

**PENGEMBANGAN SISTEM INFORMASI PELABUHAN KAPAL PT
PATRAKIM TIMUR DI PT ANDROMEDIA**

PRAKTIK KERJA LAPANGAN



FALACHUL AKHADIHIMA IBRAHAICAL

NIM: 311910026

**UNIVERSITAS
MA CHUNG**

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS MA CHUNG

MALANG

2023

**LEMBAR PENGESAHAN
PRAKTIK KERJA LAPANGAN**

**PENGEMBANGAN SISTEM INFORMASI PELABUHAN KAPAL PT
PATRAKIM TIMUR DI PT ANDROMEDIA**

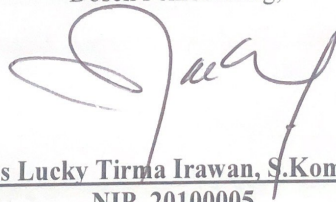
Oleh:

**FALACHUL AKHADIHIMA IBRAHAICAL
NIM. 311910026**

dari:

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS dan TEKNOLOGI
UNIVERSITAS MA CHUNG**

Dosen Pembimbing,



Paulus Lucky Tirma Irawan, S.Kom., MT.
NIP. 20100005

Dekan Fakultas Sains dan Teknologi



Dr. Kestrelia Rega Prilianti, M.Si.
NIP. 20120035

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Allah SWT yang telah melimpahkan karunia, rahmat, dan hidayah-Nya sehingga penulis dapat menyusun dan menyelesaikan laporan Praktik Kerja Lapangan (PKL) yang berjudul “Pengembangan Sistem Informasi Pelabuhan Kapal PT Patrakim Timur di PT Andromedia”. Laporan ini dibuat berdasarkan hasil dari Pratik Kerja Lapangan (PKL) di PT Andromedia melalui program Magang Bersertifikat Kampus Merdeka (MBKM) yang diselenggarakan oleh Kementrian Pendidikan, Kebudayaan, Riset, dan Teknologi Indonesia sebagai salah satu syarat untuk memenuhi kelulusan mata kuliah Praktik Kerja Lapangan (PKL). Selama proses penyusunan dan pembuatan laporan ini, penulis banyak mengalami kesulitan dalam pembuatanya. Oleh karena itu, penulis ingin berterima kasih kepada:

1. Kementerian Pendidikan dan Kebudayaan RI, selaku penyelenggara program Merdeka Belajar Kampus Merdeka.
2. Ibu Dr. Kestrilia Rega Prilianti, M.Si., selaku Dekan Fakultas Sains dan Teknologi Universita Ma Chung.
3. Bapak Paulus Lucky Tirma Irawan, S.Kom., MT., selaku Dosen Pembimbing Magang dan Praktik Kerja Lapangan Universitas Ma Chung.
4. PT. Andromedia selaku mitra program Kampus Merdeka yang memberikan kesempatan penulis untuk berproses dan belajar dalam mengembangkan ilmu di dunia kerja.
5. Kak Nizam selaku Mentor di Tim Full Stack Developer yang selalu memberikan bimbingan, arahan, saran, motivasi dan semangat kepada penulis dalam menyelesaikan program magang.
6. Kedua orang tua terkasih, yang telah memberikan dukungan dan semangat selama menjalani Praktik Kerja Lapangan.
7. Teman-teman Full Stack Developer dan semua pihak yang telah membantu penulis dalam menyelesaikan program magang ini yang tidak dapat penulis sebutkan satu persatu.

Susunan laporan praktik kerja lapangan ini dibuat dengan sebaik-baiknya. Namun, tidak menutup kemungkinan adanya kesalahan dan kekurangan, maka dari

itu penulis terbuka terhadap adanya kritik dan saran yang membangun dari semua pihak.

Malang, Januari 2023

Falachul Akhadihima Ibrahaical



UNIVERSITAS
MA CHUNG

DAFTAR ISI

LEMBAR PENGESAHAN	i
KATA PENGANTAR	ii
DAFTAR ISI.....	iv
DAFTAR GAMBAR	vi
DAFTAR TABEL.....	vii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Batasan Masalah.....	2
1.3 Rumusan Masalah	2
1.4 Tujuan.....	2
1.5 Manfaat.....	2
BAB II GAMBARAN UMUM PERUSAHAAN.....	3
2.1 PT Andromedia	3
2.2 Struktur Organisasi.....	4
2.3 Program Magang Divisi IT.....	4
2.4 Deskripsi Lingkup Kerja <i>Full-Stack Developer</i>	5
BAB III TINJAUAN PUSTAKA	6
3. 1 Tailwind CSS	6
3. 2 JavaScript	7
3. 3 TypeScript	8
3. 4 React.js	8
3. 5 Node.js.....	9
3. 6 Royal	9
3. 7 Database	9

3. 8 PostgreSQL	9
3. 9 Adminer.....	10
3. 10 <i>Object Relational Mapping</i> (ORM)	10
3. 11 Prisma	11
3. 12 <i>Version Control System</i> (VCS).....	11
3. 13 Git.....	12
3. 14 <i>Enterprise Resource Planning</i> (ERP).....	12
3. 15 <i>Unified Modelling Language</i> (UML).....	12
BAB IV HASIL DAN PEMBAHASAN	14
4.1 Analisis Kebutuhan Sistem	14
4.3 Desain Sistem.....	15
4.4 Implementasi Sistem	61
4.5 Testing.....	85
4.6 Perawatan Sistem	90
BAB V PENUTUP.....	91
5.1 Simpulan.....	91
5.2 Saran.....	91
DAFTAR PUSTAKA	92
LAMPIRAN.....	94

DAFTAR GAMBAR

Gambar 2. 1 Logo PT Andromedia.....	3
Gambar 2. 2 Struktur Organisasi PT Andromedia	4
Gambar 3. 1 Implementasi Tailwind CSS	6
Gambar 3. 2 Penggunaan CSS tanpa Tailwind CSS	7
Gambar 4. 1 Model <i>Waterfall</i>	14
Gambar 4. 2 <i>Use Case Diagram</i>	15
Gambar 4. 3 <i>Activity diagram export PDF</i>	16
Gambar 4. 4 <i>Activity diagram export Excel</i>	16
Gambar 4. 5 <i>Activity diagram mass update</i>	17
Gambar 4. 6 <i>Activity diagram mass delete</i>	18
Gambar 4. 7 <i>Entity Relationship Diagram</i> Bagian 1	19
Gambar 4. 8 <i>Entity Relationship Diagram</i> Bagian 2	20
Gambar 4. 9 Struktur <i>tag Admin</i> pada <i>framework</i> Royal	61
Gambar 4. 10 Halaman <i>Summary Training Realization</i>	66
Gambar 4. 11 Hasil Export PDF	67
Gambar 4. 12 Hasil <i>Export Excel</i>	67
Gambar 4. 13 <i>Form Training Realization</i>	68
Gambar 4. 14 <i>Property Form Framework</i> Royal	69
Gambar 4. 15 Peringatan salah input Start Date	70
Gambar 4. 16 <i>Custom Action Mass Update</i>	75
Gambar 4. 17 <i>Template Excel Fitur Mass Update</i>	76
Gambar 4. 18 Hasil pembuatan fitur mass update & delete	85

DAFTAR TABEL

Tabel 3. 1 Sudut Pandang <i>Diagram UML</i>	13
Tabel 4. 1 Rincian tabel m_cargo_color	21
Tabel 4. 2 Rincian Tabel m_cargo_grade	21
Tabel 4. 3 Rincian Tabel m_pg_personel_unit	22
Tabel 4. 4 Rincian Tabel m_unit.....	22
Tabel 4. 5 Rincian Tabel m_activity	22
Tabel 4. 6 Rincian Tabel m_area	22
Tabel 4. 7 Rincian Tabel m_asset	23
Tabel 4. 8 Rincian Tabel m_port.....	24
Tabel 4. 9 Rincian Tabel m_jetty.....	24
Tabel 4. 10 Rincian Tabel m_asset_type	25
Tabel 4. 11 Rincian Tabel m_asset_status	25
Tabel 4. 12 Rincian Tabel m_ownership	25
Tabel 4. 13 Rincian Tabel m_asset_items	26
Tabel 4. 14 Rincian Tabel m_work_center.....	26
Tabel 4. 15 Rincian Tabel m_asset_item_troubles	26
Tabel 4. 16 Rincian Tabel m_asset_type_role	27
Tabel 4. 17 Rincian Tabel p_role.....	27
Tabel 4. 18 Rincian Tabel v_housekeeping_asset	27
Tabel 4. 19 Rincian Tabel m_vessel_crane_brands.....	28
Tabel 4. 20 Rincian Tabel m_bwt_exclude	28
Tabel 4. 21 Rincian Tabel m_bariksama_note.....	28
Tabel 4. 22 Rincian Tabel m_contract_type	29
Tabel 4. 23 Rincian Tabel m_cargo	29
Tabel 4. 24 Rincian Tabel m_cargo_category	30
Tabel 4. 25 Rincian Tabel m_cargo_uom.....	30
Tabel 4. 26 Rincian Tabel m_arrival_type.....	30
Tabel 4. 27 Rincian Tabel m_cargo_status	31
Tabel 4. 28 Rincian Tabel m_cargo_procurement.....	31
Tabel 4. 29 Rincian Tabel m_cargo_color_grade	31
Tabel 4. 30 Rincian Tabel m_cargo_orde	32

