

PERANCANGAN DAN PENGEMBANGAN APLIKASI DAILY CHECK & FINDING REPORT BERBASIS WEB



member of



Disusun Oleh:

Nama : Doni Darmawan
NIK : 55000154
Departemen : Electric EI2 (PM37)

**DEPARTEMEN ELECTRIC INSTRUMENT 2
PT FAJAR SURYA WISESA Tbk**

Daftar isi

Daftar isi	2
Abstrak	3
BAB I - Pendahuluan	3
1.1 Latar Belakang Masalah	3
1.2 Rumusan Masalah	4
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	4
BAB II - Landasan Teori	4
2.1 Aplikasi Berbasis Web dan Pemrograman Web	4
2.2 <i>Entity Relationship Diagram</i> (ERD)	5
2.3 Web Server	5
2.4 Bahasa Pemrograman	5
2.5 Basis Data MySQL	5
BAB III - Perancangan Sistem	5
3.1 Tahapan Pengembangan	5
3.2 Analisis Kebutuhan	5
3.3 Analisis Sistem	6
3.4 Perancangan Sistem	6
3.4.1 Kebutuhan Fungsional	6
3.4.2 Kebutuhan Non Fungsional	7
3.4.3 Perancangan Basis Data	7
BAB IV - Implementasi Dan Pengujian	8
4.1 Spesifikasi Perangkat Keras yang Digunakan	8
4.2 Spesifikasi Perangkat Lunak yang Digunakan	8
4.3 Implementasi Antarmuka Pengguna	8
4.4 Analisis Hasil Pengujian	12
BAB V - Penutup	12
5.1 Kesimpulan	12
5.2 Saran	12

Abstrak

Kemajuan ilmu pengetahuan dan teknologi (IPTEK) adalah sesuatu yang tidak bisa dihindari dalam kehidupan dewasa ini. Setiap inovasi diciptakan untuk memberikan manfaat positif bagi kehidupan manusia. Kemajuan IPTEK memberikan banyak kemudahan, serta sebagai cara baru dalam melakukan aktivitas manusia. Manusia juga sudah menikmati banyak manfaat yang dibawa oleh inovasi-inovasi di bidang IPTEK yang telah dihasilkan dalam dekade terakhir ini.

Web application atau aplikasi berbasis web adalah program perangkat lunak yang berjalan di browser berjalan diatas protokol HTTP (Hypertext Transfer Protocol). Aplikasi berbasis web telah banyak digunakan sebagai media penyampai informasi di berbagai sektor. Namun di PT Fajar Surya Wisesa Tbk khususnya Departemen EI2 belum menerapkan penggunaan web application untuk memudahkan pekerjaan daily check & finding report. Daily check ampere & temperature masih dilakukan dengan cara konvensional yang memungkinkan terjadinya data masukkan yang tidak valid, dan masih menggunakan arsip dalam bentuk fisik yang rentan hilang atau bahkan hilang. Untuk dapat menangani permasalahan yang terjadi pada Departemen EI2 dibuatlah perancangan dan pembangunan aplikasi Sistem Informasi berbasis web. Agar dalam implementasi aplikasi tidak memakan biaya mahal, maka aplikasi dikembangkan menggunakan perangkat lunak open source. Untuk bahasa pemrograman server side menggunakan PHP sedangkan untuk basis data menggunakan MySQL, yang sudah terangkum dalam aplikasi XAMPP. Metode pengembangan aplikasi yang digunakan adalah metode SDLC (Software Development Life Cycle) model Waterfall. Hasil akhir dari perancangan dan pembangunan sistem informasi ini adalah sebuah aplikasi yang dapat memberikan kemudahan akses informasi sehingga proses preventive dan corrective dalam Departemen EI2 menjadi lebih efektif dan efisien.

Kata kunci: *Daily check, finding, data, web.*

I. PENDAHULUAN

1.1 Latar Belakang Masalah

Intranet adalah jaringan informasi yang meliputi ruang lingkup sempit, bahkan dapat dikatakan *private* apabila dibandingkan dengan internet. Sistem ini diciptakan agar ruang kerja digital suatu perusahaan lebih berfokus pada dokumen, proyek, alat, percakapan, beserta seluruh kinerja karyawannya.

