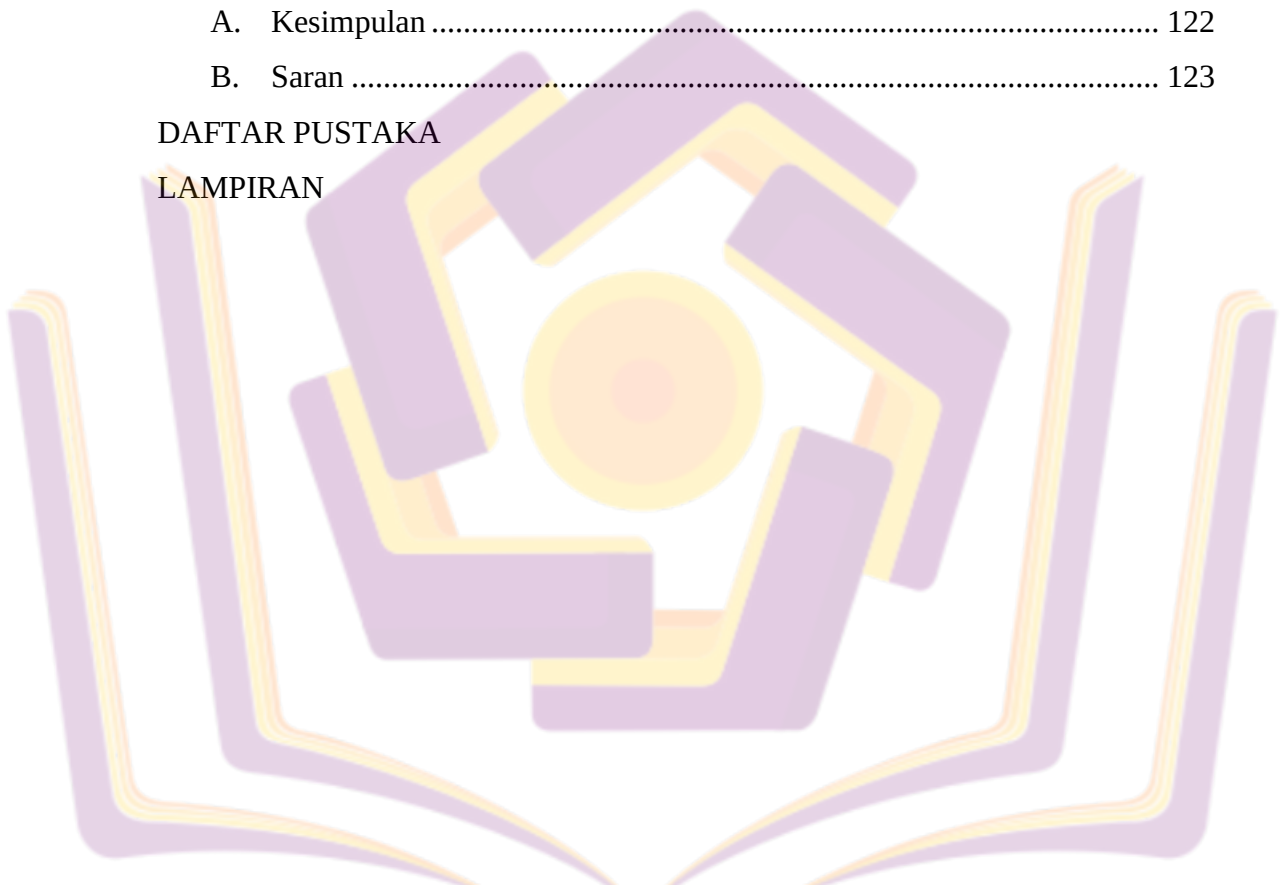


DAFTAR ISI

HALAMAN SAMPUL	i
HALAMAN JUDUL.....	ii
HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN.....	iv
HALAMAN PERNYATAAN KEASLIAN	v
HALAMAN PERSEMBAHAN	vi
HALAMAN MOTTO	viii
KATA PENGANTAR	ix
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR	xiv
DAFTAR ISTILAH	xvi
DAFTAR LAMPIRAN.....	xxi
ABSTRAK.....	xxii
ABSTRACT.....	xxiii
BAB I PENDAHULUAN	1
A. Latar Belakang Masalah	1
B. Rumusan Masalah.....	9
C. Batasan Masalah	10
D. Tujuan Penelitian	11
E. Manfaat Penelitian	11
BAB II TINJAUAN PUSTAKA.....	13
A. Landasan Teori.....	13
B. Penelitian Sebelumnya.....	36
BAB III METODE PENELITIAN.....	47
A. Tempat dan Waktu Penelitian.....	47
B. Metode Pengumpulan Data.....	47
C. Alat dan Bahan Penelitian.....	50
D. Konsep Penelitian	51

