**Laporan Tugas Besar**

**IF-929 / Advanced Java Technology**

Semester Genap 2019/2020

Dosen: Firhat Hidayat, S.Kom., M.T.



**Disusun oleh:**

|  |  |
| --- | --- |
| 1117002 | Dionisius Pratama |
| 1117019 | Richard Denilson |
| 1117022 | Irvandi Gian Wiryawan |

**PROGRAM STUDI INFORMATIKA**

**INSTITUT TEKNOLOGI HARAPAN BANGSA**

**2020**

# DAFTAR ISI

[DAFTAR ISI 2](#_Toc39068824)

[BAB 1 - PENDAHULUAN 3](#_Toc39068825)

[**1.1.** **Latar Belakang** 3](#_Toc39068826)

[**1.2.** **Identifikasi Masalah** 3](#_Toc39068827)

[**1.3.** **Batasan Masalah** 3](#_Toc39068828)

[BAB 2 - ANALISIS DAN STUDI KASUS 4](#_Toc39068829)

[**2.1.** **Fitur Aplikasi** 4](#_Toc39068830)

[**2.2.** **Analisis Sistem yang Berjalan** 4](#_Toc39068831)

[***2.3.*** ***Unified Modelling Language*** 8](#_Toc39068832)

[**2.3.1.** **Use Case Diagram** 8](#_Toc39068833)

[**2.3.2.** **Activity Diagram** 9](#_Toc39068834)

[**2.4.** **Rancangan Basis Data** 10](#_Toc39068835)

[**2.5.** **Katalog API** 11](#_Toc39068836)

[**2.6.** **Collections (Postman)** 31](#_Toc39068837)

# BAB 1 - PENDAHULUAN

1. **Latar Belakang**

Seiring dengan perkembangan teknologi web, muncul kebutuhan komunikasi antar *platform* yang mudah diakses dan dijalankan antara banyak aplikasi. Atas dasar pemikiran ini, muncullah konsep API (*Application Programming Interface*), yang memungkinkan developer untuk mengintegrasikan dua atau lebih bagian dari aplikasi atau dengan aplikasi yang berbeda secara bersamaan. Perancangan *backend* API sederhana yang melibatkan proses CRUD (*Create, Read, Update,* dan *Delete*) dan memiliki struktur yang jelas serta aman menjadi diperlukan pada pembuatan *website*.

Membuat API dapat dilakukan melalui platform mana pun. Dalam tugas besar ini, untuk memenuhi *backend* API dibangun menggunakan bantuan *framework* Spring yang berbasis bahasa pemrograman Java.

1. **Identifikasi Masalah**

Manajemen rumah sakit memerlukan sebuah sistem di mana alur transaksi data dapat dikendalikan dan di-*track* dengan baik. Rumah sakit memiliki banyak aspek proses bisnis, dan salah satunya adalah rekam data medis, yang menjadi *core* bisnis rumah sakit selain *invoicing*. Mengelola rekam data medis dianggap cukup kompleks karena banyak melibatkan entitas data, seperti pasien, dokter, obat, riwayat kunjungan, dan sebagainya. Hubungan antarentitas tersebut harus dipastikan terdefinisi dengan baik, sehingga integritas data dapat dijaga, dan data yang dihasilkan dapat berguna bagi penggunanya.

Pada tugas besar ini, kelompok mencoba melakukan analisa kebutuhan-kebutuhan tersebut untuk dapat membuat sebuah sistem sederhana yang dapat mengelola data rekam medis dengan lebih mudah dan teratur. Sistem yang dihasilkan dapat menangani pengelolaan data rekam medis secara sederhana, dengan melakukan integrasi antar beberapa entitas.

1. **Batasan Masalah**

Dalam penyusunan aplikasi untuk pengolahan rekam data medis ini, ada beberapa hal yang belum dapat dilakukan dalam sistem, yaitu:

1. Memecah pengguna dalam role-nya masing-masing, sehingga Administrator bertanggung jawab penuh atas seluruh alur data.
2. Fungsi validasi untuk melakukan login user.
3. Memanfaatkan fitur *user interface*

# BAB 2 - ANALISIS DAN STUDI KASUS

1. **Fitur Aplikasi**

Pada aplikasi manajemen rumah sakit ini, ada 8 buah fungsi yang semuanya dikontrol oleh administrator, yaitu:

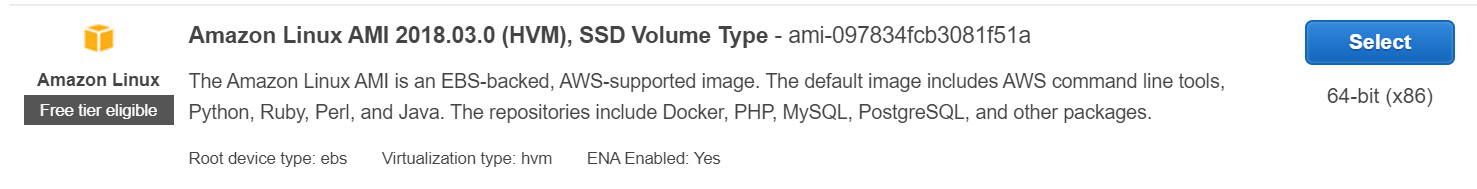
|  |  |  |
| --- | --- | --- |
| **No.** | **Nama Fitur** | **Keterangan** |
| 1 | *Patient management* | Mengelola data pasien dan melakukan operasi CRUD pada table database terkait |
| 2 | *Medical staff management* | Mengelola data staf medis dan melakukan operasi CRUD pada table database terkait |
| 3 | *Disease management* | Mengelola data penyakit yang dapat/sudah ditangani rumah sakit dan melakukan operasi CRUD pada table database terkait |
| 4 | *Medicine data management* | Mengelola data obat yang dimiliki rumah sakit dan melakukan operasi CRUD pada table database terkait |
| 5 | *Medical record management* | Mengelola data rekam medis pasien dan melakukan operasi CRUD pada table database terkait |
| 6 | *Medicine list* | Mengelola data obat yang masuk ke rekam medis pasien dan melakukan operasi CRUD pada table database terkait |
| 7 | *Appointment management* | Mengelola data perjanjian antara pasien dan dokter serta melakukan operasi CRUD pada table database terkait |
| 8 | *Schedule management* | Mengelola data jadwal *availability* dokter serta melakukan operasi CRUD pada table database terkait |

