

**Perancangan dan Implementasi *Backend*
Real-Time Runner Tracking dan *ETA Prediction*
dengan Skalabilitas dan *High Availability*
untuk ITB Ultra Marathon**

Proposal Tugas Akhir

Oleh

**Justin Lawrance
18222006**



**PROGRAM STUDI SISTEM DAN TEKNOLOGI INFORMASI
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
Desember 2025**

LEMBAR PENGESAHAN

Perancangan dan Implementasi *Backend* *Real-Time Runner Tracking* dan *ETA Prediction* dengan Skalabilitas dan *High Availability* untuk ITB Ultra Marathon

Proposal Tugas Akhir

Oleh

**Justin Lawrance
18222006**

**Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung**

Proposal Tugas Akhir ini telah disetujui dan disahkan
di Bandung, pada tanggal 4 Desember 2025

Pembimbing,

Dr. Riza Satria Perdana, S.T, M.T.

NIP. 19700609 199512 1 002

DAFTAR ISI

DAFTAR GAMBAR	iv
DAFTAR TABEL	v
DAFTAR KODE	vi
I PENDAHULUAN	1
I.1 Latar Belakang	1
I.2 Rumusan Masalah	2
I.3 Tujuan	3
I.4 Batasan Masalah	4
I.5 Metodologi	5
II STUDI LITERATUR	7
II.1 Penulisan Gambar, Tabel, Rumus, dan Kode	7
II.1.1 Gambar	7
II.1.2 Tabel	8
II.1.2.1 Tabel yang Muat dalam Satu Halaman	8
II.1.2.2 Mengimpor Tabel dari Berkas Eksternal	9
II.1.2.3 Tabel yang Sangat Panjang	9
II.1.2.4 Beberapa Contoh Penulisan Rumus atau Persamaan Matematika Menggunakan LaTeX Termasuk Penomorannya	11
II.1.3 Algoritma, Pseudocode, atau Kode	12
II.2 Beberapa Kesalahan Penulisan yang Sering Terjadi	13
II.2.1 Penggunaan Kata "di mana" atau "dimana"	13
II.2.2 Penggunaan Kata "sedangkan" dan "sehingga"	13
II.2.3 Penggunaan Istilah yang Tidak Baku	14
II.2.4 Pemisah Desimal dan Ribuan	14
II.2.5 Daftar atau <i>List</i>	14
II.2.6 Penggunaan Kata "masing-masing" dan "setiap"	14
III ANALISIS MASALAH	16
III.1 Analisis Kondisi Saat Ini	16
III.1.1 Model Sistem Berbasis RFID	16
III.1.2 Model Sistem Berbasis LoRaWan dan GPS	17
III.1.3 Kesenjangan pada Metode Prediksi ETA	17

III.2 Analisis Kebutuhan	18
III.2.1 Identifikasi Masalah Pengguna	18
III.2.2 Kebutuhan Fungsional	19
III.2.3 Kebutuhan Nonfungsional	21
III.3 Analisis Pemilihan Solusi	22
III.3.1 Alternatif Solusi	22
III.3.2 Analisis Penentuan Solusi	27
IV DESAIN KONSEP SOLUSI	38
V RENCANA SELANJUTNYA	39

DAFTAR GAMBAR

II.1 Contoh gambar jaringan	8
---------------------------------------	---

DAFTAR TABEL

II.1	Tabel harga bahan pokok	9
II.2	Tabel harga bahan sekunder	9
II.3	Tabel harga bahan tertier	9
II.4	Comprehensive Data Table Example	9
II.4	Comprehensive Data Table Example (lanjutan)	10
II.4	Comprehensive Data Table Example (lanjutan)	11
II.5	Contoh penggunaan kata "sedangkan" dan "sehingga"	13
III.1	Kebutuhan Fungsional Sistem	19
III.2	Kebutuhan Nonfungsional Sistem	21
III.3	Penilaian Tiap Alternatif Arsitektur Backend	27
III.4	Hasil Perhitungan Weighted Score Arsitektur Backend	28
III.5	Penilaian Tiap Alternatif Protokol Komunikasi Telemetri	29
III.6	Hasil Perhitungan Weighted Score Protokol Komunikasi Telemetri	29
III.7	Penilaian Tiap Alternatif Basis Data Time-Series	30
III.8	Hasil Perhitungan Weighted Score Basis Data Time-Series	30
III.9	Penilaian Tiap Alternatif Platform Deployment	31
III.10	Hasil Perhitungan Weighted Score Platform Deployment	31
III.11	Penilaian Tiap Alternatif Sistem Monitoring Metrics	32
III.12	Hasil Perhitungan Weighted Score Sistem Monitoring Metrics	32
III.13	Penilaian Tiap Alternatif Sistem Logging	33
III.14	Hasil Perhitungan Weighted Score Sistem Logging	34
III.15	Penilaian Tiap Alternatif Sistem Tracing	34
III.16	Hasil Perhitungan Weighted Score Sistem Tracing	35
III.17	Penilaian Tiap Alternatif Model Perhitungan ETA	36
III.18	Hasil Perhitungan Weighted Score Model ETA	36
III.19	Penilaian Tiap Alternatif Pendekatan Integrasi Elevasi	37
III.20	Hasil Perhitungan Weighted Score Pendekatan Integrasi Elevasi	37

DAFTAR KODE

II.1 Contoh pseudocode	12
II.2 Contoh source code Python	13

BAB I

PENDAHULUAN

I.1 Latar Belakang

Ultra-Marathon merupakan salah satu cabang olahraga lari yang kini kian populer. Hal ini ditunjukkan dengan kegiatan ITB Ultra-Marathon yang jumlah pesertanya terus meningkat sejak pertama kali diselenggarakan pada tahun 2017 (M. Naufal Hafizh September 2025). Seiring dengan meningkatnya partisipasi, kebutuhan akan sistem pelacakan pelari menjadi semakin penting. Sistem tersebut berperan dalam memastikan keselamatan, performa, dan pengalaman peserta secara keseluruhan, serta memberikan interaktivitas bagi panitia dan pendukung acara (Hochreiter 2024).

Dalam penyelenggaraan ITB Ultra-Marathon saat ini, pelacakan posisi peserta masih dilakukan secara manual. Kondisi ini menyulitkan panitia dalam memantau lokasi pelari secara *real-time* dan menyebabkan koordinasi antarpos menjadi kurang efisien. Dampaknya terlihat pada ketidakakuratan pengaturan mobil penjemput, yang kerap terlambat tiba di titik pengambilan peserta. Ketiadaan sistem pemantauan khusus juga membuat panitia dan peserta mengandalkan aplikasi umum seperti Google Maps untuk memprediksi estimasi waktu kedatangan (ETA), meskipun aplikasi tersebut tidak dirancang untuk konteks pelari. Permasalahan ini menegaskan perlunya mekanisme pelacakan yang terintegrasi, akurat, dan adaptif terhadap kebutuhan operasional marathon.

Rute marathon terdiri dari 16 segmen, dengan titik *check point* (CP) dan *water station* (WS) yang menjadi patokan untuk pemantauan *real-time*. Pelari dapat mengikuti kategori individu maupun relay, sehingga sistem harus mampu memantau pergantian anggota tim secara *real-time* dan menampilkan status anggota aktif di setiap segmen. Informasi posisi dan ETA pelari juga dapat dimanfaatkan panitia untuk mengatur transportasi dan shuttle support di jalur lomba.

Sistem komersial seperti MyLaps, ChronoTrack, dan Race Result menggunakan teknologi chip timing berbasis RFID untuk merekam waktu pelari secara otomatis di titik start, checkpoint, dan finish (Sjöbeck August 2022). Meskipun akurat dalam pencatatan waktu di titik-titik tertentu, sistem ini belum mendukung pemantauan posisi pelari secara kontinu di sepanjang lintasan. Keterbatasan ini menghambat kemampuan penyelenggara untuk melakukan tracking secara *real-time* dan memperkirakan ETA.

Sistem berbasis LoRaWAN dan GPS juga telah digunakan untuk pelacakan pelari dalam skala besar. Pada uji coba dengan lebih dari 35.000 peserta, sistem ini mampu memperbarui posisi setiap 30 detik melalui *mobile LoRaWAN gateway* pada kendaraan operasional, menunjukkan potensi untuk pelacakan *real-time*. Namun, LoRaWAN memerlukan arsitektur jaringan yang efisien dan terdistribusi untuk menjaga keandalan dan skalabilitas pengiriman data dalam sistem pelacakan berskala besar (Hochreiter 2024).

Mengenai prediksi ETA, metode yang tersedia saat ini umumnya dikembangkan untuk kendaraan, sehingga pendekatannya tidak sesuai untuk konteks pelari. Sistem ETA tersebut tidak mempertimbangkan pola pace pelari, karakteristik rute, elevasi lintasan, maupun dinamika performa yang berubah sepanjang waktu. Selain itu, metode ETA yang ada umumnya tidak dirancang untuk memproses data dari peserta dalam jumlah besar.

Keterbatasan berbagai solusi tersebut menunjukkan bahwa diperlukan sistem pelacakan yang dirancang khusus untuk kebutuhan ITB Marathon. Oleh karena itu, penelitian ini mengusulkan perancangan dan implementasi *backend* untuk *Real-Time Runner Tracking* dan *ETA Prediction* yang mampu beroperasi secara terukur (*scalable*) serta memiliki tingkat ketersediaan layanan yang tinggi (*high availability*). Sistem ini diharapkan dapat menyediakan informasi posisi dan prediksi waktu kedatangan pelari secara lebih akurat, sekaligus mendukung proses operasional kepanitiaan secara efisien dan andal.

I.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah dalam penelitian ini adalah sebagai berikut.

1. Bagaimana merancang dan mengimplementasikan sistem *backend* untuk *Real-Time Runner Tracking* yang mampu memantau posisi ribuan peserta ITB Marathon secara akurat dan *real-time*, termasuk pemantauan per segmen,

- check point, dan status peserta (*finisher* atau *Did Not Finish*)?
2. Bagaimana merancang dan mengimplementasikan modul *ETA Prediction* yang mampu menghasilkan estimasi waktu kedatangan pelari secara relevan dengan karakteristik pergerakan pelari dan kondisi rute?
 3. Bagaimana memastikan bahwa sistem *backend* yang dibangun memiliki kemampuan skalabilitas (*scalability*) dan ketersediaan layanan yang tinggi (*high availability*) sehingga dapat beroperasi secara andal selama acara berlangsung?
 4. Bagaimana mengevaluasi kinerja sistem dalam menangani beban tinggi, khususnya dari aspek *throughput*, *latency*, *error rate*, dan *resource utilization*?

