Project Report (Dummy) – Well Predict Backend Service

1. Project Overview

Well Predict adalah aplikasi kesehatan berbasis machine learning yang menyediakan prediksi penyakit secara real-time berdasarkan gejala yang diinput pengguna. Sistem terdiri dari mobile app, backend service, dan ML model di cloud.

Backend dibangun menggunakan **Golang**, **Gin Framework**, database **PostgreSQL**, dan model machine learning yang disimpan di **Google Cloud Storage**. API backend berfungsi sebagai penghubung antara aplikasi mobile dan model ML.

2. Objectives

1. Menyediakan API yang cepat, aman, dan mudah di-scale untuk kebutuhan prediksi penyakit.
2. Menyederhanakan integrasi antara aplikasi mobile dengan model machine learning.
3. Mengimplementasikan best practice backend seperti structuring project, middleware, logging, caching, dan clean architecture.
4. Mendukung kebutuhan analitik melalui pencatatan riwayat prediksi.

3. System Architecture

3.1 High-Level Architecture

- Mobile App → REST API → Backend Service → ML Model (GCP Storage)
- PostgreSQL digunakan untuk menyimpan:
    - User profile
    - Riwayat prediksi
    - Logs request

3.2 Technical Stack

Component	Technology
Backend	Golang (Gin)
Database	PostgreSQL
Cache	Redis (optional)
Cloud	Google Cloud Storage, App Engine
Auth	JWT
Logging	Zerolog / logrus

Deployment	Docker + App Engine

4. Features Summary

4.1 Core Features

- Predict Disease (ML integration)
- User Authentication (JWT)
- History Tracking
- Health Check Endpoint

4.2 Additional Features (Dummy)

- Rate limiting
- Caching prediction result (TTL 30s)
- Error monitoring (Sentry dummy)

5. API Documentation (Dummy)

5.1 POST /api/v1/predict

Description: Mengirim gejala pengguna untuk diproses oleh model ML.
Body Example

```
{
  "symptoms": "Headache, fever, nausea"
}
```

Response

```
{
  "prediction": "Dengue Fever",
  "confidence": 0.87,
  "timestamp": "2025-02-15T10:15:30Z"
}
```

5.2 POST /api/v1/auth/login

```
{
```

```
  "email": "user@example.com",
  "password": "123456"
}
```

5.3 GET /api/v1/history

Header: Authorization: Bearer <token>

Response

```
[
  {
    "disease": "Flu",
    "confidence": 0.75,
    "created_at": "2025-02-14T08:10:00Z"
  }
]
```

6. Database Schema

6.1 users

Field	Type
id	UUID
name	varchar(100)
email	varchar(100)
password	text

6.2 prediciton_history

Field	Type
id	UUID
user_id	UUID
disease	varchar(100)
confidence	float
created_at	timestamp

7. Testing Summary (Dummy)

7.1 Unit Test

Module	Status	Coverage
Controllers	Passed	82%

Services	Passed	79%
Repository	Passed	88%

7.2 API Load Testing

Tools: k6

Scenario	**Result**
100 VU	Avg latency 120ms
500VU	Avg latency 230ms
Peak Load (1000 VU)	System stable, no errors

7.3 Security Testing

- SQL Injection: **Passed**
- JWT tampering: **Blocked**
- Rate limiting: **Enabled (100 req / minute)**

8. Deployment Summary

- Build Docker image
- Push to GCP Artifact Registry
- Deploy via App Engine Standard
- Auto scaling enabled
- Error rate: **< 0.5%** after 24 hours

9. Known Issues (Dummy)

- Model response delay ±300ms pada jam traffic tinggi
- History endpoint belum mendukung pagination
- Caching layer belum diimplementasikan penuh

10. Conclusion

Backend Well Predict berhasil memenuhi kebutuhan inti aplikasi mobile, dengan performa stabil, integrasi ML lancar, dan arsitektur yang scalable. Beberapa peningkatan seperti pagination dan caching dapat dilakukan pada iterasi selanjutnya.