

Laporan Proyek Pengembangan Sistem Pelacakan dan Pemantauan

Berbasis Web untuk Tahanan Rumah dan Tahanan Kota

Disusun Untuk Memenuhi Tugas Mata Kuliah

Sistem dan Jaringan Komputer

Dosen Pengampu: Ari Sujarwo, S.Kom., MIT (Hons.)



Disusun oleh :

Niko Yoga Pranata (23523267)

Ardelia Putri Eka Rahmawati (23523037)

Mohamad Rafi Hendryansah (23523064)

Abdul Qadir Abdurrahman Mahrus (23523178)

PROGRAM STUDI INFORMATIKA

UNIVERSITAS ISLAM INDONESIA

2024/2025

Daftar Isi

BAB I Pendahuluan.....	3
A. Deskripsi Aplikasi Web.....	3
B. Tujuan.....	3
BAB II Metodologi.....	4
A. Analisis Kebutuhan.....	4
1.) Analisis Kebutuhan Pengguna:.....	4
2.) Definisi Fungsionalitas Utama:.....	5
3.) Perangkat Lunak dan Protokol Jaringan:.....	5
B. Desain Sistem.....	5
1.) Arsitektur Sistem:.....	5
2.) Prototipe Antarmuka Pengguna:.....	5
3.) Entity Relationship Diagram (ERD):.....	5
C. Pengembangan Sistem.....	5
1.) Pengembangan Frontend:.....	6
2.) Pengembangan Backend:.....	6
3.) Pengembangan Database:.....	6
D. Integrasi dan Kontainerisasi.....	6
1.) Version Control dengan GitHub:.....	6
2.) Kontainerisasi dengan Docker:.....	6
E. Pengujian Sistem.....	6
1.) Pengujian Fungsionalitas:.....	7
2.) Pengujian Komunikasi Data:.....	7
3.) Pengujian Responsivitas:.....	7
4.) Pengujian Kesehatan Database:.....	7
F. Deployment.....	7

1.) Hosting Aplikasi:.....	7
2.) Pengujian Produksi:.....	7
BAB III Implementasi dan Hasil.....	8
A. Implementasi.....	8
1.) Pengembangan Frontend.....	8
2.) Pengembangan Backend.....	8
3.) Pengembangan Database.....	9
4.) Integrasi dengan Docker.....	9
5.) Pengujian Sistem.....	15
B. Hasil.....	16
1.) Keberhasilan Pelacakan Real-Time.....	16
2.) Peringatan Geo-Fencing.....	16
3.) Antarmuka Pengguna.....	16
4.) Pengelolaan Backend dan Database.....	16
5.) Stabilitas dan Skalabilitas.....	16
6.) Hasil Uji Sistem.....	16
BAB IV Penutup.....	17
A. Kesimpulan.....	17

BAB I Pendahuluan

A. Deskripsi Aplikasi Web

WETRACK adalah platform inovatif berbasis web yang dirancang secara komprehensif untuk memfasilitasi pemantauan tahanan rumah dan tahanan kota. Platform ini memungkinkan instansi berwenang, seperti Kemenkumham, Polri, Lapas, dan Bapas, untuk mengelola serta melacak individu yang berada di bawah pengawasan mereka.

B. Tujuan

Tujuan dari Platform WETRACK adalah sebagai berikut:

1. Efisiensi Pengelolaan Data:

Berfokus pada pengelolaan data narapidana yang lebih sistematis dan terpusat. WETRACK, memberikan akses secara digital kepada berwenang terhadap data narapidana, mengurangi ketergantungan pada dokumentasi secara manual, dan mempercepat proses pengambilan informasi.

2. Pemantauan Real-Time dan Geo-Fencing:

Memungkinkan pemantauan lokasi narapidana secara real-time menggunakan teknologi GPS. Selain itu, fitur geo-fencing mengirimkan pemberitahuan ketika narapidana keluar dari area yang telah ditentukan, dan memastikan pemantauan yang lebih ketat

3. Peningkatan Transparansi dan Keamanan:

Platform ini meningkatkan transparansi dalam proses pemantauan dengan menyediakan laporan otomatis yang dapat diakses oleh pihak terkait. Hal ini mendukung akuntabilitas dan meningkatkan kepercayaan publik terhadap pengelolaan tahanan.