Selain itu, intranet adalah portal untuk menyediakan akses data yang diperlukan pekerja. Tentunya sistem ini telah dilengkapi firewall dan *password* sehingga hanya bisa di akses oleh karyawan perusahaan tersebut. Sayangnya, di Departemen EI2 sendiri belum memanfaatkan penuh jaringan

intranet yang ada di Perusahaan untuk pekerjaan *preventive* dan *corrective*.

Daily check ampere dan temperature dilakukan setiap hari guna memantau proses produksi agar tetap efisien dan analisa saat terjadi masalah pada suatu motor atau panel. Namun penginputan data masih dilakukan dengan cara konvensional, mencatatnya di lembar kertas lalu menginput data ke dalam komputer sehingga memungkinkan data yang tidak valid saat penginputan dan juga memerlukan waktu relatif lama. Penggunaan data untuk analisa juga cenderung lambat, karena data belum terintegrasi dan terkelola dengan baik. Proses seperti ini juga masih menggunakan arsip dalam bentuk fisik yang rentan mengalami kerusakan atau bahkan hilang.

Pun dengan penanganan *finding inspection*, dilakukan dengan cara mengirimkan foto *finding* di grup whatsapp lalu memilah foto dan deskripsi yang nantinya di olah oleh pengguna dan di *generate* manual berbasis file folder.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah, dibuat suatu rumusan masalah yaitu:

- a. Bagaimana merancang dan mengembangkan aplikasi *daily check & manajemen finding* berbasis web yang dapat memberikan kemudahan bagi pengguna ?.

1.3 Batasan Masalah

Untuk menghindari pembahasan yang meluas maka dalam makalah ini ditetapkan batasan-batasan masalah sebagai berikut:

- a. Aplikasi *daily check* akan menerima input data *load* dan *temperature* dari pengguna.
- b. User memasukkan data *ampere & temperature* dan divalidasi sesuai basis data *module* dan *motor* di sisi klien.
- c. Data yang sudah di input bisa di *generate* menjadi *trend* dalam rentang waktu tertentu.
- d. Aplikasi manajemen *finding* hanya menerima data foto *finding*, nama, area, equipment, funcloc, dan deskripsi *finding* dari pengguna.
- e. *Finding* yang telah di upload akan di tampilkan di halaman web dan bisa di *update* status *finding* berdasarkan level user.
- f. Data *finding* yang sudah di unggah hanya bisa di hapus oleh user level Administrator.

- g. Aplikasi ini dibangun berbasis web menggunakan bahasa pemrograman PHP dan basis data MySQL yang sudah terangkum dalam satu paket aplikasi XAMPP sedangkan tampilan aplikasi dibuat dengan HTML, CSS & JavaScript.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah merancang dan mengembangkan Aplikasi Berbasis Web untuk menampilkan *ampere*, *temperature* dan *finding report* yang telah di input oleh user melalui *web browser*.

II. LANDASAN TEORI

2.1 Aplikasi Berbasis Web dan Pemrograman Web

Aplikasi berbasis web yaitu aplikasi yang menggunakan standar *Hypertext Transfer Protocol* (HTTP) dan membutuhkan *browser* untuk menggunakannya serta diakses melalui *internet* atau *intranet*. Aplikasi berbasis *web* merupakan salah satu dari sejumlah kategori perangkat lunak yang sifatnya khas.

Ada dua jenis pemrograman *web* yaitu SSP – *Server Side Programming* (Pemrograman pada Sisi Server) dan CSP – *Client Side Programming* (Pemrograman pada Sisi Klien). Pada SSP, semua sintaks dan perintah program diberikan akan dijalankan atau diproses di *web server*, kemudian hasilnya akan dikirimkan ke *browser* pengguna dalam bentuk teks HTML biasa, sehingga pengguna tidak dapat melihat kode asli yang ditulis dalam bentuk SSP tersebut. Sebaliknya, pada CSP semua sintaks dan perintah program dijalankan di *web browser*.

2.2 Entity Relationship Diagram (ERD)

ERD atau diagram E-R adalah sebuah model data yang menggunakan beberapa notasi untuk menggambarkan data dalam hal entitas dan relasi yang digambarkan oleh data tersebut. Komponen dasar ERD terdiri dari entitas, atribut dan hubungan antar relasi.