BAB IV HASIL DAN PEMBAHASAN	60
A. <i>Requirement Analysis</i>	60
B. <i>System Design</i>	66
C. <i>Implementation</i>	75
D. <i>Testing</i>	92
E. <i>Deployment</i>	105
F. <i>Maintenance</i>	117
BAB V PENUTUP.....	122
A. Kesimpulan	122
B. Saran	123
DAFTAR PUSTAKA	
LAMPIRAN	



DAFTAR TABEL

Tabel 2.1. Klasifikasi Status Gizi Berdasarkan Indikator TB/U	14
Tabel 2.2. Komponen Utama <i>Backend System</i>	21
Tabel 2.3. Penelitian Sebelumnya	44
Tabel 4. 1 Kebutuhan Fungsional Sistem <i>Backend</i>	61
Tabel 4. 2 Kebutuhan Nonfungsional Sistem <i>Backend</i>	62
Tabel 4. 3 Analisis Kebutuhan Integrasi	64
Tabel 4. 4 <i>RESTful API Endpoint Design</i>	68
Tabel 4. 5 Tabel Hasil Evaluasi Pengujian <i>Black box API Backend</i>	104
Tabel 4. 6 Infrastruktur <i>Cloud</i> yang Digunakan pada <i>Deployment</i>	107



DAFTAR GAMBAR

Gambar 2. 1 Arsitektur Umum Integrasi <i>Frontend</i> dan <i>Backend</i> (Sumber: Magnific, 2023).....	17
Gambar 2. 2 Arsitektur Siklus <i>Request-Response</i> pada <i>RESTful API</i> (Sumber: Yu, 2026).....	23
Gambar 2. 3 Arsitektur Event Loop dan Non-Blocking I/O pada <i>Node.js</i> (Sumber: Alam, 2026).....	26
Gambar 3. 1 Tahapan Metode <i>Waterfall</i> (Sumber: BINUS UNIVERSITY Bekasi, 2024).....	53
Gambar 3. 2 Diagram Alur Penelitian.....	58
Gambar 4. 1 <i>Backend Component Architecture</i>	67
Gambar 4. 2 Authentication Flow Using <i>bcrypt</i> and <i>JSON Web Token (JWT)</i>	69
Gambar 4. 3 <i>Database Design</i>	71
Gambar 4. 4 Sequence Diagram	72
Gambar 4. 5 <i>Cloud Architecture Design</i>	74
Gambar 4. 6 Struktur Project <i>Backend</i>	76
Gambar 4. 7 Implementasi <i>Routing</i> Utama pada File <i>app.js</i>	77
Gambar 4. 8 Logika Bisnis Proses Registrasi Pengguna Baru	78
Gambar 4. 9 Logika Bisnis Proses Autentikasi Login dan Generate <i>JWT</i>	79
Gambar 4. 10 Implementasi <i>Middleware</i> Otorisasi Bearer <i>Token JWT</i>	80
Gambar 4. 11 Pendefinisian <i>Endpoint</i> Pengelolaan Profil Pengguna.....	81
Gambar 4. 12 Logika Bisnis Manipulasi Data Profil pada <i>Database</i>	82
Gambar 4. 13 Pendefinisian <i>Endpoint</i> <i>Journal API</i>	83
Gambar 4. 14 Logika Bisnis Fungsi Pembuatan <i>Journal</i>	84
Gambar 4. 15 Logika Bisnis Fungsi Menampilkan Daftar dan Detail <i>Jurnal</i>	85
Gambar 4. 16 Kode Program Konfigurasi Inisialisasi dan Validasi Ukuran Berkas <i>GCS</i>	86
Gambar 4. 17 Kode Program Logika <i>Middleware</i> Fungsi <i>Upload Image</i> ke <i>GCS Bucket</i>	87
Gambar 4. 18 Implementasi Mekanisme <i>Routing</i> pada Modul <i>Prediksi</i>	88