4. Dukungan Reintegrasi:

WETRACK mendukung proses reintegrasi sosial tahanan dengan memberikan data yang relevan kepada pihak yang bertanggung jawab, seperti Bapas. Data ini membantu dalam evaluasi perilaku tahanan dan mempersiapkan mereka untuk kembali ke masyarakat dengan lebih baik.

BAB II Metodologi

A. Analisis Kebutuhan

Tahap ini bertujuan untuk mengidentifikasi kebutuhan sistem yang diperlukan dalam membangun aplikasi berbasis web. Aktivitas yang dilakukan meliputi:

1.) Analisis Kebutuhan Pengguna:

Melakukan identifikasi terhadap kebutuhan dari berbagai jenis pengguna, yaitu admin (Kemenkumham), Petugas Berwenang (Lapas, Bapas, Polri), dan mantan narapidana. Setiap pihak yang terlibat memiliki kebutuhan spesifik yang harus dipenuhi oleh sistem.

2.) Definisi Fungsionalitas Utama:

Menentukan fitur-fitur utama yang harus ada dalam aplikasi, seperti pelacakan real-time, riwayat data, dan fitur laporan. Fungsionalitas ini dirancang untuk memastikan aplikasi dapat memenuhi kebutuhan operasional dan manajemen data secara efektif.

3.) Perangkat Lunak dan Protokol Jaringan:

Menentukan spesifikasi teknis yang dibutuhkan yaitu perangkat lunak dan protokol jaringan yang akan digunakan untuk memastikan kinerja sistem yang optimal.

B. Desain Sistem

Tahap ini fokus pada perancangan arsitektur dan komponen sistem, meliputi:

1.) Arsitektur Sistem:

Mengadopsi model client-server dengan komponen frontend dan backend. Frontend dikembangkan menggunakan HTML, CSS, JavaScript, dan Bootstrap untuk memastikan tampilan yang responsif dan interaktif. Backend dibangun dengan PHP untuk menangani logika bisnis dan integrasi dengan database MySQL yang berfungsi menyimpan data pengguna, riwayat lokasi, dan laporan.

2.) Prototipe Antarmuka Pengguna:

Mendesain wireframe antarmuka menggunakan Bootstrap untuk memastikan responsivitas dan kemudahan penggunaan. Prototipe ini berfungsi sebagai acuan dalam pengembangan antarmuka yang user-friendly.

3.) Entity Relationship Diagram (ERD):

Membuat diagram yang menggambarkan hubungan antar entitas dalam sistem, termasuk atribut dan relasi antar entitas tersebut. ERD membantu dalam merancang struktur database yang terorganisir dan memastikan hubungan data yang efisien.

C. Pengembangan Sistem

Tahap pengembangan mencakup implementasi komponen sistem sesuai dengan desain yang telah dibuat:

1.) Pengembangan Frontend:

Membangun interface pengguna dengan HTML, CSS, dan JavaScript, serta memanfaatkan Bootstrap untuk tata letak yang responsif. Tujuannya adalah menyediakan interface yang intuitif dan mudah digunakan oleh pengguna.

2.) Pengembangan Backend:

Mengimplementasikan logika bisnis menggunakan PHP, termasuk pembuatan API sederhana untuk komunikasi antara frontend dan backend. Backend bertanggung jawab untuk memproses permintaan dari frontend dan berinteraksi dengan database.

3.) Pengembangan Database:

Mendesain dan mengimplementasikan skema database MySQL untuk menyimpan data aplikasi, seperti tabel pengguna, lokasi, dan laporan. Struktur database dirancang untuk memastikan integritas dan konsistensi data.

D. Integrasi dan Kontainerisasi

Tahap ini memastikan bahwa semua komponen sistem dapat berfungsi secara harmonis:

1.) Version Control dengan GitHub:

Pengendalian versi dilakukan menggunakan GitHub untuk memastikan kolaborasi dan pengelolaan kode yang efektif. Setiap perubahan kode dilacak melalui commit, sementara pull request digunakan untuk mengelola kontribusi tim. Hal ini memastikan integritas kode tetap terjaga dan memungkinkan proses review kode berjalan dengan lancar.

2.) Kontainerisasi dengan Docker:

Aplikasi web dikemas dalam kontainer menggunakan Docker untuk memastikan konsistensi lingkungan pengembangan dan produksi. Dockerfile dibuat untuk mengemas aplikasi web ke dalam kontainer. Docker Compose digunakan untuk mengelola layanan yang melibatkan server PHP dan database MySQL, sehingga memudahkan proses deployment serta memungkinkan aplikasi untuk diskalakan dengan mudah.