2.3 Web Server

Web server adalah server internet yang melayani permintaan web dari klien dengan menggunakan HTTP untuk melayani semua proses pentransferan data yang diminta oleh klien menuju *middleware*, dan respon dari *middleware* akan dikirim kembali kepada klien peminta data.

2.4 Bahasa Pemrograman

Bahasa pemrograman merupakan bahasa yang digunakan dalam pembuatan perangkat lunak aplikasi. Aplikasi Berbasis *Web* untuk menampilkan data load, temperature dan finding.

2.5 Basis Data MySQL

Basis data adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Sistem manajemen basis data adalah kumpulan data yang saling berhubungan dan kumpulan program untuk mengakses data

MySQL adalah sebuah perangkat lunak sistem manajemen basis data dengan menggunakan standar SQL (*Structured Query Language*) atau DBMS (*Database Management System*) yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL awalnya dibuat oleh perusahaan konsultan bernama TcX yang berlokasi di Swedia. Saat ini pengembangan MySQL berada di bawah

naungan perusahaan MySQL AB. Adapun perangkat lunak MySQL dapat diunduh www.mysql.com.

III. PERANCANGAN SISTEM

3.1 Tahapan Pengembangan

Untuk mengembangkan sistem aplikasi berbasis *web* yang baik diperlukan sebuah perancangan yang matang. Perancangan yang matang dilakukan agar sistem aplikasi yang akan dibuat nantinya dapat tepat sasaran dan tepat guna. Aplikasi pada makalah ini dirancang dengan metode SDLC model *waterfall*. Model *waterfall* terdiri dari lima tahapan pengembangan: yaitu analisis kebutuhan, analisis sistem, perancangan, implementasi dan pengujian.

3.2 Analisis Kebutuhan

1. Kebutuhan: Aplikasi Daily Check dan Finding Report berbasis web

- Masalah:
Belum adanya aplikasi daily check dan finding berbasis web untuk membantu mengolah dan menampilkan informasi data ampere & temperature dan finding.
- Usulan:
Data daily check dan finding diharapkan dapat ditampilkan secara *online* sehingga memberi kemudahan bagi pengguna untuk mengakses data ketika membutuhkannya.

2. Kebutuhan: Basis Data

- Masalah:
Belum adanya basis data yang menampung data ampere, temperature dan finding yang dapat dilihat kapanpun dimanapun di area Perusahaan.
- Usulan:
Perlu dibuat aplikasi berbasis *web* dengan menggunakan basis data.

3. Kebutuhan: Jaringan WiFi

- Masalah:
Tidak adanya jaringan WiFi untuk terhubung ke aplikasi melalui *smartphone* di semua area PM3 dan PM7 khususnya area MCC.
- Usulan:
Instal kabel LAN, *Switch/Router & WiFi* yang terhubung ke jaringan *intranet*.

4. Kebutuhan: Komputer Server

- Masalah:
Tidak adanya komputer server untuk *me-running* aplikasi daily check dan manajemen finding.
- Usulan:
Pembelian satu unit komputer khusus untuk server aplikasi.

3.3 Analisis Sistem

Lingkup aplikasi sistem daily check dan finding report adalah sebagai berikut :

1. Aplikasi berbasis *web*.
2. Aplikasi dioperasikan pada sebuah *web server*.
3. Pengguna aplikasi meliputi administrator dan setiap level jabatan yang ada di Departemen EI2.
4. Pengguna bisa mengakses aplikasi menggunakan komputer perusahaan atau *smartphone* pribadi.
5. Administrator digolongkan sebagai *high level user*. Leader, Supervisor dan Department Head di golongkan sebagai *mid level user*. Dan Operator & Foreman sebagai *low level user*.

Berdasarkan hasil analisis kebutuhan, pengembang memutuskan untuk mengelompokkan aktor atau pengguna aplikasi ini menjadi tiga kategori, yaitu sebagai berikut:

1. High Level User

High Level User adalah pengguna yang memiliki hak akses penuh terhadap aplikasi. Aktivitas yang dilakukan oleh High Level User adalah:

- a. *Create, Read, Update, Delete* data master. Data master meliputi master load, temperature, dan finding.

2. Mid Level User

Mid Level User adalah pengguna yang memiliki hak akses terbatas terhadap aplikasi. Aktivitas yang dilakukan oleh Mid Level User adalah:

- a. *Create, Read, Update* data master load & temperature & data master finding.