1. **Analisis Sistem yang Berjalan**

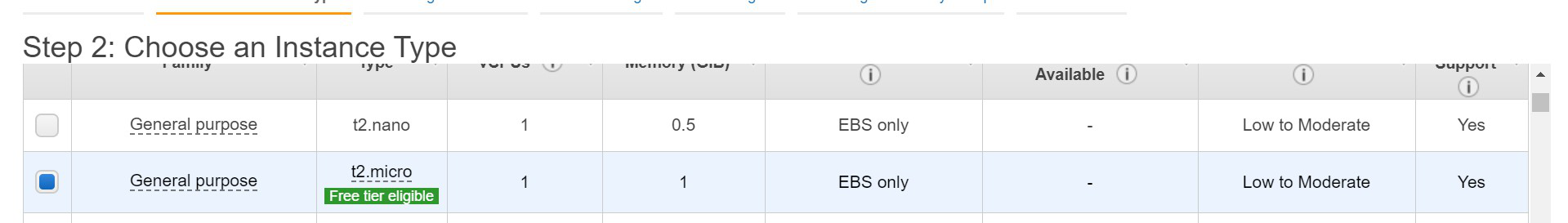
Sistem saat ini sudah dapat berjalan di *cloud*. Untuk mengakses *instance,* sistem menggunakan 1 *super user* yang berperan sebagai administrator untuk mengurus semua transaksi dan lalu lintas data yang ada di dalam sistem. Admin dapat melakukan seluruh operasi *create, update, insert, delete* (CRUD) di semua tabel database. Adapun database yang digunakan adalah MySQL yang di-*deploy* ke *cloud* menggunakan bantuan *instance* Amazon Web Service.

Instance yang dibangun memiliki beberapa perincian sebagai berikut:

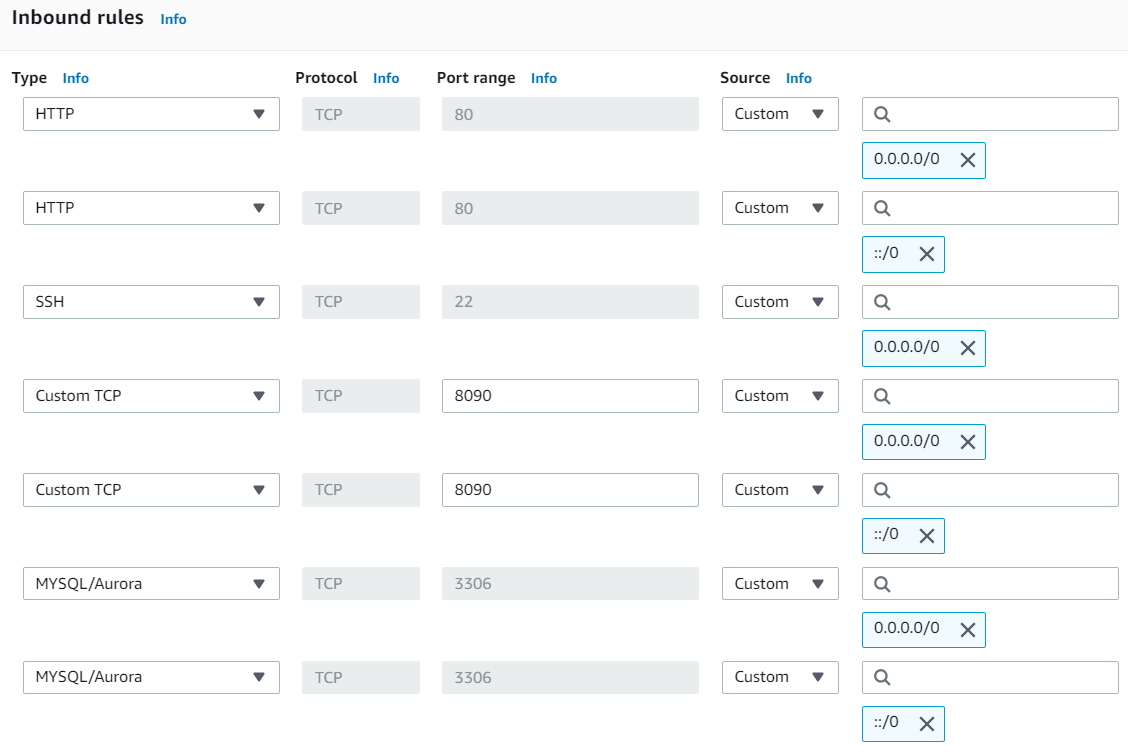
* + - 1. Sistem operasi AMI



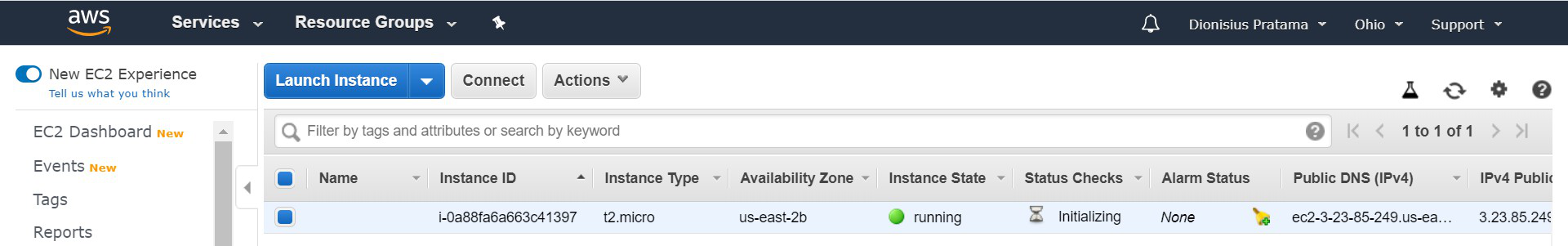
* + - 1. Instance type



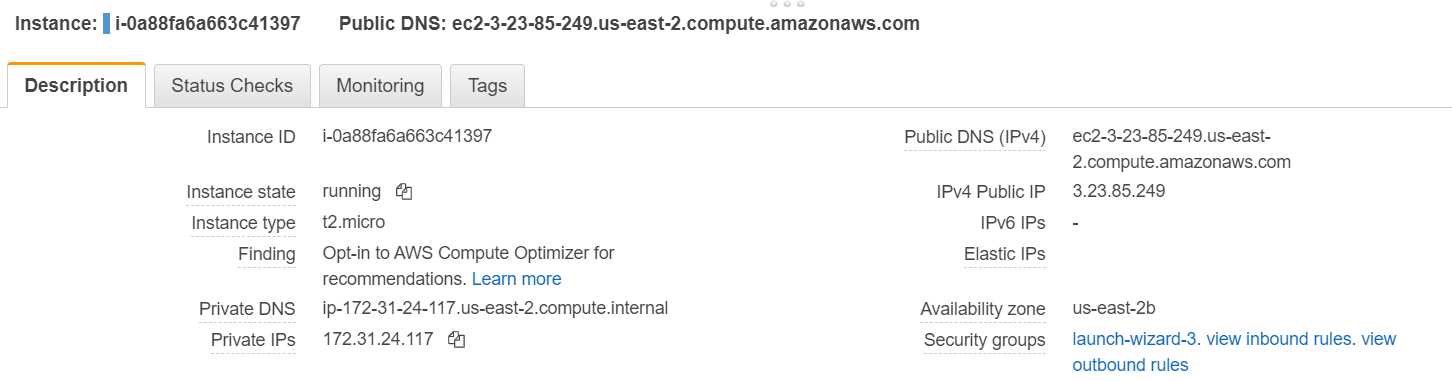
* + - 1. Security Groups setting



* + - 1. Instance preview

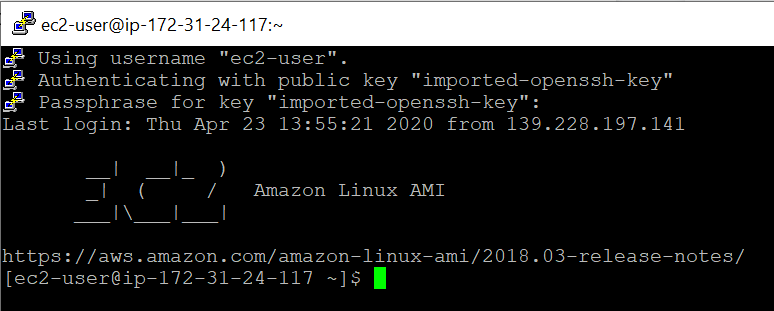


* + - 1. IP Address:



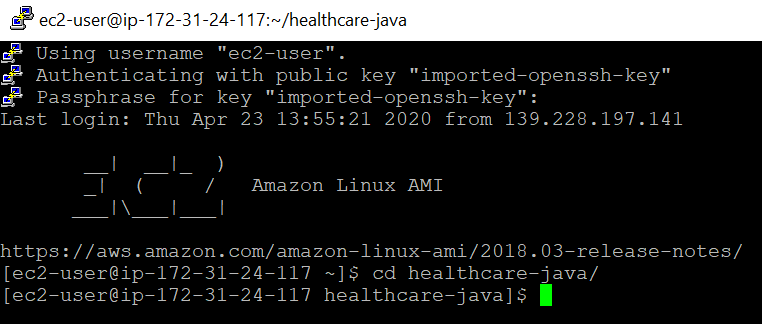
Akses ke Database dan Instance menggunakan PuTTY console:

1. Akses ke Instance

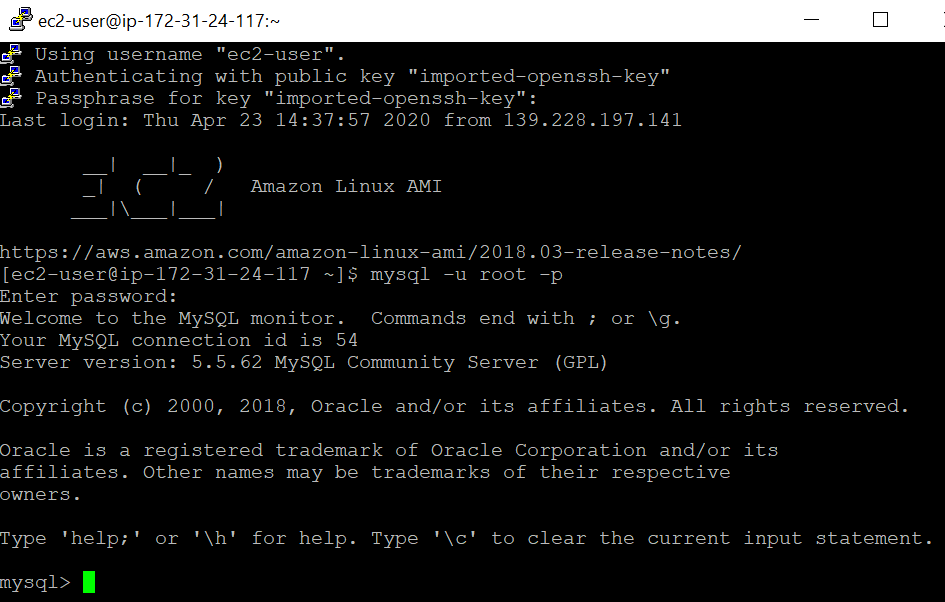


1. Akses ke Project

Project disimpan di github dengan link: <https://github.com/dionisiuspr/healthcare-java>. Lalu, project diletakkan juga di git dalam AMI.