E. Pengujian Sistem

Tahap pengujian dilakukan untuk memastikan kualitas dan kinerja aplikasi:

1.) Pengujian Fungsionalitas:

Memastikan semua fitur bekerja sesuai dengan kebutuhan, seperti login, pelacakan, dan laporan. Pengujian ini dilakukan dengan skenario yang mencakup berbagai kasus penggunaan untuk memastikan setiap fungsi berjalan dengan benar.

2.) Pengujian Komunikasi Data:

Menggunakan Docker untuk mengelola lingkungan pengujian yang terisolasi, memastikan komunikasi data antara frontend, backend, dan database berjalan dengan benar. Pengujian dilakukan untuk memastikan data dikirim dan diterima secara efisien tanpa adanya kebocoran atau kehilangan informasi selama proses komunikasi.

3.) Pengujian Responsivitas:

Memastikan aplikasi tampil baik di berbagai perangkat dan resolusi layar, memberikan pengalaman pengguna yang konsisten di desktop maupun perangkat mobile.

4.) Pengujian Kesehatan Database:

Mengecek kesehatan database melibatkan pemantauan kinerja, konsistensi data, dan integritas sistem secara keseluruhan. Langkah utama termasuk memeriksa penggunaan sumber daya seperti CPU, memori, dan *disk* untuk memastikan performa optimal

F. Deployment

Tahap ini mencakup penyebaran aplikasi ke lingkungan produksi:

1.) Hosting Aplikasi:

Menggunakan layanan hosting atau server lokal dengan Docker untuk menjalankan aplikasi, memastikan aplikasi dapat diakses oleh pengguna sesuai dengan kebutuhan.

2.) Pengujian Produksi:

Memastikan aplikasi dapat diakses secara online dan semua layanan berjalan dengan baik di lingkungan produksi, termasuk pengujian performa dan keamanan tambahan jika diperlukan.

BAB III Implementasi dan Hasil

A. Implementasi

1.) Pengembangan Frontend

- a. Antarmuka dikembangkan menggunakan HTML, CSS, JavaScript, dan dipadukan dengan Bootstrap untuk memastikan desain yang responsif.
- b. Diimplementasikan pada semua fitur antarmuka seperti halaman login, dashboard pemantauan, laporan, dsb dengan tampilan antarmuka yang user-friendly.

2.) Pengembangan Backend

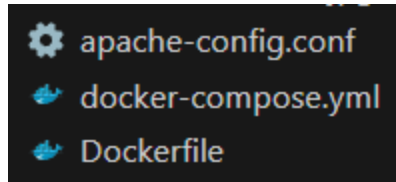
- a. Backend menggunakan PHP dengan API sederhana yang menghubungkan frontend ke database.
- b. API menangani, pengambilan data tahanan untuk ditampilkan di dashboard, pengolahan data lokasi dari GPS yang dikirim secara berkala ke server, dan pengiriman notifikasi melalui geo-fencing ketika tahanan melewati batas area wilayah yang telah ditentukan.

3.) Pengembangan Database

- a. Database dirancang menggunakan MySQL dengan struktur yang mencakup tabel pengguna yaitu untuk menyimpan data admin dan petugas maupun instansi yang terkait, tabel lokasi menyimpan data lokasi real-time tahanan, tabel laporan yaitu menyimpan catatan aktivitas tahanan untuk keperluan evaluasi, dan tabel-tabel lainnya untuk penyimpanan informasi tambahan sesuai kebutuhan sistem.
- b. Database diimplementasikan dengan menggunakan relasi antar tabel yang terstruktur dan indeks ditambahkan pada kolom utama untuk meningkatkan kinerja query.

4.) Integrasi dengan Docker

Aplikasi ini telah dikemas dalam kontainer menggunakan Docker untuk memastikan konsistensi antara lingkungan pengembangan dan produksi. Berikut adalah langkah-langkah yang dilakukan:



Gambar 1. File untuk Membangun Docker Image

a. Dockerfile:

- **Frontend:** File Dockerfile dibuat untuk menjalankan layanan frontend dengan server Apache yang mendukung PHP. Dalam Dockerfile ini, kami juga menambahkan konfigurasi khusus untuk Apache menggunakan file `apache-config.conf` untuk memastikan pengaturan server sesuai dengan kebutuhan aplikasi. File konfigurasi ini mengatur direktori root, pengaturan `mod_rewrite`, dan pengaturan lainnya yang diperlukan untuk mendukung aplikasi berbasis PHP.

