

**LAPORAN PROJEK AKHIR**  
**MATA KULIAH PEMROGRAMAN BERBASIS PLATFORM**



WIDYA OHIBOR	20240040250
--------------	-------------

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**UNIVERSITAS NUSA PUTRA**

**2026**

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Perkembangan teknologi informasi saat ini mendorong kebutuhan akan sistem informasi yang cepat, akurat, dan terintegrasi. Salah satu kebutuhan yang sering dijumpai dalam kehidupan sehari-hari adalah sistem pemesanan tiket untuk berbagai acara, seperti konser musik, seminar, dan event publik lainnya. Proses pemesanan tiket secara manual dinilai kurang efisien karena membutuhkan waktu lebih lama dan berpotensi menimbulkan kesalahan dalam pengelolaan data.

Oleh karena itu, diperlukan sebuah sistem pemesanan tiket berbasis backend API yang mampu mengelola data pengguna, event, serta transaksi pemesanan secara terstruktur dan aman. Selain itu, untuk meningkatkan kualitas dan kelengkapan data event, sistem backend dapat diintegrasikan dengan API publik dari pihak ketiga. Salah satu API publik yang dapat dimanfaatkan adalah Ticketmaster API, yang menyediakan data event secara real-time dari berbagai wilayah.

Dengan adanya integrasi API publik, sistem tidak hanya bergantung pada data internal, tetapi juga mampu menampilkan informasi event terbaru yang bersumber langsung dari penyedia resmi. Hal ini diharapkan dapat meningkatkan keandalan sistem serta memberikan pengalaman yang lebih baik bagi pengguna.

### **1.2. Rumusan Masalah**

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan masalah dalam pengembangan sistem ini adalah sebagai berikut:

1. Bagaimana merancang dan membangun backend API untuk sistem pemesanan tiket berbasis Node.js dan Express.js?
2. Bagaimana mengintegrasikan API publik Ticketmaster ke dalam sistem backend menggunakan API Key?
3. Bagaimana cara mengelola endpoint backend agar dapat diakses dan diuji menggunakan Postman?
4. Bagaimana sistem backend menangani pengambilan data event publik secara aman dan terstruktur?

### **1.3. Tujuan Penelitian**

Tujuan dari pengembangan sistem backend pemesanan tiket ini adalah:

1. Membangun backend API yang mampu mengelola data pengguna, event, dan pemesanan tiket.
2. Mengintegrasikan API publik Ticketmaster untuk memperoleh data event secara real-time.
3. Menerapkan penggunaan API Key sebagai mekanisme keamanan dalam integrasi API publik.

4. Menyediakan endpoint backend yang dapat diuji menggunakan Postman untuk keperluan pengujian dan dokumentasi.

#### 1.4. Manfaat Penelitian

Adapun manfaat dari pengembangan sistem ini adalah sebagai berikut:

##### 1. Bagi Mahasiswa

- Menambah pemahaman mengenai pengembangan backend API menggunakan Node.js dan Express.js.
- Melatih kemampuan integrasi API publik dengan sistem internal.
- Memahami penggunaan API Key dan pengelolaan environment variable.

##### 2. Bagi Pengguna

- Memahami penggunaan API Key dan pengelolaan environment variable.
- Mempermudah proses pemesanan tiket melalui sistem terkomputerisasi.

##### 3. Bagi Pengembang Sistem

- Menjadi dasar pengembangan sistem pemesanan tiket yang lebih kompleks.
- Dapat dikembangkan lebih lanjut dengan fitur pembayaran dan notifikasi.

#### 1.5. Ruang Lingkup

Ruang lingkup dalam pengembangan sistem ini meliputi:

- Pembuatan backend API menggunakan Node.js dan Express.js.
- Integrasi API publik Ticketmaster menggunakan API Key.
- Pengelolaan endpoint untuk pengambilan data event publik.
- Pengujian endpoint menggunakan Postman.
- Sistem difokuskan pada sisi backend dan tidak mencakup pengembangan frontend.