1. Akses ke Database

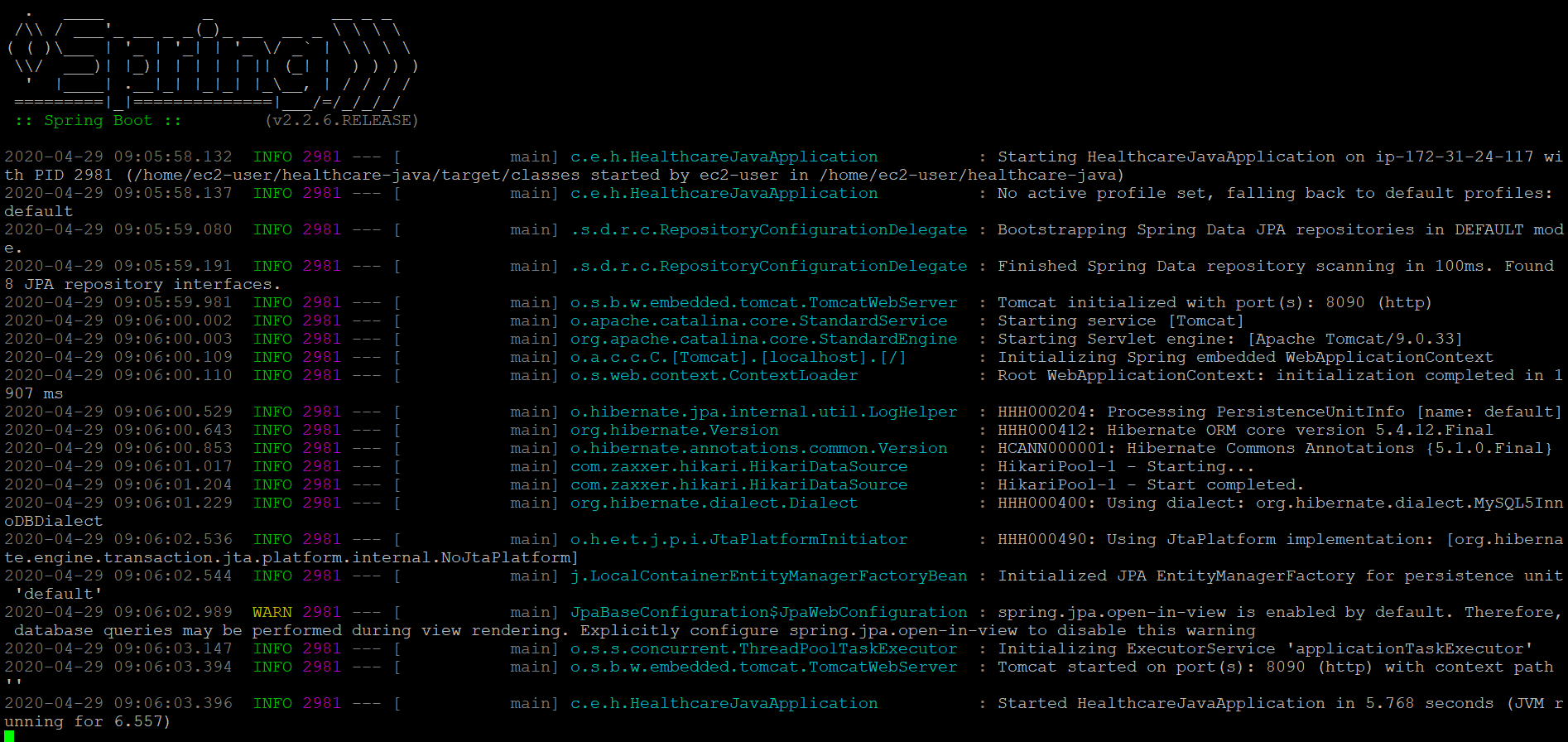


1. Run project

Sebelum dirun <mvn spring-boot:run>, pada instance, harus lebih dahulu diinstall java jdk (kelompok menggunakan version 8) dan apache-maven (kelompok menggunakan versi 3.6.3)



Jika project sudah berjalan, maka akan ada tertulis “started” pada console PuTTY seperti berikut:

