

# **Implementasi NestJS dan Prisma pada pengembangan Backend Monolitik pada aplikasi web Antria**

**Proposal Tugas Akhir**

**Kelas MK Penulisan Proposal (CII4A2)**

**1302204044**

**Muhammad Rovino Sanjaya**



**Program Studi Sarjana Rekayasa Perangkat  
Lunak**

**Fakultas Informatika**

**Universitas Telkom**

**Bandung**

**2023**

## Lembar Persetujuan

Implementasi NestJS dan Prisma pada pengembangan Backend  
Monolitik pada aplikasi web Antria

*Implementation of Monolithic Backend Development with NestJS  
and Prisma for Antria's Web Application*

NIM: 1302204044  
Muhammad Rovino Sanjaya

Proposal ini diajukan sebagai usulan pembuatan tugas akhir pada  
Program Studi Sarjana Rekayasa Perangkat Lunak  
Fakultas Informatika Universitas Telkom

Bandung, 7 Desember 2023  
Menyetujui

Calon Pembimbing 1

Calon Pembimbing 2

(Dr. Mira Kania Sabariah, S.T., M.T.)  
NIP: 14770011

(Monterico Adrian, S.T., M.T.)  
NIP: 20870024

## Abstrak

Abstrak berisikan resume yang menggambarkan keseluruhan rencana TA yang akan dikerjakan yang meliputi permasalahan, metodologi dan hipotesis awal. Secara umum poin penting yang harus ada didalam abstrak sebuah proposal TA adalah sebagai berikut:

1. Deskripsi singkat permasalahan (1-2 kalimat)
2. Tujuan utama
3. Metode yang digunakan atau solusi yang ditawarkan (1-3 kalimat)
4. Rencana sumber dan/atau jenis data dan/atau studi kasus yang digunakan
5. Hipotesis awal

Selain itu, pada abstrak harus dituliskan kata kunci atau keyword, yang berisikan kata-kata yang medeskripsikan isi tulisan dan ditulis dengan huruf non kapital. Kata kunci maksimum sebanyak 6 kata.

**Kata Kunci:** keyword-1, keyword-2, keyword-3.

# Daftar Isi

<b>Abstrak</b>	<b>i</b>
<b>Daftar Isi</b>	<b>ii</b>
<b>I Pendahuluan</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Perumusan Masalah . . . . .	2
1.3 Tujuan . . . . .	3
1.4 Rencana Kegiatan . . . . .	3
1.5 Jadwal Kegiatan . . . . .	3
<b>II Kajian Pustaka</b>	<b>5</b>
2.1 NodeJs . . . . .	5
2.2 NestJs . . . . .	5
2.3 Object Relational Mapping . . . . .	5
2.4 PrismaJs . . . . .	5
2.5 Arsitektur Monolitik . . . . .	6
2.6 JSON Web Token . . . . .	6
2.7 Anti Pattern . . . . .	6
2.8 RESTful API . . . . .	6
<b>III Metodologi dan Desain Sistem</b>	<b>7</b>
3.1 Flowchart sistem . . . . .	7
3.2 Algoritma . . . . .	7
<b>Daftar Pustaka</b>	<b>10</b>
<b>Lampiran</b>	<b>12</b>

# Bab I

## Pendahuluan

Bagian pendahuluan memuat beberapa substansi sebagai berikut:

### 1.1 Latar Belakang

Dalam meningkatnya populasi di Indonesia, semakin banyak orang-orang yang mengantri untuk mendapatkan layanan di bank, restoran, rumah sakit, dan tempat penyedia jasa lainnya. Mengantri merupakan kegiatan yang membosankan dan menguras waktu. Mengantri juga dapat mempengaruhi kualitas layanan dari suatu tempat. Pelanggan yang lama mengantri mempunyai kemungkinan untuk pindah ke kompetitor, atau jika ada urusan lain yang lebih penting maka pelanggan akan keluar dari tempat antrian, meninggalkan antrian nya [8][4][16]. Solusi yang ada pada bank, kantor pos, dan rumah sakit saat ini menggunakan ticketing nomor antrian secara manual, dimana antrian yang sedang dilayani ditampilkan di layar pada ruang tunggu. Hal ini kurang efektif karena pelanggan harus berada di ruang tunggu[4].