I.3 Tujuan

Berdasarkan rumusan masalah yang telah disebutkan sebelumnya, tujuan dari penelitian ini adalah sebagai berikut.

1. Menghasilkan rancangan dan implementasi sistem *backend* untuk *Real-Time Runner Tracking* yang mampu memantau posisi ribuan peserta ITB Marathon secara akurat dan *real-time*, termasuk pemantauan per segmen, check point, dan status peserta (*finisher/DNF*).
2. Menghasilkan modul *ETA Prediction* yang mampu memberikan estimasi waktu kedatangan pelari secara relevan dengan pola pergerakan pelari, kondisi rute, dan dinamika performa pelari.
3. Menghasilkan arsitektur sistem *backend* yang mampu mendukung skalabilitas (*scalability*) dan ketersediaan layanan yang tinggi (*high availability*) sehingga sistem tetap dapat beroperasi secara andal selama acara berlangsung.
4. Menghasilkan evaluasi kinerja sistem dalam menangani beban tinggi, khususnya terkait *throughput*, *latency*, *error rate*, dan *resource utilization*.

Kriteria keberhasilan dari penelitian ini ditetapkan sebagai berikut.

1. Sistem *Real-Time Runner Tracking* mampu memperbarui posisi pelari secara *real-time* dengan tingkat keterlambatan pembaruan (update delay) yang berada dalam batas operasional yang dapat diterima oleh panitia.
2. Modul *ETA Prediction* mampu menghasilkan estimasi waktu kedatangan pelari dengan tingkat kesalahan prediksi yang rendah berdasarkan uji validasi pada data pergerakan pelari.
3. Arsitektur sistem mampu menangani skala pengguna sesuai jumlah peserta

ITB Marathon dan tetap beroperasi tanpa gangguan (*downtime*) selama simulasi atau pengujian beban.

4. Sistem memenuhi batas performa minimum pada pengujian beban, yang mencakup metrik *throughput*, *latency*, *error rate*, dan *resource utilization* sesuai target yang telah ditentukan.

I.4 Batasan Masalah

Dalam penelitian ini, terdapat beberapa batasan yang digunakan untuk memfokuskan ruang lingkup pekerjaan dan memastikan hasil penelitian tetap relevan dengan tujuan yang telah ditetapkan. Batasan-batasan tersebut adalah sebagai berikut.

1. Tugas akhir ini dikerjakan oleh dua orang mahasiswa, yaitu Dinda Thalia Fahira (18222055) dan Justin Lawrance (18222006), dengan pembagian fokus bahwa pengembangan *backend* dilakukan oleh Justin Lawrance, sedangkan pengembangan *frontend* dilakukan oleh Dinda Thalia Fahira.
2. Implementasi yang dibahas pada laporan ini hanya mencakup pengembangan sistem *backend* untuk *Real-Time Runner Tracking* dan *ETA Prediction*. Pengembangan *frontend* aplikasi tidak termasuk ruang lingkup pembahasan teknis pada laporan ini.
3. Sistem *backend* yang dikembangkan dibatasi pada fungsionalitas inti yang diperlukan untuk mendukung proses pelacakan dan prediksi, termasuk:
 - penerimaan dan pemrosesan data lokasi pelari.
 - penyimpanan data posisi secara *real-time*.
 - penyediaan *API* untuk konsumsi *frontend*.
 - modul prediksi waktu kedatangan (ETA).
 - pemantauan status pelari per segmen dan Checkpoint/Water Station.
 - deteksi *off-route*.
4. Sistem hanya memproses rute lomba yang telah ditentukan panitia, termasuk segmentasi jalur dan lokasi Checkpoint/Water Station. Perubahan rute saat lomba tidak ditangani sistem.
5. Backend mendukung kategori pelari individu dan tim relay dengan segmentasi tertentu. Pergantian pelari dicatat oleh sistem sesuai aturan panitia, namun status DNF atau penalti sepenuhnya ditentukan manual oleh panitia.
6. Evaluasi nonfungisional difokuskan pada metrik *performance* yang meliputi *throughput*, *latency*, *error rate*, dan *resource utilization*. Evaluasi aspek lain seperti keamanan, biaya operasional, atau konsumsi energi tidak dibahas

secara mendalam.

7. Pengujian dilakukan menggunakan data simulasi yang merepresentasikan pergerakan pelari dalam skala besar. Pengujian tidak dilakukan dalam kondisi event sesungguhnya.
8. Sistem tidak mencakup integrasi dengan perangkat pelacakan khusus (misalnya chip RFID atau sensor profesional) dan hanya memproses data lokasi berbasis koordinat yang dikirimkan dari aplikasi *frontend*.
9. Sistem tidak menangani kualifikasi dan pembayaran peserta. Akses terhadap fitur pendaftaran hanya diberikan oleh panitia kepada peserta yang terkualifikasi dan sudah melunasi pendaftaran.
10. Sistem mencatat waktu, posisi, dan status peserta, namun tidak akan mengelola peringkat.

I.5 Metodologi

Metodologi yang digunakan pada penelitian ini adalah pendekatan *Waterfall Model* dari *Software Development Life Cycle (SDLC)*. Pemilihan metodologi ini didasarkan pada batasan ruang lingkup sistem yang telah ditetapkan serta kebutuhan tahapan pengembangan yang sistematis dan terstruktur. Pengembangan dilakukan secara linear dan sekuensial, di mana setiap tahap harus diselesaikan sebelum melanjutkan ke tahap berikutnya, sehingga memudahkan pengukuran progres dan evaluasi hasil. Berikut tahapan-tahapan yang dilakukan:

1. Analisis Kebutuhan

Pada tahap ini dilakukan pengumpulan dan analisis kebutuhan sistem backend untuk *Real-Time Runner Tracking* dan *ETA Prediction*. Kegiatan dilakukan melalui studi literatur terkait sistem pelacakan, algoritma prediksi ETA, serta arsitektur backend yang scalable dan memiliki ketersediaan layanan yang tinggi. Selain itu, dilakukan observasi dan wawancara dengan pengguna untuk memahami alur operasional, pembagian segmen lintasan, *Water Station*, dan kebutuhan data posisi peserta. Hasil dari tahap ini adalah spesifikasi kebutuhan fungsional dan non-fungsional sistem.

2. Perancangan Sistem

Berdasarkan hasil analisis kebutuhan, tahap perancangan mencakup penyusunan arsitektur sistem baik secara high-level maupun low-level, model database, spesifikasi API, *sequence diagram*, serta arsitektur deployment yang dirancang untuk mendukung skalabilitas dan *high availability*. Selain itu, perancangan juga mempertimbangkan rencana pengujian, pipeline deployment, dan mekanisme observability untuk memudahkan monitoring

dan evaluasi sistem.

3. Implementasi

Tahap implementasi mencakup pengembangan backend sesuai desain yang telah dibuat. Proses dimulai dengan setup lingkungan pengembangan dan infrastruktur, kemudian diikuti dengan implementasi modul penerimaan dan pemrosesan data posisi pelari secara *real-time*, penyimpanan data, penyediaan API, serta perhitungan prediksi ETA. Setiap modul dikembangkan secara sistematis agar backend berfungsi secara optimal dan sesuai spesifikasi.

4. Deployment

Tahap ini mencakup penyebaran sistem backend pada lingkungan *cloud* dengan arsitektur deployment yang mendukung (*high availability*) dan skalabilitas. Konfigurasi sistem dirancang agar dapat berjalan stabil saat menerima beban ribuan peserta. Tahap ini juga memastikan backend dapat diintegrasikan dengan frontend dan komponen lain untuk tahap pengujian.

5. Pengujian

Pengujian dilakukan untuk memastikan sistem berjalan sesuai spesifikasi. Tahap pengujian memiliki dua fokus utama.

(a) Pengujian Fungsional

Pengujian fungsional dilakukan untuk memastikan seluruh kebutuhan fungsional yang telah didefinisikan pada tahap perancangan berjalan sesuai spesifikasi. Tahap ini mencakup pengecekan alur data, integrasi antar modul, validasi input dan output, serta mekanisme pelacakan dan perhitungan ETA yang menjadi inti dari sistem backend.

(b) Pengujian Non-Fungsional

Pengujian non-fungsional difokuskan pada evaluasi kinerja sistem dalam menangani beban tinggi dan menjaga kualitas layanan. Metode yang digunakan meliputi load testing dan simulasi data ribuan peserta. Metrik yang diukur mencakup *throughput*, *latency*, *error rate*, dan *resource utilization*. Hasil pengujian didokumentasikan secara detail untuk mendukung evaluasi efektivitas dan skalabilitas sistem.

6. Evaluasi

Tahap evaluasi meliputi analisis hasil pengujian untuk menilai efektivitas sistem backend. Analisis mencakup perbandingan performa prediksi ETA, akurasi data posisi pelari, dan pemenuhan kriteria keberhasilan yang telah ditetapkan. Hasil evaluasi digunakan sebagai dasar dokumentasi temuan, pembelajaran, dan rekomendasi perbaikan sistem.