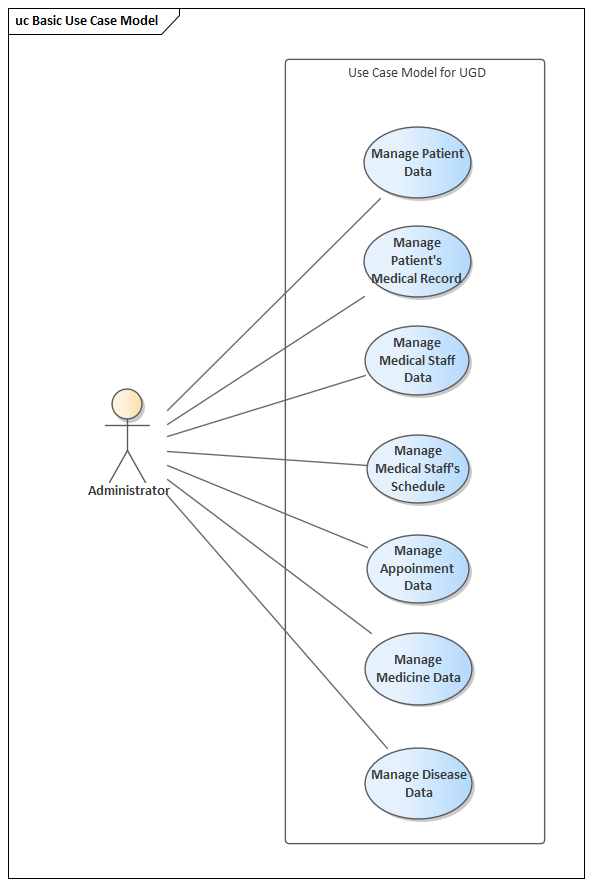


1. ***Unified Modelling Language***

Berikut adalah beberapa gambaran UML sistem yang berjalan untuk aplikasi ini.

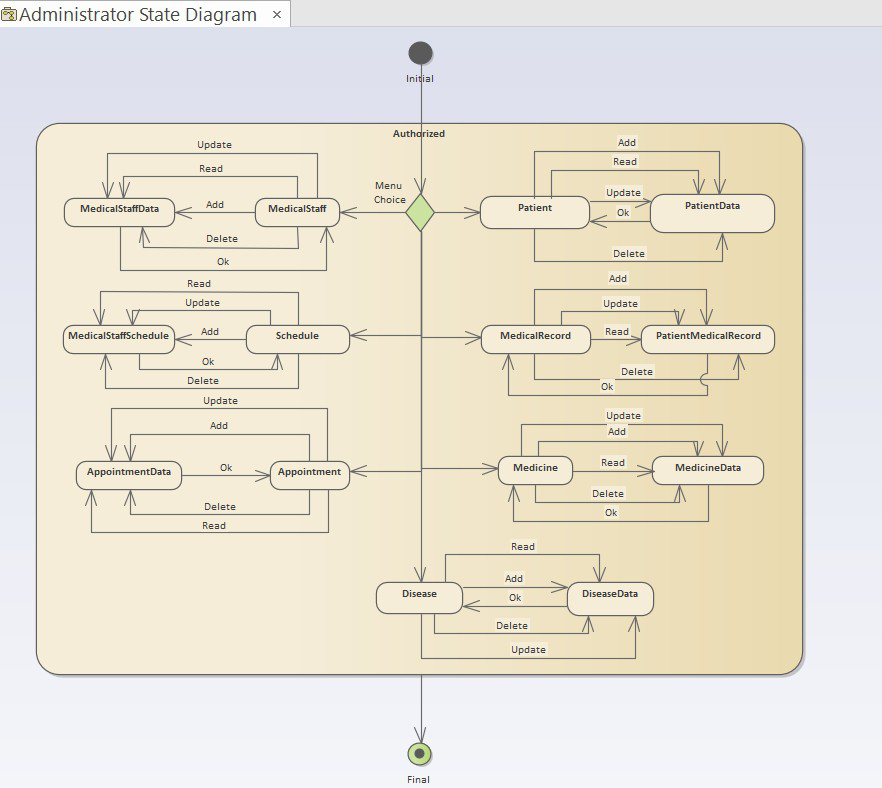
1. **Use Case Diagram**

Berikut adalah use case diagram dari sistem yang dibangun. Sistem memakai 1 super user yang berperan sebagai administrator dan dapat mengontrol semua transaksi data yang ada



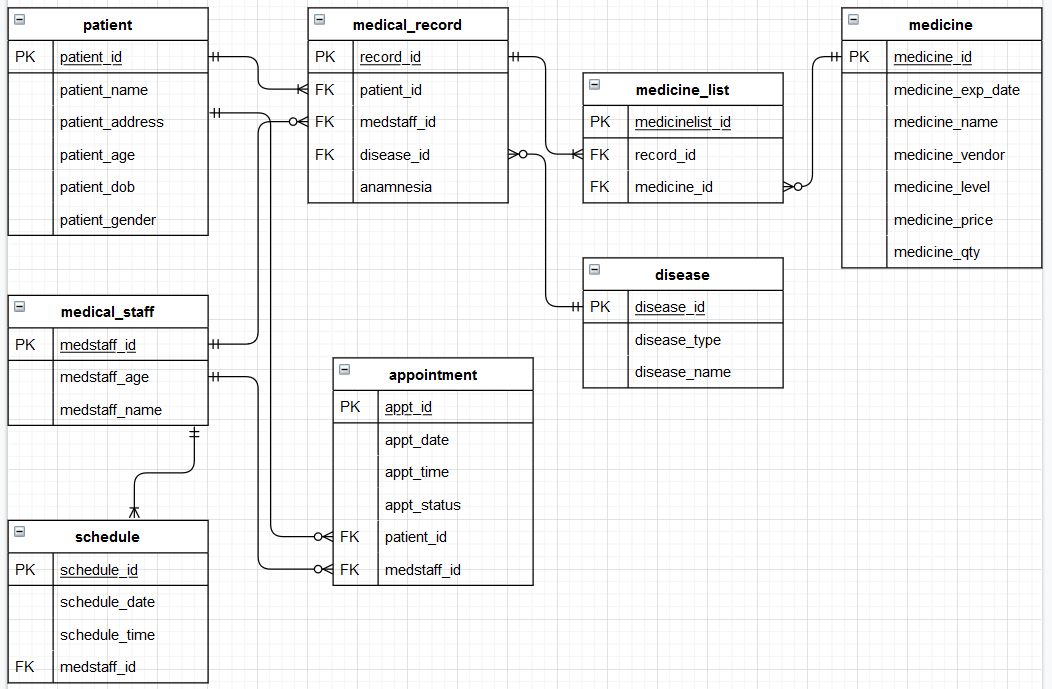
1. **Activity Diagram**

Berikut adalah diagram aktivitas yang berjalan di sistem saat ini.