Tabel 4. 31 Rincian Tabel m_wms_safety_stock	32
Tabel 4. 32 Rincian Tabel m_cargo_owner	33
Tabel 4. 33 Rincian Tabel p_user	33
Tabel 4. 34 Rincian Tabel m_checksheets_type_ops	34
Tabel 4. 35 Rincian Tabel m_currency_convert	35
Tabel 4. 36 Rincian Tabel m_currency	35
Tabel 4. 37 Rincian Tabel m_document_type	35
Tabel 4. 38 Rincian Tabel m_document_region	36
Tabel 4. 39 Rincian Tabel t_wms_forecast_authority	37
Tabel 4. 40 Rincian Tabel m_wms_authority	37
Tabel 4. 41 Rincian Tabel m_housekeeping_group	37
Tabel 4. 42 Rincian Tabel m_target_cargo	38
Tabel 4. 43 Rincian Tabel m_holiday	38
Tabel 4. 44 Rincian Tabel m_vessel_hatch_type	38
Tabel 4. 45 Rincian Tabel m_isps	39
Tabel 4. 46 Rincian Tabel m_laytime	39
Tabel 4. 47 Rincian Tabel m_activity_laytime	39
Tabel 4. 48 Rincian Tabel m_laytime_start_count	40
Tabel 4. 49 Rincian Tabel m_liquid_logsheets	40
Tabel 4. 50 Rincian Tabel m_location	40
Tabel 4. 51 Rincian Tabel m_wms_material_number	41
Tabel 4. 52 Rincian Tabel m_wms_nopol	41
Tabel 4. 53 Rincian Tabel m_wms_transportir	41
Tabel 4. 54 Rincian Tabel m_origin	42
Tabel 4. 55 Rincian Tabel m_pg_personel	42
Tabel 4. 56 Rincian Tabel m_pg_personel_scope	43
Tabel 4. 57 Rincian Tabel m_pg_personel_position	43
Tabel 4. 58 Rincian Tabel m_pg_personel_unit	43
Tabel 4. 59 Rincian Tabel m_housekeeping_payrate	44
Tabel 4. 60 Rincian Tabel m_housekeeping_payrate_asset	44
Tabel 4. 61 Rincian Tabel m_project_aspek_penilaian	44
Tabel 4. 62 Rincian Tabel m_project_permit	45

Tabel 4. 63 Rincian Tabel m_wms_plant	45
Tabel 4. 64 Rincian Tabel m_wms_plant_warehouse	45
Tabel 4. 65 Rincian Tabel m_wms_sloc	46
Tabel 4. 66 Rincian Tabel m_project_location	46
Tabel 4. 67 Rincian Tabel m_route	46
Tabel 4. 68 Rincian Tabel m_warehouse	47
Tabel 4. 69 Rincian Tabel m_route_compatible_cargos	48
Tabel 4. 70 Rincian Tabel m_route_conveyors	48
Tabel 4. 71 Rincian Tabel m_route_cargo_specifier	49
Tabel 4. 72 Rincian Tabel m_signature	49
Tabel 4. 73 Rincian Tabel m_standby_cause	50
Tabel 4. 74 Rincian Tabel m_surveyor_service	50
Tabel 4. 75 Rincian Tabel m_surveyor_service_unit	51
Tabel 4. 76 Rincian Tabel m_surveyor_service_category	51
Tabel 4. 77 Rincian Tabel m_temp_pressure_indicator	51
Tabel 4. 78 Rincian Tabel m_trouble	52
Tabel 4. 79 Rincian Tabel m_productivity_tw	52
Tabel 4. 80 Rincian Tabel m_vendor_personel	52
Tabel 4. 81 Rincian Tabel m_vendor_personel_role	53
Tabel 4. 82 Rincian Tabel m_vessel	53
Tabel 4. 83 Tabel m_vessel_region	54
Tabel 4. 84 Rincian Tabel m_vessel_type	55
Tabel 4. 85 Rincian Tabel m_vessel_hatch	55
Tabel 4. 86 Rincian Tabel m_vessel_grab	55
Tabel 4. 87 Rincian Tabel m_vessel_crane	56
Tabel 4. 88 Rincian Tabel m_wms_stock_maximum	57
Tabel 4. 89 Rincian Tabel m_wms_maximum_cargo	57
Tabel 4. 90 Rincian Tabel m_warehouse_type	58
Tabel 4. 91 Rincian Tabel m_warehouse_cargo	58
Tabel 4. 92 Rincian Tabel m_wms_pillars	58
Tabel 4. 93 Rincian Tabel m_training	59
Tabel 4. 94 Rincian Tabel t_training	60

Tabel 4. 95 Rincian Tabel t_training_participant	60
Tabel 4. 96 Hasil Uji Coba Fitur <i>Website</i>	85



UNIVERSITAS
MA CHUNG

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan sistem informasi pada saat ini sangat cepat dan pesat, dengan menggunakan sistem informasi dapat mempermudah dan membantu dalam bekerja. Namun demikian, masih banyak perusahaan yang belum secara maksimal memanfaatkan teknologi informasi. Akibatnya, pengelolaan data yang dimiliki perusahaan masih dilakukan secara manual sehingga menyebabkan pemborosan waktu dan tenaga (Anitasari, 2021).

Salah satu bentuk teknologi sistem informasi yang mudah dikembangkan adalah berbasis *website*, media informasi berbasis jaringan komputer yang terintegrasi dengan internet. Sistem informasi berbasis *website* tidak sekedar hanya untuk menampilkan sebuah informasi saja. Namun dapat digunakan untuk mengolah data, seperti menambah, merubah, bahkan menghapus sebuah data. Sistem informasi berbasis *website* merupakan sebuah sarana yang telah dilengkapi dengan fitur-fitur dan didesain sedemikian rupa sesuai dengan kebutuhan yang akan digunakan untuk mengolah suatu data tertentu yang bertujuan untuk mempermudah, mempercepat dan mengakuratkan data yang diolah.

PT Patrakim Timur merupakan salah satu perusahaan yang sudah menerapkan sistem informasi berbasis *website*. Perusahaan ini bergerak di bidang pertanian dengan memproduksi berbagai macam pupuk dan bahan kimia untuk solusi agroindustri. PT Patrakim Timur bekerja sama dengan Penyelenggara Pelabuhan Gresik untuk mengelola terminal untuk kepentingan sendiri di dalam Daerah Lingkungan Kerja dan Daerah Lingkungan Kepentingan Pelabuhan Gresik guna menunjang kegiatan usaha di bidang industri pupuk buatan. Namun seiring dengan berjalanya waktu banyak fitur yang harus dikembangkan atau ditambah untuk menyesuaikan dengan kebutuhan yang dimiliki perusahaan dan perlu dilakukan perawatan secara berkala untuk meminimalisir *bug* atau *error* pada *website*.

Pada laporan ini akan dibahas lebih detail mengenai bagian apa saja yang penulis kembangkan pada *website* sistem informasi pelabuhan kapal PT

Patrakim Timur sehingga diharapkan dapat membantu dan mempermudah perusahaan dalam melakukan pengolahan data yang dimiliki perusahaan.

1.2 Batasan Masalah

Batasan masalah dalam pengembangan *website* ini adalah sebagai berikut:

- a. Pengembangan *website* menggunakan *framework* Royal.
- b. Metode yang digunakan adalah *System Development Life Cycle* (SDLC).
- c. Pengembangan *website* meliputi bagian halaman *Master Data*, *Training Realization*, dan *Training & Sharing Knowledge Summary*.
- d. *Database* yang digunakan pada *website* adalah PostgreSQL.

