

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama menjalani kegiatan magang di PT Patria Maritime Lines penugasan sebagai seorang *Programmer* pada divisi *IT & Digitalization* yang bertanggung jawab dalam penulisan kode dalam pengembangan website berdasarkan instruksi dan rancangan yang telah ada.

Penulis bekerja di bawah arahan langsung dari *site supervisor* dan mendapatkan bimbingan teknis maupun non-teknis dari seorang mentor. Selain itu, penulis juga berkolaborasi dengan seorang rekan yang berperan sebagai *Project Manager*. Kolaborasi ini meliputi proses analisis, identifikasi kebutuhan pengguna, perancangan fitur, hingga tahap pengujian sistem.

Koordinasi antaranggota tim dilakukan baik secara langsung di kantor (*offline*) maupun secara daring (*online*) melalui media komunikasi seperti WhatsApp, guna memastikan kelancaran dalam proses pengembangan serta penyelesaian tugas. Seluruh proses pengembangan sistem dilaksanakan di lingkungan perusahaan karena akses terhadap server dilakukan melalui koneksi *remote SSH* yang hanya dapat dijalankan melalui jaringan internet internal perusahaan.

3.2 Tugas yang Dilakukan

Selama menjalani kegiatan magang di PT Patria Maritime Lines, penulis diberikan berbagai tugas yang berkaitan dengan pengembangan website *Fleet Management System*. Adapun rincian tugas yang dilakukan selama masa magang adalah sebagai berikut:

1. Membuat tampilan tabel hasil (*report*) konsumsi bahan bakar berdasarkan data tabel 'timesheet', ekspor data ke format Excel pada fitur pemantauan konsumsi bahan bakar.
2. Membuat fitur *filter* data berdasarkan rute pelayaran pada fitur pemantauan konsumsi bahan bakar.
3. Membantu proses migrasi data dan penyesuaian database perusahaan dengan database yang diperoleh dari sistem PT CLS Argos Indonesia.

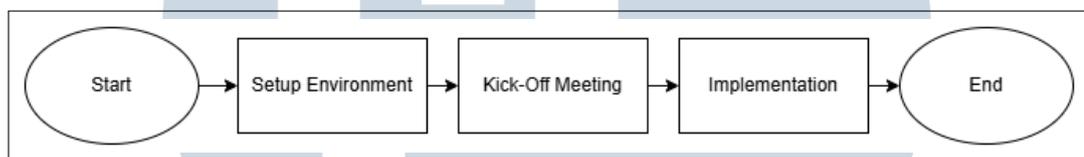
3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Melakukan pengenalan terhadap lingkungan kerja di kantor.
2	Melakukan persiapan lingkungan pengembangan (<i>environment setup</i>) termasuk konfigurasi server.
3	Mengikuti proses <i>onboarding</i> proyek sistem pemantauan konsumsi bahan bakar.
4	Menyusun dokumentasi proyek awal, termasuk pembuatan <i>flowchart</i> dan <i>entity relationship diagram</i> (ERD).
5	Membuat tampilan tabel <i>report</i> konsumsi bahan bakar berdasarkan data tabel 'timesheet'.
6	Menyelesaikan masalah duplikasi data pada tampilan tabel <i>report</i> konsumsi bahan bakar berdasarkan data tabel 'timesheet'.
7	Melakukan pembaruan progres bersama <i>supervisor</i> .
8	Melakukan penyesuaian struktur database perusahaan dengan data yang diperoleh dari sistem PT CLS Argos Indonesia.
9	Mengembangkan fitur <i>filter</i> berdasarkan rute pelayaran (<i>voyage route</i>).
10	Melakukan revisi pada tampilan <i>report</i> konsumsi bahan bakar berdasarkan data 'timesheet'.
11	Membuat <i>Endpoint</i> baru untuk melakukan <i>filter</i> data berdasarkan rute perjalanan.
12	Menambahkan tabel daftar rute pelayaran kapal untuk membantu pengguna dalam proses <i>data</i> .
13	Melakukan revisi pada fitur ekspor data ke dalam format Excel.
14	Mengembangkan tampilan <i>popup</i> interaktif untuk menampilkan detail kegiatan berdasarkan data tabel 'timesheet'.
15	Melakukan pengujian fitur yang telah dikembangkan.
16	Menambahkan validasi data dan melakukan finalisasi pengembangan fitur.

Gambar 3.1 menunjukkan secara rinci tahapan dalam proses pengembangan fitur pemantauan konsumsi bahan bakar. Tahap pertama adalah *Setup Environment*, yang mencakup penyiapan server untuk keperluan kolaborasi serta konfigurasi lingkungan kerja pada Visual Studio Code agar dapat terhubung ke server melalui remote SSH. Tahap berikutnya adalah *Kick-Off Meeting*, yang bertujuan untuk melakukan *briefing* awal terkait pengembangan fitur dan pembagian tugas. Selanjutnya adalah tahap *Implementation*, yang mencakup kegiatan menulis, membaca, dan memahami kode program berdasarkan arahan dan modul yang telah ditentukan.



Gambar 3.1. *Flowchart* proses pengembangan fitur

3.3.1 Perangkat Penunjang

Perangkat pendukung yang digunakan dalam proses pengembangan sistem ini diklasifikasikan menjadi dua kategori, yaitu perangkat keras dan perangkat lunak. Adapun perangkat keras yang digunakan meliputi:

Perangkat Keras:

- 1 unit laptop pribadi yang digunakan untuk pengembangan dan pengujian fitur
- 1 unit komputer milik perusahaan yang difungsikan sebagai server pengembangan

Perangkat Lunak:

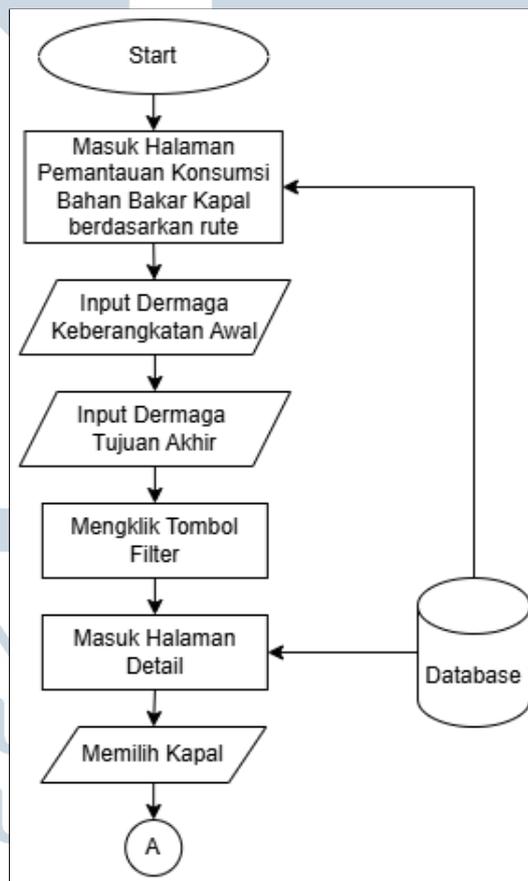
- Sistem operasi Linux pada komputer server
- *Framework* Yii 1 sebagai kerangka kerja pengembangan *backend* berbasis PHP
- XAMPP sebagai paket server lokal yang mencakup Apache, PHP, dan MySQL
- Postman sebagai alat bantu untuk pengujian dan dokumentasi API
- Visual Studio Code sebagai editor kode

3.3.2 Fitur Pemantauan Konsumsi Bahan Bakar

A. Requirements

Kebutuhan dari perusahaan adalah untuk menampilkan hasil *report* konsumsi bahan bakar kapal berdasarkan data tabel 'timesheet'. Fitur yang dikembangkan harus mampu menampilkan informasi konsumsi bahan bakar berdasarkan kegiatan-kegiatan kapal, seperti *Waiting Repair*, *Waiting Loading*, atau *Arrived at Loading Port*. Selain itu, sistem juga harus menyediakan fitur *filter* berdasarkan rute pelayaran tertentu, sehingga pengguna dapat memantau dan menganalisis efisiensi penggunaan bahan bakar pada masing-masing kapal berdasarkan rute *voyage*.