1. **Rancangan Basis Data**

Bab ini berisi rancangan basis data. Rancangan digambarkan dalam bentuk *Entity Relationship Diagram* (ERD). Seluruh akses ke database dilakukan melalui AWS Instance, seperti yang sudah dijelaskan pada bagian sebelumnya.



1. **Katalog API**

Berikut adalah daftar API yang digunakan dalam aplikasi. List API ini juga dapat dilihat pada file .xlsx yang dikumpulkan pada *link* berikut: <https://classroom.google.com/u/0/c/NTI1MzQ2OTg3MzBa/a/OTI5OTg1MTc5MTZa/details>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nama API** | **Fungsi** | **Base URL** | **Method** | **Parameter** | **Sample Output** |
| **1** | Add Patient | addNewPatient() | 3.23.85.249:8090/healthcare/patient/add | POST | patient\_name, patient\_address, patient\_age, patient\_dob, patient\_gender | {  "patient\_id": 11,  "patient\_name": "Ricad",  "patient\_address": "Holis",  "patient\_age": 23,  "patient\_dob": "1997/05/26",  "patient\_gender": "M"  } |
| **2** | Add Appointment | addAppointment() | 3.23.85.249:8090/healthcare/appointment/add | POST | appointment\_date, appointment\_time, appointment\_status, medstaff\_id, patient\_id | {  "appointment\_id": 9,  "appointment\_date": "2020/09/12",  "appointment\_time": "19:21",  "appointment\_status": true,  "medicalStaff": {  "medstaff\_id": 3,  "medstaff\_age": 40,  "medstaff\_name": "Bambang",  "hibernateLazyInitializer": {}  },  "patient": {  "patient\_id": 1,  "patient\_name": "Dionis",  "patient\_address": "Kopo",  "patient\_age": 66,  "patient\_dob": "1982-09-12",  "patient\_gender": "M",  "hibernateLazyInitializer": {}  }  } |
| **3** | Add Medical Staff | addNewMedstaff() | 3.23.85.249:8090/healthcare/medstaff/add | POST | medstaff\_name, medstaff\_age | {  "medstaff\_id": 12,  "medstaff\_age": 38,  "medstaff\_name": "Yohan"  } |
| **4** | Add Disease | addNewDiesease() | 3.23.85.249:8090/healthcare/disease/add | POST | disease\_type, disease\_name | {  "disease\_id": 13,  "disease\_type": "infectious",  "disease\_name": "covid-19"  } |
| **5** | Add Medicine | addNewMedicine() | 3.23.85.249:8090/healthcare/medicine/add | POST | med\_date, med\_name, med\_vendor, med\_level, med\_price, med\_qty | {  "medicine\_id": 14,  "medicine\_exp\_date": "2025-01-01",  "medicine\_name": "Methylprednisolone 4mg",  "medicine\_vendor": "Sanbe",  "medicine\_level": 3,  "medicine\_price": 40000,  "medicine\_qty": 100  } |
| **6** | Add Schedule | addSchedule() | 3.23.85.249:8090/healthcare/schedule/add | POST | medstaff\_id, schedule\_date, schedule\_time | {  "schedule\_id": 15,  "schedule\_date": "2020/05/03",  "schedule\_time": "16:15",  "medicalStaff": {  "medstaff\_id": 3,  "medstaff\_age": 40,  "medstaff\_name": "Bambang"  }  } |
| **7** | Add Medical Record | addMedicalRecord() | 3.23.85.249:8090/healthcare/record/add | POST | anamnesia, disease\_id, medstaff\_id, patient\_id | {  "record\_id": 16,  "anamnesia": "Kemungkinan kena koped-20",  "patient": {  "patient\_id": 1,  "patient\_name": "Dionis",  "patient\_address": "Kopo",  "patient\_age": 66,  "patient\_dob": "1982-09-12",  "patient\_gender": "M"  },  "medicalStaff": {  "medstaff\_id": 3,  "medstaff\_age": 40,  "medstaff\_name": "Bambang"  },  "disease": {  "disease\_id": 13,  "disease\_type": "infectious",  "disease\_name": "covid-19"  }  } |
| **8** | Add Medicine List | addMedicineList() | 3.23.85.249:8090/healthcare/medicinelist/add | POST | record\_id, medicine\_id | {  "medicinelist\_id": 18,  "medicalRecord": {  "record\_id": 16,  "anamnesia": "Positif Koped-20",  "patient": {  "patient\_id": 1,  "patient\_name": "Dionis",  "patient\_address": "Kopo",  "patient\_age": 66,  "patient\_dob": "1982-09-12",  "patient\_gender": "M",  "hibernateLazyInitializer": {}  },  "medicalStaff": {  "medstaff\_id": 3,  "medstaff\_age": 40,  "medstaff\_name": "Bambang",  "hibernateLazyInitializer": {}  },  "disease": {  "disease\_id": 13,  "disease\_type": "infectious",  "disease\_name": "covid-19",  "hibernateLazyInitializer": {}  }  },  "medicine": {  "medicine\_id": 14,  "medicine\_exp\_date": "2025-01-01",  "medicine\_name": "Methylprednisolone 4mg",  "medicine\_vendor": "Sanbe",  "medicine\_level": 3,  "medicine\_price": 40000,  "medicine\_qty": 100  }  } |
| **9** | Get Patients | getAllPatient() | 3.23.85.249:8090/healthcare/patient | GET | None | [  {  "patient\_id": 1,  "patient\_name": "Dionis",  "patient\_address": "Kopo",  "patient\_age": 66,  "patient\_dob": "1982-09-12",  "patient\_gender": "M"  },  {  "patient\_id": 11,  "patient\_name": "Ricad",  "patient\_address": "Holis",  "patient\_age": 23,  "patient\_dob": "1997/05/26",  "patient\_gender": "M"  }  ] |