1.3 Rumusan Masalah

Berdasarkan latar belakang permasalahan tersebut, maka rumusan masalahnya adalah bagaimana mengembangkan sebuah *website* sistem informasi pelabuhan kapal PT Patrakim Timur yang dapat membantu pihak perusahaan agar dapat dengan mudah mengelola data yang dimiliki.

1.4 Tujuan

Tujuan mengembangkan sebuah *website* sistem informasi pelabuhan kapal PT Patrakim Timur adalah untuk membantu pihak perusahaan agar dapat dengan mudah mengelola data secara efisien dan efektif yang dapat mendukung serta meningkatkan proses bisnis yang telah berjalan.

1.5 Manfaat

Manfaat yang diperoleh dalam pelaksanaan Praktik Kerja Lapangan (PKL) ini adalah sebagai berikut:

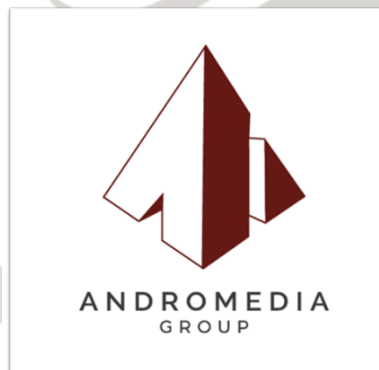
1. Bagi PT Patrakim Timur, manfaat yang didapatkan adalah mempermudah mengelola informasi atau data yang dapat mendukung serta meningkatkan proses bisnis yang telah berjalan.
2. Bagi penulis, manfaat yang didapatkan adalah kesempatan terjun langsung ke dunia kerja untuk memperoleh ilmu serta mengasah kemampuan dalam mengembangkan sebuah *website*.
3. Bagi Universitas Ma Chung khususnya Program Studi Teknik Informatika, manfaat yang didapatkan adalah sebagai sarana kerjasama dengan PT Andromedia dalam menyusun kerja praktik, serta mempersiapkan lulusan mahasiswa yang kompeten dengan pengalaman di dunia kerja.

BAB II

GAMBARAN UMUM PERUSAHAAN

2.1 PT Andromedia

PT Andromedia adalah perusahaan yang bergerak di bidang IT *consulting*, *software development*, dan implementasi ERP (*Enterprise Resource Planning*) dengan karyawan berjumlah kurang lebih sebanyak 64 karyawan yang bekerja secara *full time*. PT Andromedia didirikan pada tahun 2008 oleh A. Ghofur yang merupakan alumni mahasiswa jurusan Sistem Informasi dari Universitas Institut Teknologi Sepuluh Nopember (ITS) tahun 2001. Perusahaan ini terletak di Jl. Jemur Andayani XV No.3B, Jemur Wonosari, Kec. Wonocolo, Kota SBY, Jawa Timur 60237. Kemudian, dapat dilihat pada gambar 2.1 merupakan logo perusahaan PT Andromedia yang didesain menyerupai bentuk huruf “A” dengan arti “Andromedia” yang membuat logo ini terkesan identik dengan identitas perusahaan.

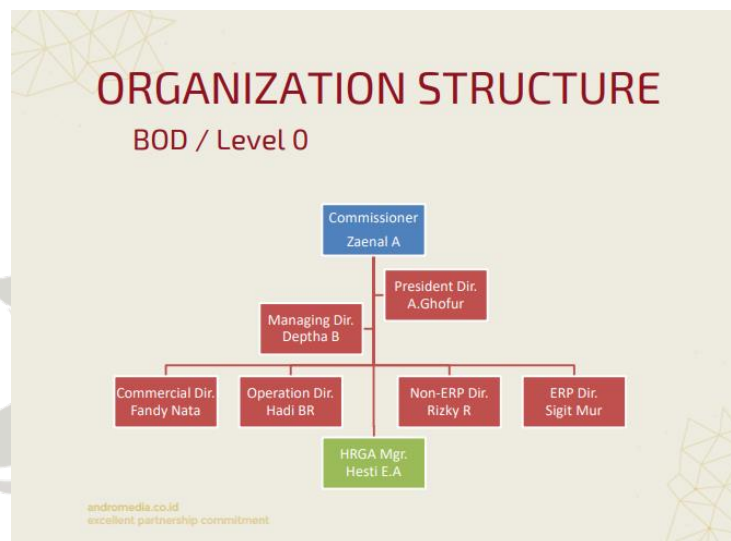


Gambar 2. 1 Logo PT Andromedia

Bisnis utama yang dijalankan oleh PT Andromedia dibagi menjadi dua, yaitu ERP (*Enterprise Resource Planning*) dan *Taylorred Software*. Kemudian, untuk ERP di PT Andromedia menggunakan *software* iDempiere yaitu *software* yang berbasis ERP yang digunakan sebagai alat untuk membantu manajemen perusahaan, perencanaan, hingga mengolah data secara efektif dan efisien, sedangkan untuk *Taylorred Software*, mengembangkan *software* dari 0 dengan menggunakan Java, PHP, Microsoft Sharepoint, dst. Bisa juga dengan menyesuaikan permintaan dari klien.

2.2 Struktur Organisasi

Sebuah sistem yang disebut struktur organisasi digunakan untuk mendefinisikan hierarki di dalam organisasi dengan tujuan untuk menentukan bagaimana setiap organisasi harus berfungsi dan membantu organisasi dalam mencapai tujuan yang ditetapkan pada periode sebelumnya. Misalnya, bagaimana pekerjaan dikelompokkan, dibagi, dan dikomunikasikan secara formal. Berikut ini adalah struktur organisasi di PT Andromedia:



Gambar 2. 2 Struktur Organisasi PT Andromedia

Pada gambar 2.2 dapat dilihat bahwa *Commissioner* merupakan jabatan tertinggi yang ada di PT Andromedia. Kemudian, *Commissioner* membawahi *President Dir.* *President Dir.* membawahi *Managing Dir.* yang dimana *Managing Dir.* membawahi 4 bagian yaitu *Commercial Dir.*, *Operation Dir.*, *Non-ERP Dir.*, dan *ERP Dir.*

2.3 Program Magang Divisi IT

PT. Andromedia merupakan perusahaan yang bergerak dibidang teknologi mulai dari mengembangkan, mengimplementasikan, mengintegrasikan, dan meningkatkan sistem informasi berupa Aplikasi *Web* dan *Mobile*. Peserta diberi kesempatan pembelajaran magang melalui berbagai posisi, yaitu *Business Analyst*, *UI/UX Design*, dan *Full Stack Developer*. Program magang di PT Andromedia dilaksanakan selama kurang lebih 5 Bulan mulai tanggal 18 Agustus 2022 – 31 Desember 2022.

2.4 Deskripsi Lingkup Kerja *Full-Stack Developer*

Pelaksanaan program magang di PT Andromedia, dimulai pada tanggal 18 Agustus 2022 - 31 Desember 2022. Jadwal kegiatan magang adalah hari senin-jumat mulai pukul 09.00 WIB - 16.00 WIB. Berikut merupakan *timeline* kegiatan magang di PT Andromedia:

2.4.1 *On Boarding* (18 Agustus – 19 Agustus 2022)

On Boarding diselenggarakan oleh pihak Kampus Merdeka untuk pelepasan peserta angkatan 2 dan penyambutan angkatan 3.

2.4.2 Pembekalan Materi dan *Mini Challenge* (22 Agustus – 30 September 2022)

Melakukan pengerjaan *mini challenge* yang telah diberikan oleh mentor untuk pelatihan skill selama kurang lebih satu bulan dengan pendampingan mentor. Target pencapaian adalah penyelesaian *mini challenge* yang telah diberikan dan pemahaman konsep *library* React.js, *Framework* Tailwind CSS, dan OpenCv.js.

2.4.3 Pembekalan *Mini Project* (03 Oktober – 21 Oktober 2022)

Melakukan *cloning project* salah satu klien PT Andromedia yang berbasis *website*. Pengembangan *website* dilakukan dengan menggunakan *framework* Royal dan OpenCv.js.

2.4.4 *Project Magang Full-Stack Developer* (24 Oktober – 31 Desember 2022)

Full-Stack Developer bertugas untuk mengerjakan *front-end* maupun *back-end* dari aplikasi yang dikembangkan, dan memastikan apakah tampilan aplikasi yang dibuat sudah memenuhi kebutuhan klien. Kegiatan ini merupakan implementasi materi secara nyata dengan *project* yang sudah ditentukan oleh mentor yang dikerjakan peserta magang khususnya Tim *Full-Stack Developer*. Peserta magang diminta untuk mengerjakan beberapa *task* seperti, membuat fitur tambahan, merubah *design*, dan perbaikan *bug* pada bagian *project* yang sedang dikerjakan. Selama pengerjaan *project* akan didampingi oleh mentor dari divisi *Full-Stack Developer* dan *Business Analyst*.