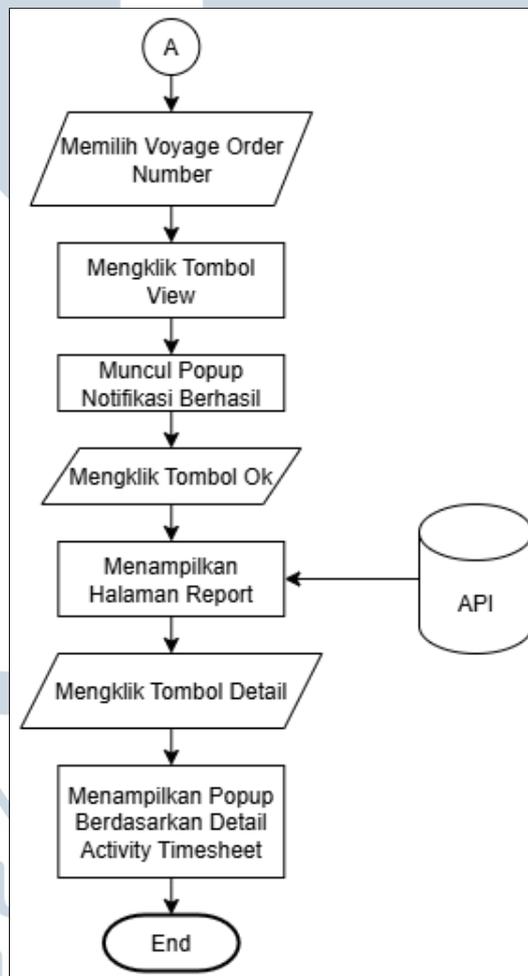
B. Flowchart



Gambar 3.2. Flowchart fitur pemantauan konsumsi bahan bakar

Gambar 3.2 menunjukkan proses dimulai dari tahap mulai, kemudian pengguna masuk ke halaman pemantauan konsumsi bahan bakar. Di halaman ini, pengguna diminta untuk melakukan input dermaga awal dan dermaga tujuan akhir. Setelah kedua lokasi dermaga diinputkan, kemudian mengklik tombol *filter* untuk mencari data kapal yang sesuai dengan rute pelayaran kapal tersebut. Proses pencarian ini melibatkan komunikasi dengan database, tempat semua data rute dan kapal disimpan.

Setelah hasil *filter* ditampilkan, pengguna dapat masuk ke halaman detail, yaitu halaman yang menampilkan informasi lebih lengkap dari setiap kapal yang tersedia. Pada tahap ini, pengguna dapat memilih kapal yang diinginkan. Setelah kapal dipilih, proses dilanjutkan ke bagian berikutnya.

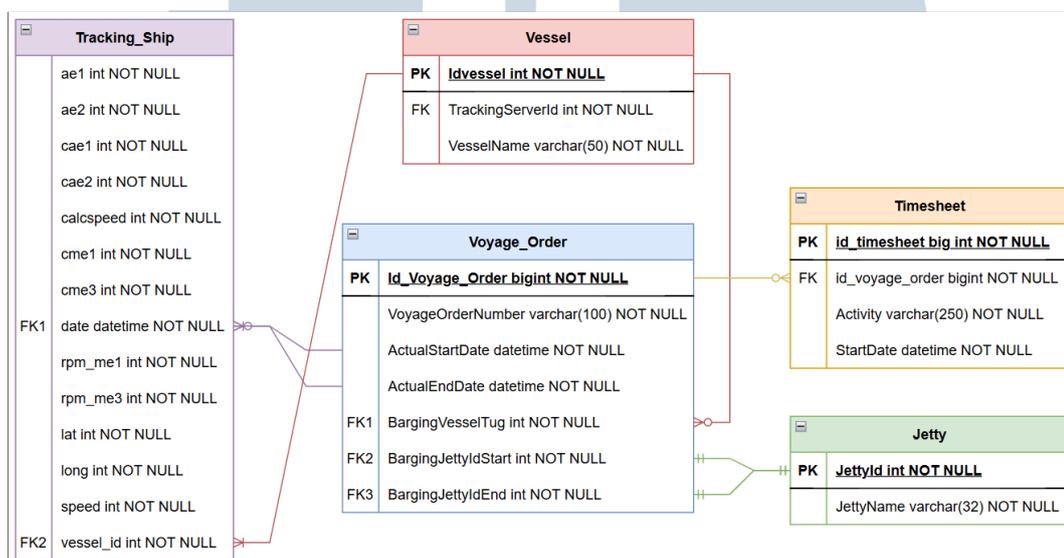


Gambar 3.3. Flowchart fitur pemantauan konsumsi bahan bakar lanjutan

Selanjutnya pada Gambar 3.3 pengguna memilih *voyage order number*. Setelah pemilihan ini, terjadi tindakan mengklik tombol *view*. Selanjutnya, sistem

akan menampilkan notifikasi berhasil. Pengguna kemudian melanjutkan dengan mengklik tombol ok. Setelah mengklik tombol ok, sistem akan menampilkan halaman laporan, yang berinteraksi dengan API (*Application Programming Interface*) kemungkinan untuk mengambil data yang diperlukan untuk laporan konsumsi bahan bakar dari kapal tersebut.

C. Entity Relationship Diagram



Gambar 3.4. Diagram ERD fitur pemantauan konsumsi bahan bakar

Entity Relationship Diagram ini menggambarkan struktur database yang digunakan untuk sistem pemantauan konsumsi bahan bakar. Terdapat lima tabel utama yaitu 'Vessel', 'Voyage_Order', 'Timesheet', 'Jetty', dan 'Tracking_Ship'.

C.1 'Vessel'

Tabel ini menyimpan data kapal, dengan 'Idvessel' sebagai *primary key*. Informasi kapal juga mencakup 'TrackingServerId' dan 'VesselName'. Entitas ini memiliki relasi dengan 'Voyage_Order', ditunjukkan oleh keterkaitan antara 'TrackingServerId' kapal dengan 'Id.Voyage_Order' dan 'IdVessel' dengan 'BargingVesselTug' pada tabel 'Voyage_Order'.