| **10** | Get Medical Staff | getMedstaff() | 3.23.85.249:8090/healthcare/medstaff | GET | None | [  {  "medstaff\_id": 3,  "medstaff\_age": 40,  "medstaff\_name": "Bambang"  },  {  "medstaff\_id": 4,  "medstaff\_age": 48,  "medstaff\_name": "Ricad"  },  {  "medstaff\_id": 5,  "medstaff\_age": 55,  "medstaff\_name": "Richard"  },  {  "medstaff\_id": 12,  "medstaff\_age": 38,  "medstaff\_name": "Yohan"  }  ] |
| **11** | Get Disease | getDisease() | 3.23.85.249:8090/healthcare/disease | GET | None | [  {  "disease\_id": 13,  "disease\_type": "infectious",  "disease\_name": "covid-19"  }  ] |
| **12** | Get Medicine | getMedicine() | 3.23.85.249:8090/healthcare/medicine | GET | None | [  {  "medicine\_id": 2,  "medicine\_exp\_date": "2020/05/30",  "medicine\_name": "panadol",  "medicine\_vendor": "PT.APOTEK",  "medicine\_level": 1,  "medicine\_price": 15000,  "medicine\_qty": 20  },  {  "medicine\_id": 14,  "medicine\_exp\_date": "2025-01-01",  "medicine\_name": "Methylprednisolone 4mg",  "medicine\_vendor": "Sanbe",  "medicine\_level": 3,  "medicine\_price": 40000,  "medicine\_qty": 100  }  ] |
| **13** | Get Schedule | getSchedule() | 3.23.85.249:8090/healthcare/schedule | GET | None | [  {  "schedule\_id": 1,  "schedule\_date": "2020/06/02",  "schedule\_time": "14:53",  "medicalStaff": {  "medstaff\_id": 3,  "medstaff\_age": 40,  "medstaff\_name": "Bambang",  "hibernateLazyInitializer": {}  }  },  {  "schedule\_id": 3,  "schedule\_date": "2020/07/02",  "schedule\_time": "10:50",  "medicalStaff": {  "medstaff\_id": 3,  "medstaff\_age": 40,  "medstaff\_name": "Bambang",  "hibernateLazyInitializer": {}  }  },  {  "schedule\_id": 6,  "schedule\_date": "2020/12/12",  "schedule\_time": "12:12",  "medicalStaff": {  "medstaff\_id": 3,  "medstaff\_age": 40,  "medstaff\_name": "Bambang",  "hibernateLazyInitializer": {}  }  },  {  "schedule\_id": 7,  "schedule\_date": "2020/08/22",  "schedule\_time": "20:20",  "medicalStaff": {  "medstaff\_id": 3,  "medstaff\_age": 40,  "medstaff\_name": "Bambang",  "hibernateLazyInitializer": {}  }  },  {  "schedule\_id": 8,  "schedule\_date": "2020/05/27",  "schedule\_time": "20:40",  "medicalStaff": {  "medstaff\_id": 3,  "medstaff\_age": 40,  "medstaff\_name": "Bambang",  "hibernateLazyInitializer": {}  }  },  {  "schedule\_id": 15,  "schedule\_date": "2020/05/03",  "schedule\_time": "16:15",  "medicalStaff": {  "medstaff\_id": 3,  "medstaff\_age": 40,  "medstaff\_name": "Bambang",  "hibernateLazyInitializer": {}  }  }  ] |
| **14** | Get Appointment | getAppointment() | 3.23.85.249:8090/healthcare/appointment | GET | None | [  {  "appointment\_id": 9,  "appointment\_date": "2020/09/12",  "appointment\_time": "19:21",  "appointment\_status": true,  "medicalStaff": {  "medstaff\_id": 3,  "medstaff\_age": 40,  "medstaff\_name": "Bambang",  "hibernateLazyInitializer": {}  },  "patient": {  "patient\_id": 1,  "patient\_name": "Dionis",  "patient\_address": "Kopo",  "patient\_age": 66,  "patient\_dob": "1982-09-12",  "patient\_gender": "M",  "hibernateLazyInitializer": {}  }  }  ] |
| **15** | Get Medical Record | getRecord() | 3.23.85.249:8090/healthcare/record | GET | None | [  {  "record\_id": 16,  "anamnesia": "Kemungkinan kena koped-20",  "patient": {  "patient\_id": 1,  "patient\_name": "Dionis",  "patient\_address": "Kopo",  "patient\_age": 66,  "patient\_dob": "1982-09-12",  "patient\_gender": "M",  "hibernateLazyInitializer": {}  },  "medicalStaff": {  "medstaff\_id": 3,  "medstaff\_age": 40,  "medstaff\_name": "Bambang",  "hibernateLazyInitializer": {}  },  "disease": {  "disease\_id": 13,  "disease\_type": "infectious",  "disease\_name": "covid-19",  "hibernateLazyInitializer": {}  }  }  ] |
| **16** | Get Medicine List | getAllMedicineList | 3.23.85.249:8090/healthcare/medicinelist | GET | None | [  {  "medicinelist\_id": 18,  "medicalRecord": {  "record\_id": 16,  "anamnesia": "Positif Koped-20",  "patient": {  "patient\_id": 1,  "patient\_name": "Dionis",  "patient\_address": "Kopo",  "patient\_age": 66,  "patient\_dob": "1982-09-12",  "patient\_gender": "M",  "hibernateLazyInitializer": {}  },  "medicalStaff": {  "medstaff\_id": 3,  "medstaff\_age": 40,  "medstaff\_name": "Bambang",  "hibernateLazyInitializer": {}  },  "disease": {  "disease\_id": 13,  "disease\_type": "infectious",  "disease\_name": "covid-19",  "hibernateLazyInitializer": {}  },  "hibernateLazyInitializer": {}  },  "medicine": {  "medicine\_id": 14,  "medicine\_exp\_date": "2025-01-01",  "medicine\_name": "Methylprednisolone 4mg",  "medicine\_vendor": "Sanbe",  "medicine\_level": 3,  "medicine\_price": 40000,  "medicine\_qty": 100,  "hibernateLazyInitializer": {}  }  }  ] |