Dengan adanya perkembangan teknologi dimana penggunaan perangkat pintar atau smartphone merupakan hal lumrah, banyak bermunculan aplikasi antrian virtual seperti Antrique, Qiwee, ExaQue dimana pengguna dapat mengantri dari jarak jauh melalui aplikasi maupun website. Para pengguna aplikasi tersebut juga bisa melakukan hal lain saat mengantri sebelum giliran nya. Namun aplikasi-aplikasi tersebut memiliki banyak kelemahan seperti *UI/UX* yang tidak bagus, sering *crash* dan *freeze*, tidak ada estimasi waktu antrian, dan masih belum ada yang berfokus ke sektor *food and beverage*.

Oleh karena itu perlunya dikembangkan sebuah aplikasi yang memiliki fitur yang sama ataupun lebih dengan menutup kekurangan-kekurangan pada aplikasi tersebut. pengembangan aplikasi yang direncanakan menggunakan arsitektur monolitik dikarenakan mudahnya untuk dibuat dan dideploy secara cepat untuk di iterasikan. Namun arsitektur monolitik memiliki kelemahan seperti sulitnya untuk di *maintenance*, *scale*, dan reliabilitasnya. Oleh karena itu perlu diperhatikan bagaimana *scope* aplikasi kedepannya dan perlunya migrasi ke arsitektur microservice [6] [7].

Dalam pengembangan aplikasi web, pemilihan bahasa pemrograman untuk

digunakan di backend sangatlah penting karena dapat mempengaruhi performa aplikasi yang dibangun. Dalam pemilihan bahasa pemrograman backend, banyak pilihan yang tersedia seperti PHP, Python, Ruby, PERL, dan banyak lagi. NodeJs merupakan tools yang memungkinkan bahasa javascript dapat dijalankan pada sisi backend. Dalam sisi performa NodeJs lebih unggul dibanding PHP dan Python dalam sisi kecepatan melayani *request* dari client [17] [11].

NestJs merupakan framework backend dari Nodejs yang menggunakan bahasa typescript, dan bisa digunakan untuk pengembangan arsitektur berbasis microservice dan monolitik, jadi jika aplikasi dikembangkan pada arsitektur monolitik dapat dengan mudah di migrasikan ke microservice. NestJs juga bisa digunakan bersamaan dengan framework PrismaJs untuk mengelola database [10]. PrismaJs merupakan framework Object Relational Mapping (ORM) [13], digunakan untuk mempercepat, dan mempermudah pengembangan aplikasi yang database nya memiliki relasi yang kompleks dan sulit di *maintenance* jika menggunakan Structured Query Language (SQL) [18].

Implementasi Application Programming Interface (API) yang digunakan adalah Representational State Transfer (RESTful) API, RESTful API adalah arsitektur untuk mempermudah komunikasi client server agar efektif untuk transaksi data. Namun pada implementasi RESTful API ada beberapa hal yang perlu diperhatikan seperti Keamanan saat transaksi atau komunikasi [3], dan Design pattern dalam penggunaan bahasa untuk endpoint API nya agar tidak terjadi Anti Pattern. Anti Pattern terjadi saat penamaan API tidak sesuai dengan fungsi, atau ada fungsi sejenis tapi penamaan nya berbeda jauh. Dengan menghindari anti pattern dapat berakibat ke aplikasi yang lebih mudah di sustain dan di maintain [1] [2].

## 1.2 Perumusan Masalah

Aplikasi antria memerlukan backend developer untuk mengimplementasikan fungsi fungsi API dan database management nya. Maka dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana merancang dan mengimplementasikan Entity Relational Diagram (ERD) pada PrismaJs.
2. Bagaimana merancang RESTful API yang aman dari anti-pattern untuk menjaga sustainability dari aplikasi.
3. Bagaimana merancang sistem keamanan aplikasi agar pengguna dapat bertransaksi dengan aman terutama pada sisi RESTful API nya.
4. Bagaimana melakukan unit testing dan integration testing pada framework NestJs.

### **1.3 Tujuan**

Tujuan dari pengerjaan Tugas Akhir ini yaitu:

1. Implementasi ERD menggunakan Prisma ORM untuk meningkatkan sustainability.
2. Merancang dan mengimplementasikan RESTful API dengan menggunakan NestJs yang menghindari anti-pattern.
3. Menjamin keamanan data user saat memakai aplikasi menggunakan JWT untuk session control dan Adaptive Security untuk RESTful API nya.
4. Menguji kualitas masing masing fungsi dengan menggunakan Unit testing dan integration testing.