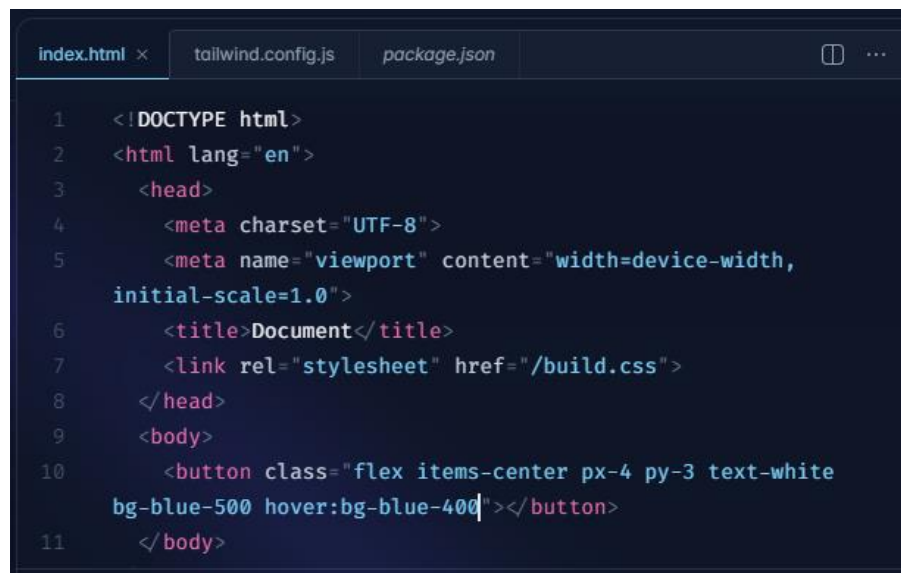
BAB III

TINJAUAN PUSTAKA

3.1 Tailwind CSS

Tailwind CSS merupakan *framework* CSS yang menawarkan konsep baru dengan konsep *utility first* di mana memiliki tujuan untuk mempermudah dalam menentukan format dan tampilan pada suatu komponen HTML tertentu (Roy, 2022). Dengan menggunakan tailwind CSS sangat memungkinkan untuk mengembangkan sebuah *website* tanpa bolak-balik dari dokumen HTML ke dokumen CSS maupun sebaliknya karena Tailwind CSS menyediakan berbagai *class* yang siap dipakai tanpa nama yang aneh dan spesifik. Contohnya seperti berikut:

- a. **Flex:** digunakan untuk menerapkan flexbox.
- b. **Items-center:** untuk menerapkan properti CSS align-items: center.
- c. **Text-white:** untuk merubah warna font menjadi putih.
- d. **Px:** untuk menerapkan properti CSS padding bagian x: kiri-kanan
- e. **Py:** untuk menerapkan properti CSS padding bagian y: atas-bawah. Serta masih banyak lagi *class utilitasnya*.



```
index.html x  tailwind.config.js  package.json  [icon] ...  
1  <!DOCTYPE html>  
2  <html lang="en">  
3    <head>  
4      <meta charset="UTF-8">  
5      <meta name="viewport" content="width=device-width,  
initial-scale=1.0">  
6      <title>Document</title>  
7      <link rel="stylesheet" href="/build.css">  
8    </head>  
9    <body>  
10     <button class="flex items-center px-4 py-3 text-white  
bg-blue-500 hover:bg-blue-400"></button>  
11   </body>
```

Gambar 3. 1 Implementasi Tailwind CSS

(Sumber: <https://tailwindcss.com/>)



```

14  .px-4 {
15    padding-left: 1rem;
16    padding-right: 1rem
17  }
18
19  .py-3 {
20    padding-top: 0.75rem;
21    padding-bottom: 0.75rem
22  }
23
24  .text-white {
25    --tw-text-opacity: 1;
26    color: rgb(255 255 255 / var(--tw-text-opacity))
27  }
28
29  .hover\:bg-blue-400:hover {
30    --tw-bg-opacity: 1;
31    background-color: rgb(96 165 250 / var(--tw-bg-opacity))
32  }
33

```

Gambar 3. 2 Penggunaan CSS tanpa Tailwind CSS

(Sumber: <https://tailwindcss.com/>)

Implementasi penggunaan *framework* Tailwind CSS dapat dilihat pada gambar 3.2.1, terlihat bahwa penggunaan Tailwind CSS lebih efektif dan lebih efisien daripada tanpa menggunakan *framework* Tailwind CSS karena membutuhkan banyak kode yang rumit serta harus membuat dokumen CSS tambahan.

3. 2 JavaScript

JavaScript merupakan bahasa pemrograman yang dirilis pada tahun 1995. Bahasa ini diciptakan oleh seorang *programmer* bernama Brandan Eich yang bertujuan untuk membuat situs dengan konten *website* yang dinamis (Puspa, 2021). Konten dinamis artinya konten dapat bergerak atau berubah di depan layar tanpa memuat ulang halaman, seperti gambar animasi, fitur *slideshow image*, mengganti warna tema halaman, dan lainnya.

Umumnya penulisan kode JavaScript dapat dilakukan dengan dua cara yaitu penulisan dengan *Embed* dan *Eksternal Line*. Penulisan dengan *Embed* memanfaatkan *tag script* untuk menyisipkan kode JavaScript ke HTML,

tepatnya dalam *tag body* atau *head*. Sedangkan *External Line* dapat dibuat dengan cara membuat dokumen JavaScript ditulis dengan ekstensi “.js” secara terpisah (Aprilia, 2022).

3. 3 TypeScript

Menurut Oktriwina (2020), TypeScript adalah bahasa pemrograman *open source* yang dibangun di atas JavaScript, dikembangkan oleh Anders Hejlsberg dari Microsoft dengan menyertakan serangkaian fitur, seperti *strong typing* dan *Object Oriented Programming* (OOP) yang sangat rapi dan kompleks untuk diterapkan. TypeScript dapat digunakan pada *framework front-end* maupun *beck-end*. Pada *front-end* TypeScript dapat digunakan untuk *framework*, seperti VueJs dan ReactJs. Kemudian, pada *beck-end* dapat digunakan untuk *framework* Node.js dan ExpressJs.

TypeScript bisa dikatakan sebagai *superset* dari JavaScript, artinya semua fitur yang ada di JavaScript juga bisa diimplementasikan di TypeScript. Namun, untuk menjalankannya, TypeScript membutuhkan kompiler yang kemudian mengubahnya menjadi dokumen JavaScript. Selain itu, TypeScript dapat berjalan di sisi klien dan juga sisi server (Bauroziq, 2022).

3. 4 React.js

Menurut Pamungkas (2020), ReactJS merupakan sebuah *library* Javascript yang digunakan untuk membuat atau membangun *User Interface* (UI) sebuah *website*. React.js dapat dianggap perpustakaan yang berisi sekumpulan kode program yang telah ditulis sebelumnya (*pre-written*), sehingga bisa digunakan tanpa membuat program tersebut dari awal.

React.js bersifat *open source* dengan mempunyai beberapa keunggulan, seperti kemampuan *reuse component*, pemuatan ulang secara *realtime*, dan ekosistem yang baik. React.js menggunakan kerangka *virtual DOM* yang membuat sebuah *website* berjalan lebih cepat. Dengan kerangka ini, antarmuka pengguna yang kompleks dapat dipecah menjadi beberapa komponen. Dengan begitu, proses pengembangan bisa berjalan lebih cepat karena memungkinkan banyak pengguna bekerja pada komponen secara bersamaan.

3. 5 Node.js

Menurut Puspasari (2022), Node.js pertama kali diciptakan dan diperkenalkan untuk pengguna pada sistem Linux pada tahun 2009 oleh Ryan Dahl. Node.js adalah platform perangkat lunak yang didesain untuk mengembangkan aplikasi berbasis web dan ditulis menggunakan bahasa pemrograman JavaScript. Node.js dibangun menggunakan *engine* JavaScript V8 milik Google yang sifatnya *open source* dan *cross platform*. Node.js bertugas untuk mengeksekusi kode JavaScript sebelum halaman website ditampilkan di browser.