BAB II

STUDI LITERATUR

II.1 Penulisan Gambar, Tabel, Rumus, dan Kode

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

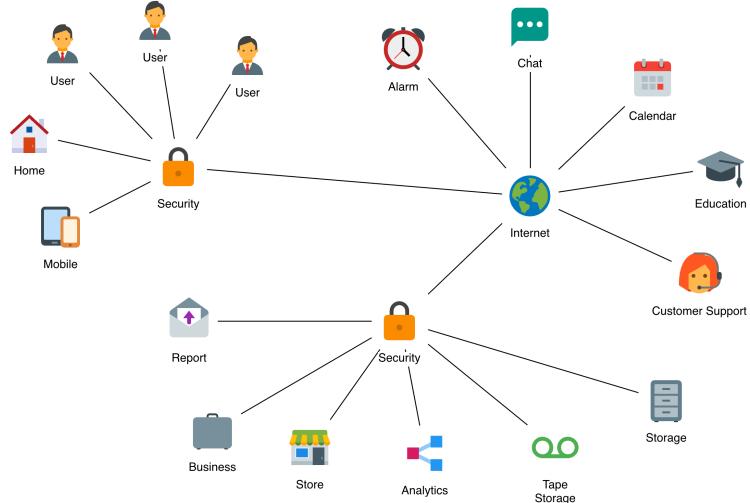
II.1.1 Gambar

Contoh gambar dapat dilihat pada Gambar II.1. Gambar dan judulnya diposisikan di tengah. Nomor gambar tidak diakhiri tanda titik. Gambar tersebut dibuat menggunakan aplikasi draw.io dan disimpan ke format PNG setelah dengan zoom setting pada angka 300%. Ukuran gambar yang ditampilkan dapat diatur dengan mengubah nilai *width* dalam sintaks *includegraphics*.

Gambar umumnya tidak jelas atau kabur jika gambar tersebut:

- a. diperoleh dari hasil cropping pada suatu halaman buku atau situs web;
- b. hasil pembesaran gambar yang gambar aslinya sebenarnya berukuran kecil; atau
- c. disimpan dalam resolusi kecil

Ketidakjelasan gambar ini dapat dilihat pada garis-garis diagram yang tidak tegas



Gambar II.1 Contoh gambar jaringan

dan tulisan-tulisan dalam gambar yang tampak kabur dan kurang jelas terbaca.

Untuk mendapatkan gambar yang tidak kabur (*blur*), langkah-langkah berikut dapat digunakan:

- (a) Gambar yang didapat di suatu pustaka atau referensi sebaiknya digambar ulang, misalnya menggunakan PowerPoint, Canva, Figma, draw.io, atau yang lainnya.
- (b) Jika diagram atau ilustrasi digambar menggunakan draw.io, saat gambar disimpan ke format PNG atau JPG (*export as*), lakukan *zoom* ke minimal 300% (*the default value is 100%*).
- (c) Jika diagram digambar dengan menggunakan PowerPoint, gambar dapat langsung di-*copy-paste* ke Word.

II.1.2 Tabel

Tabel ada dua jenis, yaitu tabel yang bisa termuat dalam satu halaman dan tabel yang sangat panjang sehingga tidak muat dalam satu halaman.

II.1.2.1 Tabel yang Muat dalam Satu Halaman

Contoh tabel dapat dilihat pada Tabel II.1 dan II.2. Tabel dan judulnya dibuat rata kiri dan judul tabel diletakkan di atas tabel. Usahakan tabel dapat ditulis dalam satu halaman, tidak terpotong ke halaman berikutnya.

Tabel II.1 Tabel harga bahan pokok

Nama	Satuan	Harga
Buku	Exemplar	25000
Komputer	Unit	2500000
Pensil	Buah	118900

Tabel II.2 Tabel harga bahan sekunder

Nama	Satuan	Harga
Buku	Exemplar	25000
Komputer	Unit	2500000
Pensil	Buah	118900

II.1.2.2 Mengimpor Tabel dari Berkas Eksternal

Tabel II.3 diimpor dari berkas eksternal *table/tabell.tex* menggunakan perintah *input*. Dengan demikian, jika tabel tersebut perlu diubah, cukup mengubah pada berkas eksternal tersebut tanpa perlu mengubah pada berkas utama ini.

Tabel II.3 Tabel harga bahan tertier

Nama	Satuan	Harga
Buku	Exemplar	25000
Komputer	Unit	2500000
Pensil	Buah	118900

II.1.2.3 Tabel yang Sangat Panjang

Jika tabel terlalu panjang sehingga tidak muat dalam satu halaman, gunakan paket *longtable* untuk membuat tabel yang dapat terpotong ke halaman berikutnya, seperti pada Tabel II.4.

Tabel II.4 Comprehensive Data Table Example

ID	Name	Score	Rank
1	Alice Smith	89	5
2	Bob Johnson	93	3
3	Carol Davis	95	2

Bersambung ke halaman berikutnya

Tabel II.4 Comprehensive Data Table Example (lanjutan)

ID	Name	Score	Rank
4	Daniel Wilson	88	6
5	Eve Thompson	97	1
6	Frank Brown	85	7
7	Grace Lee	91	4
8	Henry Miller	80	9
9	Irene Garcia	83	8
10	Jack Robinson	78	10
11	Kevin Harris	76	11
12	Laura Martin	75	12
13	Michael Clark	74	13
14	Natalie Lewis	73	14
15	Olivia Walker	72	15
16	Peter Hall	71	16
17	Quinn Allen	70	17
18	Rachel Young	69	18
19	Samuel King	68	19
20	Tina Wright	67	20
21	Uma Scott	66	21
22	Victor Green	65	22
23	Wendy Adams	64	23
24	Xavier Nelson	63	24
25	Yolanda Carter	62	25
26	Zachary Perez	61	26
27	Amelia Baker	60	27
28	Benjamin Rivera	59	28
29	Charlotte Rogers	58	29
30	David Murphy	57	30
31	Ethan Cooper	56	31
32	Fiona Reed	55	32
33	George Bailey	54	33
34	Hannah Cox	53	34
35	Isaac Howard	52	35
36	Julia Ward	51	36

Bersambung ke halaman berikutnya

Tabel II.4 Comprehensive Data Table Example (lanjutan)

ID	Name	Score	Rank
37	Kyle Flores	50	37
38	Lily Bell	49	38
39	Mason Sanders	48	39
40	Nora Patterson	47	40
41	Owen Ramirez	46	41
42	Penelope Torres	45	42
43	Quentin Foster	44	43
44	Rebecca Gonzales	43	44
45	Sebastian Bryant	42	45
46	Taylor Alexander	41	46
47	Ursula Russell	40	47
48	Vincent Griffin	39	48
49	William Diaz	38	49
50	Zoe Simmons	37	50

II.1.2.4 Beberapa Contoh Penulisan Rumus atau Persamaan Matematika Menggunakan LaTeX Termasuk Penomorannya

Contoh rumus matematika dapat ditulis seperti pada Persamaan II.1 di bawah ini. Penomoran persamaan diletakkan di sebelah kanan, dan rumus ditulis dalam mode *display math*.

$$E = mc^2 \quad (\text{II.1})$$

Contoh lain penulisan rumus matematika yang lebih kompleks dapat ditulis seperti pada Persamaan II.3.

$$f(x) = ax^2 + bx + c \quad (\text{II.2})$$

$$\begin{aligned} f'(x) &= \frac{d}{dx}(ax^2 + bx + c) \\ &= 2ax + b \end{aligned} \quad (\text{II.3})$$

Jika rumus terlalu panjang untuk ditulis dalam satu baris, gunakan lingkungan

multiline seperti pada Persamaan II.4 di bawah ini.

$$y = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7 \\ + a_8x^8 + a_9x^9 + a_{10}x^{10} \quad (\text{II.4})$$

Jika ada penurunan rumus yang terdiri dari beberapa baris, namun tidak memerlukan penomoran pada setiap baris, gunakan lingkungan *align**, misalnya:

$$\begin{aligned} S &= \sum_{i=1}^n i^2 \\ &= 1^2 + 2^2 + 3^2 + \cdots + n^2 \\ &= \frac{n(n+1)(2n+1)}{6} \end{aligned}$$

Contoh lainnya adalah rumus untuk mencari nilai rata-rata fungsi $f(x)$ pada interval $[p, q]$:

$$\begin{aligned} \bar{f} &= \frac{1}{q-p} \int_p^q f(x) dx \\ &= \frac{1}{q-p} \int_p^q (ax^2 + bx + c) dx \\ &= \frac{1}{q-p} \left[\frac{a}{3}x^3 + \frac{b}{2}x^2 + cx \right]_p^q \\ &= \frac{a(q^3 - p^3)}{3(q-p)} + \frac{b(q^2 - p^2)}{2(q-p)} + c \end{aligned}$$

II.1.3 Algoritma, Pseudocode, atau Kode

Contoh penulisan algoritma atau pseudocode dapat ditulis seperti pada Kode II.1 di bawah ini. Gunakan paket *listings* untuk menulis source code dalam bahasa pemrograman tertentu, seperti pada Kode II.2.

Kode II.1 Contoh pseudocode

```
ALGORITHM HelloWorld
    PRINT "Hello, World!"
END ALGORITHM
```

Tabel II.5 Contoh penggunaan kata "sedangkan" dan "sehingga"

Kata	Salah	Benar
sedangkan	Sedangkan sistem lama masih digunakan oleh banyak pengguna.	Sistem lama masih digunakan oleh banyak pengguna, sedangkan sistem baru belum siap.
sehingga	Sehingga sistem lama masih digunakan oleh banyak pengguna.	Sistem lama masih digunakan oleh banyak pengguna sehingga sistem baru belum siap.

Kode II.2 Contoh source code Python

```
def hello_world():
    print("Hello, World!")
hello_world()
```

II.2 Beberapa Kesalahan Penulisan yang Sering Terjadi

II.2.1 Penggunaan Kata "di mana" atau "dimana"

Banyak yang menuliskan kata "di mana" atau "dimana" sebagai pengganti kata "which" dalam bahasa Inggris. Padahal, penggunaan kata "di mana" atau "dimana" tidak tepat dalam konteks tersebut. Demikian juga untuk kata serupa, misalnya "yang mana". Kata "di mana" atau "dimana" ini harus diganti dengan kata lain, seperti "dengan", "tempat", "yang", dan sebagainya tergantung kalimatnya. Penjelasan lengkap dapat dilihat pada (*Buku Praktis Bahasa Indonesia 1/Kata - Wikisumber bahasa Indonesia 2024*).