C.2 'Voyage_Order'

Tabel ini menjadi inti dari sistem karena berisi informasi terkait kegiatan pelayaran kapal. 'Id_Voyage_Order' berfungsi sebagai *primary key* dan menjadi referensi utama dalam tabel 'Timesheet'. Data yang disimpan dalam tabel ini mencakup 'VoyageOrderNumber', informasi kapal dan dermaga yang digunakan 'BargingVesselTug', 'BargingJettyIdStart', 'BargingJettyIdEnd', serta 'TrackingServerId' untuk kebutuhan pelacakan. Tabel 'Voyage_Order' memiliki beberapa relasi, yaitu: (1) 'Id_Voyage_Order' dengan *field* 'id_voyage_order' pada tabel 'Timesheet', (2) 'BargingVesselTug' dengan *field* 'IdVessel' pada tabel 'Vessel', dan (3) 'BargingJettyIdStart' serta 'BargingJettyIdEnd' dengan *field* 'JettyId' pada tabel 'Jetty'.

C.3 'Jetty'

Tabel ini menyimpan informasi detail mengenai dermaga seperti 'JettyId' sebagai *primary key*, 'JettyName'. Tabel ini terhubung dengan tabel 'Voyage_Order' melalui atribut 'JettyId' dengan *field* 'BaringJettyIdStart' dan 'BargingJettyIdEnd', yang mengindikasikan dermaga yang terkait dengan perjalanan kapal.

C.4 'Timesheet'

Tabel ini berfungsi mencatat detail kegiatan selama pelayaran. Setiap entri memiliki 'id_timesheet' sebagai *primary key*, 'id_voyage_order' yang menghubungkannya relasi 'id_voyage_order' dengan *field* 'id_voyage_order' pada tabel 'Voyage_Order'. Informasi lainnya meliputi 'Activity', tanggal mulai ('StartDate'), dan durasi aktivitas ('Duration'), yang semuanya berguna untuk pemantauan operasional kapal secara lebih rinci.

C.5 'Tracking_Ship'

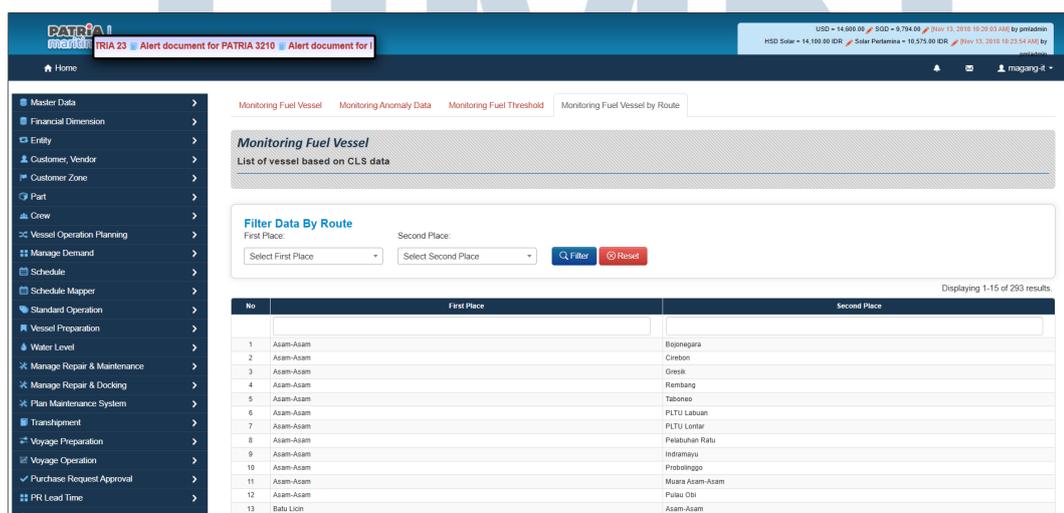
Pada Tabel ini data yang didapatkan berasal dari API yang sudah disediakan dari sistem milik PT CLS Argos Indonesia. Tabel ini berfungsi sebagai data yang digunakan untuk menampilkan hasil *report* konsumsi bahan bakar. 'vessel_id' memiliki relasi dengan 'Idvessel' pada tabel Vessel dan 'date' memiliki relasi dengan 'ActualStartDate' dan 'ActualEndDate' pada tabel Voyage_Order

D. Pengembangan

Fitur pemantauan konsumsi bahan bakar dikembangkan pada website *Fleet Management System* milik perusahaan, yang menerapkan pendekatan *Multi-Page Application* (MPA) dengan pemanfaatan kombinasi data dari sumber lokal dan eksternal melalui API. Dalam pelaksanaan proyek ini, penulis berfokus pada pengembangan dan penyesuaian fitur yang tersedia di server lokal perusahaan. Sementara itu, API untuk pengambilan data dari sistem eksternal telah disediakan oleh pihak perusahaan, sehingga penulis hanya perlu memanfaatkan API tersebut tanpa perlu mengembangkan ulang layanan *backend*-nya.

Software yang digunakan dalam pengembangan sistem ini menggunakan *framework* Yii 1 dengan bahasa pemrograman PHP sebagai sisi *backend*. Pada sisi *frontend*, digunakan Bootstrap untuk membangun tampilan antarmuka yang interaktif dan responsif. Selain itu, jQuery dimanfaatkan untuk berbagai keperluan, seperti pengelolaan tabel, penerapan fitur select2, manipulasi DOM, pengiriman permintaan AJAX, serta pengolahan data dalam format JSON yang diperoleh dari API. Proyek ini juga menggunakan berbagai *widget* bawaan dari Yii, seperti komponen tabel, serta mengintegrasikan library eksternal seperti SweetAlert untuk menampilkan notifikasi yang lebih menarik dan informatif kepada pengguna.

D.1 Halaman *Filter* Berdasarkan Rute



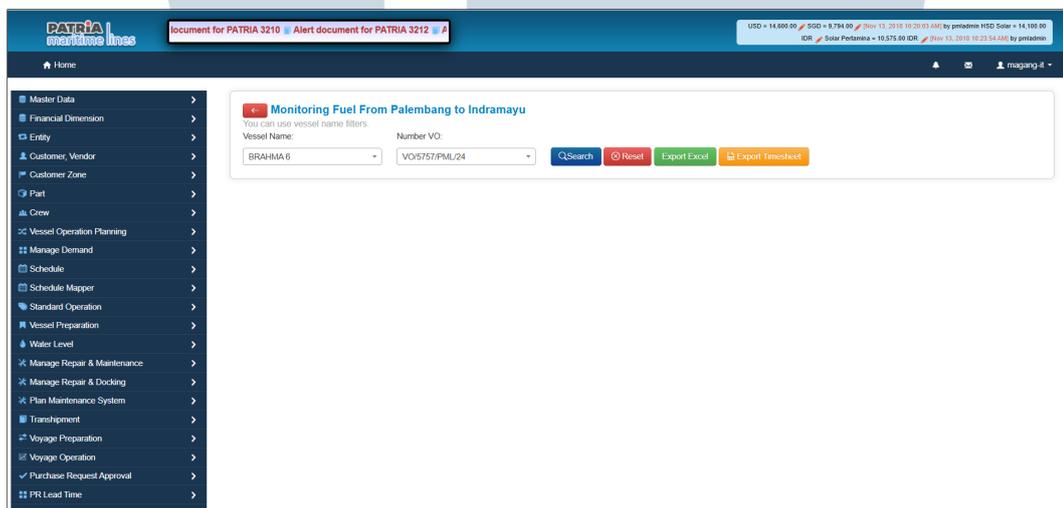
The screenshot displays the 'Monitoring Fuel Vessel' interface. At the top, there are navigation tabs: 'Monitoring Fuel Vessel', 'Monitoring Anomaly Data', 'Monitoring Fuel Threshold', and 'Monitoring Fuel Vessel by Route'. Below the tabs, the main content area is titled 'Monitoring Fuel Vessel' and contains a 'Filter Data By Route' section. This section has two dropdown menus labeled 'First Place' and 'Second Place', each with a 'Select' button. To the right of these dropdowns are 'Filter' and 'Reset' buttons. Below the filter section is a table with the following data:

No	First Place	Second Place
1	Asam-Asam	Roponegara
2	Asam-Asam	Cirebon
3	Asam-Asam	Gresik
4	Asam-Asam	Rembang
5	Asam-Asam	Taboneo
6	Asam-Asam	PLTU Labuan
7	Asam-Asam	PLTU Lontar
8	Asam-Asam	Perabuhan Ratu
9	Asam-Asam	Indramayu
10	Asam-Asam	Probolinggo
11	Asam-Asam	Muara Asam-Asam
12	Asam-Asam	Pelau Ohi
13	Batu Luhu	Asam-Asam

Gambar 3.5. Halaman *filter* berdasarkan rute

Gambar 3.5 menunjukkan halaman *filter* berdasarkan rute berfungsi untuk melakukan *filter* data kapal berdasarkan rute perjalanan yang dipilih dengan memilih *First Place* dan *Second Place* dari *dropdown* yang tersedia, kemudian menekan tombol *filter* untuk menampilkan data kapal yang sesuai dengan kombinasi rute tersebut. Sebaliknya, tombol *reset* digunakan untuk menghapus *filter* dan menampilkan kembali seluruh data yang tersedia. Terdapat tabel yang berisi daftar rute perjalanan berdasarkan rute perjalanan yang sudah pernah ditempuh yang.

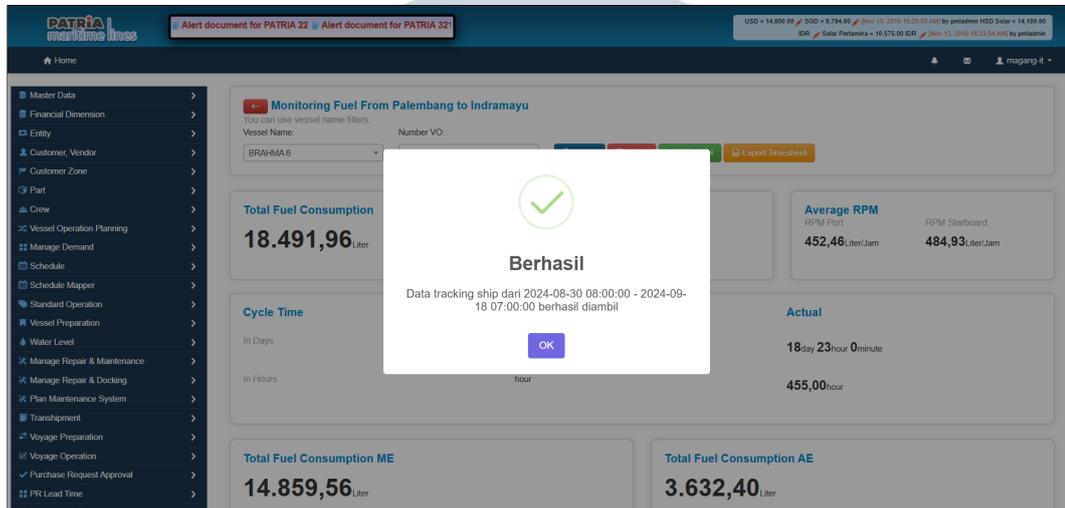
D.2 Halaman Memilih *Vessel* dan *Voyage Order Number*



Gambar 3.6. Halaman memilih *Vessel* dan *Voyage Order Number*

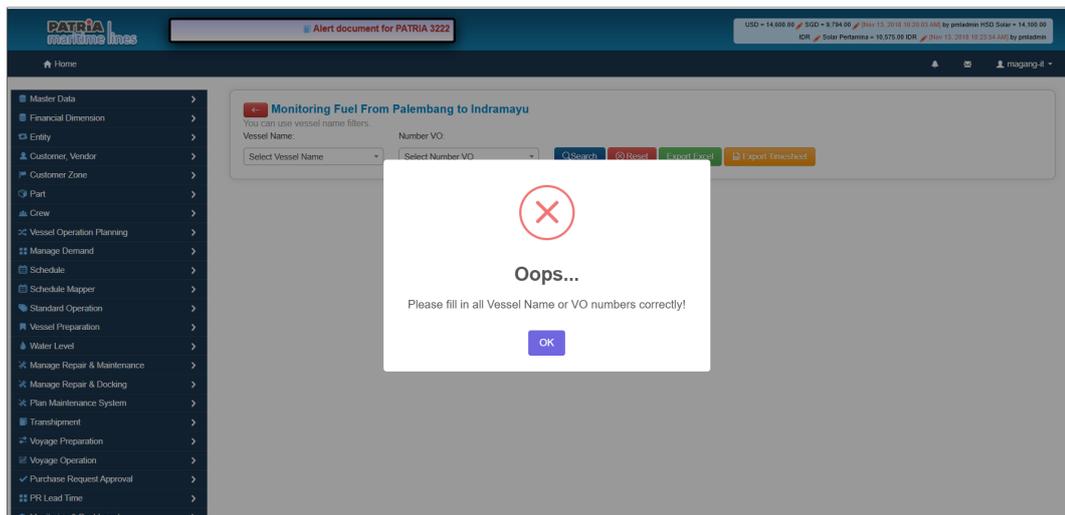
Pada Gambar 3.6 halaman ini berfungsi untuk memilih *Vessel Name* dan *Voyage Order Number* dari *dropdown* yang tersedia untuk menyaring informasi sesuai kebutuhan. Setelah memilih data yang diinginkan, pengguna dapat menekan tombol *search* untuk menampilkan informasi yang diinginkan. Apabila ingin menghapus *filter* dan kembali ke tampilan awal, tersedia tombol *reset*.

D.3 Notifikasi Berhasil, *Empty Error Handler*, dan *Error Data Not Found*



Gambar 3.7. Notifikasi berhasil

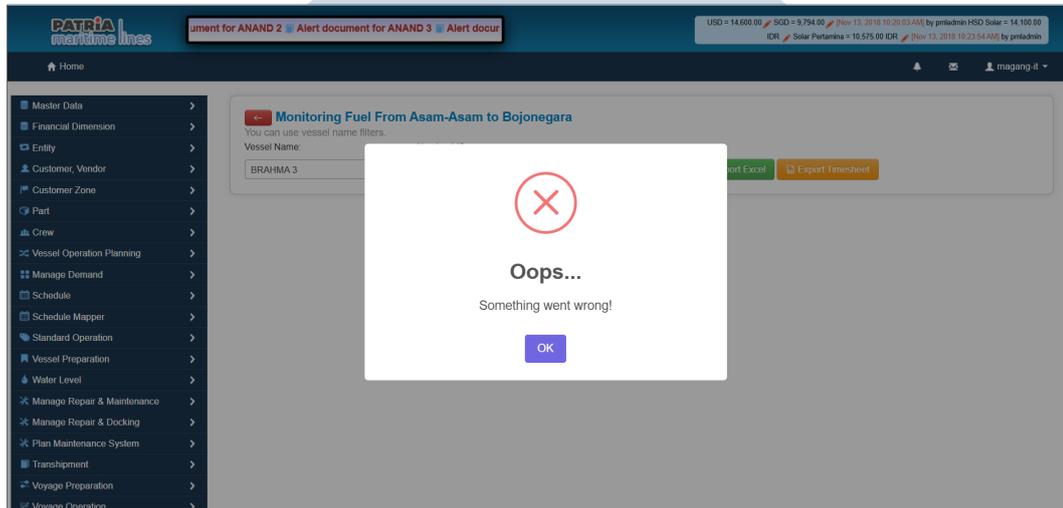
Notifikasi yang ditampilkan pada Gambar 3.7 menunjukkan bahwa proses pengambilan data *tracking ship* telah berhasil dilakukan. Notifikasi ini memberikan kejelasan kepada pengguna bahwa sistem telah berhasil memuat data sesuai dengan parameter waktu yang diminta.