3. 6 Royal

Royal adalah sebuah *framework* yang dikembangkan oleh perusahaan PT Andromedia yang berbasis pada bahasa pemrograman Typescript menggunakan *database* PostgreSQL. *Framework* Royal dapat digunakan *developer* untuk mengembangkan *website* baik *front-end* maupun *beck-end*. *Framework* ini memiliki struktur yang mencakup implementasi React.js, Node.js, Tailwind CSS, dan Prisma.

3. 7 Database

Database atau basis data adalah kumpulan data yang terorganisir atau terstruktur. *Database* dapat diakses dan disimpan pada sistem komputer. Untuk mengelola *database*, dibutuhkan prangka lunak *Database Management System* (DBMS). Dalam *database*, data akan diatur dalam tabel yang memiliki baris dan kolom untuk mendefinisikan jenis informasi apa yang harus disimpan. Data akan diberi indeks agar dapat diperbarui, dihapus, atau diperluas dengan mudah.

3. 8 PostgreSQL

Menurut Iswara (2022), PostgreSQL atau Postgres adalah *Object Relational Database Management System* (ORDBMS) yang bersifat *open source*, dikembangkan oleh Michael Stonebraker dari Universitas California di Berkley. PostgreSQL juga menyediakan dukungan untuk berbagai bahasa pemrograman, seperti C, C ++, Python, Java, dan banyak lagi lainnya, sehingga mudah digunakan dalam berbagai proyek. Sistem manajemen *database* ini memiliki performa stabil, keamanan tinggi, serta fitur melimpah.

PostgreSQL cocok untuk mengembangkan sistem dalam skala besar yang membutuhkan *database* untuk mengeksekusi berbagai query kompleks, seperti *Geographical Information System* (GIS), dan *Business Intelligence System* (BIS), Fungsi utama PostgreSQL adalah tempat untuk menyimpan dan mengelola data melalui perintah atau *query* SQL seperti:

- Membuat tabel atau memanipulasi tabel dengan *Data Definition Language* (DDL) menggunakan *query* seperti *create*, *drop*, *alter*.
- Memanipulasi data atau nilai dari tabel dengan *Data Manipulation Language* (DML) menggunakan *query* seperti *insert*, *update*, *delete*.
- Mengelola *database* dengan *Data Control Language* (DCL) menggunakan *query* seperti *grant*, *revoke*, *commit*.

3. 9 Adminer

Menurut Priatna (2022), Adminer adalah *tool* untuk mengelola *database* yang dibuat menggunakan bahasa PHP dengan berbagai fitur yang mendukung, seperti *Data Definition Language* (DDL), *Data Manipulation Language* (DML) dan SQL *command* lainnya, *security* yang ketat, fitur *export*, *import* dan banyak fitur lainnya. Selain untuk mengelola *database* MariaDB atau MySQL, Adminer dapat digunakan untuk mengelola *database* lain, seperti PostgreSQL, SQLite, MS SQL, Oracle, Elasticsearch, MongoDB dan lain-lain.

3. 10 Object Relational Mapping (ORM)

Object Relational Mapping (ORM) merupakan metode untuk merubah suatu tabel menjadi sebuah *object* dengan *field property* yang sama seperti pada tabel sehingga memungkinkan untuk melakukan *query* dan memanipulasi data di *database* menggunakan *object oriented* (Agnes, 2020). Salah satu keunggulan ORM adalah mempermudah programmer agar tidak perlu lagi menuliskan perintah-perintah SQL ketika memerlukan koneksi dengan *database*, tetapi cukup dengan menggunakan fungsi-fungsi yang sudah disediakan oleh ORM tersebut.

ORM akan membantu sebagai jembatan antara *object* yang didefinisikan dalam kode program dengan *database*. Dengan ORM ini, programmer dibantu untuk melakukan aksi-aksi yang diperlukan terkait komunikasi *object* ketika

program dijalankan dengan *database* seperti bagaimana *object* itu disimpan, diambil, dihapus, diubah dan sebagainya.

Berikut ini adalah contoh penggunaan ORM dengan tabel Mahasiswa,

Jika menggunakan ORM:

```
Person.all  
Person.find(1)
```

Tanpa Menggunakan ORM:

```
SELECT * FROM Mahasiswa  
SELECT * FROM Mahasiswa WHERE id = 1
```

3. 11 Prisma

Prisma adalah ORM *open source* untuk Node.js dan TypeScript. Penggunaan Prisma memungkinkan untuk mengakses *database* melalui metode dan objek JavaScript biasa tanpa harus menulis kueri dalam bahasa *database* itu sendiri sehingga lebih mudah untuk menulis aplikasi CRUD (*create, update, edit, dan delete*) (Prasad, 2021). Prisma saat ini mendukung pembuatan *database* sendiri seperti PostgreSQL, MySQL, SQL Server, SQLite, MongoDB dan CockroachDB (Pratinjau). Meskipun Prisma dapat digunakan dengan JavaScript biasa, Prisma lebih disarankan untuk menggunakan TypeScript dengan tujuan memberikan tingkat jenis keamanan yang dapat melampaui ORM lain.

3. 12 Version Control System (VCS)

Version Control System (VCS) merupakan sebuah sistem yang mencatat atau menginformasikan setiap perubahan terhadap sebuah dokumen atau struktur kode sehingga pada suatu saat seorang *developer* dapat kembali kepada salah satu versi dari berkas atau dokumen yang diinginkan (sebelum dirubah). Pengembangan sebuah sistem atau *software* secara kolaboratif juga didukung oleh VCS, setiap anggota tim *development* dapat menulis atau merubah kode programnya yang kemudian akan digabungkan menjadi satu menggunakan VCS. Tempat penyimpanan atau *database* untuk kode pada VCS dikenal dengan sebutan *repository*. Selain itu terdapat istilah seperti *commit, push, fetch*, dll.

3. 13 Git

Menurut Muhandian (2017), Git adalah tipe *distributed* VCS yang dapat digunakan secara lokal ataupun menggunakan server, artinya *database* Git tidak hanya disimpan dalam satu tempat saja, semua orang yang terlibat dalam pengembangan sebuah proyek akan menyimpan *database* Git, sehingga akan memudahkan dalam mengelola proyek baik secara *online* maupun *offline*.

Saat ini, Git merupakan salah satu *software open source* yang sering dipakai para *developer* dalam mengembangkan sebuah proyek. Dengan menggunakan Git, *programmer* akan sangat terbantu karena Git akan mencatat setiap perubahan dokumen baik itu kode, *comment*, dan lain-lain. Beberapa contoh server yang dapat digunakan untuk git adalah Github, Gitlab, dan Bitbucket.

3. 14 Enterprise Resource Planning (ERP)

Enterprise Resource Planning (ERP) adalah jenis sistem perangkat lunak yang membantu organisasi mengotomatisasi dan mengelola proses bisnis inti untuk kinerja yang optimal. Perangkat lunak ERP mengoordinasikan aliran data di antara proses bisnis perusahaan dan merampingkan operasi di seluruh perusahaan. Perangkat lunak ini mampu menghubungkan aktivitas keuangan, rantai pasokan, operasional, perdagangan, pelaporan, manufaktur, dan sumber daya manusia perusahaan dalam satu sistem (Mahatmavidya, 2022).

ERP merupakan aset dalam hal perencanaan dan koordinasi. Karyawan dapat melihat inventaris yang tersedia saat ini dan pesanan pelanggan secara rinci, kemudian membandingkan pesanan pembelian pemasok dan perkiraan permintaan di masa mendatang. Jika perlu, mereka dapat melakukan penyesuaian untuk mengatasi masalah. Perangkat lunak ERP juga meningkatkan komunikasi dan kolaborasi karena para pekerja dapat memeriksa status departemen lain untuk membantu pengambilan tindakan. Sistem ERP juga menyediakan sejumlah laporan dan analisis yang dapat menjadi penentu keputusan bisnis (McCue, 2022).

3. 15 Unified Modelling Language (UML)

Unified Modelling Language (UML) merupakan suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem,

desain sistem dapat diwujudkan dalam bentuk simbol dan diagram. Pada UML terdapat 9 diagram, yang terbagi atas 4 kategori yaitu sudut pandang permintaan, sudut pandang logikal, sudut pandang dinamis, dan sudut pandang fisikal (Juliarto, 2021).