3. Low Level User

Low Level User adalah pengguna yang memiliki hak akses yang sangat terbatas terhadap aplikasi. Aktivitas yang dilakukan oleh Low Level User adalah:

- a. *Create, Read* data master

3.4 Perancangan Sistem

3.4.1 Kebutuhan Fungsional

Kebutuhan fungsional meliputi fungsi-fungsi yang dapat dilakukan oleh sistem, yaitu:

1. Adanya halaman login bagi *user*.
2. Adanya halaman input load & temperature.
3. Adanya *shortcut* menuju form input ampere & temperature pada *MCC* tertentu.
4. Adanya halaman form *trend* record ampere & temperature.
5. Adanya halaman *trend* record ampere & temperature.
6. Adanya halaman form input finding.

7. Adanya halaman finding data yang menampilkan foto dan keterangan dari finding yang telah di *upload*.

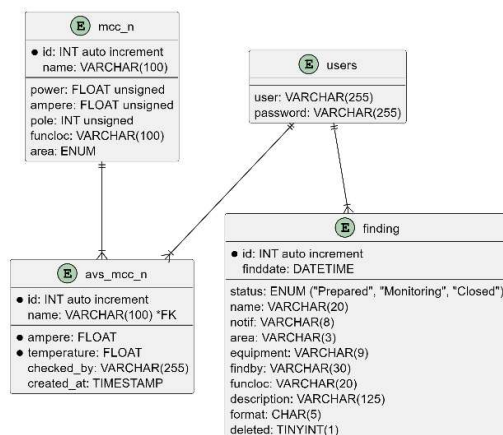
3.4.2 Kebutuhan Non Fungsional

Rumusan kebutuhan non fungsional meliputi:

1. Aplikasi harus mampu melakukan validasi data yang dimasukkan oleh pengguna di sisi klien.
2. Hanya pengguna yang terdaftar yang bisa menggunakan aplikasi.
3. Aplikasi harus mampu melakukan autentikasi dan otorisasi pengguna.

3.4.3 Perancangan Basis Data

Tahap selanjutnya adalah merancang basis data untuk menyimpan data-data yang akan dipergunakan. Dalam perancangan basis data dibutuhkan untuk menggambarkan hubungan antar data. Hubungan tersebut dapat dinyatakan dalam bentuk ERD (*Entity Relationship Diagram*), yaitu diagram yang menyajikan data E-R, dimana E berarti *Entity* (entitas) dan R berarti *Relationship* (hubungan).



Gambar.1

Pada Gambar.1 diatas ditunjukkan empat *entity* pokok, yaitu user, mcc_n, avs_mcc_n & finding.

Pada *entity* user memiliki atribut User & Password, atribut User adalah *username* pengguna. Password adalah kat sandi pengguna yang terdaftar sebagai pengguna aplikasi dan digunakan saat melakukan login pada aplikasi.

Pada *entity* mcc_n merepresentasikan total mcc yang setiap *instance* mcc berisikan data nama *module*, lalu atribut *power* atau daya, maksimal *ampere* atau *nominal* motor, jumlah kutub motor induksi atau *pole*, nama *function* *location* SAP, dan area motor.

Pada *entity* avs_mcc_n berisikan atribut nama *module*, data ampere yang dimasukkan oleh pengguna, data temperature yang dimasukkan oleh pengguna. Atribut *checked_by* yang berisikan data pengguna yang memasukkan data ampere dan temperature dan *created_at* yang berisikan data waktu kapan data itu dibuat, yang semua data tersebut akan ditampilkan di halaman *browser*.

Pada *entity* finding berisikan atribut status, dalam atribut ini hanya memiliki tiga nilai yang bisa dimasukkan oleh pengguna yaitu: Prepared yang berarti finding baru saja dibuat dan *notif* belum dibuat, Monitoring berarti finding yang di unggah sudah dibuat *notif* dan menunggu waktu yang tepat untuk pengerjaan dan Closed berarti finding telah selesai dikerjakan. Atribut *name* adalah nama dari finding, atribut *notif* adalah *notif* pekerjaan dari finding, attribute area adalah area dari finding, atribut *equipment* adalah nama *equipment* dari finding, atribut *description* adalah deskripsi dari finding, atribut format adalah file format yang diunggah oleh pengguna, dan format yang diterima hanya bertipe .jpeg, .jpg dan .png.