Gambar 4. 19 Kode Program Fungsi predict pada Komponen <i>Controller</i>	88
Gambar 4. 21 Implementasi Mekanisme <i>Routing</i> pada Modul Rekomendasi Nutrisi	90
Gambar 4. 22 Kode Program Logika <i>Rule-Based Filtering</i> Data Makanan pada <i>Controller</i>	91
Gambar 4. 23 Hasil Pengujian <i>Postman</i> pada <i>Endpoint</i> /auth/register	93
Gambar 4. 24 Hasil Pengujian <i>Postman</i> pada <i>Endpoint</i> /auth/login.....	94
Gambar 4. 25 Hasil Pengujian <i>Postman</i> pada <i>Endpoint</i> /profile (GET).....	95
Gambar 4. 26 Hasil Pengujian <i>Postman</i> pada <i>Endpoint</i> /profile (PUT)	96
Gambar 4. 27 Hasil Pengujian <i>Postman</i> pada <i>Endpoint</i> /profile (DELETE).....	97
Gambar 4. 28 Hasil Pengujian <i>Postman</i> pada <i>Endpoint</i> /prediction (POST)	98
Gambar 4. 29 Hasil Pengujian <i>Postman</i> pada <i>Endpoint</i> /prediction (GET)	99
Gambar 4. 30 Hasil Pengujian <i>Postman</i> pada <i>Endpoint</i> /journal (POST)	100
Gambar 4. 31 Hasil Pengujian <i>Postman</i> pada <i>Endpoint</i> /journal (GET)	101
Gambar 4. 32 Hasil Pengujian <i>Postman</i> pada <i>Endpoint</i> /journal/:id	102
Gambar 4. 33 Hasil Pengujian <i>Postman</i> pada <i>Endpoint</i> GET /nutrition/classification	103
Gambar 4. 34 <i>Dashboard</i> Infrastruktur <i>Google Cloud platform</i>	107
Gambar 4. 35 Konfigurasi <i>Deployment</i> Menggunakan <i>app.yaml</i>	108
Gambar 4. 36 Proses <i>Deployment Backend</i> Menggunakan <i>Google Cloud SDK</i>	109
Gambar 4. 37 Konfigurasi Koneksi <i>Google Cloud SQL</i>	111
Gambar 4. 38 Konfigurasi Bucket <i>Google Cloud Storage</i>	112
Gambar 4. 39 Konfigurasi Service <i>Google Cloud Run</i>	114
Gambar 4. 40 Halaman <i>Dashboard Cloud App Engine</i>	115
Gambar 4. 41 <i>Monitoring</i> Aktivitas <i>Cloud App Engine</i> Menggunakan <i>Logs Explorer</i>	118
Gambar 4. 42 <i>Monitoring</i> Performa <i>Google Cloud SQL</i>	119
Gambar 4. 43 <i>Monitoring</i> Layanan <i>Machine learning</i> pada <i>Google Cloud Run</i>	120
Gambar 4. 44 <i>Monitoring</i> <i>Google Cloud Storage</i>	121

DAFTAR ISTILAH

API (Application Programming Interface) : Antarmuka yang memungkinkan dua atau lebih aplikasi saling berkomunikasi dan bertukar data.

Backend : Bagian aplikasi yang berjalan di sisi server untuk mengelola logika bisnis, pemrosesan data, dan komunikasi dengan basis data.

Black-Box Testing : Metode pengujian perangkat lunak yang berfokus pada fungsi sistem tanpa melihat kode program di dalamnya.

Blueprint : Rancangan atau cetak biru teknis yang menjadi acuan dalam proses implementasi sistem.

Callback : Fungsi yang dipanggil kembali setelah suatu proses selesai dijalankan, terutama pada pemrograman asinkron.

Client : Perangkat atau aplikasi yang mengirimkan permintaan kepada server untuk memperoleh layanan atau data.

Cloud computing : Model penyediaan layanan komputasi melalui internet yang mencakup server, penyimpanan, basis data, jaringan, dan layanan lainnya.

Cloud Run : Layanan Google Cloud yang digunakan untuk menjalankan aplikasi berbasis container secara serverless.

Cloud SQL : Layanan basis data relasional terkelola yang disediakan oleh Google Cloud platform.

Cloud Storage : Layanan penyimpanan objek berbasis Cloud untuk menyimpan berbagai jenis berkas.

Coding : Proses menulis kode program menggunakan bahasa pemrograman.

CRUD (Create, Read, Update, Delete) : Empat operasi dasar yang digunakan dalam pengelolaan data pada suatu sistem.

Dataset : Sekumpulan data yang digunakan sebagai sumber informasi atau pelatihan model machine learning.

Deployment : Proses menerapkan aplikasi ke lingkungan produksi sehingga dapat digunakan oleh pengguna.

Endpoint : Alamat atau jalur akses spesifik pada API yang digunakan untuk mengakses suatu layanan.

Event-driven : Pendekatan pemrograman yang menjalankan proses berdasarkan peristiwa atau kejadian tertentu.

Event Loop : Mekanisme pada Node.js yang mengatur eksekusi proses asinkron secara efisien.

Express.js : Framework berbasis Node.js yang digunakan untuk membangun aplikasi web dan RESTful API.

Failover : Mekanisme pemindahan layanan secara otomatis ke sistem cadangan ketika terjadi gangguan.