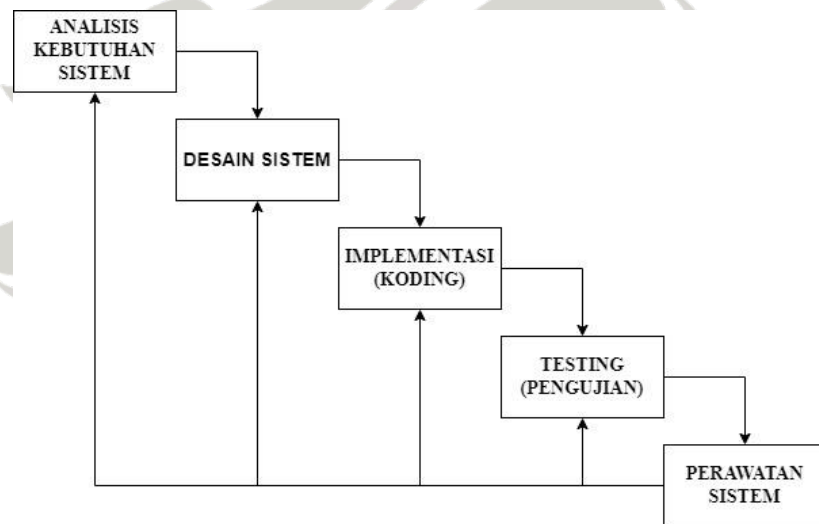
Tabel 3. 1 Sudut Pandang Diagram UML

Sudut Pandang	Nama Diagram	Deskripsi
Permintaan	<i>Use Case</i>	Menunjukkan sekumpulan actor dan use case, serta hubungan di antara keduanya. Diagram use case berkontribusi kepada model pengorganisasian yang lebih efektif dengan memperlihatkan perilaku sistem.
Logikal	<i>Class</i>	Menggambarkan kelas, paket, dan hubungan antara aspek dari sistem.
	<i>Object</i>	Menjelaskan hubungan statis yang ada antara objek-objek.
Dinamis	<i>Collaboration</i>	Menjelaskan urutan organisasi dari objek yang mengirim dan menerima pesan. Ekuivalen dengan diagram <i>sequence</i> .
	<i>Sequence</i>	Menjelaskan pesan-pesan berbasis waktu yang dikirim tiap objek. Ekuivalen dengan diagram <i>collaboration</i> .
	<i>State</i>	Menjelaskan perilaku yang berhubungan dengan keadaan dari objek. Transisi dari keadaan membantu mengidentifikasi perilaku yang kompleks.
	<i>Activity</i>	Menjelaskan aliran aktifitas antars proses. Berguna untuk merinci perilaku <i>use case</i> .
Fisikal	<i>Component</i>	Menjelaskan hubungan statik antara komponen perangkat lunak yang ada.
	<i>Deployment</i>	Digunakan pada bagian awal proses perancangan sistem untuk mendokumentasikan arsitektur fisik sebuah sistem.

BAB IV

HASIL DAN PEMBAHASAN

Pengembangan *website* PT Patrakim Timur, dimulai dari tahapan analisis kebutuhan sistem, desain sistem, implementasi, *testing*, dan perawatan sistem. Pada tahap *testing* akan di uji apakah sebuah sistem yang dikembangkan sudah berjalan sesuai analisis, desain, dan implementasi di sistem. Kemudian tahap terakhir adalah melakukan perawatan sistem dengan tujuan untuk menjaga agar sistem tetap beroperasi dengan optimal. Model *waterfall* pengembangan *website* ditunjukkan oleh gambar 4. 1



Gambar 4. 1 Model *Waterfall*

4.1 Analisis Kebutuhan Sistem

Proses pengembangan *website* dalam kegiatan Praktik Kerja Lapangan dimulai dengan tahap analisis kebutuhan sistem. Analisis kebutuhan dilakukan dengan cara pemberian tugas oleh mentor *Full-Stack Developers* dan *Business Analyst* PT Andromedia kepada peserta magang. Mentor akan memberikan penugasan lewat Excel kemudian peserta magang dapat bertanya kepada mentor apabila masih kurang jelas mengenai penugasan yang telah diberikan sebelumnya. Beberapa tugas yang diberikan adalah sebagai berikut:

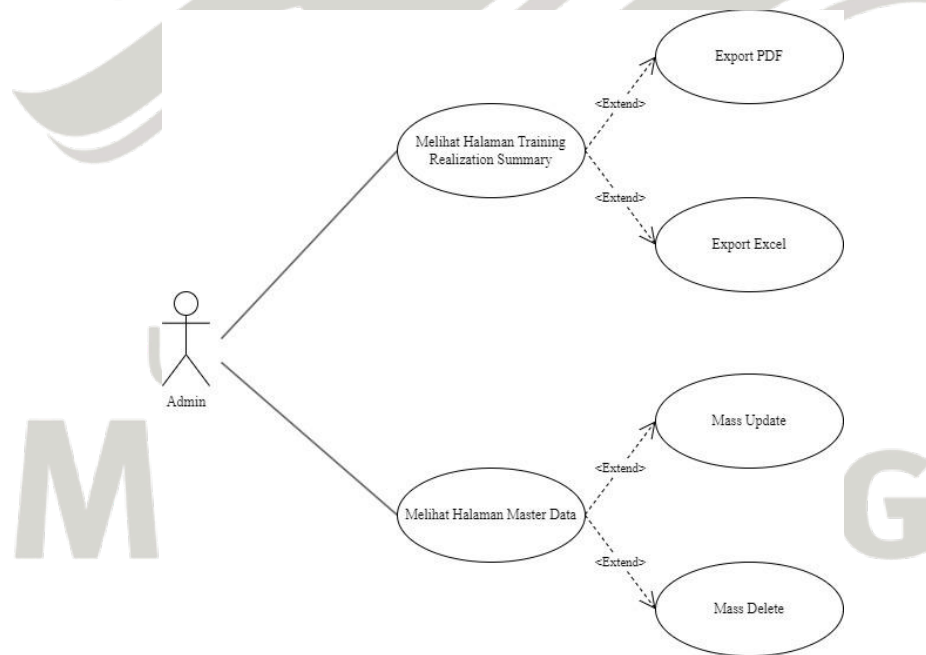
1. Penambahan halaman baru *Summary Training Realization* yang dilengkapi dengan fitur *export* PDF & Excel.

2. Melakukan perbaikan pada salah satu input di halaman *Training Realization* yaitu input *Start Date* karena tidak sesuai dengan data tahun *Master Data Training*.
3. Penambahan fitur mass update & delete pada halaman *Master Data*.

4.3 Desain Sistem

4.3.1 Use Case Diagram

Use case adalah interaksi atau hubungan yang saling berkaitan antara aktor dengan sistem. *Use case* merupakan teknik yang digunakan ketika mengembangkan perangkat lunak untuk mengetahui kebutuhan fungsional dari suatu sistem yang dirancang. *Use case diagram* merupakan proses untuk menggambarkan untuk merepresentasikan hubungan atau relasi antara aktor atau pengguna dengan sistem yang dirancang. Penggunaan *use case diagram* ditunjukkan pada gambar 4. 2



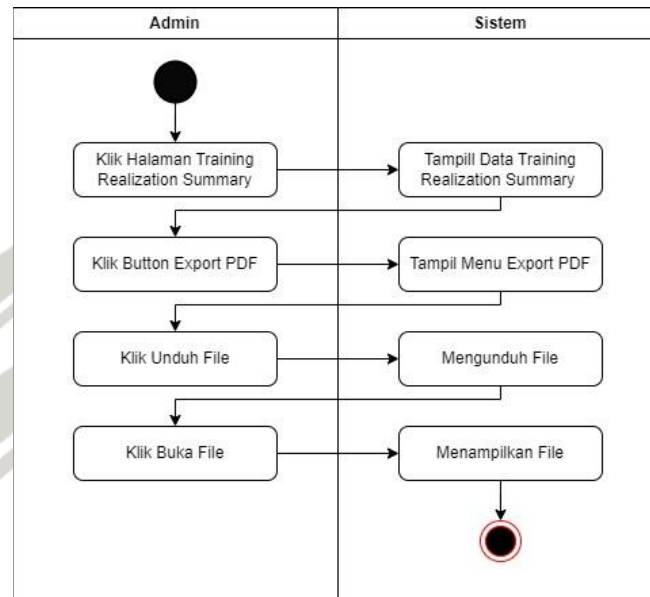
Gambar 4. 2 Use Case Diagram

Gambar 4. 2 menunjukkan *use case diagram* halaman baru *Training Realization Summary* beserta fitur *export PDF & Excel* dan halaman *Master Data* beserta fitur *mass update & delete*.