II.2.2 Penggunaan Kata "sedangkan" dan "sehingga"

Kata "sedangkan" dan "sehingga" adalah kata hubung atau konjungsi. Konjungsi adalah kata atau ungkapan yang menghubungkan satuan bahasa (kata, frasa, klausa, dan kalimat). Konjungsi dapat dibagi menjadi konjungsi intrakalimat dan antarkalimat. Kata "sedangkan" menghubungkan dua klausa yang bersifat kontrasif, sedangkan "sehingga" menghubungkan dua klausa yang bersifat kausal. Dalam ragam formal, kata hubung "sedangkan" dan "sehingga" hanya dapat digunakan sebagai konjungsi intrakalimat sehingga kedua konjungsi itu **tidak dapat diletakkan pada awal kalimat**. Selain itu, penggunaan kata "sedangkan" harus didahului oleh koma (,), sedangkan kata "sehingga" tidak perlu didahului oleh koma (.). Contoh penggunaan yang benar dan salah dapat dilihat pada Tabel II.5.

II.2.3 Penggunaan Istilah yang Tidak Baku

Ada beberapa istilah yang sering digunakan dalam pembicaraan sehari-hari, tetapi tidak baku dalam penulisan ilmiah. Beberapa istilah tersebut antara lain:

1. analisa → analisis
2. eksisting atau existing → yang ada atau saat ini
3. bisnis proses → proses bisnis
4. user → pengguna
5. system → sistem
6. database → basis data
7. aktifitas → aktivitas
8. efektifitas → efektivitas
9. sosial media → media sosial

II.2.4 Pemisah Desimal dan Ribuan

Tanda pemisah desimal dalam bahasa Indonesia adalah tanda koma, contoh:

1. (Salah) Akurasi naik menjadi 50.6%
2. (Benar) Akurasi naik menjadi 50,6%

II.2.5 Daftar atau *List*

Ada beberapa aturan penulisan daftar atau *list* yang perlu diperhatikan, antara lain:

- a) Jika memungkinkan, hindari penggunaan “bullet points” atau sejenisnya. Sebaiknya, gunakan angka (1, 2, 3, ...) atau huruf (a, b, c, ...). Dengan demikian, pembaca dapat dengan mudah melihat jumlah *item* atau *list*.
- b) Jika dalam daftar hanya ada satu item, tidak perlu menggunakan nomor urut.
- c) Penjelasan atau deskripsi suatu item sebaiknya menyatu dengan judul item tersebut, tidak berbeda halaman. Contoh yang salah: judul item ada di halaman 10, namun deskripsinya di halaman 11. Sebaiknya pindahkan judul tersebut ke halaman 11.
- d) Jika penjelasan atau deskripsi suatu item cukup panjang, misalnya lebih dari 1 halaman atau terdiri atas beberapa paragraf, sebaiknya setiap item tersebut dijadikan judul subbab, kecuali jika level subbab sudah mencapai level 4.

II.2.6 Penggunaan Kata ”masing-masing” dan ”setiap”

Kata ”masing-masing” digunakan di belakang kata yang diterangkan, misalnya ”Setiap proses menggunakan algoritma masing-masing”. Kata ”tiap-tiap” atau

“setiap” ditempatkan di depan kata yang diterangkan, misalnya ”Setiap proses menggunakan algoritma tertentu”.

BAB III

ANALISIS MASALAH

III.1 Analisis Kondisi Saat Ini

Seiring dengan meningkatnya kebutuhan akan sistem pelacakan pelari yang akurat dan dapat diakses secara *real-time* dalam penyelenggaraan ITB Ultra-Marathon, penting untuk memahami terlebih dahulu kondisi teknologi dan pendekatan yang saat ini digunakan di industri maupun penelitian. Analisis ini diperlukan untuk mengidentifikasi bagaimana sistem pencatatan waktu konvensional bekerja, sejauh mana teknologi pelacakan kontinu telah berkembang, serta apa saja keterbatasan pada metode prediksi ETA yang tersedia. Pemahaman menyeluruh terhadap kondisi eksisting menjadi dasar dalam merumuskan rancangan *backend* yang mampu menangani skala besar, mendukung prediksi yang lebih kontekstual, dan memenuhi kebutuhan operasional maraton secara spesifik.

III.1.1 Model Sistem Berbasis RFID

Standar industri dalam manajemen lomba lari massal saat ini adalah sistem *timing* berbasis *Radio Frequency Identification* (RFID), khususnya RFID pasif UHF. Model konseptual sistem ini mengandalkan interaksi antara *tag* yang dibawa pelari dengan infrastruktur pembaca yang dipasang pada titik-titik tertentu (Sjöbeck August 2022).

Komponen utama dalam sistem ini meliputi:

1. Transponder Pasif (Tag): *Chip* tanpa baterai yang ditempelkan pada nomor dada pelari dan hanya aktif ketika menerima energi dari pembaca.
2. Infrastruktur *Reader/Mats*: Antena atau karpet pembaca yang ditempatkan pada titik diskrit seperti garis *start*, *split* setiap beberapa kilometer, dan garis *finish*.
3. Dekoder dan Perangkat Lunak: Perangkat yang mengonversi sinyal menjadi

ID pelari dan stempel waktu untuk diproses lebih lanjut.

Sistem RFID memiliki keterbatasan mendasar berupa sifat data yang diskrit. Posisi pelari hanya diketahui ketika melewati titik pembaca, sehingga terdapat zona buta antar *checkpoint* (Hochreiter 2024). Akibatnya, estimasi posisi maupun kecepatan bergantung pada interpolasi sederhana. Selain itu, pemasangan infrastruktur RFID memerlukan biaya dan tenaga yang signifikan sehingga jumlah titik baca tidak dapat diperbanyak secara fleksibel, terutama pada rute maraton yang panjang.

III.1.2 Model Sistem Berbasis LoRaWan dan GPS

Sebagai perbaikan atas rendahnya resolusi spasial RFID, riset terbaru mengeksplorasi teknologi *Low-Power Wide-Area Network* (LPWAN) seperti LoRaWAN untuk pelacakan kontinu. Riset mengusulkan model di mana pelari membawa pelacak GPS berbasis LoRa yang mengirimkan data posisi secara periodik, tidak terbatas pada titik diskrit(Hochreiter 2024).

Model konseptual dalam studi tersebut mencakup:

1. *Tracker Node*: Perangkat LoRa berbasis GPS yang mengirimkan pembaruan posisi secara berkala (misalnya setiap 30 detik).
2. *Mobile Gateway*: *Gateway* bergerak yang dipasang pada kendaraan atau sepeda untuk meningkatkan cakupan sinyal di lingkungan lomba yang dinamis.
3. *LoRaWAN Network Server (LNS)*: Server yang menerima, mendekode, dan menyimpan paket data untuk visualisasi dan analisis lebih lanjut.

Eksperimen pada Vienna City Marathon menunjukkan bahwa pendekatan ini mampu menyediakan data posisi dengan resolusi jauh lebih tinggi dibandingkan RFID, yaitu median interval sekitar 31 detik. Namun, pendekatan ini menuntut infrastruktur *backend* yang sanggup menangani aliran data telemetri yang terus-menerus dan berjumlah besar.

III.1.3 Kesenjangan pada Metode Prediksi ETA

Walaupun pelacakan kontinu semakin memungkinkan, metode prediksi ETA yang digunakan saat ini belum mampu memenuhi kebutuhan konteks lomba lari jarak jauh. Terdapat beberapa kesenjangan utama:

1. Model yang Kurang Relevan: Banyak metode ETA yang tersedia dikembangkan untuk kendaraan atau logistik sehingga mengasumsikan kecepatan relatif stabil, bukan perubahan performa fisiologis pelari.

2. Kurangnya Pertimbangan Konteks Lari: Metode konvensional sering kali tidak memperhitungkan karakteristik seperti elevasi rute, strategi *pacing*, maupun dampak kelelahan.
3. Kendala Skalabilitas dan Ketersediaan: Sistem ETA eksisting umumnya tidak dirancang untuk menghitung prediksi bagi ribuan peserta secara bersamaan setiap kali data baru diterima. Dalam *event* berskala besar, diperlukan arsitektur *backend* yang mampu menjamin *high availability* dan pemrosesan *real-time*.

Dengan demikian, terdapat kebutuhan akan sistem *backend* yang mampu mengolah data pelacakan kontinu sekaligus menyediakan prediksi ETA yang cerdas, skalabel, dan andal.

III.2 Analisis Kebutuhan

Tahap analisis kebutuhan dilakukan untuk memperoleh gambaran yang komprehensif mengenai tujuan, batasan, serta karakteristik sistem yang akan dibangun. Pada tahap ini, dilakukan identifikasi terhadap permasalahan dan kebutuhan pengguna yang muncul dalam operasional ITB Ultra-Marathon. Selain itu, dirumuskan kebutuhan fungsional yang mendefinisikan kemampuan utama yang harus disediakan sistem, serta kebutuhan nonfungsional yang memastikan sistem mampu beroperasi secara andal, aman, dan skalabel sesuai standar yang dibutuhkan. Hasil dari analisis kebutuhan ini menjadi dasar dalam proses perancangan arsitektur dan implementasi sistem pada tahap berikutnya.

III.2.1 Identifikasi Masalah Pengguna

Berdasarkan observasi dan wawancara mengenai kondisi penyelenggaraan ITB Ultra-Marathon saat ini, terdapat sejumlah permasalahan utama yang dialami oleh panitia, peserta, maupun pihak pendukung acara. Permasalahan tersebut dapat diidentifikasi sebagai berikut.

1. Pelacakan posisi pelari masih manual dan tidak real-time.
Panitia dan tim supporter harus meminta pelari melakukan *share location* melalui aplikasi perpesanan untuk mengetahui posisi terkini, sehingga proses pemantauan menjadi tidak efisien.
2. Mobil penjemput dan *drop bag* sulit untuk tepat waktu karena ETA tidak akurat.
Keterlambatan terjadi karena tidak tersedianya data posisi dan kecepatan pelari yang diperbarui secara kontinu untuk memperkirakan estimasi waktu

kedatangan dengan tepat.

3. Metode ETA yang tersedia (misalnya Google Maps) tidak sesuai untuk konteks pelari.

Pola *pace*, kelelahan, elevasi lintasan, serta karakteristik rute tidak diperhitungkan, sehingga estimasi waktu kedatangan sering meleset dan tidak dapat diandalkan.

4. Peta digital lomba masih dibuat secara manual.

Peta rute biasanya dibuat menggunakan Google Maps secara manual dan tidak terintegrasi dengan sistem pelacakan, meskipun pelari menggunakan aplikasi tersebut untuk mendapatkan arah rute dan estimasi waktu tempuh.