### **1.4 Rencana Kegiatan**

Rencana kegiatan adalah penjelasan mengenai rencana langkah-langkah yang akan dilakukan dalam pengerjaan Tugas Akhir yang memuat: kajian pustaka, cara pengumpulan data (kualitatif, kuantitatif), rancangan penelitian (mencakup prosedur penelitian dan perancangan sistem), cara menguji hasil penelitian (cara penafsiran dan penyimpulan hasil penelitian).

### **1.5 Jadwal Kegiatan**

Jadwal pelaksanaan dibuat berdasarkan metodologi penyelesaian masalah yang digunakan. Bar-chart bisa dibuat per bulan atau per minggu. Dibawah ini adalah merupakan contoh bar-chart:

Tabel 1.1: Jadwal kegiatan proposal tugas akhir

No	Kegiatan	Bulan ke-																										
		1				2				3				4				5				6						
1	Studi Literatur																											
2	Pengumpulan Data																											
3	Analisis dan Perancangan Sistem																											
4	Implementasi Sistem																											
5	Analisa Hasil Implementasi																											
6	Penulisan Laporan																											



## Bab II

# Kajian Pustaka

### 2.1 NodeJs

NodeJs adalah runtime javascript yang basis nya dibangun dari V8 Javascript Engine. NodeJs berjalan dalam bentuk *event-driven*, dan menggunakan model non blocking I/O. meskipun menggunakan event-driven untuk menangani request, NodeJs dapat menangani jutaan koneksi dalam bersamaan secara *asynchronous* [15].

### 2.2 NestJs

NestJs merupakan framework untuk Nodejs yang dikembangkan oleh Kamil Myśliwiec yang bertujuan untuk membuat aplikasi NodeJs yang efektif dan *scalable*. NodeJs mendukung penggunaan bahasa typescript dan javascript. NestJs juga menggabungkan komponen-komponen dari Functional Programming, Object Oriented Programming, dan Functional Reactive Programming [12] [10].

### 2.3 Object Relational Mapping

Object Relational Mapping (ORM) adalah sebuah teknologi yang memetakan table database ke dalam objek, biasanya dipakai dalam bahasa yang berbasis Object Oriented Programming. Dengan menggunakan ORM, developer dapat berfokus ke business logic tanpa mengkhawatirkan penggunaan akses database yang rumit [9].

### 2.4 PrismaJs

Prisma adalah ORM Open Source, biasanya digunakan sebagai alternatif dari menggunakan Structured Query Language (SQL) secara langsung. Prisma Mendukung penggunaan database MySQL, PostgreSQL, SQLite, SQL Server, CockroachDB, dan MongoDB. Prisma digunakan untuk mempermudah pengembangan database yang memiliki relasi yang kompleks dan besar, dengan cara memberikan API yang type-safe untuk query database nya dan mengembalikan hasil query dalam bentuk javascript Object Notation (JSON) [13].

## 2.5 Arsitektur Monolitik

Arsitektur Monolitik adalah arsitektur sebuah software dimana beberapa fungsi komponen yang berbeda seperti fungsi otorisasi, business logic, notifikasi, pembayaran. Berada dalam satu program dan platform yang sama [6].

## 2.6 JSON Web Token

JSON Web Token (JWT) adalah sebuah token berbentuk string json yang dapat digunakan untuk melakukan otorisasi. Ukuran JWT tergolong kecil jadi dapat dengan cepat di transfer antar client dan server. JWT menggunakan algoritma HMAC atau RSA untuk meng-enkripsi digital signature yang digunakan. JWT memiliki 3 bagian pada string nya yang dipisahkan menggunakan ".", bagian ini berupa *header*, *payload*, dan *signature* [14].

## 2.7 Anti Pattern

Anti Pattern terjadi jika pembuatan nama sebuah objek tidak konsisten dengan yang lain. Objek disini dapat berupa endpoint API, nama variable, nama fungsi, dan nama lain yang penggunaan nya bersifat publik. Terjadinya anti pattern dapat mengakibatkan sulitnya untuk mengkomprehensi suatu aplikasi [1] [2].