Framework : Kerangka kerja perangkat lunak yang menyediakan struktur dan komponen untuk mempermudah pengembangan aplikasi.

Frontend : Bagian aplikasi yang berinteraksi langsung dengan pengguna melalui antarmuka.

HTTP (Hypertext Transfer Protocol) : Protokol komunikasi yang digunakan untuk pertukaran data di internet.

HTTPS (Hypertext Transfer Protocol Secure) : Versi aman dari HTTP yang menggunakan enkripsi dalam proses pertukaran data.

Input : Data atau informasi yang dimasukkan ke dalam sistem untuk diproses.

JSON (JavaScript Object Notation) : Format pertukaran data berbasis teks yang ringan dan mudah diproses oleh berbagai platform.

JWT (JSON Web Token) : Standar token yang digunakan untuk proses autentikasi dan otorisasi pengguna.

Machine learning : Cabang kecerdasan buatan yang memungkinkan sistem mempelajari pola dari data untuk membuat prediksi atau keputusan.

Maintenance : Tahap pemeliharaan sistem setelah diimplementasikan agar tetap berjalan dengan baik.

Middleware : Komponen perantara pada Express.js yang memproses request sebelum diteruskan ke fungsi utama.

Mobile Development : Proses pengembangan aplikasi yang berjalan pada perangkat bergerak seperti smartphone atau tablet.

Node.js : Lingkungan eksekusi JavaScript di sisi server yang digunakan untuk membangun aplikasi backend.

Non-Blocking I/O : Mekanisme pemrosesan input dan output tanpa menghentikan eksekusi proses utama.

Official Documentation : Dokumentasi resmi yang diterbitkan oleh pengembang atau penyedia teknologi sebagai acuan penggunaan.

Output : Hasil pemrosesan data yang dihasilkan oleh sistem.

Performance Monitoring : Proses pemantauan performa sistem untuk memastikan layanan berjalan optimal.

Postman : Perangkat lunak yang digunakan untuk menguji dan mendokumentasikan API.

Request : Permintaan yang dikirim oleh client kepada server untuk memperoleh atau mengirimkan data.

Response : Balasan yang diberikan server terhadap request dari client.

REST (Representational State Transfer) : Gaya arsitektur yang menjadi dasar pembangunan layanan web berbasis HTTP.

RESTful API : API yang menerapkan prinsip-prinsip arsitektur REST dalam komunikasi data.

Routing : Mekanisme yang menghubungkan URL dengan fungsi tertentu pada aplikasi.

Runtime Environment : Lingkungan yang menyediakan fasilitas agar suatu program dapat dijalankan.

Scalable : Kemampuan sistem untuk menangani peningkatan beban kerja tanpa menurunkan kinerja.

Scalability : Kemampuan suatu sistem untuk meningkatkan kapasitas sesuai kebutuhan.

Sequence Diagram : Diagram UML yang menggambarkan urutan interaksi antarobjek dalam suatu proses.

Server : Komputer atau layanan yang menyediakan sumber daya dan melayani permintaan dari client.

Server-Side : Bagian sistem yang dijalankan pada sisi server.

Service Layer : Lapisan pada arsitektur perangkat lunak yang menangani logika layanan dan komunikasi antarkomponen.

SQL (Structured Query Language) : Bahasa standar yang digunakan untuk mengelola basis data relasional.

Stateless : Karakteristik REST yang mengharuskan setiap request diproses secara independen tanpa menyimpan status sebelumnya.

Thread Pool : Kumpulan thread yang digunakan Node.js untuk menjalankan operasi berat secara paralel.

UI (User Interface) : Antarmuka yang menjadi media interaksi antara pengguna dan aplikasi.

UI/UX (User Interface/User Experience) : Aspek tampilan antarmuka dan pengalaman pengguna dalam menggunakan aplikasi.

URI (Uniform Resource Identifier) : Identitas unik yang digunakan untuk mengidentifikasi suatu sumber daya pada web.

URL (Uniform Resource Locator) : Alamat yang menunjukkan lokasi suatu sumber daya di internet.

V8 JavaScript Engine : Mesin JavaScript yang digunakan oleh Node.js untuk mengeksekusi kode JavaScript.

Waterfall : Model pengembangan perangkat lunak yang dilakukan secara berurutan mulai dari analisis kebutuhan hingga pemeliharaan.

Web Service : Layanan yang memungkinkan pertukaran data antar aplikasi melalui jaringan.

DAFTAR LAMPIRAN

Lampiran 1. Kartu Bimbingan Dosen Pembimbing 1

Lampiran 2. Kartu Bimbingan Dosen Pembimbing 2

Lampiran 3. *Source Code Backend System*