File `apache-config.conf` ini berisi konfigurasi untuk mengoptimalkan server Apache dalam menangani permintaan aplikasi web Anda, seperti:

- Menentukan `DocumentRoot` ke direktori aplikasi.
- Mengaktifkan `mod_rewrite` untuk mendukung URL yang ramah SEO.
- Menyesuaikan pengaturan lainnya sesuai kebutuhan aplikasi.
- **Backend:** File Dockerfile mengatur konfigurasi PHP yang diperlukan untuk aplikasi backend.
- **Database:** Menggunakan image MySQL resmi yang dikonfigurasi untuk menyimpan data aplikasi.

b. Docker Compose:

- Docker Compose digunakan untuk mengatur beberapa layanan aplikasi, termasuk server PHP (web) dan MySQL (database). Semua layanan diatur untuk saling terhubung melalui jaringan virtual Docker. Pengaturan ini memungkinkan komunikasi antara layanan web dan database secara lancar.

Struktur Docker Compose

- **Web Server:** Container berbasis Apache dengan PHP, dengan file konfigurasi Apache (apache-config.conf) diatur untuk memastikan server berjalan dengan pengaturan yang tepat.
- **Database Server:** Container berbasis MySQL yang dihubungkan ke layanan web melalui jaringan virtual Docker.

c. Struktur Docker Compose:

1. Web Server:

- Base Image: php:8.0-apache untuk mendukung aplikasi PHP dengan Apache.
- Menggunakan port 80:80 untuk mengakses HTTP.
- Menyediakan layanan web untuk aplikasi WETRACK.
- Direktori kode aplikasi di-host di volume container.

2. Database Server:

- Base Image: mysql:8.0 untuk layanan MySQL.
- Menggunakan port: 3306 untuk koneksi ke database.
- Menyimpan data aplikasi di database wetrack.
- Inisialisasi: File wetrack.sql untuk membuat struktur database awal
- Database diakses oleh aplikasi melalui nama layanan db dalam jaringan Docker.

3. Virtual Network:

- Jaringan khusus bernama wetrack-network dibuat untuk memastikan komunikasi antar-container yang aman dan terisolasi.

d. Proses Building Image:

Proses membangun image untuk aplikasi dilakukan menggunakan perintah **Docker Compose** yang memastikan bahwa container dibangun tanpa menggunakan cache dari image sebelumnya, sehingga memastikan bahwa semua perubahan terbaru pada Dockerfile dan konfigurasi lainnya diterapkan dengan benar.




1. **Build Image:** Perintah ini digunakan untuk membangun image Docker tanpa cache, memastikan bahwa semua file dan konfigurasi terbaru diterapkan ke dalam image. Kami menggunakan perintah `docker compose build --no-cache`, Perintah ini akan membaca file `docker-compose.yml` dan `Dockerfile` yang terkait untuk membangun image terbaru untuk setiap layanan (web, database).

Gambar 2. Proses membangun Docker Image


```
time="2024-12-22T14:59:23+07:00" level=warning msg="C:\xampp\htdocs\wetrack\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
```

```
[+] Running 4/4
✔ Network wetrack_wetrack-network Created 0.1s
✔ Volume "wetrack_mysql_data" Created 0.0s
✔ Container wetrack-db-1 Healthy 32.5s
✔ Container wetrack-web-1 Started 32.3s
```



Gambar 3. Proses Menjalankan Kontainer

	Name	Container ID	Image	Port(s)
	wetrack	-	-	-
	web-1	48391d5fd4b1	wetrack-we	53:53 (UDP) 80:8...
	db-1	3635b092b745	mysql:8.0	3306:3306 ↗

Gambar 4. Kontainer wetrack




wetrack-web-1



 48391d5fd4b1  [wetrack-web:latest](#) (was wetrack-web:<none>)

53:53 (UDP) [80:80](#) ↗

Gambar 5. Kontainer wetrack-web-1





wetrack-db-1

 3635b092b745  [mysql:8.0](#)

[3306:3306](#) ↗

Gambar 6. Kontainer wetrack-db-1





wetrack-web-1

 48391d5fd4b1  [wetrack-web:latest](#) (was wetrack-web:<none>)

53:53 (UDP) [80:80](#) ↗

STATUS

Running (4 hours ago)