4.3.2 Activity Diagram Halaman Training Realization Summary

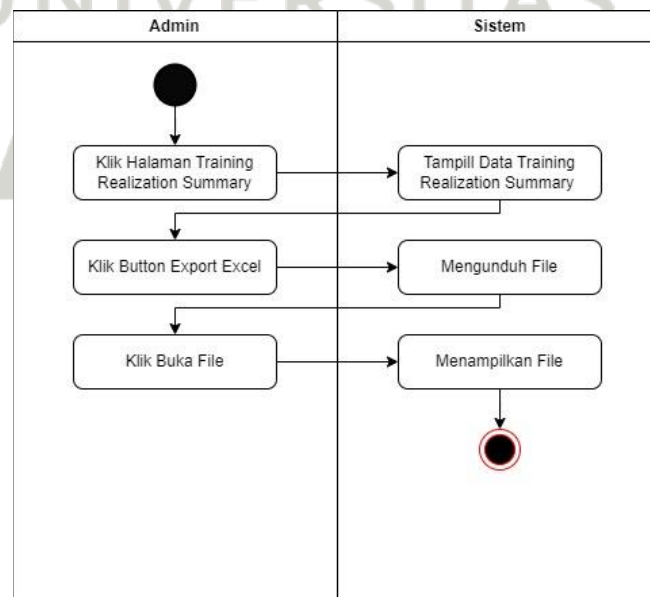
Activity diagram halaman *Training Realization Summary* merupakan gambaran dari aliran kerja/workflow dari aktifitas dalam sistem yang dilakukan oleh admin. Berikut gambaran dari aktifitas halaman *Training Realization Summary*.

1. Activity diagram export PDF



Gambar 4. 3 Activity diagram export PDF

2. Activity diagram export Excel

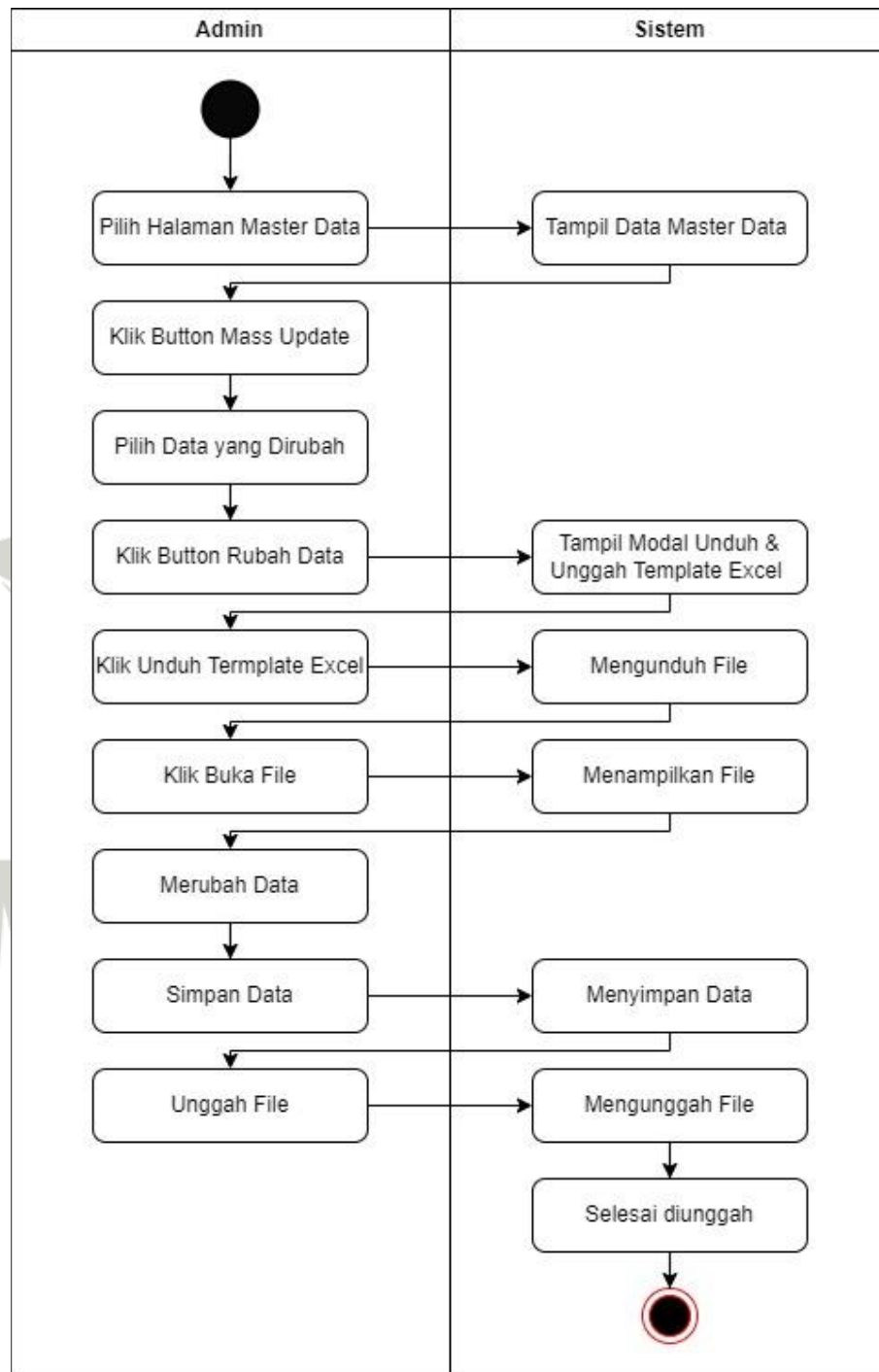


Gambar 4. 4 Activity diagram export Excel

4.3.3 Activity Diagram Fitur Mass Update & Delete Halaman Master Data

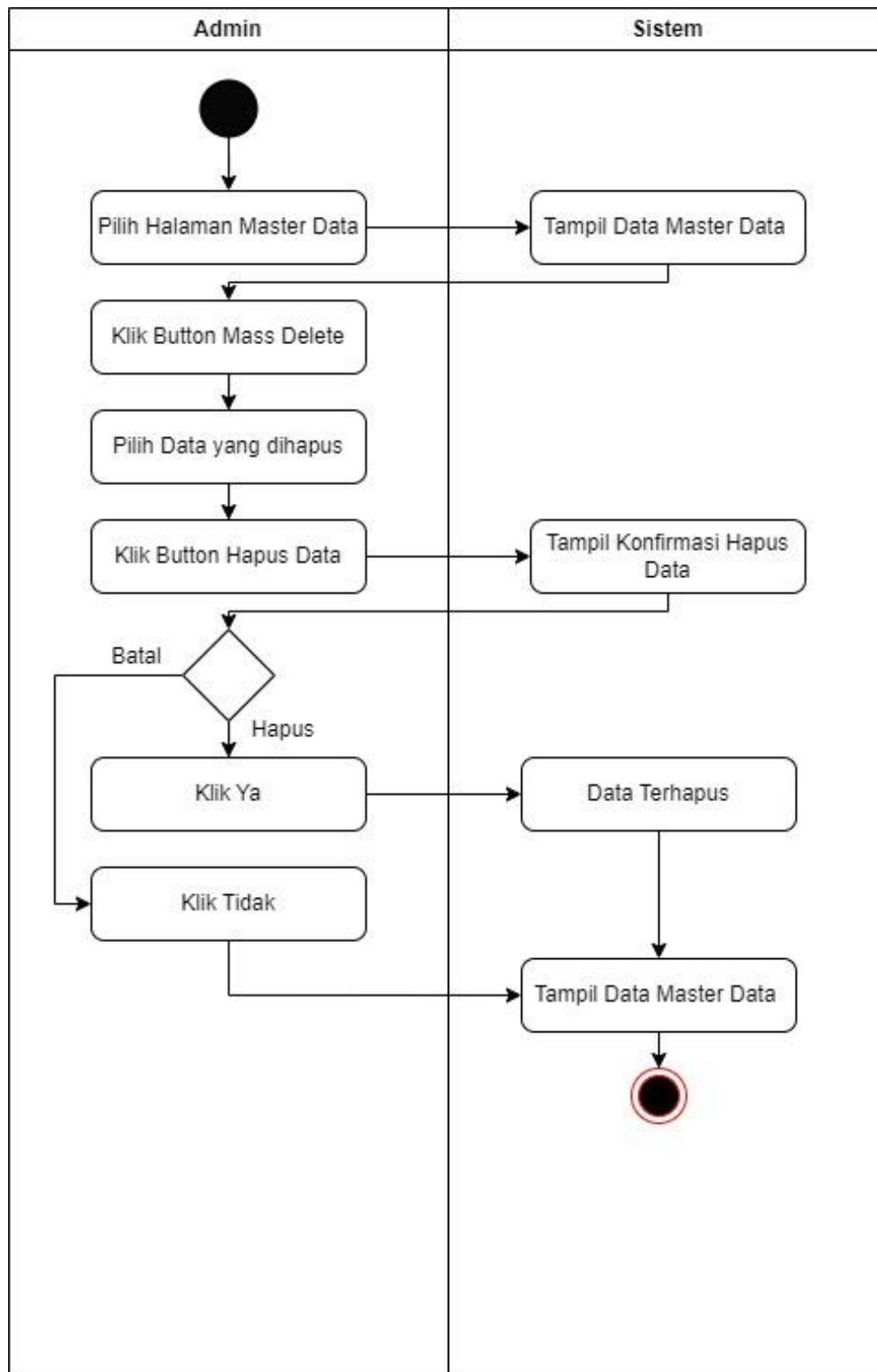
Berikut aliran kerja/*workflow* dari fitur yang telah dibuat pada halaman *Master Data*.

