol (HTTP)* (Adi Pranata *et al.*, 2018). *Framework Spring Boot*, bersifat *open source*, digunakan secara luas dalam pembuatan aplikasi berbasis *Java* dan mendukung pengembangan aplikasi layanan *web RESTful* (Umbu Dagha, 2021). Langkah-langkah ini diterapkan sebagai respons terhadap tuntutan perkembangan teknologi, mencerminkan implementasi praktis di lingkungan bisnis yang berorientasi pada kehandalan dan efisiensi penyimpanan data. *Java* merupakan bahasa pemrograman tingkat tinggi yang didasarkan pada konsep *Object Oriented Programming*. Keunggulan *Java* terletak pada kinerjanya yang tinggi, fleksibilitas, dan tingkat keamanannya, menjadikannya dapat digunakan secara luas di berbagai *platform*, termasuk dalam pengembangan *framework* seperti *Spring Boot* (Gata, 2013). Dalam pembuatan *RESTful API*, *Spring Boot* dapat didukung dengan menggunakan ekstensi lainnya seperti *MyBatis*. *MyBatis* merupakan salah satu *framework* dalam rumpun *Database Bridging Layer*, yang dikenal mampu menyimpan data sementara yang dapat merelaksasi kinerja *harddisk* (Ginanjar *et al.*, 2021). Berdasarkan permasalahan yang telah disebutkan, penelitian dengan judul “Pembuatan *REST API* Manajemen Data Karyawan Berbasis *Website* Menggunakan *Spring Boot*” dilakukan. Fokus penelitian ini terletak pada penerapan *API* dengan metode *REST* dalam aplikasi berbasis *website* menggunakan *framework Spring Boot* untuk manajemen data karyawan. Dengan pembuatan *REST API* ini, diharapkan bahwa pengelolaan data karyawan dalam bidang administrasi dapat dipermudah, memungkinkan proses tersebut berlangsung dengan cepat dan efisien.

2. Metode Penelitian

Dalam penelitian ini, empat tahapan penelitian akan dilalui, yaitu analisis kebutuhan, perancangan sistem, implementasi desain, dan pengujian sistem.



Gambar 1. Model Proses Penelitian

Langkah-langkah penelitian ditunjukkan dalam gambar 1. Langkah pertama terlibat dalam analisis kebutuhan, di mana kebutuhan yang diperlukan untuk membangun sistem informasi dianalisis secara cermat oleh peneliti. Tujuan dari tahap ini adalah memberikan dasar yang kokoh dalam perancangan sistem, memastikan bahwa aspek-aspek penting telah dipertimbangkan dengan baik. Selanjutnya, tahap kedua adalah perancangan sistem. Setelah pemahaman yang komprehensif tentang kebutuhan diperoleh, sistem dirancang oleh peneliti sesuai dengan hasil analisis dari tahap sebelumnya. Langkah ini memegang peran krusial dalam penciptaan kerangka kerja yang kokoh dan efisien, memudahkan proses implementasi. Setelah sistem dirancang, langkah ketiga adalah implementasi desain. Pada tahap ini, konsep dan desain yang telah dibuat sebelumnya akan diwujudkan secara nyata. Proses ini melibatkan perubahan konsep teoritis menjadi suatu sistem yang dapat berfungsi dalam praktik. Langkah terakhir dalam penelitian ini adalah pengujian sistem. Pada tahap ini, sistem yang telah diimplementasikan akan diuji secara menyeluruh oleh peneliti. Tujuannya adalah memastikan kinerja sistem optimal, mengidentifikasi potensi masalah pada sistem, dan mengatasi masalah tersebut dengan efektif. Pengujian sistem merupakan langkah krusial dalam memastikan keberhasilan dan kehandalan suatu sistem. Oleh karena itu, keseluruhan proses perancangan dan implementasi sistem ini membentuk sebuah pendekatan sistematis untuk mencapai tujuan penelitian ini.