Gambar 3.8. Notifikasi *empty error handler*

Notifikasi pada Gambar 3.8 menunjukkan bahwa sistem tidak dapat memproses permintaan karena input yang dibutuhkan belum diisi dengan benar. Notifikasi ini muncul untuk mencegah pencarian atau proses lebih lanjut dilakukan

tanpa adanya data penting yang diperlukan, dan bertujuan untuk mengarahkan pengguna agar melengkapi informasi yang diminta terlebih dahulu sebelum melanjutkan.



Gambar 3.9. Notifikasi *error data not found*

Notifikasi pada Gambar 3.9 ini berarti bahwa telah terjadi kesalahan saat sistem mencoba memproses permintaan pengguna. Meskipun pengguna telah memilih *Vessel Name* dan *Voyage Order Number*, sistem tetap gagal memproses permintaan tersebut. Ini kemungkinan disebabkan oleh kesalahan internal aplikasi, data yang tidak valid, atau tidak tersedianya data terkait kombinasi kapal dan *Voyage Order Number* yang dipilih.

D4. Hasil *Report* Konsumsi Bahan Bakar

Alur pseudocode yang digunakan untuk mendapatkan data konsumsi bahan dilihat pada Kode 3.1.

```

1 START
2   DECLARE VoyageOrderNumber : STRING
3   DECLARE StartDate, EndDate : DATETIME
4   DECLARE vesselId, startTimestamp, endTimestamp, i : INTEGER
5   DECLARE voyageOrder, response, item, aktivitas : RECORD
6   DECLARE timesheets, sortedTimesheets, dataFromTimesheet,
   dataTracking : ARRAY OF RECORD
7
8   GET StartDate, EndDate, vesselId, VoyageOrderNumber
9
10  SET timesheets <- CariTimesheet(voyageOrder.id_voyage_order)
11
12  IF timesheets != EMPTY THEN
13    FOR setiap ts IN timesheets DO

```

```

14     item.Activity = ts.Activity
15     item.Duration = ts.Duration
16     item.StartDateActivity = KonversiKeTimestamp(ts.StartDate
)
17     ADD item TO sortedTimesheets
18 ENDFOR
19
20     SORT sortedTimesheets BY StartDateActivity ASCENDING
21
22     FOR i FROM 0 TO LENGTH(sortedTimesheets) - 1 DO
23         SET startTimestamp = sortedTimesheets[i].
StartDateActivity
24
25         IF i + 1 < LENGTH(sortedTimesheets) THEN
26             SET endTimestamp = sortedTimesheets[i + 1].
StartDateActivity
27         ELSE
28             SET endTimestamp = startTimestamp + 1
29         ENDIF
30
31         aktivitas.activity = sortedTimesheets[i].Activity
32         aktivitas.duration = sortedTimesheets[i].Duration
33         aktivitas.StartDateActivity = KonversiKeDatetime(
startTimestamp)
34         aktivitas.EndDateActivity = KonversiKeDatetime(
endTimestamp)
35
36         ADD aktivitas TO dataFromTimesheet
37     ENDFOR
38 ENDIF
39
40 SET response = AmbilDataTracking(StartDate, EndDate, vesselId)
41 SET dataTracking = response.data
42
43 response.data = dataTracking
44 response.timesheet_data = dataFromTimesheet
45 response.meta.code = 200
46 response.meta.message = JikaTersedia(response.message, "Success
")
47 response.meta.status = JikaTersedia(response.status, "OK")
48 DISPLAY JSON(response)
49 END

```

Kode 3.1: Alur pseudocode mendapatkan data konsumsi bahan bakar

Pseudocode pada Kode 3.1 menjelaskan alur sistem dalam proses pengambilan dan pengolahan data konsumsi bahan bakar kapal berdasarkan parameter input dari pengguna. Proses dimulai dengan pengguna yang memasukkan data berupa 'StartDate', 'EndDate', 'vesselId', dan 'VoyageOrderNumber'. Berdasarkan nomor tersebut, sistem akan mencari data tabel 'timesheet' yang terkait, kemudian menyusunnya menjadi rangkaian aktivitas pelayaran yang terurut secara kronologis berdasarkan waktu mulai aktivitas. Setiap aktivitas akan memiliki waktu mulai ('StartDateActivity') dan waktu selesai ('EndDateActivity') yang dihitung dari aktivitas berikutnya atau

ditambah durasi *default* satu detik jika berada di akhir untuk mengatasi masalah duplikasi data atau tidak terambil karena 'EndDateActivity' sama dengan 'date' pada data 'Tracking_Ship'.

Setelah aktivitas terbentuk, sistem akan mengambil data *tracking* menggunakan *request API* dari sumber eksternal berdasarkan parameter waktu dan kapal. Data tracking ini dapat dilihat pada Tabel 3.4 yang memuat informasi posisi kapal, kecepatan, serta nilai konsumsi bahan bakar mesin utama (*Main Engine/ME*) dan mesin bantu (*Auxiliary Engine/AE*). Seluruh hasil dari proses tersebut dikemas dalam struktur *response*, yang terdiri dari data tracking (*response.data*), data aktivitas (*response.timesheet_data*), serta metadata berupa kode status dan pesan. Data ini akan dikembalikan dalam format JSON dan digunakan pada perhitungan selanjutnya.

Berikut adalah alur pseudocode perhitungan data konsumsi bahan bakar dilihat pada pada Kode 3.2

```

1 START
2   SET datas = response.data
3
4   DECLARE totalCME1, totalCME3, totalCAE1, totalCAE2, totalFuel,
   count, totalHours : INTEGER = 0
5
6   IF LENGTH(datas) > 1 THEN
7     SORT datas BERDASARKAN tanggal ASCENDING      sortedData
8
9     SET startDate = KONVERSI_DATETIME(sortedData[0].date)
10    SET endDate = KONVERSI_DATETIME(sortedData[LENGTH(sortedData)
    - 1].date)
11
12    SET diffMillis = endDate - startDate
13    SET totalMinutes = FLOOR(diffMillis / (1000 * 60))
14    SET totalHours = totalMinutes / 60
15  ENDIF
16
17  FOR setiap data IN datas DO
18    SET cme1 = data.cme1 OR 0
19    SET cme3 = data.cme3 OR 0
20    SET cae1 = data.cae1 OR 0
21    SET cae2 = data.cae2 OR 0
22
23    totalCME1 = totalCME1 + cme1
24    totalCME3 = totalCME3 + cme3
25    totalCAE1 = totalCAE1 + cae1
26    totalCAE2 = totalCAE2 + cae2
27
28    totalFuel = totalFuel + cme1 + cme3 + cae1 + cae2
29    count = count + 1
30  ENDFOR
31
32  IF totalHours > 0 THEN
33    SET averageME = (totalCME1 + totalCME3) / totalHours
34    SET averageAE = (totalCAE1 + totalCAE2) / totalHours