5. Tidak ada sistem yang mendukung kebutuhan operasional lomba secara menyeluruh.

Hal-hal seperti manajemen tim, deteksi pergantian pelari untuk kategori relay, pengelolaan pelanggaran dan penalti, pencatatan kehadiran, perkiraan jadwal penjemputan, *drop bag* masih dilakukan secara manual dan rentan terhadap kesalahan.

III.2.2 Kebutuhan Fungsional

Pada tahap ini, disusun kebutuhan fungsional berdasarkan permasalahan yang telah diidentifikasi pada bagian sebelumnya. Selain itu, kebutuhan ini juga mencerminkan fungsi-fungsi dasar yang harus disediakan oleh sebuah sistem pelacakan marathon modern agar mampu memenuhi kebutuhan panitia, pelari, dan tim supporter secara efektif dan konsisten.

Tabel III.1 Kebutuhan Fungsional Sistem

Kode	Kebutuhan	Deskripsi
FR-01	API Pendaftaran Peserta	Backend menyediakan API untuk pendaftaran peserta, termasuk validasi data identitas dan penyimpanan informasi dasar peserta ke dalam database.
FR-02	API Manajemen Tim	Backend menyediakan endpoint untuk membuat, mengelola, dan menghubungkan peserta ke dalam tim, termasuk sinkronisasi struktur tim dengan modul lomba dan modul pelacakan.

Kode	Kebutuhan	Deskripsi
FR-03	API Manajemen Lomba	Backend menyediakan API bagi panitia untuk membuat, memperbarui, dan mengambil konfigurasi event seperti jadwal, kategori lomba, serta aturan dan parameter lomba.
FR-04	API Pengelolaan Trek	Backend menyediakan API untuk menerima data rute lomba (GPX/TCX), pengaturan checkpoint dan water station, serta pendefinisian jam buka, jam tutup, dan cutoff time untuk setiap titik.
FR-05	Akuisisi dan Distribusi Data Tracking	Backend menerima data lokasi peserta melalui endpoint ingestion dan menyediakan API streaming atau REST untuk menyalurkan data posisi terbaru kepada pihak yang berwenang.
FR-06	Layanan Backend Perhitungan ETA	Backend menjalankan komponen perhitungan ETA berdasarkan pace, progres rute, elevasi, dan dinamika performa pelari, serta menyediakan API untuk pengambilan nilai ETA.
FR-07	API Deteksi Pergantian Pelari	Backend menyediakan mekanisme serta API untuk mencatat dan mendeteksi pergantian pelari dalam satu tim berdasarkan checkpoint atau input panitia.
FR-08	API Manajemen Status Peserta	Backend menyediakan endpoint untuk memperbarui status peserta (DNS, DNF, finished, disqualified, running), mencatat waktu start, checkpoint, penalti, dan waktu selesai.
FR-09	API Permintaan Bantuan	Backend menyediakan endpoint bagi peserta untuk mengirim sinyal bantuan, permintaan berhenti, atau permintaan evakuasi, serta mendistribusikan event tersebut ke modul panitia.
FR-10	Manajemen Pengguna dan RBAC	Backend menyediakan autentikasi, otorisasi, manajemen token, dan pembatasan akses sehingga pengguna hanya dapat melihat data sesuai perannya (panitia, peserta, supporter).
FR-11	Penyimpanan Riwayat dan Audit Log	Backend menyimpan riwayat posisi, kecepatan, perubahan status, aktivitas panitia, serta log penting lainnya yang dapat diakses kembali melalui API.

Kode	Kebutuhan	Deskripsi
FR-12	Sinkronisasi dan Buffering Offline	Backend menyediakan mekanisme penerimaan data tertunda akibat koneksi buruk serta melakukan sinkronisasi ulang ketika koneksi pulih.

III.2.3 Kebutuhan Nonfungsional

Pada bagian sebelumnya telah dijabarkan kebutuhan fungsional yang menggambarkan perilaku dan layanan utama yang harus disediakan oleh sistem. Namun, pemenuhan fungsi-fungsi tersebut tidak akan berjalan efektif tanpa dukungan kualitas sistem yang memadai. Oleh karena itu, pada bagian ini disusun kebutuhan nonfungsional yang berfokus pada karakteristik teknis, performa, keandalan, keamanan, dan aspek operasional lain yang harus dipenuhi oleh sistem.

Kebutuhan nonfungsional ini memastikan bahwa setiap fungsi yang telah dirumuskan tidak hanya berjalan dengan benar, tetapi juga konsisten, aman, responsif, serta mampu beroperasi dalam skala yang diperlukan oleh kegiatan lomba marathon yang dinamis dan intensif terhadap data. Dengan demikian, bagian ini mendefinisikan standar kualitas yang menjadi fondasi bagi desain arsitektur dan implementasi sistem pada tahap selanjutnya.

Tabel III.2 Kebutuhan Nonfungsional Sistem

Kode	Kebutuhan	Deskripsi
NF-01	Ketersediaan Sistem	Sistem backend harus memiliki tingkat <i>availability</i> minimal 99.9% selama periode lomba, sehingga layanan inti seperti pelacakan real-time, ETA, dan dashboard panitia tetap dapat diakses tanpa gangguan.
NF-02	Skalabilitas	Sistem harus mampu menangani peningkatan jumlah pengguna, klien pelacakan, dan request API secara dinamis dengan mendukung mekanisme autoscaling dan arsitektur terdistribusi di lingkungan cloud.

Kode	Kebutuhan	Deskripsi
NF-03	Performa API	Endpoint API harus merespons dalam waktu kurang dari 300 ms untuk operasi utama seperti pengambilan posisi pelari, status, dan data ETA, demi menjamin pengalaman real-time.
NF-04	Keamanan	Sistem harus menerapkan autentikasi dan otorisasi berbasis token, enkripsi data dalam transit (HTTPS/TLS), pembatasan akses berdasarkan peran, serta perlindungan terhadap akses tidak sah dan penyalahgunaan API.
NF-05	Fault Tolerance	Sistem harus tetap beroperasi meskipun terjadi kegagalan pada salah satu komponen (server, pod, database replica, dsb.) dengan menerapkan <i>redundancy</i> dan <i>graceful degradation</i> .
NF-06	Monitoring dan Observability	Sistem menyediakan logging terstruktur, metrik performa, tracing request, dan dashboard monitoring untuk mendeteksi anomali secara cepat.

III.3 Analisis Pemilihan Solusi

Pada tahap sebelumnya telah dijabarkan kebutuhan fungsional dan nonfungsional yang harus dipenuhi oleh sistem pelacakan real-time dan prediksi ETA. Tahap berikutnya adalah menentukan pendekatan teknis yang paling sesuai untuk memenuhi kebutuhan tersebut. Bab ini menyajikan proses pemilihan solusi secara sistematis, dimulai dari identifikasi alternatif yang memungkinkan, analisis komparatif berdasarkan kriteria tertentu, hingga penetapan solusi akhir yang akan digunakan dalam perancangan sistem.

III.3.1 Alternatif Solusi

Bagian ini menguraikan berbagai alternatif solusi yang relevan untuk setiap komponen utama sistem, seperti arsitektur backend, metode akuisisi telemetri, model ETA, penyimpanan data, dan platform deployment. Identifikasi alternatif dilakukan untuk memastikan bahwa seluruh pendekatan yang layak

dipertimbangkan secara objektif sebelum dilakukan analisis pemilihan pada tahap berikutnya.

Arsitektur Backend

Beberapa alternatif arsitektur dipertimbangkan untuk mendukung kebutuhan sistem pelacakan real-time dan perhitungan ETA pada ITB Ultra-Marathon. Setiap pendekatan memiliki karakteristik berbeda terkait skalabilitas, kompleksitas, serta kemudahan pengembangan.

- Layered Architecture

Pendekatan yang memisahkan berdasarkan lapisan (presentation, service, repository), dengan masing-masing lapisan dapat dideploy secara terpisah (misal front-end, back-end, dan database). Struktur ini mudah dipahami dan cocok untuk tim kecil, serta memungkinkan skalabilitas lebih baik dibanding monolith tunggal, meskipun modul ingest data telemetri mungkin masih menjadi bottleneck jika trafik sangat tinggi.

- Modular Monolithic

Seluruh fungsi dideploy dalam satu unit, tetapi dipisahkan secara modular berdasarkan domain. Memberikan latensi rendah dan kompleksitas operasional minimal, sekaligus tetap terstruktur.

- Microservices

Setiap domain (tracking, event, ETA, tim, notifikasi) berjalan sebagai layanan terpisah. Mendukung penskalaan selektif, namun meningkatkan kebutuhan orkestrasi dan observabilitas.

- Event-Driven Architecture

Komunikasi melalui *event bus*. Ideal untuk pemrosesan telemetri, deteksi checkpoint, dan pembaruan ETA secara streaming.

Protokol Komunikasi Telemetri

Beberapa opsi protokol dipertimbangkan untuk mengirimkan data lokasi pelari ke backend secara real-time. Masing-masing memiliki karakteristik terkait latensi, efisiensi jaringan, dan kompleksitas implementasi.

- HTTPS REST Interval-based

Pelari mengirimkan data lokasi setiap X detik melalui permintaan HTTP POST. Pendekatan ini sederhana, mudah diimplementasikan, dan memiliki kompatibilitas tinggi dengan berbagai platform.

- WebSocket

Mengirimkan telemetri secara kontinu tanpa overhead *HTTP handshake* berulang. Cocok untuk kebutuhan pelacakan *real-time* dengan interval 1–2 detik dan volume koneksi besar.

- **MQTT**

Protokol IoT yang ringan dan dirancang untuk koneksi jaringan yang tidak stabil. Ideal untuk area rural dengan kualitas internet yang sering fluktuatif.

Penyimpanan Data Telemetri (Time-Series)

Beberapa opsi basis data deret-waktu dipertimbangkan untuk menyimpan lokasi pelari, kecepatan, serta data telemetri lainnya. Setiap solusi memiliki kekuatan berbeda dalam hal performa query, skalabilitas, dan kemudahan integrasi.

- **TimescaleDB (PostgreSQL-based TSDB)**

Memudahkan eksekusi query per pelari, jarak, maupun checkpoint. Cocok digunakan untuk beban menengah dengan kebutuhan analisis yang fleksibel.