3. Hasil dan Pembahasan

Pada bagian ini, dijelaskan hasil dari perancangan *REST API* manajemen data karyawan berbasis *website*, bertujuan untuk memudahkan pengelolaan data karyawan pada suatu perusahaan, baik skala besar ataupun kecil, berdasarkan metodologi penelitian yang telah dilakukan. *Website* ini menggunakan *framework springboot*, dikarenakan *springboot* merupakan salah satu *framework* yang baik dalam membangun aplikasi terutama berbasis *website*. *Website* ini dirancang untuk satu pengguna, yaitu admin, yang memiliki akses penuh dalam mengelola data karyawan.

3.1 Proses dan Pembuatan Aplikasi

REST API Manajemen Data Karyawan merupakan aplikasi berbasis website yang dirancang untuk mengelola data lengkap karyawan. Aplikasi ini menyediakan berbagai fitur yang memudahkan penggunaannya dalam mengelola data, serta menawarkan tampilan yang sederhana dan mudah digunakan. Pembuatan aplikasi ini dibagi menjadi dua bagian, yaitu:

3.1.1 Pembuatan Antarmuka Aplikasi

Enam halaman antarmuka disediakan oleh website ini, melibatkan halaman tabel data sebagai halaman utama, dua halaman formulir pembuatan, dua halaman formulir pembaruan, dan halaman tabel pemberhentian kerja.

No	Code	Employee Name	Email	Address	NIN	Position	Phone	Action
8	KY007	Gunarso	gunarso@gmail.com	Jl.Kemaren sore No.40	2020015123111	Programmer	0895620058408	[edit] [delete]
9	KY008	Gunarsonri	gunarsonri@gmail.com	Jl.Kemaren sore No.40	123123123	Direktur	0895620058408	[edit] [delete]
10	KY009	Mellina	123@gmail.com	Jl.Kemaren sore No.47	11223344556677	Direktur	0895620058408	[edit] [delete]
11	KY010	Melari	123456@gmail.com	Jl.Kemaren sore No.47	11223344556688	Direktur Keuangan	0895620058408	[edit] [delete]
12	KY011	Maureen	maureen@gmail.com	Jl.Kemaren sore No.77	9890901000	Sekretaris	0895620058408	[edit] [delete]
15	KY014	Angung Hapsah	gungah@gmail.com	Jl.Kemaren sore No.99	9890901001	Bendahara	0895620058408	[edit] [delete]
16	KY015	Sugeng	sugeng@gmail.com	Jl.Kemaren sore No.80	9890901032	Bendahara	0895620058408	[edit] [delete]
17	KY016	Sugeng	sugeng@gmail.com	Jl.Kemaren sore No.80	9890901032	Bendahara	0895620058408	[edit] [delete]
18	KY016	Sugeng	sugeng@gmail.com	Jl.Kemaren sore No.80	9890901032	Bendahara	0895620058408	[edit] [delete]
19	KY020	Sugeng	sugeng@gmail.com	Jl.Kemaren sore No.80	9890901032	Bendahara	0895620058408	[edit] [delete]

Gambar 2. Tampilan Halaman Utama

Tampilan untuk halaman utama yang langsung menampilkan tabel berisi data – data karyawan ditunjukkan oleh Gambar 2 di atas. Pada halaman ini, terdapat menu pencarian yang digunakan untuk mencari karyawan berdasarkan nama atau kode karyawan. Selain itu, tersedia Tombol *Resign Table* untuk beralih ke halaman *resign table*, Tombol *Create*, dan juga tombol *action* yang berisi opsi *update* dan *delete* di setiap baris datanya.