Logs

Inspect

Bind mounts

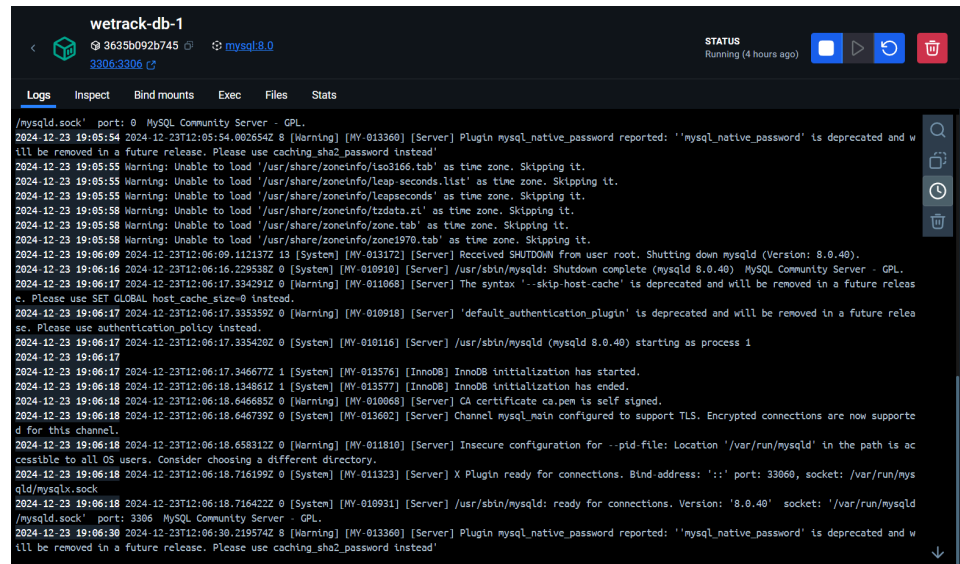
Exec

Files

Stats

```
2024-12-23 19:06:13 172.19.0.1 - - [23/Dec/2024:12:06:13 +0000] "GET /public/landing-page/images/man-with-tablet-20-1-201-20-1-p-1080.png HTTP/1.1" 200 616262 "http://mywebsite.local/public/landing-page/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36"
2024-12-23 19:06:13 172.19.0.1 - - [23/Dec/2024:12:06:13 +0000] "GET /public/landing-page/images/logo-white-p-500.png HTTP/1.1" 200 14681 "http://mywebsite.local/public/landing-page/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36"
2024-12-23 19:06:13 172.19.0.1 - - [23/Dec/2024:12:06:13 +0000] "GET /public/landing-page/images/kemenkumham.png HTTP/1.1" 200 23370 "http://mywebsite.local/public/landing-page/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36"
2024-12-23 19:06:13 172.19.0.1 - - [23/Dec/2024:12:06:13 +0000] "GET /public/landing-page/images/lapas-p-500.png HTTP/1.1" 200 83884 "http://mywebsite.local/public/landing-page/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36"
2024-12-23 19:06:13 172.19.0.1 - - [23/Dec/2024:12:06:13 +0000] "GET /public/landing-page/images/polri.png HTTP/1.1" 200 351805 "http://mywebsite.local/public/landing-page/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36"
2024-12-23 19:16:34 172.19.0.1 - - [23/Dec/2024:12:16:34 +0000] "GET / HTTP/1.1" 302 576 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36"
2024-12-23 19:16:34 172.19.0.1 - - [23/Dec/2024:12:16:34 +0000] "GET /public/landing-page/ HTTP/1.1" 200 3815 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36"
2024-12-23 19:06:02 AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.19.0.3. Set the 'ServerName' directive globally to suppress this message
2024-12-23 19:06:02 AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.19.0.3. Set the 'ServerName' directive globally to suppress this message
2024-12-23 19:06:02 [Mon Dec 23 12:06:02.358662 2024] [mpm_prefork:notice] [pid 19] AH00163: Apache/2.4.56 (Debian) PHP/8.0.30 configured -- resuming normal operations
2024-12-23 19:06:02 [Mon Dec 23 12:06:02.358742 2024] [core:notice] [pid 19] AH00994: Command line: 'apache2 -D FOREGROUND'
2024-12-23 23:26:45 172.19.0.1 - - [23/Dec/2024:16:26:44 +0000] "GET /wetrack/public/landing-page/index.php HTTP/1.1" 200 3816 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36"
2024-12-23 23:26:45 172.19.0.1 - - [23/Dec/2024:16:26:45 +0000] "GET /wetrack/public/landing-page/css/webflow-style.css HTTP/1.1" 304 250 "http://localhost/wetrack/public/landing-page/index.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36"
2024-12-23 23:26:45 172.19.0.1 - - [23/Dec/2024:16:26:45 +0000] "GET /wetrack/public/landing-page/images/wide-2520feature-p-500.png HTTP/1.1" 200 31972 "http://localhost/wetrack/public/landing-page/index.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36"
2024-12-23 23:26:45 172.19.0.1 - - [23/Dec/2024:16:26:45 +0000] "GET /wetrack/public/landing-page/images/man-with-tablet-20-1-201-20-1-p-500.png HTTP/1.1" 200 135637 "http://localhost/wetrack/public/landing-page/index.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36"
```