- **InfluxDB**

Basis data yang dioptimalkan khusus untuk data *time-series*. Memberikan performa query sangat cepat untuk kebutuhan analisis historis.

Platform Deployment

Berbagai opsi platform deployment dipertimbangkan untuk menyeimbangkan fleksibilitas, skalabilitas, dan kompleksitas operasional sistem.

- **Cloud VM (IaaS)**

Menggunakan layanan seperti GCP Compute Engine atau AWS EC2. Memberikan kontrol penuh terhadap lingkungan dan konfigurasi, serta menawarkan fleksibilitas tertinggi.

- **Container Orchestration (Kubernetes)**

Cocok untuk arsitektur dengan banyak *service*. Mendukung *auto-scaling* pada modul ingest telemetri dan memudahkan pengelolaan deployment berskala besar.

- **Serverless**

Menggunakan layanan seperti Cloud Run atau AWS Lambda. Ideal untuk modul yang berjalan tidak kontinu, seperti layanan ETA API, notifikasi, atau API pendaftaran.

Monitoring (Metrics)

Sistem memerlukan pemantauan metrik untuk memastikan proses ingest telemetri, perhitungan ETA, dan API tetap beroperasi secara optimal. Beberapa opsi monitoring yang dipertimbangkan adalah:

- Prometheus + Grafana
Solusi open-source standar industri. Mendukung pemantauan metrik seperti *ingestion rate*, *ETA latency*, dan *API throughput*.
- Datadog Metrics
Layanan *managed* yang menawarkan integrasi cepat dan operasional sederhana. Cocok untuk acara jangka pendek yang membutuhkan setup minim.
- Cloud-Native Monitoring (AWS CloudWatch / GCP Cloud Monitoring)
Integrasi langsung dengan layanan cloud tanpa perlu instalasi tambahan. Mendukung pengumpulan metrik, log, dan *alerts* secara otomatis untuk VM, Kubernetes, maupun serverless.

Logging (Event & System Logs)

Sistem membutuhkan mekanisme pencatatan log yang andal untuk debugging, audit, dan analisis event selama lomba berlangsung. Beberapa opsi yang dipertimbangkan antara lain:

- ELK Stack (Elasticsearch + Logstash + Kibana)
Solusi populer yang sangat powerful untuk pencarian dan analisis log skala besar. Cocok untuk kebutuhan *event tracking* yang kompleks.
- Loki + Grafana
Pendekatan yang lebih ringan dengan fokus pada *log streaming*. Sederhana, efisien, dan lebih hemat sumber daya dibanding ELK.
- Cloud-Native Logging (AWS CloudWatch Logs / GCP Cloud Logging)
Integrasi bawaan dari platform cloud. Mendukung pengumpulan log otomatis dari VM, Kubernetes, maupun layanan serverless, dengan konfigurasi minimal dan *managed retention*.

Tracing (Distributed Tracing)

Tracing diperlukan untuk menganalisis alur request antar layanan, terutama jika sistem menggunakan pendekatan microservice atau modul-modul terpisah. Beberapa opsi tracing yang dipertimbangkan:

- Jaeger

Solusi open-source populer untuk *distributed tracing*. Jaeger dapat diintegrasikan dengan OpenTelemetry sebagai layer instrumentasi, maupun dengan ekosistem Prometheus, Grafana, dan Loki (melalui Grafana Tempo atau dashboard Jaeger). Cocok untuk sistem dengan banyak layanan yang saling berkomunikasi.

- Datadog APM

Layanan *managed* dengan instalasi sederhana dan integrasi otomatis. Menyediakan tampilan *end-to-end tracing* dengan konfigurasi minimal, ideal untuk event jangka pendek.

- Cloud-Native Tracing (AWS X-Ray / GCP Cloud Trace)

Layanan tracing terkelola penuh dari penyedia cloud. Terintegrasi langsung dengan layanan cloud-native seperti serverless, load balancer, dan service mesh, sehingga mengurangi overhead operasional dalam pengelolaan infrastruktur tracing.

Model Perhitungan ETA Marathon

Beberapa pendekatan umum yang banyak digunakan pada aplikasi lari dan perangkat GPS dalam menghitung estimasi waktu tiba (ETA) pelari marathon.

- Ekstrapolasi Pace Rata-Rata

Menggunakan pace rata-rata sejauh ini untuk memproyeksikan waktu tempuh sisa jarak. Sederhana dan cepat, namun tidak mempertimbangkan kelelahan atau variasi medan.

- Model Pace Berbasis Segmen

Mengambil pace terbaru (mis. beberapa kilometer terakhir) dan memprediksi sisa jarak per segmen. Dapat memasukkan faktor pelambatan bertahap atau kondisi rute sehingga hasil lebih realistik.

- Model Berbasis Data Historis

Menggabungkan data performa sebelumnya, kondisi lingkungan, serta profil rute dalam model regresi atau pembelajaran mesin. Lebih personal dan akurat, tetapi memerlukan data dan proses kalibrasi tambahan.

Integrasi Elevasi dalam Perhitungan ETA

Elevasi merupakan faktor penting dalam memproyeksikan performa pelari karena tanjakan dan turunan dapat mengubah kebutuhan usaha secara signifikan. Beberapa pendekatan umum untuk memasukkan faktor ini adalah:

- Grade-Adjusted Pace (GAP)

Mengoreksi pace aktual berdasarkan kemiringan lintasan sehingga waktu tempuh pada segmen menanjak atau menurun dapat disetarakan dengan kondisi datar. Hasil pace yang telah dikoreksi digunakan dalam model ETA apa pun.

- Elevation-Corrected Segment ETA

Menghitung ETA per segmen dengan menambahkan penalti atau bonus waktu bergantung pada elevasi segmen tersebut. Pendekatan ini umum pada aplikasi tracking berbasis rute.

III.3.2 Analisis Penentuan Solusi

Analisis penentuan solusi pada penelitian ini menggunakan metode Multi-Criteria Decision Analysis (MCDA) dengan pendekatan Weighted Scoring. Metode ini dipilih karena mampu mengevaluasi alternatif solusi teknis berdasarkan bobot kriteria yang relevan seperti performa, skalabilitas, reliabilitas, kompleksitas implementasi, dan biaya.

Arsitektur Backend

- **Skalabilitas** (35%): Kemampuan arsitektur untuk menangani peningkatan jumlah data dan pengguna secara efisien.
- **Kompleksitas Pengembangan dan Operasional** (25%): Tingkat kesulitan implementasi dan pemeliharaan sistem.
- **Latensi Pemrosesan Telemetri** (25%): Kecepatan arsitektur dalam memproses data telemetri secara real-time.
- **Kesesuaian dengan Real-Time Event/Streaming** (15%): Kemampuan arsitektur untuk mendukung aliran data berbasis event secara kontinu.

Penilaian tiap alternatif dilakukan dengan skala 1–10, di mana nilai lebih tinggi menunjukkan performa yang lebih baik. Hasil penilaian disajikan pada tabel berikut:

Tabel III.3 Penilaian Tiap Alternatif Arsitektur Backend

Arsitektur	Skalabilitas	Kompleksitas	Latensi	Real-Time Events
Layered Architecture	6	8	7	5
Modular Monolithic	4	8	8	7
Microservices	9	5	8	9
Event-Driven Architecture	9	6	9	10

Berdasarkan bobot kriteria, skor tertimbang tiap alternatif dihitung dengan rumus:

$$\text{Skor Total} = \sum(\text{nilai kriteria} \times \text{bobot})$$

Tabel III.4 Hasil Perhitungan Weighted Score Arsitektur Backend

Arsitektur	Skor Total	Ranking
Event-Driven Architecture	8.40	1
Microservices	7.75	2
Layered Architecture	6.60	3
Modular Monolithic	6.45	4

Berdasarkan hasil MCDA weighted scoring, **Event-Driven Architecture** merupakan arsitektur backend yang paling sesuai untuk sistem ITB Ultra-Marathon. Pendekatan ini unggul dalam menangani data telemetri secara real-time, mendukung aliran event streaming, serta memiliki kemampuan skalabilitas tinggi. Meskipun kompleksitas pengembangan sedikit lebih tinggi dibandingkan Layered Architecture dan Modular Monolithic, keuntungan dalam performa real-time dan fleksibilitas penskalaan membuat Event-Driven Architecture menjadi pilihan optimal. Microservices menjadi alternatif kedua yang layak dipertimbangkan jika modularitas layanan terpisah menjadi prioritas.

Protokol Komunikasi Telemetri

Untuk menentukan protokol komunikasi telemetri yang paling sesuai dalam pengiriman data lokasi pelari ke backend secara real-time, digunakan pendekatan Multi-Criteria Decision Analysis (MCDA) dengan metode weighted scoring. Kriteria yang digunakan beserta bobotnya adalah sebagai berikut:

- **Latensi Pengiriman Data** (35%): Kecepatan protokol dalam menyampaikan data lokasi secara real-time.
- **Efisiensi Jaringan** (30%): Penggunaan bandwidth dan overhead protokol dalam kondisi jaringan terbatas.
- **Kompleksitas Implementasi** (20%): Tingkat kesulitan pengembangan dan integrasi protokol.
- **Kestabilan Koneksi pada Kondisi Fluktuatif** (15%): Kemampuan protokol menjaga koneksi stabil di area dengan kualitas jaringan tidak konsisten.

Penilaian tiap protokol dilakukan dengan skala 1–10, di mana nilai lebih tinggi

menunjukkan performa yang lebih baik. Hasil penilaian disajikan pada tabel berikut:

Tabel III.5 Penilaian Tiap Alternatif Protokol Komunikasi Telemetri

Protokol	Latensi	Efisiensi Jaringan	Kompleksitas	Kestabilan Koneksi
HTTPS REST Interval-based	6	7	9	6
WebSocket	9	8	7	7
MQTT	8	9	8	9

Berdasarkan bobot kriteria, skor tertimbang tiap alternatif dihitung dengan rumus:

$$\text{Skor Total} = \sum (\text{nilai kriteria} \times \text{bobot})$$

Tabel III.6 Hasil Perhitungan Weighted Score Protokol Komunikasi Telemetri

Protokol	Skor Total	Ranking
MQTT	8.50	1
WebSocket	8.10	2
HTTPS REST Interval-based	7.05	3

Berdasarkan hasil MCDA weighted scoring, **MQTT** menjadi protokol komunikasi telemetri yang paling sesuai untuk sistem pelacakan ITB Ultra-Marathon. Protokol ini unggul dalam efisiensi jaringan dan kestabilan koneksi di area dengan kualitas internet yang berfluktuasi, serta memiliki latensi yang rendah dan kompleksitas implementasi yang moderat. WebSocket menjadi alternatif kedua yang baik untuk kebutuhan real-time dengan volume koneksi besar, sementara HTTPS REST Interval-based tetap layak dipertimbangkan untuk implementasi sederhana dan kompatibilitas platform yang tinggi.