No	Code	Employee Name	Email	Address	NIN	Position	Phone	reason for retirement	Action
6	KY006	Bagas Saputra	bagas@gmail.com	Jl.Kemaren sore No.29	33255209657	Programmer	0895620058408	Sakit Kanker	[edit] [delete]
1	KY001	Angela Kurniawan	angelakurniawan2024@gmail.com	Jl.Pemotongan no.5	9890901032	Sekretaris	62895123456789	Masa magang sudah selesai	[edit] [delete]
5	KY012	Maura	maura@gmail.com	Jl.Kemaren sore No.77	9890901000	Sekretaris	0895620058408	Masa magang sudah selesai	[edit] [delete]
4	KY013	Agus	agus@gmail.com	Jl.Kemaren sore No.79	9890901000	Bendahara	0895620058408	Masa magang sudah selesai	[edit] [delete]
3	KY013	Agus	agus@gmail.com	Jl.Kemaren sore No.79	9890901000	Bendahara	0895620058408	Masa magang sudah selesai	[edit] [delete]

Gambar 3. Tampilan Halaman Resign

Tampilan untuk halaman resign yang memiliki tampilan yang sama dengan tampilan halaman utama diilustrasikan oleh Gambar 3 di atas. Perbedaannya terletak pada isi tabel, di mana tabel data *resign* memiliki satu kolom tambahan yang berisi alasan pengunduran diri dari karyawan.

Gambar 4. Tampilan Halaman *Form Create*

Tampilan halaman *create* diilustrasikan oleh gambar 4. Pembentukan halaman pembuatan data dilakukan dengan menggunakan struktur formulir yang berisi label dan input, dilengkapi dengan tombol *submit* untuk membuat data dan memasukkannya ke dalam *database*. *Form create* ini juga digunakan pada tombol *create* yang terdapat di halaman *resign*, namun hanya berbeda pada judul halaman dan *endpoint*-nya saja.

Gambar 5. Tampilan Halaman *Form Update*

Tampilan halaman untuk update, yang diilustrasikan oleh gambar 5, digunakan untuk menampilkan halaman *update* pada data yang dipilih. Pada halaman ini, input pada formulir sudah diisi dengan nilai sesuai data yang ada di dalam *database*, memungkinkan pengguna untuk memperbaiki kesalahan dalam penginputan data. Bentuk dan struktur formulir ini mirip dengan formulir *create*, hanya berbeda pada *endpoint* dan metodenya yang menggunakan *"/update"*. Formulir ini juga digunakan pada halaman *update* untuk tabel *resign*.

The screenshot shows a web form with the following fields and values:

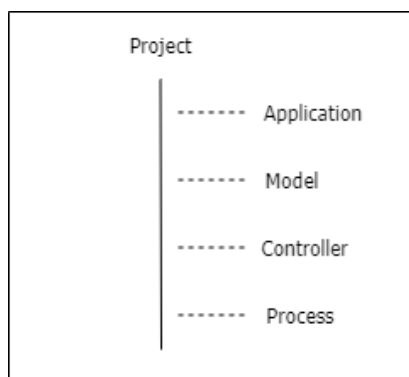
- Employee Id: KY007
- Name: Gunarno
- Email: gunarno@gmail.com
- Address: Jl.Kemaren sore No.40
- NIN: 2020015123111
- Position: Programmer
- Phone Number: 0895620058408

Gambar 6. Tampilan Halaman Detail

Tampilan pada halaman detail, sesuai dengan kode karyawan yang ditekan, diilustrasikan oleh gambar 6. Halaman ini berfungsi untuk menampilkan seluruh data karyawan secara rinci. Selain itu, halaman ini juga berguna untuk mencegah kelalaian input atau perubahan data yang tidak disengaja ketika pengguna ingin melihat data karyawan, karena pada halaman detail yang berbentuk formulir ini, data tidak dapat diubah.

3.1.2 Pembuatan *Backend* Aplikasi

Pembuatan *backend* pada *website* ini dilakukan dengan penerapan konsep *Model View Control (MVC)* yang terdiri dari empat *package*, yaitu *Application*, *Model*, *Controller*, dan *Process*, masing-masing memiliki perannya sendiri. Struktur program dapat dijelaskan sebagai berikut:



Gambar 7. *Model View Control (MVC)*

Dalam pembuatan proyek, langkah pertama yang dilakukan adalah pembuatan *package* dan *class Application*, diikuti dengan pembuatan *package Model*, *Controller*, dan *Process*, sebagaimana tergambar pada gambar 7 di atas.