```

```

35     SET averageTotal = averageME + averageAE
36 ELSE
37     SET averageME = 0
38     SET averageAE = 0
39     SET averageTotal = 0
40 ENDIF
41 END

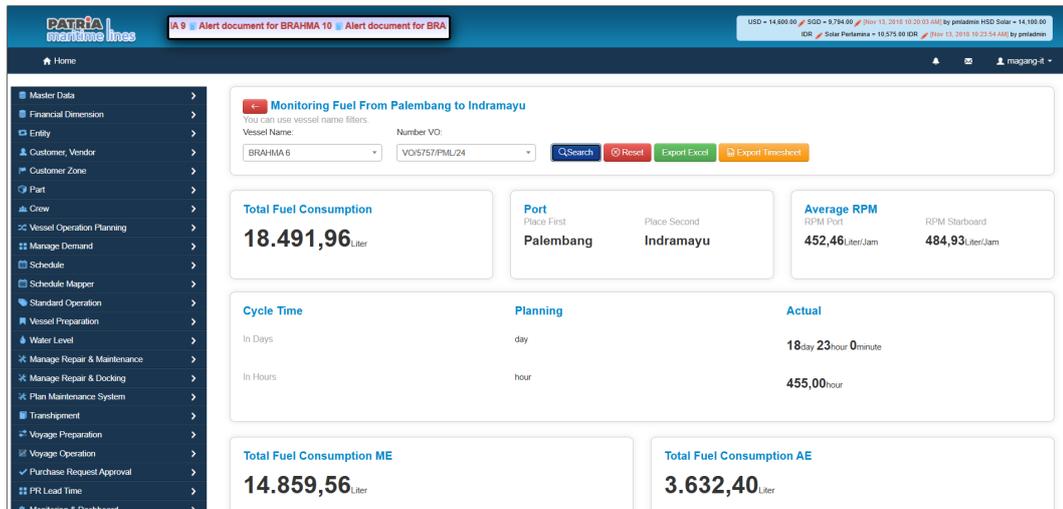
```

Kode 3.2: Alur pseudocode perhitungan data konsumsi bahan bakar

Pseudocode pada Kode 3.2 menjelaskan proses perhitungan total dan rata-rata konsumsi bahan bakar berdasarkan data pelacakan kapal yang diperoleh dari sistem. Proses diawali dengan mengambil seluruh data dari respons server yang disimpan ke dalam variabel datas. Kemudian, sistem mengecek apakah jumlah data lebih dari satu, yang diperlukan untuk menghitung durasi pelayaran. Jika ya, data tersebut diurutkan berdasarkan tanggal secara menaik, dan dihitung selisih waktu antara tanggal awal dan akhir. Selisih ini diubah dari milidetik menjadi menit, lalu dikonversi menjadi jam dan disimpan dalam variabel totalHours.

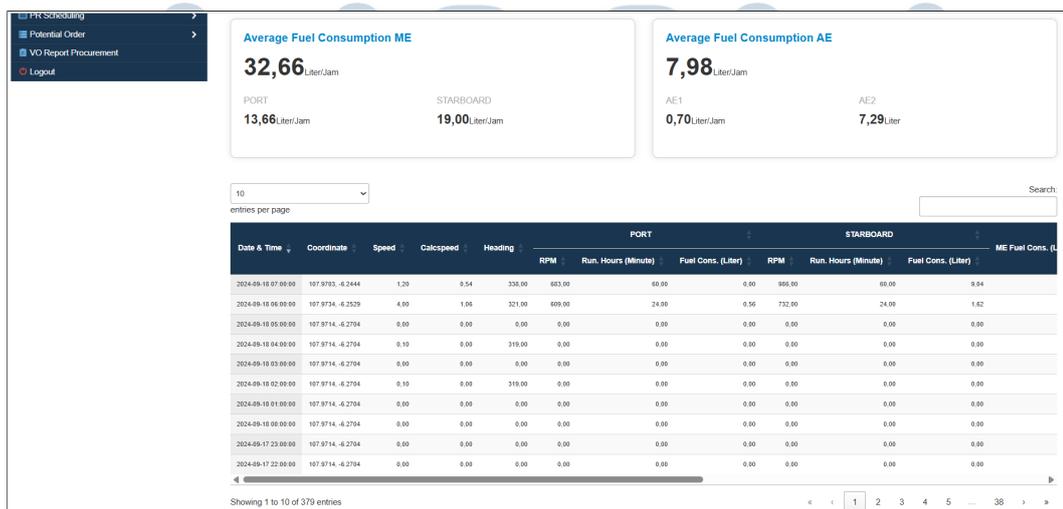
Selanjutnya, sistem melakukan iterasi terhadap setiap data untuk menjumlahkan konsumsi bahan bakar mesin utama (ME) yang terdiri dari 'cme1' dan 'cme3', serta mesin bantu (AE) dari cae1 dan cae2. Seluruh nilai tersebut dijumlahkan ke dalam variabel akumulator, termasuk 'totalFuel' untuk keseluruhan konsumsi dan 'count' sebagai jumlah data. Setelah proses akumulasi selesai, sistem memeriksa apakah 'totalHours' lebih dari nol. Jika ya, maka rata-rata konsumsi bahan bakar per jam untuk ME dan AE dihitung masing-masing dengan membagi total konsumsi ME dan AE dengan durasi pelayaran ('totalHours'). Total rata-rata konsumsi ('averageTotal') dihitung dari penjumlahan kedua rata-rata tersebut. Jika tidak ada durasi waktu yang valid, semua nilai rata-rata akan diset ke nol.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.10. Hasil *report* konsumsi bahan bakar

Pada Gambar 3.10 merupakan hasil *report* konsumsi bahan bakar setelah melakukan proses pengolahan data yang dapat dilihat pada Kode 3.1 mulai dari proses pengolahan data sampai perhitung data yang dapat dilihat pada Kode 3.2. Informasi yang disajikan meliputi total konsumsi bahan bakar kapal, lokasi port, *average revolutions per minute* (RPM), *data cycle time*, jumlah konsumsi bahan bakar oleh *main engine* (ME), dan jumlah konsumsi bahan bakar oleh *auxiliary engine* (AE).



Gambar 3.11. Fitur pemantauan konsumsi bahan bakar lanjutan

Selanjutnya pada Gambar 3.11, terdapat dua kartu informasi utama, yaitu *average fuel consumption main engine* (ME) dan *average fuel consumption*

auxiliary engine (AE), yang masing-masing menampilkan rata-rata konsumsi bahan bakar per durasi jam dalam satuan liter. Di bagian bawah, terdapat tabel data *tracking ship* yang masih bersifat mentah. Tabel ini menampilkan detail operasional seperti tanggal dan waktu, koordinat lokasi kapal, kecepatan kapal, *calspeed*, *heading*, serta parameter mesin di sisi *port* dan *starboard*, seperti *RPM*, durasi operasi dalam menit, dan konsumsi bahan bakar. Kolom terakhir memperlihatkan konsumsi bahan bakar dari *main Engine (ME)*.