IV. IMPLEMENTASI DAN PENGUJIAN

4.1 Spesifikasi Perangkat Keras yang Digunakan

Spesifikasi perangkat keras yang digunakan dalam implementasi sistem adalah:

1. *Processor: Intel Core i5 2.9 GHz*
2. *RAM: 8GB*
3. *Hardisk: 1TB*
4. *System type: 64-bit Win OS*
5. *Address: http://10.55.11.117/*

4.2 Spesifikasi Perangkat Lunak yang Digunakan

Sistem aplikasi daily check dan finding berbasis *web* dibuat dengan sedemikian rupa agar pengguna merasa dimudahkan saat menggunakan aplikasi ini. Aplikasi ini diimplementasi dengan menggunakan:

1. *Sistem Operasi: Microsoft Windows 10*
2. *Web Server: Apache*
3. *Database: MySQL*
4. *Editor: Visual Studio Code*
5. *Browser: Mozilla Firefox*

4.3 Implementasi Antarmuka Pengguna

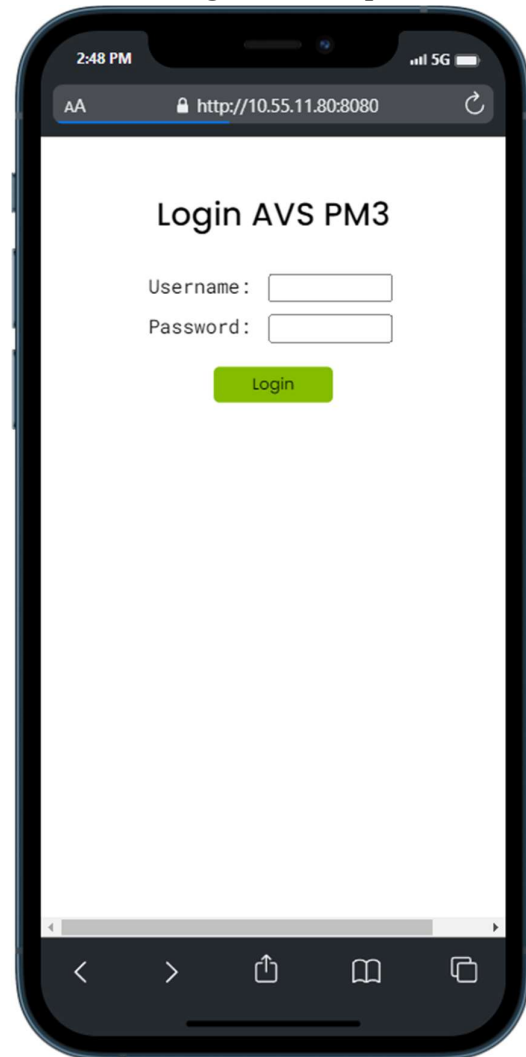
a. Halaman Login



Pengguna diminta *login* saat pertama kali menggunakan aplikasi, setelah

sukses *login* pengguna akan diarahkan menuju halaman input load & temperature.