Kode Program 1. Contoh Kode Program *Class Application*

```
@SpringBootApplication
public class SpringbootmybatiscrudApplication {

    public static void main(String[] args) {
        SpringApplication.run(SpringbootmybatiscrudApplication.class, args);
    }

}
```

Dalam Kode Program 1 di atas, terdapat contoh kode program untuk *class Application*. Pada awal program, anotasi *@SpringBootApplication* digunakan sebagai penanda bahwa aplikasi *Spring Boot* akan dijalankan oleh *class Application*.

Kode Program 2. Contoh Kode Program *Class Model*

```
@Getter
@Setter
public class Employee {
    private int no;
    private String employe_code;
}
```

Dalam Kode Program 2 di atas, terdapat contoh kode program dari *class Model*. *Class* ini berisi parameter dan *variable* sesuai dengan *entity database*. Anotasi *@Getter* dan *@Setter* digunakan untuk mengambil nilai dari *database* dan mengirimkan nilai ke *database*.

Kode Program 3. Contoh Kode Program *Class Controller*

```
@Controller
public class EmployeeController {
    @GetMapping("/")
    private String emp(){
        return "redirect:/emp-list";
    }
}
```

Dalam Kode Program 3 di atas, terdapat contoh kode program dari *class Controller*. Kode program ini dilengkapi dengan anotasi *@Controller* untuk menandai *class* sebagai bagian dari lapisan *control* pada *Model View Control (MVC)*. *@Controller* bertanggung jawab untuk menerima permintaan dari pengguna dan memprosesnya. *Class Controller* juga dapat menggunakan anotasi lain, seperti *@GetMapping*, yang berfungsi untuk menjalankan perintah *Get* untuk mengambil data atau mengarahkan pengguna ke *endpoint* yang berbeda.

Kode Program 4. Contoh Kode Program *Class Process*

```
@Mapper
@Service
public interface EmployeeMapper {
    @Select("SELECT * FROM employee_table")
    List<Employee> findAll();
}
```

Dalam Kode Program 4 di atas, terdapat contoh kode program untuk *class Process*. *Class Process* menggunakan anotasi *@Service* untuk menandai *Class Process* sebagai bagian dari lapisan *service* dalam *Model View Control (MVC)*. *Class Process* berfungsi sebagai tempat untuk memproses perintah dari permintaan pengguna yang berasal dari *controller*. Pada *class* ini juga digunakan anotasi *@Mapper* yang berasal dari modul *MyBatis* yang berfungsi untuk membuat *query* menggunakan anotasi yang disediakan oleh *MyBatis*.

3.2 Pengujian Sistem

Dalam bagian ini, akan dijelaskan hasil pengujian sistem dari aplikasi berbasis *website* yang telah dibuat. Metode pengujian dilaksanakan dengan menggunakan metode *Black Box*. Berikut adalah hasil pengujian sistem yang telah dilaksanakan:

Tabel 1. Hasil Pengujian Sistem

Pengujian	Hasil Yang Diharapkan	Hasil	Kesimpulan
Mencari data karyawan berdasarkan Id atau nama.	Bisa menampilkan data karyawan yang dicari.	Proses berhasil	sukses
Membuat data karyawan baru.	Data yang sudah dibuat masuk ke database dan di tampilkan di tabel.	Proses berhasil	sukses
Mengubah data karyawan.	Data yang sudah dirubah disimpan di database menimpa data sebelumnya.	Proses berhasil	sukses
Menghapus data karyawan.	Data terhapus dari database dan tampilan pada tabel	Proses berhasil	sukses
Masuk ke halaman detail karyawan.	Masuk kedalam halaman detai dan bisa menampilkan seluruh datanya sesuai dengan databasnya	Proses berhasil	sukses
Berpindah ke halaman selanjutnya pada tabel.	Bisa berpindah ke halaman tabel selanjutnya ketika tabel mencapai batas maksimum per halaman	Proses berhasil	sukses