D5. Tabel Hasil Report Konsumsi Bahan Bakar Berdasarkan Data 'timesheet'

```

1 START
2   FOR setiap aktivitas IN timesheetData DO
3     SET startDate = aktivitas.StartDateActivity
4     SET endDate = aktivitas.EndDateActivity
5
6     SET filteredData = FILTER dataTracking WHERE date
startDate AND date < endDate
7
8     DECLARE sumCME1, sumCME3, sumCAE1, sumCAE2 = 0
9
10    FOR setiap data IN filteredData DO
11      sumCME1 += (data.cme1 OR 0)
12      sumCME3 += (data.cme3 OR 0)
13      sumCAE1 += (data.cae1 OR 0)
14      sumCAE2 += (data.cae2 OR 0)
15    ENDFOR
16
17    aktivitas.total_me_fuel = sumCME1 + sumCME3
18    aktivitas.total_ae_fuel = sumCAE1 + sumCAE2
19    aktivitas.total_fuel = aktivitas.total_me_fuel + aktivitas.
total_ae_fuel
20    aktivitas.jumlahData = JUMLAH data dalam filteredData
21  ENDFOR
22
23  FOR setiap aktivitas IN timesheetData DO
24    SET totalMinutes = SELISIH_MENIT(aktivitas.StartDateActivity
, aktivitas.EndDateActivity)
25    SET totalHours = totalMinutes / 60
26    aktivitas.totalHours = totalHours
27
28    IF totalHours != 0 THEN
29      aktivitas.averageME = aktivitas.total_me_fuel /
totalHours
30      aktivitas.averageAE = aktivitas.total_ae_fuel /
totalHours
31    ELSE
32      aktivitas.averageME = 0
33      aktivitas.averageAE = 0
34    ENDIF
35  ENDFOR
36
37  DECLARE groupedData = GROUP timesheetData BY aktivitas.
activityName

```

```

38
39  FOR setiap group IN groupedData DO
40      group.total_me_fuel = SUM total_me_fuel dari semua aktivitas
        dalam group
41      group.total_ae_fuel = SUM total_ae_fuel dari semua aktivitas
        dalam group
42      group.total_fuel = group.total_me_fuel + group.total_ae_fuel
43      group.total_duration = SUM totalHours dari semua aktivitas
        dalam group
44
45      IF group.total_duration != 0 THEN
46          group.averageTotal = group.total_fuel / group.
total_duration
47      ELSE
48          group.averageTotal = 0
49      ENDIF
50  ENDFOR
51  END

```

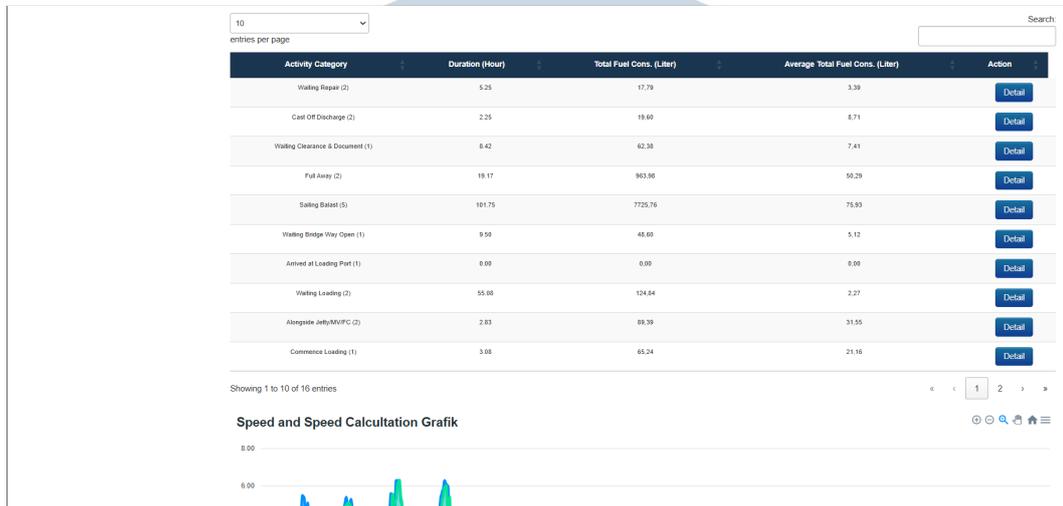
Kode 3.3: Alur pseudocode perhitungan data konsumsi bahan bakar berdasarkan timesheet

Pseudocode pada Kode 3.3 menjelaskan proses perhitungan konsumsi bahan bakar kapal berdasarkan 'timesheet'. Setiap entri dalam 'timesheetData' mewakili satu aktivitas kapal dengan waktu mulai dan selesai tertentu. Pertama, sistem melakukan *filter* data *tracking* yang waktu pencatatannya berada di antara 'StartDateActivity' dan 'EndDateActivity' dari masing-masing aktivitas. Dari data hasil *filter* tersebut, sistem menghitung total konsumsi bahan bakar dari mesin utama (*Main Engine/ME*) yang terdiri dari 'cme1' dan 'cme3', serta mesin bantu (*Auxiliary Engine/AE*) yang terdiri dari 'cae1' dan 'cae2'. Hasil penjumlahan konsumsi ME dan AE disimpan sebagai 'total_me_fuel' dan 'total_ae_fuel', kemudian dijumlahkan menjadi 'total_fuel'. Selain itu, jumlah data yang digunakan juga dihitung untuk referensi analisis.

Langkah berikutnya adalah menghitung durasi dari masing-masing aktivitas dengan mencari selisih waktu dalam menit, kemudian dikonversi ke dalam satuan jam ('totalHours'). Berdasarkan durasi ini, dihitung pula rata-rata konsumsi bahan bakar per jam untuk ME ('averageME') dan AE ('averageAE'). Jika durasi bernilai nol, maka nilai rata-rata otomatis diset ke nol untuk menghindari pembagian tidak valid.

Selanjutnya, seluruh aktivitas dikelompokkan berdasarkan jenis aktivitas yang sama menggunakan 'activityName'. Untuk setiap grup aktivitas, sistem menghitung total konsumsi bahan bakar dan total durasi dari semua aktivitas dalam kelompok tersebut. Dari sana, dihitung juga rata-rata total konsumsi bahan bakar per jam dalam setiap kelompok. Proses ini memungkinkan sistem memberikan informasi terperinci mengenai pola konsumsi bahan bakar berdasarkan

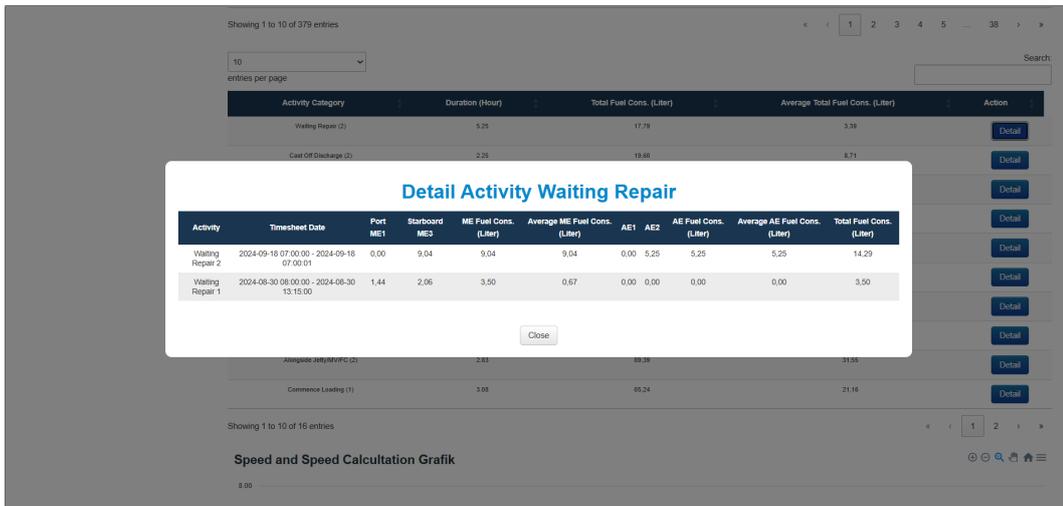
jenis aktivitas kapal seperti *waiting repair*, *cast off discharge*, dan *commence loading*, sehingga dapat digunakan sebagai dasar evaluasi efisiensi operasional.