b. Halaman Login di Smartphone

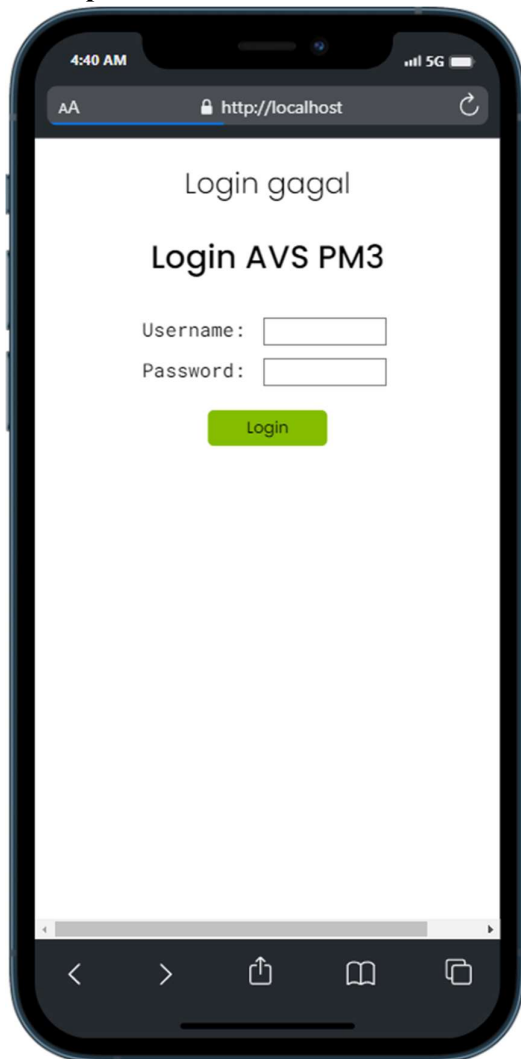


c. Halaman Login ketika gagal

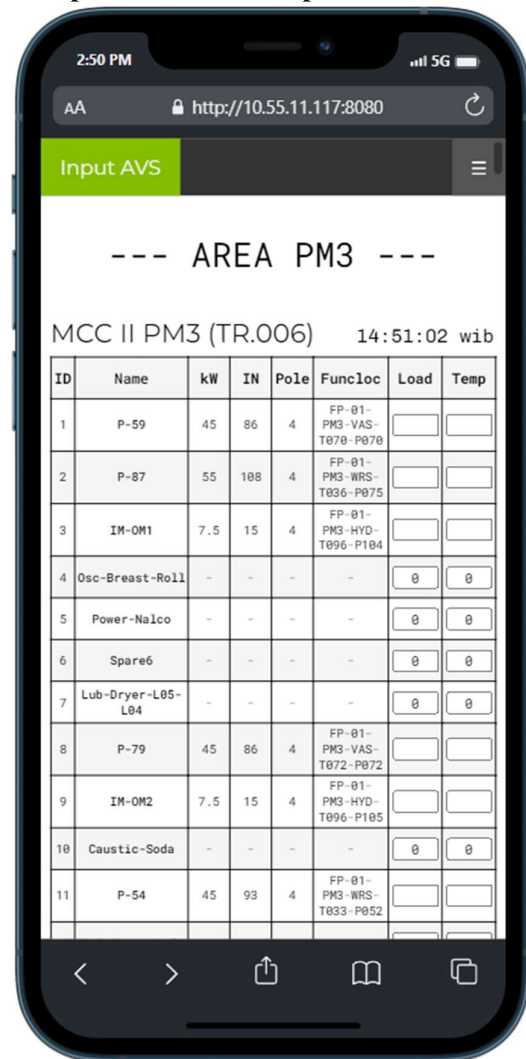


Ketika pengguna salah memasukkan kata sandi akan muncul keterangan *login* gagal.