| **17** | Delete Patient | deletePatient() | 3.23.85.249:8090/healthcare/patient/{id} | DELETE | id | Record dengan id terkait akan dihapus dari tabel database |
| **18** | Delete Medical Staff | deleteMedstaff() | 3.23.85.249:8090/healthcare/medstaff/{id} | DELETE | id | Record dengan id terkait akan dihapus dari tabel database |
| **19** | Delete Disease | deleteDisease() | 3.23.85.249:8090/healthcare/disease/{id} | DELETE | id | Record dengan id terkait akan dihapus dari tabel database |
| **20** | Delete Medicine | deleteMedicine() | 3.23.85.249:8090/healthcare/medicine/{id} | DELETE | id | Record dengan id terkait akan dihapus dari tabel database |
| **21** | Delete Schedule | deleteSchedule() | 3.23.85.249:8090/healthcare/schedule/{id} | DELETE | id | Record dengan id terkait akan dihapus dari tabel database |
| **22** | Delete Appointment | deleteAppointment() | 3.23.85.249:8090/healthcare/appointment/{id} | DELETE | id | Record dengan id terkait akan dihapus dari tabel database |
| **23** | Delete Medical Record | deleteMedicalRecord() | 3.23.85.249:8090/healthcare/record/{id} | DELETE | id | Record dengan id terkait akan dihapus dari tabel database |
| **24** | Delete Medicine List | deleteMedicineList() | 3.23.85.249:8090/healthcare/medicinelist/{id} | DELETE | id | Record dengan id terkait akan dihapus dari tabel database |
| **25** | Update Patients | updatePatient() | 3.23.85.249:8090/healthcare/patient/{patient\_id} | PUT | patient\_id (@PathVariable) | Record dengan patient\_id terkait akan dihapus dari tabel database |
| **26** | Update Medical Staff | updateMedstaff() | 3.23.85.249:8090/healthcare/medstaff/{medstaff\_id} | PUT | medstaff\_id (@PathVariable) | Record dengan medstaff\_Id terkait akan dihapus dari tabel database |
| **27** | Update Disease | updateDisease() | 3.23.85.249:8090/healthcare/disease/{disease\_id} | PUT | disease\_id (@PathVariable) | Record dengan disease\_id terkait akan dihapus dari tabel database |
| **28** | Update Medicine | updateMedicine() | 3.23.85.249:8090/healthcare/medicine/{medicine\_id} | PUT | med\_id (@PathVariable) | Record dengan med\_id terkait akan dihapus dari tabel database |
| **29** | Update Schedule | updateSchedule() | 3.23.85.249:8090/healthcare/schedule/{schedule\_id} | PUT | schedule\_id (@PathVariable) | Record dengan schedule\_id terkait akan dihapus dari tabel database |
| **30** | Update Appointment | updateAppointment() | 3.23.85.249:8090/healthcare/appointment/{appointment\_id} | PUT | appointment\_id (@PathVariable) | Record dengan appointment\_id terkait akan dihapus dari tabel database |
| **31** | Update Medical Record | updateRecord() | 3.23.85.249:8090/healthcare/record/{record\_id} | PUT | record\_id (@PathVariable) | Record dengan record\_id terkait akan dihapus dari tabel database |
| **32** | Update Medicine List | updateMedicineList() | 3.23.85.249:8090/healthcare/medicinelist/{medicinelist\_id} | PUT | medicinelist\_id (@PathVariable) | Record dengan medicinelist\_id terkait akan dihapus dari tabel database |
| **33** | Get Patient | getPatient() | 3.23.85.249:8090/healthcare/patient/{id} | GET | id | Record dengan id terkait akan diambil dari table jika ada |
| **34** | Get Medical Staff | getMedstaff() | 3.23.85.249:8090/healthcare/medstaff/{id} | GET | id | Record dengan id terkait akan diambil dari table jika ada |
| **35** | Get Disease | getDisease() | 3.23.85.249:8090/healthcare/disease/{id} | GET | id | Record dengan id terkait akan diambil dari table jika ada |
| **36** | Get Medicine | getMedicine() | 3.23.85.249:8090/healthcare/medicine/{id} | GET | id | Record dengan id terkait akan diambil dari table jika ada |
| **37** | Get Schedule | getSchedule() | 3.23.85.249:8090/healthcare/schedule/{id} | GET | id | Record dengan id terkait akan diambil dari table jika ada |
| **38** | Get Appointment | getAppointment() | 3.23.85.249:8090/healthcare/appointment/{id} | GET | id | Record dengan id terkait akan diambil dari table jika ada |
| **39** | Get Medical Record | getRecord() | 3.23.85.249:8090/healthcare/record/{id} | GET | id | Record dengan id terkait akan diambil dari table jika ada |
| **40** | Get Medicine List | getMedicineList() | 3.23.85.249:8090/healthcare/medicinelist/{id} | GET | id | Record dengan id terkait akan diambil dari table jika ada |

1. **Collections (Postman)**

Collections dalam Postman adalah kumpulan *pre-built request* yang dapat diatur ke dalam folder, dan dapat dengan mudah diekspor dan dibagikan dengan orang lain. Collection dalam tugas besar ini diekspor ke file .json dan dikumpulkan pada submission terpisah.