Gambar 7. Log Kontainer wetrack-web-1



```
wetrack-db-1
3639b092b745 mysql:8.0
3306:3306

Logs Inspect Bind mounts Exec Files Stats

/mysqld.sock' port: 0 MySQL Community Server - GPL.
2024-12-23 19:05:54 2024-12-23T12:05:54.002654Z 0 [Warning] [MY-013360] [Server] Plugin mysql_native_password reported: 'mysql_native_password' is deprecated and will be removed in a future release. Please use caching_sha2_password instead'
2024-12-23 19:05:55 Warning: Unable to load '/usr/share/zoneinfo/iso3166.tab' as time zone. Skipping it.
2024-12-23 19:05:55 Warning: Unable to load '/usr/share/zoneinfo/leap-seconds.list' as time zone. Skipping it.
2024-12-23 19:05:55 Warning: Unable to load '/usr/share/zoneinfo/leapseconds' as time zone. Skipping it.
2024-12-23 19:05:58 Warning: Unable to load '/usr/share/zoneinfo/tzdata.zi' as time zone. Skipping it.
2024-12-23 19:05:58 Warning: Unable to load '/usr/share/zoneinfo/zone.tab' as time zone. Skipping it.
2024-12-23 19:05:58 Warning: Unable to load '/usr/share/zoneinfo/zone1970.tab' as time zone. Skipping it.
2024-12-23 19:06:09 2024-12-23T12:06:09.112137Z 13 [System] [MY-013172] [Server] Received SHUTDOWN from user root. Shutting down mysqld (Version: 8.0.40).
2024-12-23 19:06:16 2024-12-23T12:06:16.229538Z 0 [System] [MY-010910] [Server] /usr/sbin/mysqld: Shutdown complete (mysqld 8.0.40) MySQL Community Server - GPL.
2024-12-23 19:06:17 2024-12-23T12:06:17.334291Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and will be removed in a future release. Please use SET GLOBAL host_cache_size=0 instead.
2024-12-23 19:06:17 2024-12-23T12:06:17.335359Z 0 [Warning] [MY-010918] [Server] 'default_authentication_plugin' is deprecated and will be removed in a future release. Please use authentication_policy instead.
2024-12-23 19:06:17 2024-12-23T12:06:17.335420Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.40) starting as process 1
2024-12-23 19:06:17
2024-12-23 19:06:17 2024-12-23T12:06:17.346677Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2024-12-23 19:06:18 2024-12-23T12:06:18.134861Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2024-12-23 19:06:18 2024-12-23T12:06:18.646685Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2024-12-23 19:06:18 2024-12-23T12:06:18.646739Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections are now supported for this channel.
2024-12-23 19:06:18 2024-12-23T12:06:18.658312Z 0 [Warning] [MY-011010] [Server] Insecure configuration for --pid-file: Location '/var/run/mysqld' in the path is accessible to all OS users. Consider choosing a different directory.
2024-12-23 19:06:18 2024-12-23T12:06:18.716199Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: '::' port: 33060, socket: /var/run/mysqld/mysqld.sock
2024-12-23 19:06:18 2024-12-23T12:06:18.716422Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.0.40' socket: /var/run/mysqld/mysqld.sock' port: 3306 MySQL Community Server - GPL.
2024-12-23 19:06:30 2024-12-23T12:06:30.219574Z 0 [Warning] [MY-013360] [Server] Plugin mysql_native_password reported: 'mysql_native_password' is deprecated and will be removed in a future release. Please use caching_sha2_password instead'
```