## 2.8 RESTful API

Representational State Transfer (RESTful) Application Programming Interface (API) adalah arsitektur untuk mempermudah komunikasi client server agar efektif untuk transaksi data, tipe data yang paling sering digunakan untuk transaksi client server adalah JSON. Karakteristik RESTful seperti : 1) Client Server, 2) Stateless, 3) Layered Architecture, 4) Caching, 5) Code on Demand, dan 6) Uniform Interface [5]

## **Bab III**

### **Metodologi dan Desain Sistem**

Judul bab 3 dapat dipilih salah satu yaitu Perancangan sistem atau Alur Pemodelan. Penjelasan untuk setiap judul pada bab 3 dijelaskan sebagai berikut : Perancangan Sistem berisi rancangan dari sistem yang akan dibangun, berupa diagram block proses atau flowchart beserta penjelasannya. Rancangan sistem dapat berisikan rencana kebutuhan data (pengumpulan dan pre-processing data), serta skenario pengujian yang akan dilakukan. Ilustrasi proses pengolahan data dapat ditambahkan dalam bagian ini untuk memperjelas kegiatan yang anda lakukan dalam pengerjaan TA. Alur Pemodelan berisikan alur pembuatan model, dapat berupa diagram block proses atau flowchart, beserta penjelasannya. Alur model berisikan rencana tahapan yang akan dilakukan berdasarkan metode yang dipilih, termasuk di dalamnya metode dan skenario pengujian yang akan dilakukan. Ilustrasi masing-masing tahapan dapat ditambahkan dalam bagian ini untuk memperjelas kegiatan yang anda lakukan dalam pengerjaan TA.

#### **3.1 Flowchart sistem**

#### **3.2 Algoritma**

Atau dalam bentuk algoritma seperti contoh pada Algoritma 1 berikut ini:

---

**Algorithm 1** Prosedur simulasi dinamika lalu lintas menggunakan FVDM.

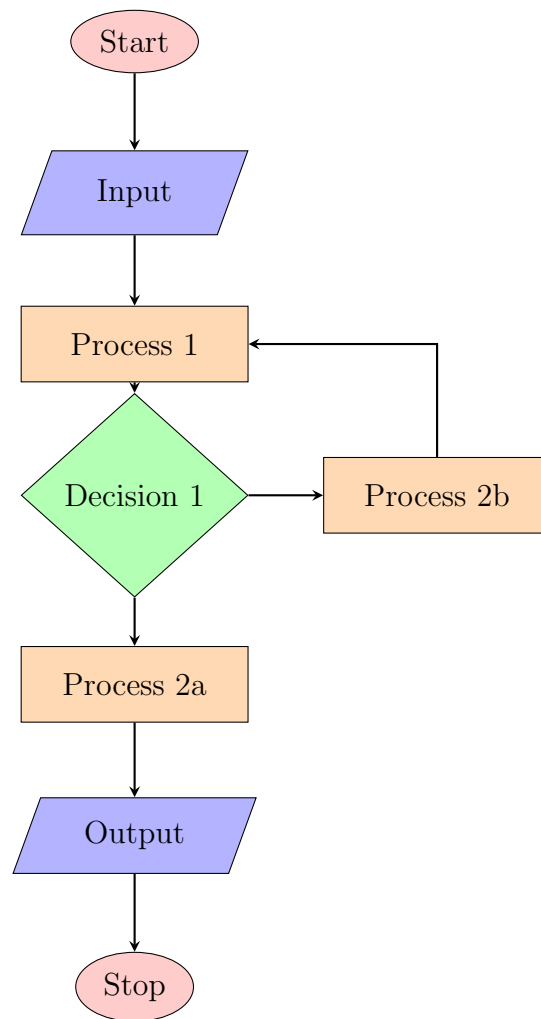
---

```

1: procedure FVDM( $T_{final}, \Delta t$ )
2:   Start
3:   For  $n = 1 : N$  do                                      $\triangleright$  Pemberian nilai awal
4:     Input nilai  $x[n]$ 
5:     Input nilai  $v[n]$ 
6:   EndFor
7:   time=0
8:   while  $time < T_{final}$  do
9:      $time = time + \Delta t$ 
10:    Hitung jarak bumper menggunakan rumus untuk  $n = 2, \dots, N$ 
11:    If ( $S(n) \leq 0m$ ) then return End If.
12:    Tentukan  $\lambda$  menggunakan.
13:    Hitung kecepatan optimal  $v_o(t)$  menggunakan.
14:    Hitung percepatan  $a_n(time)$  menggunakan .
15:    Hitung kecepatan baru dengan  $v_n(time) = v_n(time - \Delta t) +$ 
       $a_n(time)\Delta t$ .
16:    Hitung posisi baru dengan  $x_n(time) = x_n(time - \Delta t) +$ 
       $v_n(time)\Delta t$ .
17:    If ( $\Delta v \leq 10^{-5} \& \& a_n(time) \leq 10^{-5}$ ) then
18:      OUTPUT Cetak hasil data  $a_n, v_n, x_n$ .
19:    return.
20:    End If.
21:  end while
22:  End
23: end procedure