Penyimpanan Data Telemetri (Time-Series)

Untuk menentukan solusi basis data deret-waktu yang paling sesuai dalam menyimpan lokasi pelari, kecepatan, dan data telemetri lainnya, digunakan pendekatan Multi-Criteria Decision Analysis (MCDA) dengan metode weighted scoring. Kriteria yang digunakan beserta bobotnya adalah sebagai berikut:

- **Performa Query** (40%): Kecepatan eksekusi query, baik untuk analisis historis maupun per pelari/checkpoint.

- **Skalabilitas** (30%): Kemampuan basis data menangani peningkatan jumlah data dan pengguna secara efisien.
- **Kemudahan Integrasi dan Pemeliharaan** (30%): Tingkat kesulitan integrasi dengan sistem backend dan kompleksitas operasional.

Penilaian tiap alternatif dilakukan dengan skala 1–10, di mana nilai lebih tinggi menunjukkan performa yang lebih baik. Hasil penilaian disajikan pada tabel berikut:

Tabel III.7 Penilaian Tiap Alternatif Basis Data Time-Series

Basis Data	Performa Query	Skalabilitas	Integrasi & Pemeliharaan
TimescaleDB	8	7	8
InfluxDB	9	8	7

Berdasarkan bobot kriteria, skor tertimbang tiap alternatif dihitung dengan rumus:

$$\text{Skor Total} = \sum (\text{nilai kriteria} \times \text{bobot})$$

Tabel III.8 Hasil Perhitungan Weighted Score Basis Data Time-Series

Basis Data	Skor Total	Ranking
InfluxDB	8.20	1
TimescaleDB	7.75	2

Berdasarkan hasil MCDA weighted scoring, **InfluxDB** merupakan pilihan basis data deret-waktu yang paling sesuai untuk penyimpanan telemetri ITB Ultra-Marathon. InfluxDB unggul dalam performa query untuk analisis historis dan mampu diskalakan dengan baik seiring pertumbuhan jumlah data. TimescaleDB tetap menjadi alternatif yang layak dipertimbangkan jika kebutuhan analisis fleksibel per pelari atau checkpoint lebih diutamakan, serta jika integrasi dengan ekosistem PostgreSQL menjadi prioritas.

Platform Deployment

Untuk menentukan platform deployment yang paling sesuai bagi sistem ITB Ultra-Marathon, digunakan pendekatan Multi-Criteria Decision Analysis (MCDA) dengan metode weighted scoring. Kriteria yang digunakan beserta bobotnya adalah sebagai berikut:

- **Fleksibilitas Konfigurasi** (30%): Kemampuan platform untuk mendukung konfigurasi khusus dan pengaturan lingkungan sesuai kebutuhan.
- **Skalabilitas** (35%): Kemampuan platform menangani peningkatan beban, baik dari sisi jumlah pengguna maupun volume data telemetri.
- **Kompleksitas Operasional** (20%): Tingkat kesulitan dalam pengelolaan, pemeliharaan, dan monitoring sistem.
- **Efisiensi Biaya** (15%): Perbandingan biaya operasional relatif terhadap performa dan kapasitas yang diperoleh.

Penilaian tiap alternatif dilakukan dengan skala 1–10, di mana nilai lebih tinggi menunjukkan performa yang lebih baik. Hasil penilaian disajikan pada tabel berikut:

Tabel III.9 Penilaian Tiap Alternatif Platform Deployment

Platform	Fleksibilitas	Skalabilitas	Kompleksitas Operasional
Cloud VM (IaaS)	9	7	6
Container Orchestration (Kubernetes)	8	9	7
Serverless	6	8	9

Berdasarkan bobot kriteria, skor tertimbang tiap alternatif dihitung dengan rumus:

$$\text{Skor Total} = \sum (\text{nilai kriteria} \times \text{bobot})$$

Tabel III.10 Hasil Perhitungan Weighted Score Platform Deployment

Platform	Skor Total	Ranking
Container Orchestration (Kubernetes)	8.50	1
Serverless	7.95	2
Cloud VM (IaaS)	7.25	3

Berdasarkan hasil MCDA weighted scoring, **Container Orchestration (Kubernetes)** menjadi pilihan platform deployment yang paling sesuai untuk sistem ITB Ultra-Marathon. Platform ini unggul dalam hal skalabilitas dan fleksibilitas konfigurasi, serta memudahkan pengelolaan layanan berskala besar. Serverless merupakan alternatif yang baik untuk modul yang berjalan tidak kontinu, dengan efisiensi biaya tinggi dan kompleksitas operasional rendah. Cloud VM (IaaS) tetap layak dipertimbangkan jika kontrol penuh terhadap lingkungan dan konfigurasi menjadi prioritas, meskipun skalabilitas dan efisiensi biayanya lebih terbatas dibandingkan opsi lainnya.

Monitoring Metrics

Untuk memastikan proses ingest telemetri, perhitungan ETA, dan API beroperasi secara optimal, sistem memerlukan pemantauan metrik yang efektif. Pendekatan Multi-Criteria Decision Analysis (MCDA) dengan metode weighted scoring digunakan untuk menilai beberapa opsi monitoring. Kriteria yang digunakan beserta bobotnya adalah sebagai berikut:

- **Kemudahan Integrasi** (30%): Tingkat kesederhanaan integrasi sistem monitoring dengan platform yang digunakan.
- **Kelengkapan Fitur Metrik** (35%): Kemampuan sistem untuk memantau berbagai metrik, termasuk ingestion rate, ETA latency, dan API throughput.
- **Kompleksitas Operasional** (20%): Tingkat kesulitan dalam pengelolaan, konfigurasi, dan pemeliharaan monitoring.
- **Biaya** (15%): Efisiensi biaya dalam penggunaan sistem monitoring, termasuk lisensi dan operasional.

Penilaian tiap alternatif dilakukan dengan skala 1–10, di mana nilai lebih tinggi menunjukkan performa yang lebih baik. Hasil penilaian disajikan pada tabel berikut:

Tabel III.11 Penilaian Tiap Alternatif Sistem Monitoring Metrics

Sistem Monitoring	Kemudahan Integrasi	Kelengkapan Fitur Metrik	Kompleksitas Operasional
Prometheus + Grafana	7	9	7
Datadog Metrics	9	8	9
Cloud-Native Monitoring	8	8	8

Berdasarkan bobot kriteria, skor tertimbang tiap alternatif dihitung dengan rumus:

$$\text{Skor Total} = \sum (\text{nilai kriteria} \times \text{bobot})$$

Tabel III.12 Hasil Perhitungan Weighted Score Sistem Monitoring Metrics

Sistem Monitoring	Skor Total	Ranking
Prometheus + Grafana	8.15	1
Cloud-Native Monitoring	7.85	2
Datadog Metrics	7.80	3

Berdasarkan hasil MCDA weighted scoring, **Prometheus + Grafana** merupakan pilihan sistem monitoring yang paling sesuai untuk ITB Ultra-Marathon. Solusi ini

unggul dalam hal kelengkapan fitur metrik dan memiliki kompleksitas operasional yang moderat, sehingga memungkinkan pemantauan ingest telemetri, perhitungan ETA, dan API secara efektif. Cloud-Native Monitoring menjadi alternatif kedua dengan kemudahan integrasi dan biaya yang seimbang, sedangkan Datadog Metrics cocok untuk acara jangka pendek yang membutuhkan setup cepat meskipun biaya relatif lebih tinggi.

Logging (Event & System Logs)

Untuk mendukung debugging, audit, dan analisis event selama lomba ITB Ultra-Marathon, sistem memerlukan mekanisme pencatatan log yang andal. Pendekatan Multi-Criteria Decision Analysis (MCDA) dengan metode weighted scoring digunakan untuk menilai beberapa opsi sistem logging. Kriteria yang digunakan beserta bobotnya adalah sebagai berikut:

- **Kemampuan Analisis dan Pencarian Log** (35%): Kapabilitas sistem dalam melakukan pencarian, agregasi, dan analisis log secara efektif.
- **Kompleksitas Operasional** (25%): Tingkat kesulitan dalam instalasi, konfigurasi, pemeliharaan, dan monitoring sistem logging.
- **Integrasi dengan Infrastruktur** (20%): Kemudahan integrasi dengan VM, Kubernetes, serverless, dan layanan lain yang digunakan.
- **Efisiensi Sumber Daya / Biaya** (20%): Penggunaan sumber daya dan biaya operasional relatif terhadap performa sistem.

Penilaian tiap alternatif dilakukan dengan skala 1–10, di mana nilai lebih tinggi menunjukkan performa yang lebih baik. Hasil penilaian disajikan pada tabel berikut:

Tabel III.13 Penilaian Tiap Alternatif Sistem Logging

Sistem Logging	Analisis & Pencarian	Kompleksitas Operasional	Integrasi Infrastruktur
ELK Stack	9	6	7
Loki + Grafana	7	8	8
Cloud-Native Logging	8	8	9

Berdasarkan bobot kriteria, skor tertimbang tiap alternatif dihitung dengan rumus:

$$\text{Skor Total} = \sum(\text{nilai kriteria} \times \text{bobot})$$

Tabel III.14 Hasil Perhitungan Weighted Score Sistem Logging

Sistem Logging	Skor Total	Ranking
Cloud-Native Logging	8.15	1
Loki + Grafana	7.75	2
ELK Stack	7.65	3

Berdasarkan hasil MCDA weighted scoring, **Cloud-Native Logging** menjadi pilihan sistem logging yang paling sesuai untuk ITB Ultra-Marathon. Solusi ini menawarkan integrasi yang mudah dengan infrastruktur yang digunakan, kompleksitas operasional moderat, serta kemampuan analisis log yang memadai. Loki + Grafana menjadi alternatif kedua yang efisien dan hemat sumber daya, sedangkan ELK Stack cocok untuk kebutuhan analisis log yang kompleks, tetapi memerlukan sumber daya dan konfigurasi lebih tinggi.