Berdasarkan hasil pengujian yang dilakukan pada Tabel 1 di atas, dapat disimpulkan bahwa operasional aplikasi pengelolaan data karyawan berbasis *website* berjalan dengan baik dan sesuai dengan ekspektasi. Analisis data uji coba menunjukkan bahwa fungsi-fungsi utama aplikasi dapat dijalankan dengan efisien, mencapai tingkat kinerja yang diinginkan, dan memenuhi harapan yang telah diantisipasi. Kesuksesan uji coba ini menegaskan bahwa kemampuan aplikasi dalam mengelola data karyawan dapat diandalkan.

4. Kesimpulan

Aplikasi Pengelolaan Data Karyawan Berbasis *Website* dirancang untuk memfasilitasi pengelolaan data karyawan dengan mudah dan cepat. Solusi terkini ini mendukung pekerjaan karyawan dalam mengelola data dengan efisien. Dengan fokus pada manajemen data karyawan, kontribusinya terhadap pencapaian efisiensi dan efektivitas di dalam sebuah organisasi menjadi signifikan. Fitur-fitur inovatif yang disediakan menjadikan pengelolaan dan akses data lebih mudah, meningkatkan akurasi, serta memberikan dasar yang kuat untuk pengambilan keputusan yang tepat. Keunggulan Aplikasi Pengelolaan Data Karyawan Berbasis *Website* tidak hanya terletak pada kemudahan penggunaannya, tetapi juga pada dampak positifnya terhadap kinerja organisasi secara keseluruhan. Peningkatan produktivitas secara keseluruhan terlihat melalui pengalaman pemakai yang lebih baik dalam manajemen, penyimpanan, dan akses data karyawan. Dengan demikian, aplikasi ini menjadi solusi praktis untuk memenuhi kebutuhan manajemen data karyawan. Saran untuk pengembangan selanjutnya adalah meningkatkan kompleksitas dan kualitas tampilan aplikasi ini. Meskipun tampilan saat ini sudah sederhana, perlu adanya pengembangan lebih lanjut agar dapat memberikan

pengalaman visual yang lebih mendalam dan memikat. Beberapa aspek yang dapat diperhatikan dalam pengembangan tampilan ini melibatkan desain antarmuka pengguna (UI) dan desain pengalaman pengguna (UX).

5. Ucapan Terima Kasih

Peneliti berterimakasih yang sebesar – besarnya kepada Tuhan Yang Maha Esa serta pihak-pihak terkait yang turut membantu peneliti dalam mengembangkan penelitian ini.