## BAB II

### PEMBAHASAN

#### 2.1. Sistem Pemesanan Tiket Berbasis Backend API

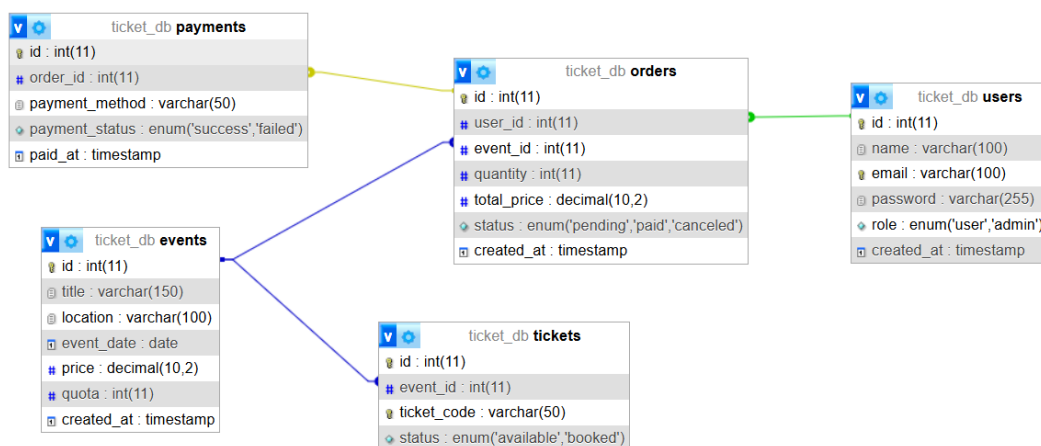
Sistem pemesanan tiket merupakan suatu sistem informasi yang dirancang untuk mengelola proses pemilihan event, pemesanan tiket, hingga pencatatan transaksi secara terkomputerisasi. Dalam perkembangannya, sistem pemesanan tiket modern umumnya dibangun menggunakan arsitektur client-server, di mana backend API berperan sebagai pengelola utama logika bisnis dan data.

Sistem pemesanan tiket merupakan suatu sistem informasi yang dirancang untuk mengelola proses pemilihan event, pemesanan tiket, hingga pencatatan transaksi secara terkomputerisasi. Dalam perkembangannya, sistem pemesanan tiket modern umumnya dibangun menggunakan arsitektur client-server, di mana backend API berperan sebagai pengelola utama logika bisnis dan data.

Pada sistem pemesanan tiket, backend API berfungsi sebagai penghubung antara database dan client. Backend akan menerima permintaan (request) dari client, memproses data sesuai dengan kebutuhan, lalu mengirimkan hasil (response) dalam format yang mudah dipahami, seperti JSON. Pendekatan ini memungkinkan sistem untuk menangani banyak pengguna secara bersamaan dengan lebih efisien.

Selain itu, penggunaan backend API juga meningkatkan aspek keamanan sistem. Data sensitif seperti informasi pengguna dan transaksi tidak langsung diakses oleh client, melainkan dikelola melalui API yang telah dilengkapi dengan mekanisme autentikasi dan otorisasi. Dengan demikian, sistem pemesanan tiket menjadi lebih aman dan andal.