Tracing (Distributed Tracing)

Untuk menganalisis alur request antar layanan, terutama pada arsitektur microservices, sistem memerlukan mekanisme distributed tracing yang efektif. Pendekatan Multi-Criteria Decision Analysis (MCDA) dengan metode weighted scoring digunakan untuk menilai beberapa opsi. Kriteria dan bobotnya adalah sebagai berikut:

- **Kemampuan End-to-End Tracing** (35%): Kapabilitas sistem untuk melacak alur request antar layanan secara lengkap.
- **Kemudahan Integrasi** (25%): Tingkat kesederhanaan integrasi dengan layanan dan sistem monitoring/metrik yang digunakan.
- **Kompleksitas Operasional** (20%): Tingkat kesulitan instalasi, konfigurasi, dan pemeliharaan sistem tracing.
- **Biaya / Efisiensi** (20%): Efisiensi biaya relatif terhadap fitur dan kapasitas tracing yang diberikan.

Penilaian tiap alternatif dilakukan dengan skala 1–10, di mana nilai lebih tinggi menunjukkan performa yang lebih baik. Hasil penilaian disajikan pada tabel berikut:

Tabel III.15 Penilaian Tiap Alternatif Sistem Tracing

Sistem Tracing	End-to-End Tracing	Kemudahan Integrasi	Kompleksitas Operasional
Jaeger	9	7	7
Datadog APM	8	9	9
Cloud-Native Tracing	8	8	8

Berdasarkan bobot kriteria, skor tertimbang tiap alternatif dihitung dengan rumus:

$$\text{Skor Total} = \sum(\text{nilai kriteria} \times \text{bobot})$$

Tabel III.16 Hasil Perhitungan Weighted Score Sistem Tracing

Sistem Tracing	Skor Total	Ranking
Jaeger	8.10	1
Cloud-Native Tracing	7.90	2
Datadog APM	7.80	3

Berdasarkan hasil MCDA weighted scoring, **Jaeger** menjadi pilihan utama untuk sistem distributed tracing ITB Ultra-Marathon. Solusi ini unggul dalam kemampuan end-to-end tracing antar layanan dan integrasi dengan ekosistem observabilitas seperti Prometheus, Grafana, dan Loki. Cloud-Native Tracing menjadi alternatif kedua yang baik dengan kemudahan integrasi dan kompleksitas moderat, sedangkan Datadog APM cocok untuk setup cepat dan event jangka pendek meskipun biaya relatif lebih tinggi.

Model Perhitungan ETA Marathon

Untuk menentukan model perhitungan estimasi waktu tiba (ETA) pelari marathon yang paling sesuai, digunakan pendekatan Multi-Criteria Decision Analysis (MCDA) dengan metode weighted scoring. Kriteria yang digunakan beserta bobotnya adalah sebagai berikut:

- **Akurasi Prediksi** (40%): Kemampuan model memprediksi ETA secara realistik, termasuk mempertimbangkan variasi pace dan kondisi rute.
- **Kompleksitas Implementasi** (25%): Tingkat kesulitan pengembangan, integrasi, dan kalibrasi model.
- **Kebutuhan Data** (20%): Ketersediaan data yang dibutuhkan untuk menjalankan model secara efektif.
- **Kecepatan Perhitungan** (15%): Waktu yang dibutuhkan model untuk menghitung ETA secara real-time.

Penilaian tiap alternatif dilakukan dengan skala 1–10, di mana nilai lebih tinggi menunjukkan performa yang lebih baik. Hasil penilaian disajikan pada tabel berikut:

Tabel III.17 Penilaian Tiap Alternatif Model Perhitungan ETA

Model ETA	Akurasi Prediksi	Kompleksitas Implementasi	Kebutuhan Data
Ekstrapolasi Pace Rata-Rata	6	9	9
Model Pace Berbasis Segmen	8	7	7
Model Berbasis Data Historis	9	5	6

Berdasarkan bobot kriteria, skor tertimbang tiap alternatif dihitung dengan rumus:

$$\text{Skor Total} = \sum(\text{nilai kriteria} \times \text{bobot})$$

Tabel III.18 Hasil Perhitungan Weighted Score Model ETA

Model ETA	Skor Total	Ranking
Model Pace Berbasis Segmen	7.70	1
Model Berbasis Data Historis	7.05	2
Ekstrapolasi Pace Rata-Rata	7.05	3

Berdasarkan hasil MCDA weighted scoring, **Model Pace Berbasis Segmen** menjadi pilihan utama untuk perhitungan ETA pelari marathon. Model ini menawarkan akurasi yang baik dengan kompleksitas implementasi moderat, serta dapat memperhitungkan variasi pace dan kondisi rute per segmen. Model Berbasis Data Historis memberikan prediksi yang sangat akurat namun memerlukan data tambahan dan proses kalibrasi, sedangkan Ekstrapolasi Pace Rata-Rata sederhana dan cepat, tetapi kurang mempertimbangkan kelelahan atau variasi medan.

Integrasi Elevasi dalam Perhitungan ETA

Untuk memperhitungkan faktor elevasi yang mempengaruhi performa pelari, digunakan pendekatan Multi-Criteria Decision Analysis (MCDA) dengan metode weighted scoring. Kriteria yang digunakan beserta bobotnya adalah sebagai berikut:

- **Akurasi Perhitungan ETA (40%)**: Kemampuan pendekatan untuk menghasilkan estimasi waktu tiba yang realistik dengan mempertimbangkan tanjakan dan turunan.
- **Kompleksitas Implementasi (30%)**: Tingkat kesulitan integrasi pendekatan ke dalam model ETA yang digunakan.
- **Kebutuhan Data Elevasi (20%)**: Ketersediaan data elevasi yang diperlukan dan kemudahan penggunaannya.

- **Kecepatan Perhitungan** (10%): Waktu yang dibutuhkan pendekatan untuk menghitung ETA secara real-time.

Penilaian tiap alternatif dilakukan dengan skala 1–10, di mana nilai lebih tinggi menunjukkan performa yang lebih baik. Hasil penilaian disajikan pada tabel berikut:

Tabel III.19 Penilaian Tiap Alternatif Pendekatan Integrasi Elevasi

Pendekatan	Akurasi ETA	Kompleksitas Implementasi	Kebutuhan Data
Grade-Adjusted Pace (GAP)	7	8	8
Elevation-Corrected Segment ETA	9	6	7

Berdasarkan bobot kriteria, skor tertimbang tiap alternatif dihitung dengan rumus:

$$\text{Skor Total} = \sum(\text{nilai kriteria} \times \text{bobot})$$

Tabel III.20 Hasil Perhitungan Weighted Score Pendekatan Integrasi Elevasi

Pendekatan	Skor Total	Ranking
Elevation-Corrected Segment ETA	7.95	1
Grade-Adjusted Pace (GAP)	7.70	2

Berdasarkan hasil MCDA weighted scoring, **Elevation-Corrected Segment ETA** menjadi pendekatan yang lebih unggul untuk integrasi faktor elevasi dalam perhitungan ETA pelari marathon. Pendekatan ini memberikan akurasi prediksi yang lebih tinggi dengan mempertimbangkan penalti atau bonus waktu per segmen. Grade-Adjusted Pace (GAP) tetap layak dipertimbangkan karena kompleksitas implementasinya lebih rendah dan perhitungannya lebih cepat, namun akurasi prediksi sedikit lebih rendah dibandingkan pendekatan per segmen.

BAB IV

DESAIN KONSEP SOLUSI

Ilustrasikan desain konsep solusi dalam bentuk model konseptual dan penjelasan secara ringkas, beserta perbedaannya dengan sistem saat ini. Ilustrasi harus dapat dibandingkan (*before and after*). Karena masih berupa proposal, bab ini hanya berisi gambar desain konsep solusi tersebut dan penjelasan perbandingannya dengan gambar sistem yang ada saat ini (yang tergambar di awal Bab III).

BAB V

RENCANA SELANJUTNYA

Jelaskan secara detail langkah-langkah rencana selanjutnya, hal-hal yang diperlukan atau akan disiapkan, dan risiko dan mitigasinya, yang meliputi:

1. Rencana implementasi, termasuk alat dan bahan yang diperlukan, lingkungan, konfigurasi, biaya, dan sebagainya.
2. Desain pengujian dan evaluasi, misalnya metode verifikasi dan validasi.
3. Analisis risiko dan mitigasi, misalnya tindakan selanjutnya jika ada yang tidak berjalan sesuai rencana.

DAFTAR PUSTAKA

- Buku Praktis Bahasa Indonesia 1/Kata - Wikisumber bahasa Indonesia.* 2024. Diakses pada October 22, 2025. https://id.wikisource.org/wiki/Buku_Praktis_Bahasa_Indonesia_1/Kata.
- Hochreiter, Dominik. 2024. “Athlete Tracking at a Marathon Event with LoRa: A Performance Evaluation with Mobile Gateways”. *Multidisciplinary Digital Publishing Institute (MDPI)*, <https://doi.org/10.1016/j.jom.2023.01.005>.
- M. Naufal Hafizh, S.S. September 2025. “wondr ITB Ultra Marathon 2025 Sukses Digelar, Alumni sampai Guru SD Turut Menyumbang untuk Dana Lestari”. Disunting oleh S.S. M. Naufal Hafizh. Accessed November 30, 2025, *ITB Official Website* (). <https://itb.ac.id/berita/wondr-itb-ultra-marathon-2025-sukses-digelar-alumni-hingga-guru-sd-turut-menyumbang-untuk-dana-lestari/62873>.
- Sjöbeck, Erik. August 2022. “How Does an RFID Chip Timing System Work?” Accessed Desember 1, 2025, *RaceID* (). <https://raceid.com/organizer/timing/how-does-an-rfid-chip-timing-system-work/>.