```

---



Gambar 3.1: Caption flowchart

## Daftar Pustaka

- [1] AGHAJANI, E., NAGY, C., BAVOTA, G., AND LANZA, M. A large-scale empirical study on linguistic antipatterns affecting apis. In *Soft System Stakeholder Analysis Methodology* (11 2018), Institute of Electrical and Electronics Engineers Inc., pp. 25–35.
- [2] ALSHRAIEDEH, F. S., AND KATUK, N. A uri parsing technique and algorithm for anti-pattern detection in restful web services. *International Journal of Web Information Systems* 17 (1 2021), 1–17.
- [3] BEER, M. I., AND HASSAN, M. F. Adaptive security architecture for protecting restful web services in enterprise computing environment. *Service Oriented Computing and Applications* 12 (6 2018), 111–121.
- [4] GHAZAL, M., HAMOUDA, R., AND ALI, S. A smart mobile system for the real-time tracking and management of service queues. *International Journal of Computing and Digital Systems* 5 (7 2016), 305–313.
- [5] GIESSLER, P., GEBHART, M., SARANCIN, D., STEINEGGER, R., AND ABECK, S. Best practices for the design of restful web services. In *International Conferences of Software Advances (ICSEA)* (2015), pp. 392–397.
- [6] GOS, K., AND ZABIEROWSKI, W. The comparison of microservice and monolithic architecture. In *2020 IEEE XVIth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH)* (2020), IEEE, pp. 150–153.
- [7] JATKIEWICZ, P., AND OKRÓJ, S. Differences in performance, scalability, and cost of using microservice and monolithic architecture. In *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing* (2023), pp. 1038–1041.
- [8] KHONG, Y. L., OOI, B. C., TAN, K. E., IBRAHIM, S. A. B., AND TEE, P. L. E-queue mobile application. In *SHS Web of Conferences* (2017), vol. 33, EDP Sciences, p. 00033.

- [9] LORENZ, M., RUDOLPH, J.-P., HESSE, G., UFLACKER, M., AND PLATTNER, H. Object-relational mapping revisited-a quantitative study on the impact of database technology on o/r mapping strategies. *Proceedings of the 50th Hawaii International Conference on System Sciences* (2017).
- [10] MYSLIWIEC, K. Nestjs documentation.
- [11] ODENIRAN, Q., WIMMER, H., AND REBMAN, C. M. Node.js or php? determining the better website server backend scripting language. *Issues in Information Systems* 24 (2023), 328–341.
- [12] PHAM, A. D. Developing back-end of a web application with nestjs framework: Case: Integrify oy’s student management system. *Theseus* (2020).
- [13] PRISMA DATA, I. Prismajs documentation.
- [14] RAHMATULLO, A., ALDYA, A. P., AND ARIFIN, M. N. Stateless authentication with json web tokens using rsa-512 algorithm. *JURNAL INFOTEL* 11, 2 (2019), 36–42.
- [15] SHAH, H., AND SOOMRO, T. R. Node. js challenges in implementation. *Global Journal of Computer Science and Technology* 17, 2 (2017), 73–83.
- [16] UDDIN, M. N., RASHID, M., MOSTAFA, M., SALAM, S., NITHE, N., AND AHMED, S. Z. Automated queue management system, 2016.
- [17] WILLIAM, G., ANTHONY, R., AND PURNAMA, J. Development of node-js based backend system with multiple storefronts for batik online store. *ACM International Conference Proceeding Series* (2020). Cited by: 0.
- [18] ZMARANDA, D., POP-FELE, L.-L., GYÖRÖDI, C., GYÖRÖDI, R., AND PECHERLE, G. Performance comparison of crud methods using net object relational mappers: A case study, 2020.

## Lampiran