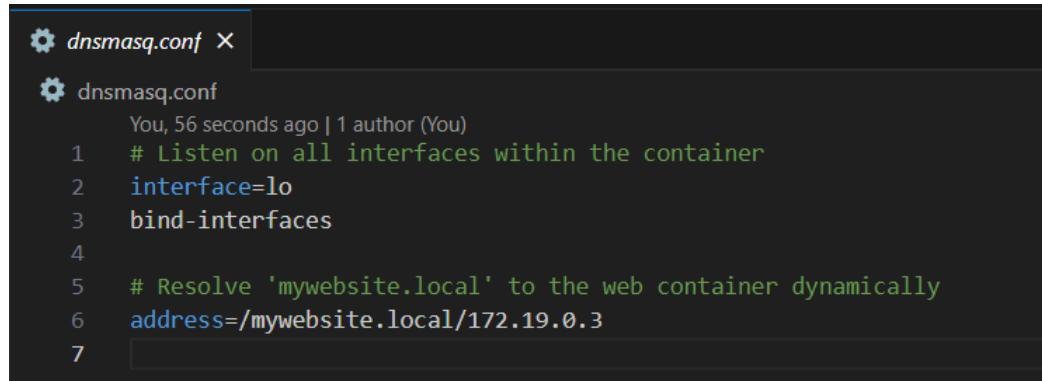
Gambar 8. Log Kontainer wetrack-db-1

e. DNS Resolver:

Aplikasi web dapat diakses melalui browser dengan menggunakan URL `http://mywebsite.local` pada sistem operasi host laptop. URL ini telah dikonfigurasi dengan DNS lokal menggunakan tool `dnsmasq`. Langkah-langkah berikut dilakukan:

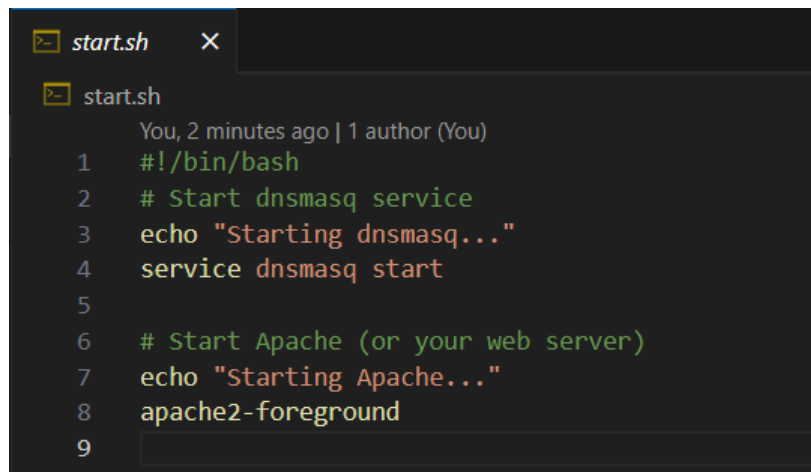
1. Konfigurasi DNS:

- File Konfigurasi: `dnsmasq.conf` untuk memetakan domain `mywebsite.local` ke IP kontainer webserver.
- DNS menggunakan `port 53/udp` yang agar dapat digunakan oleh client lokal.
- Entrypoint: `start.sh` untuk memulai layanan DNS sebelum Apache berjalan.



```
dnsmasq.conf
You, 56 seconds ago | 1 author (You)
1 # Listen on all interfaces within the container
2 interface=lo
3 bind-interfaces
4
5 # Resolve 'mywebsite.local' to the web container dynamically
6 address=/mywebsite.local/172.19.0.3
7
```

Gambar 9. File dnsmasq.conf



```
start.sh
You, 2 minutes ago | 1 author (You)
1 #!/bin/bash
2 # Start dnsmasq service
3 echo "Starting dnsmasq..."
4 service dnsmasq start
5
6 # Start Apache (or your web server)
7 echo "Starting Apache..."
8 apache2-foreground
9
```

Gambar 10. File start.sh

2. Host OS:

- Memetakan **mywebsite.local** ke **IP lokal** (127.0.0.1)
- Untuk mengakses aplikasi di kontainer Docker yang berjalan di server lokal.

```
127.0.0.1 mywebsite.local
```

Gambar 11. Penulisan 127.0.0.1 mywebsite.local di dalam file hosts

f. Akses Dari Client:



Gambar 12. Pengaksesan Web Melalui Domain Lokal

Langkah-langkah berikut telah dilakukan untuk memastikan sistem berjalan dengan baik:

1. Koneksi Jaringan Antar-Container:

- Pengujian koneksi antara layanan web dan database menggunakan perintah ping dan koneksi PHP ke MySQL.