Gambar 3.12. Tabel hasil *report* bahan bakar berdasarkan data 'timesheet'

Gambar 3.12 tersebut menampilkan tabel *report* konsumsi bahan bakar berdasarkan data 'timesheet' yang telah diklasifikasikan menurut kategori kegiatan kapal proses mendapatkan data dan perhitungan data dapat dilihat pada Kode 3.3, lengkap dengan informasi durasi, total konsumsi bahan bakar, serta rata-rata konsumsi bahan bakar untuk setiap aktivitas. Tabel ini merupakan bagian dari sistem pemantauan operasional kapal yang merinci berbagai kegiatan seperti *waiting repair*, *cast off discharge*, *sailing ballast*, hingga *commence loading*.

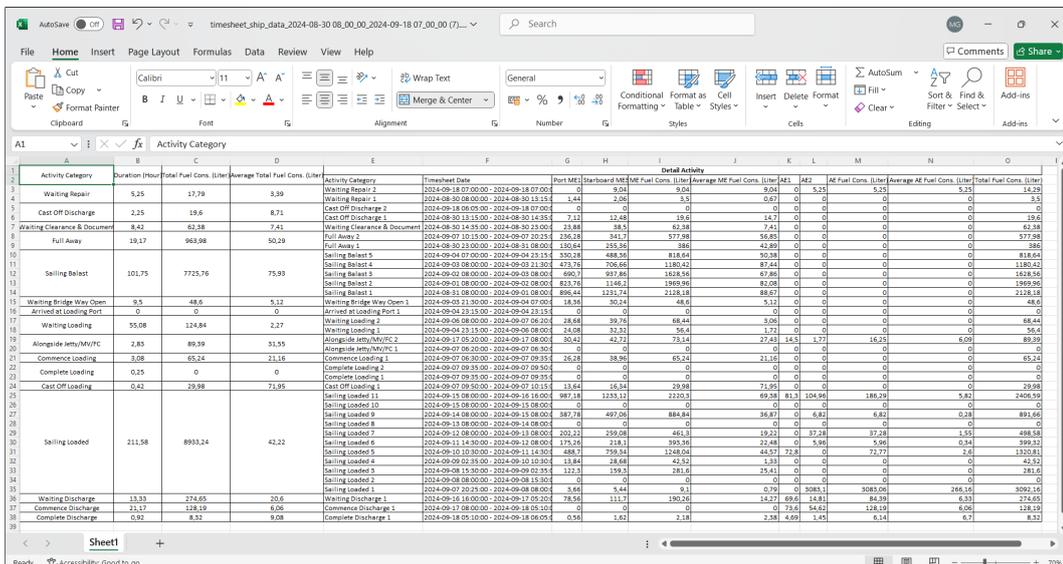
Setiap baris dalam tabel menampilkan kategori kegiatan, jumlah entri untuk masing-masing kegiatan, durasi (dalam jam), jumlah konsumsi bahan bakar dalam liter (*total fuel consumption*), serta rata-rata konsumsi bahan bakar per jam (*average total fuel Consumption*). Pada kolom paling kanan, terdapat tombol detail untuk setiap baris kegiatan yang akan mengarahkan pengguna ke tampilan *popup* berisi rincian lebih lanjut mengenai kegiatan tersebut.



Gambar 3.13. *Popur* detail berdasarkan data 'timesheet'

Gambar 3.13 menunjukkan *Popur* detail kegiatan pada gambar yang berfungsi untuk menampilkan rincian konsumsi bahan bakar berdasarkan data pada tabel 'timesheet'.

D6. Tampilan Hasil Ekspor Excel Berdasarkan Data 'timesheet'



Gambar 3.14. Hasil ekspor Excel berdasarkan data 'timesheet'

Berdasarkan hasil ekspor Excel dari data tabel 'timesheet' yang ditampilkan pada Gambar 3.14, data mencatat kegiatan kapal dalam berbagai kategori

operasional beserta durasi, konsumsi bahan bakar mesin utama (*main engine* (ME)) dan mesin bantu (*Auxiliary Engine* (AE)), serta rata-rata konsumsi selama periode waktu tertentu.

Secara keseluruhan, *timesheet* ini memberi gambaran menyeluruh atas penggunaan mesin dan konsumsi bahan bakar dalam berbagai kondisi operasional kapal, yang sangat penting untuk pemantauan dan analisis efisiensi bahan bakar dan pemeliharaan mesin.

E. Testing

Pengujian dilakukan oleh rekan penulis yang bertugas sebagai *Project Manager*, bersama dengan *supervisor* secara langsung saat sesi presentasi. Pengujian dilakukan dengan mengevaluasi kesesuaian fitur terhadap kebutuhan pengguna dan skenario penggunaan aktual. Hasil pengujian menunjukkan bahwa fitur berjalan dengan baik dan sesuai harapan, sehingga direncanakan untuk segera dideploy ke dalam server utama perusahaan.

3.4 Kendala dan Solusi yang Ditemukan

Selama proses pengembangan website *fleet management system*, ada beberapa kendala dalam mengerjakan fitur pemantauan konsumsi bahan bakar ini. Berikut adalah kendala yang didapatkan yaitu:

- Penulis perlu mempelajari kembali dasar-dasar serta struktur kerja dari *framework* Yii 1 karena belum memiliki pengalaman sebelumnya dalam menggunakan *framework* tersebut.
- Terjadi penyesuaian antara struktur database perusahaan dengan database yang diperoleh dari sistem PT CLS Argos Indonesia, yang menyebabkan keterlambatan dalam proses integrasi data.
- Gangguan pada koneksi internet dan API, serta kerusakan pada *hard drive* server lokal perusahaan, menghambat akses terhadap data dan mengganggu kelancaran pengembangan serta penulisan kode program.

Dengan kendala-kendala ini, penulis menemukan solusi untuk kendala tersebut yaitu:

- Penulis mempelajari *framework* Yii 1 secara mandiri melalui dokumentasi resmi dan video tutorial di YouTube.
- Penulis melakukan analisis struktur database dari sistem PT CLS Argos Indonesia dan mencocokkannya dengan struktur database perusahaan, serta melakukan penyesuaian data dan skema secara bertahap agar integrasi data dapat berjalan dengan lancar.
- Sambil menunggu koneksi API, jaringan internet, dan PC server lokal kembali normal, penulis tetap melanjutkan pekerjaan dengan melakukan analisis sistem dan *debugging* secara lokal.