d. Halaman Login ketika Gagal di Smartphone



f. Halaman Input Ampere & Temperature di Smartphone

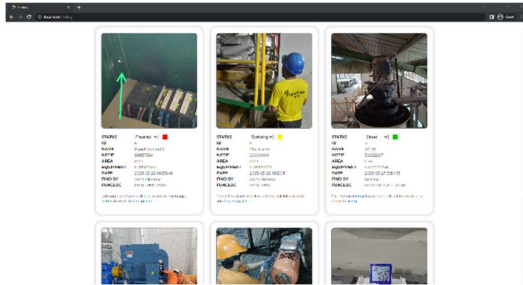


e. Halaman Input Ampere & Temperature

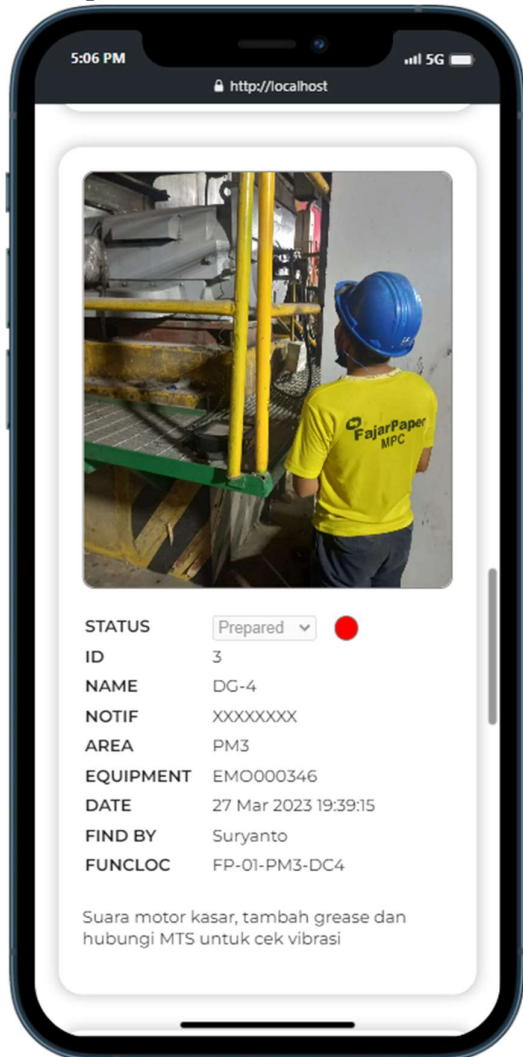


Pada halaman input load & temperature, data yang dimasukkan oleh pengguna divalidasi di browser sebelum dikirim ke *server*. Data ampere atau load tidak akan melebihi nominal motor dan data temperature tidak akan melebihi 150 derajat celcius. Dengan aplikasi ini pengguna tidak perlu mencatat di kertas lalu memasukkan kembali ke dalam komputer. Sehingga pekerjaan menjadi lebih mudah, dan penggunaan kertas bisa berkurang.

g. Halaman Finding Report



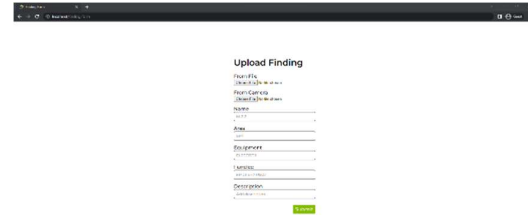
h. Halaman Finding Report di Smartphone



Pada halaman finding upload, perubahan status finding dan *notif* hanya bisa dilakukan oleh *mid level user* yaitu *level group leader* keatas. Dan *high level*

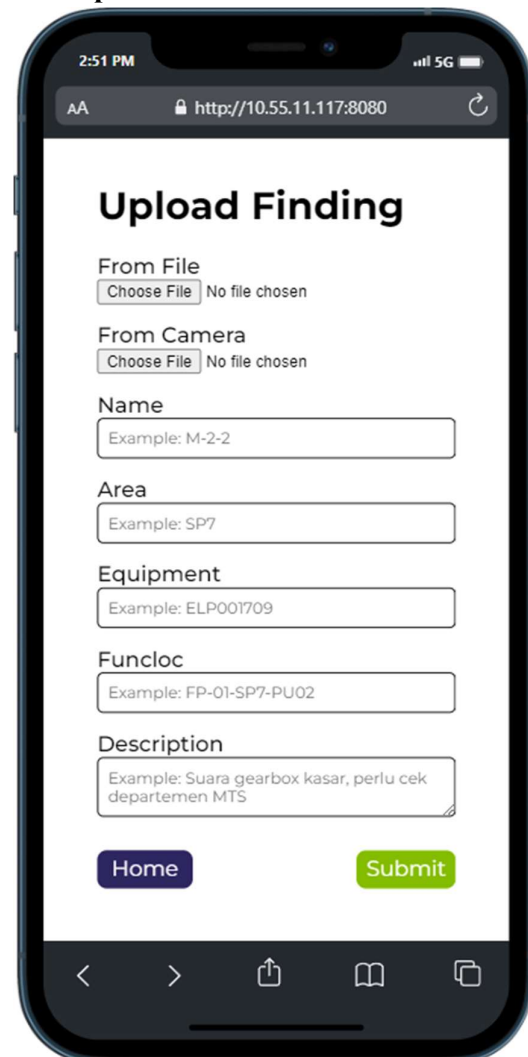
user bisa mengubah semua keterangan begitu juga menghapusnya.

i. Halaman Finding Upload



Pada halaman *form upload* finding *user* bisa memilih mengunggah file dari penyimpanan atau dari kamera perangkat.