2. Akses Browser:

- Aplikasi diakses melalui browser untuk memverifikasi bahwa aplikasi dapat diakses melalui `http://mywebsite.local`.

3. Fungsi Aplikasi:

- Aplikasi diuji untuk memastikan semua fitur berjalan sesuai harapan.

5.) Pengujian Sistem

- a. Pengujian Fungsionalitas, setiap fitur diuji menggunakan skenario yang mencakup berbagai penggunaan, seperti login, pelacakan, dan pelaporan.

- b. Pengujian Geo-Fencing, memastikan lokasi terlacak dan notifikasi terkirim saat tahanan melanggar area yang telah ditentukan.
- c. Pengujian Responsivitas, menggunakan perangkat dengan berbagai resolusi untuk memastikan aplikasi tampil optimal di desktop maupun mobile.

B. Hasil

1.) Keberhasilan Pelacakan Real-Time

Data lokasi tahanan dapat ditampilkan secara real-time dengan akurasi yang baik dan mendukung pemantauan efektif.

2.) Peringatan Geo-Fencing

Sistem mampu mengirimkan notifikasi kepada admin saat tahanan keluar dari zona yang ditentukan untuk meningkatkan keamanan (ini tambahkan atau gimana soalnya kan alert itu dari 2 pov sistem sm admin kan?)

3.) Antarmuka Pengguna

Halaman login dan dashboard telah berfungsi sesuai dengan desain awal yang interaktif dan mudah digunakan dan menerima umpan balik positif dari tahap pengembangan awal oleh user saat simulasi. Fitur pelacakan real-time berhasil menampilkan lokasi tahanan dengan akurasi tinggi (isi sendiri deh persen).

4.) Pengelolaan Backend dan Database

Data pengguna, lokasi, dan laporan tersimpan dengan baik di database dengan akses yang mudah melalui API serta integrasi backend dengan frontend berjalan tanpa hambatan.

5.) Stabilitas dan Skalabilitas

Kontainerisasi dengan Docker memastikan aplikasi dapat dengan mudah di maintenance dan skalabilitasnya dapat ditingkatkan di masa depan

6.) Hasil Uji Sistem

Aplikasi dapat digunakan di berbagai perangkat desktop maupun mobile dengan konsisten dan semua fitur utama berfungsi dengan baik sesuai kebutuhan.

BAB IV Penutup

A. Kesimpulan

Proyek aplikasi web WETRACK dengan integrasi Docker telah berhasil diselesaikan dengan sejumlah pencapaian utama. Sistem pelacakan dan pemantauan tahanan rumah serta tahanan kota berbasis web berhasil dirancang dan diimplementasikan, memberikan solusi inovatif untuk pengelolaan data dan pemantauan real-time melalui teknologi GPS dan geo-fencing. Sistem ini mendukung peningkatan transparansi, keamanan, dan reintegrasi tahanan ke masyarakat. Dalam hal integrasi teknologi, aplikasi berhasil dikemas dalam kontainer Docker yang mencakup layanan web dan database, sementara DNS lokal diterapkan menggunakan dnsmasq untuk memudahkan akses aplikasi dengan domain kustom. Sistem ini telah diuji dan berfungsi sesuai dengan spesifikasi proyek.

Fitur utama, seperti pelacakan real-time dan geo-fencing, berfungsi optimal, sementara pelaporan otomatis terintegrasi dengan baik. DNS resolver juga berhasil diimplementasikan untuk meningkatkan kemudahan akses aplikasi. Ke depan, proyek ini akan fokus pada pengembangan berkelanjutan untuk meningkatkan performa, skalabilitas, dan kompatibilitas sistem. Evaluasi lebih lanjut akan dilakukan untuk mengoptimalkan sistem, dengan beberapa keterbatasan teknis dan sumber daya yang telah diidentifikasi untuk perbaikan di masa depan. Melalui proyek ini, WETRACK telah berhasil mengintegrasikan teknologi modern, seperti Docker dan DNS resolver, dengan kebutuhan praktis pemantauan tahanan, serta memberikan landasan yang kuat untuk pengembangan sistem di masa depan..