6. Daftar Pustaka

- Aidhawani, A., & Nasution, M. I. P. (2023). Sistem Informasi Manajemen Menggunakan Database Karyawan Dalam Melakukan Perhitungan Penggajian Pada Percetakan UD Wahyu. *Jurnal Manuhara: Pusat Penelitian Ilmu Manajemen dan Bisnis*, 1(3), 41-46. DOI: <https://doi.org/10.61132/manuhara.v1i3.52>.
- Asgar, H., & Hartono, B. (2022). Rancang Bangun Rest Api Aplikasi Weshare Sebagai Upaya Mempermudah Pelayanan Donasi Kemanusiaan. *Jurnal Informatika Teknologi dan Sains (Jinteks)*, 4(1), 8-14. DOI: <https://doi.org/10.51401/jinteks.v4i1.1474>.
- Choirinisa, A. A., & Ikhwan, K. (2022). Pengaruh Penggunaan Aplikasi Digital Terhadap Efektivitas Kerja Pegawai. *Transekonomika: Akuntansi, Bisnis Dan Keuangan*, 2(5), 483-492. DOI: <https://doi.org/10.55047/transekonomika.v2i5.239>.
- Dagha, W. C. U. (2021). Web Event, Spring Boot, Java Pembangunan Aplikasi Web Event menggunakan Framework Spring Boot di PT XYZ. *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, 8(3), 1457-1469. DOI: <https://doi.org/10.35957/jatisi.v8i3.1052>.
- Darmi, Y., Chandra, S. P. D., Diana, D., & Pahrizal, P. (2023). Sistem Pengarsipan Data Pegawai Menggunakan Metode Mixturemodelling Berbasis Web. *Jurnal Media Infotama*, 19(1), 217-223. DOI: <https://doi.org/10.37676/jmi.v19i1.3507>.
- Dhika, H., Akhirina, T., Mustari, D., & Destiwati, F. (2019). Pemanfaatan Teknologi Cloud Computing sebagai Media Penyimpanan Data. *Jurnal PkM Pengabdian Kepada Masyarakat*, 2(03), 221-226. DOI: <http://dx.doi.org/10.30998/jurnalpkm.v2i03.3144>.
- Gata, W., & Gata, G. (2013). Penerapan Bahasa Pemrograman Java Dalam Sistem Informasi Penjualan Versi Desktop. *Bit (Fakultas Teknologi Informasi Universitas Budi Luhur)*, 10(1). DOI: <https://dx.doi.org/10.36080/bit.v10i1.507>.
- Ginanjari, A., Sari, W. P., & Dwipriyoko, E. (2021). Perbandingan Keandalan Operasi CRUD Menggunakan Perpaduan Spring dan MyBatis Framework serta Algoritma Cache Engine. *Jurnal Tiarsie*, 18(1), 11-18. DOI: <https://doi.org/10.32816/tiarsie.v18i1.90>.
- Pangestu, H. A., Kurniadi, D., & Septiana, Y. (2022). Aplikasi Pengelolaan Data Pegawai Berbasis REST API untuk Transfer Data Real Time dengan Framework Codeigniter. *Jurnal Algoritma*, 19(1), 313-322. DOI: <https://doi.org/10.33364/algoritma/v.19-1.1090>.

- Pranata, B. A. (2017). Perancangan Application Programming Interface (Api) Berbasis Web Menggunakan Gaya Arsitektur Representational State Transfer (Rest) Untuk Pengembangan Sistem Informasi Administrasi Pasien Klinik Perawatan Kulit.
- Putra, F. A. D. T. (2022). Rancangan Sistem Informasi Berbasis Web Untuk Pengelolaan Data Transaksi Nasabah Pada Perusahaan Pegadaian. *Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi*, 11(1), 123-130. DOI: 10.35889/jutisi.v11i1.821.
- Rumetna, M. S. (2018). Pemanfaatan cloud computing pada dunia bisnis: studi literatur. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 5(3), 305-314.
- Saputra, M. F. S., Pamungkas, A. Y. M., Faisal, S. D., & Rakhmawati, N. A. (2019). Pentingnya Memahami Etika dalam Mengambil dan Mengolah Data. *Jurnal Komunikasi Profesional*, 3(1). DOI: <https://doi.org/10.25139/jkp.v3i1.1187>.
- Umar, M. A. H. (2023). MANFAAT PENGGUNAAN DATABASE TERHADAP KINERJA BISNIS DIGITAL. *Journal Digital Business, Management, Finance, Economics (DBISMAFIEN)*, 1(2), 43-49.
- Wijaya, W. M., & Subekti, Z. M. (2019). Penerapan Aplikasi Database pada Kegiatan Manajemen Sekolah. *Jurnal Manajemen Pendidikan: Jurnal Ilmiah Administrasi, Manajemen dan Kepemimpinan Pendidikan*, 1(2), 157-166. DOI: <https://doi.org/10.21831/jump.v1i2.42352>.