#### 2.2. Database



## 2.3. CMD

```
Command Prompt
Microsoft Windows [Version 10.0.26100.7462]
(c) Microsoft Corporation. All rights reserved.

C:\Users\AHMET>cd Documents
C:\Users\AHMET\Documents>cd ProjectPBP
The system cannot find the path specified.
C:\Users\AHMET\Documents>mkdir ProjectPBP
C:\Users\AHMET\Documents>cd ProjectPBP
C:\Users\AHMET\Documents\ProjectPBP>npm init -y
Wrote to C:\Users\AHMET\Documents\ProjectPBP\package.json:

{
  "name": "projectpbp",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

C:\Users\AHMET\Documents\ProjectPBP>npm install express mysql
added 75 packages, and audited 76 packages in 4s

22 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\Users\AHMET\Documents\ProjectPBP>npm install express mysql2 jsonwebtoken bcryptjs dotenv axios
```

```
C:\Users\AHMET\Documents\ProjectPBP>npm install express mysql
added 75 packages, and audited 76 packages in 4s

22 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\Users\AHMET\Documents\ProjectPBP>npm install express mysql2 jsonwebtoken bcryptjs dotenv axios
added 36 packages, and audited 112 packages in 3s

26 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

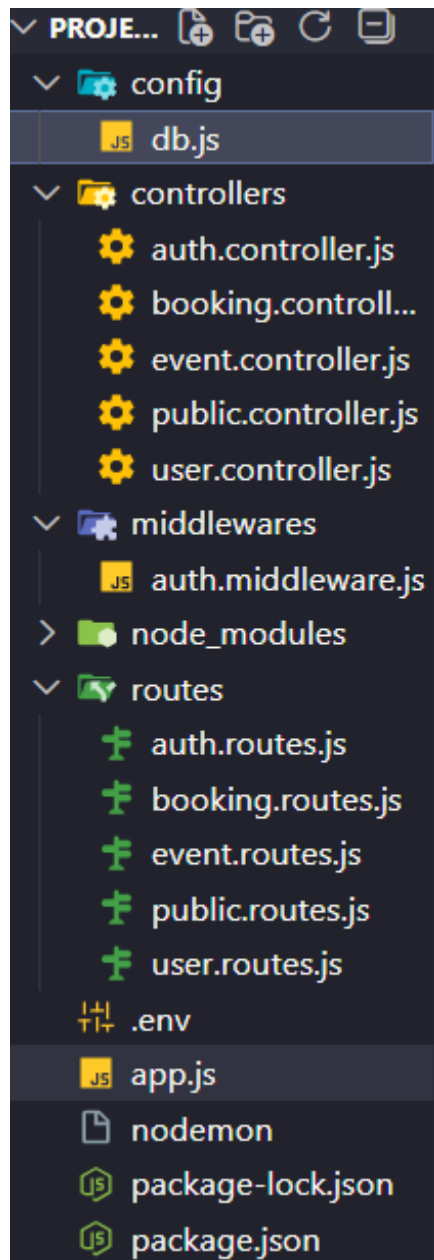
C:\Users\AHMET\Documents\ProjectPBP>npm install nodemon --save-dev
added 26 packages, and audited 138 packages in 4s

30 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\Users\AHMET\Documents\ProjectPBP>code .
```

## 2.4. Struktur Folder



## **BAB III PENUTUP**

### **3.1. Kesimpulan**

Berdasarkan hasil perancangan dan implementasi sistem autentikasi menggunakan JSON Web Token (JWT), dapat disimpulkan bahwa JWT merupakan metode yang efektif untuk mengamankan akses ke suatu aplikasi berbasis web atau API. JWT memungkinkan proses autentikasi dilakukan secara stateless, sehingga server tidak perlu menyimpan sesi pengguna secara terus-menerus.

Pada implementasinya, proses autentikasi dimulai dari registrasi dan login pengguna. Data password dienkripsi menggunakan bcrypt untuk menjaga keamanan informasi pengguna. Setelah pengguna berhasil login, sistem akan menghasilkan token JWT yang berisi informasi pengguna dan memiliki batas waktu kedaluwarsa sesuai dengan konfigurasi yang ditentukan pada file environment.

Middleware autentikasi berperan penting dalam memvalidasi token yang dikirimkan melalui header Authorization. Jika token tidak ditemukan, tidak valid, atau telah kedaluwarsa, maka sistem akan menolak akses dan mengembalikan pesan kesalahan yang sesuai. Hal ini membuktikan bahwa middleware JWT mampu meningkatkan keamanan dengan memastikan hanya pengguna yang terautentikasi yang dapat mengakses endpoint tertentu.

Dengan demikian, penerapan JWT pada sistem ini telah berhasil memberikan mekanisme autentikasi dan otorisasi yang aman, efisien, serta mudah dikembangkan lebih lanjut, terutama jika nantinya diintegrasikan dengan database dan fitur keamanan tambahan lainnya.