j. Halaman Finding Upload di Smartphone



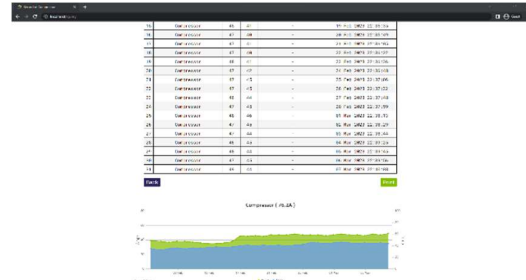
k. Halaman Form Query Record

l. Halaman Form Query Record di Smartphone

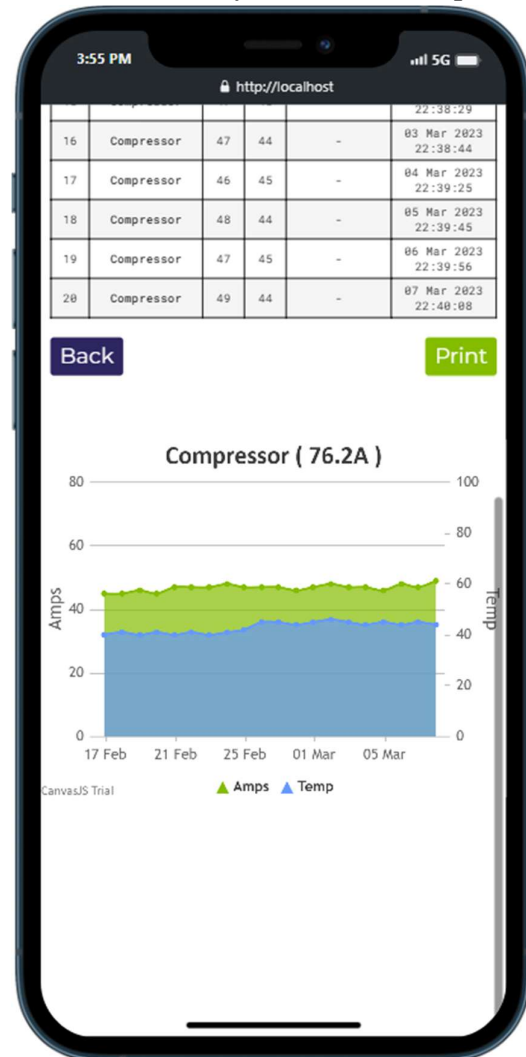
Pada halaman ini terdapat empat masukkan yang bisa di pilih oleh pengguna. *Start date* adalah awal waktu membaca *record* data dan *End date* adalah akhir waktu membaca *record* data yang nantinya akan ditampilkan dalam bentuk *trend*. *MCC*

name dan *Module name* adalah *MCC* (Motor Control Center) dan *Module* mana yang ingin di tampilkan *trend* nya.

m. Halaman Query Result



n. Halaman Query Result di Smartphone



Trend yang ditampilkan bisa di *print* dengan format .pdf.

4.4 Analisis Hasil Pengujian

Setelah melakukan tindak pengujian terhadap semua unit (menu pilihan) yang terdapat pada Aplikasi Daily Check dan Finding Report, dapat disimpulkan bahwa hasil keluaran (*output*) yang ditunjukkan selama proses pengujian sesuai dengan rancangan aplikasi program ini. Maka dapat dikatakan bahwa Aplikasi Berbasis Web dapat berfungsi dengan baik. Kesimpulan ini telah dibuktikan dari hasil pengujian.

V. PENUTUP

5.1 Kesimpulan

Dari pembahasan pada bab-bab sebelum ini, kesimpulan yang dapat diambil adalah sebagai berikut:

“Dari hasil pengujian yang telah dilakukan, diketahui bahwa hasil keluaran (*output*) yang ditunjukkan selama proses pengujian sesuai dengan rancangan sistem aplikasi ini. Maka dapat disimpulkan bahwa Aplikasi Berbasis *Web* Daily Check dan Finding Report berhasil dikembangkan dan dapat berfungsi dengan baik”.

5.2 Saran

Berikut ini adalah saran yang penulis harap akan berguna dalam mendukung kelancaran berjalannya sistem yang diusulkan:

1. Sistem dipelihara dengan baik supaya mampu terus berjalan sesuai dengan yang diharapkan. Pemeliharaan dapat dilakukan dengan cara mengecek kondisi perangkat keras dan perangkat lunak sistem secara teratur.
2. Pengecekan berkala pada *storage* finding *images* karena setiap foto finding yang diunggah memakan memori sekitar 4Mb.
3. Dilakukan *backup* basis data secara berkala minimal setiap satu tahun sekali.