1. Activity diagram mass update



Gambar 4. 5 Activity diagram mass update

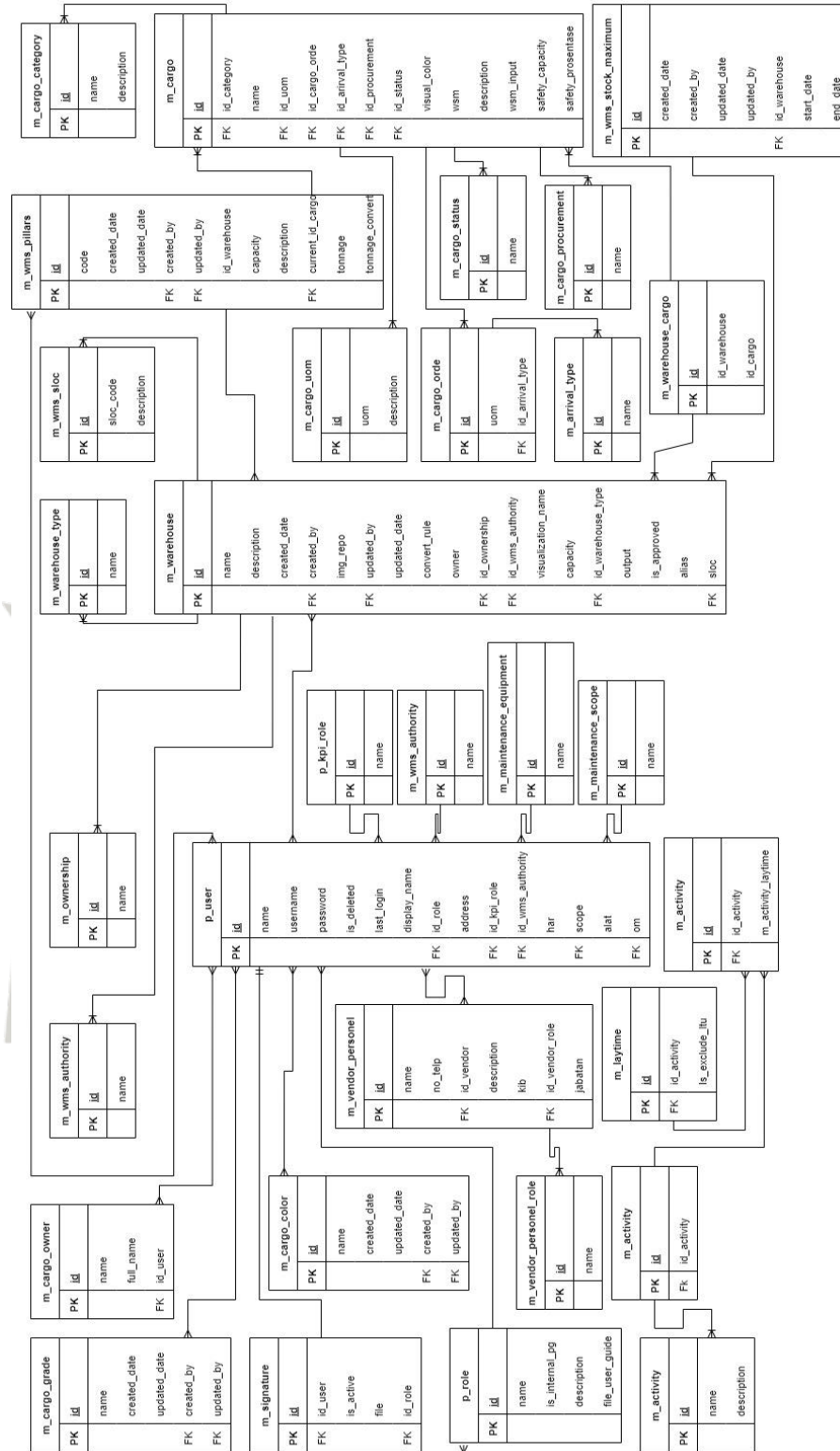
2. Activity diagram mass delete



Gambar 4. 6 Activity diagram mass delete

4.3.4 *Entity Relationship Diagram (ERD)*

Berikut adalah ERD dari bagian website yang dikerjakan.



Gambar 4. 7 *Entity Relationship Diagram* Bagian 1



Gambar 4. 8 Entity Relationship Diagram Bagian 2

4.3.5 Database

Database yang digunakan untuk mengembangkan proyek ini adalah PostgreSQL. Terdapat 98 tabel yang digunakan untuk implementasi pengembangan *website* yang dikerjakan oleh penulis. Masing-masing dari tabel memiliki atribut dan tipe data yang telah disesuaikan dengan

kebutuhan pengguna untuk mengelola data yang telah dimiliki. Berikut adalah rincian atribut dan tipe data dari tabel-tabel tersebut:

1. Tabel m_cargo_color

Tabel m_cargo_color utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Cargo Color*. Rincian dari tabel m_cargo_color dapat dilihat dari tabel 4.1.

Tabel 4. 1 Rincian tabel m_cargo_color

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-
created_date	timestampz	-	-
created_by	integer	foreign key	p_user(id)
updated_date	timestampz (null)	-	-
updated_by	integer (null)	foreign key	p_user(id)

2. Tabel m_cargo_grade.

Tabel m_cargo_grade utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Cargo Grade*. Rincian dari tabel m_cargo_grade dapat dilihat dari tabel 4.2.

Tabel 4. 2 Rincian Tabel m_cargo_grade

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-
created_date	timestampz	-	-
created_by	integer	foreign key	p_user(id)
updated_date	timestampz (null)	-	-
updated_by	integer (null)	foreign key	p_user(id)

3. Tabel m_pg_personel_unit

Tabel m_pg_personel_unit utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Hard File*. Rincian dari tabel m_pg_personel_unit dapat dilihat dari tabel 4.3.

Tabel 4. 3 Rincian Tabel m_pg_personel_unit

Atribut	Tipe Data	Keterangan	Target
id	integer	primary	-
name	character varying	-	-

4. Tabel m_unit

Tabel m_unit utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Unit*. Rincian dari tabel m_unit dapat dilihat dari tabel 4.4.

Tabel 4. 4 Rincian Tabel m_unit

Atribut	Tipe Data	Keterangan	Target
id	integer	primary	-
name	character varying	-	-

5. Tabel m_activity

Tabel m_activity utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Activity*. Rincian dari tabel m_activity dapat dilihat dari tabel 4.5.

Tabel 4. 5 Rincian Tabel m_activity

Atribut	Tipe Data	Keterangan	Target
id	integer	primary	-
name	character varying	-	-
description	text (null)	-	-

6. Tabel m_area

Tabel m_area utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Area* (Op). Rincian dari tabel m_area dapat dilihat dari tabel 4.6.

Tabel 4. 6 Rincian Tabel m_area

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

Tabel 4. 6 Rincian Tabel m_area (Lanjutan)

Atribut	Tipe Data	Keterangan	Target
description	text (null)	-	-
timestamp	timestampz (null)	-	-

7. Tabel m_asset

Tabel m_asset utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Asset*. Rincian dari tabel m_asset dapat dilihat dari tabel 4.7.

Tabel 4. 7 Rincian Tabel m_asset

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-
description	text (null)	-	-
status	character varying	-	-
id_type	integer	foreign key	m_asset_type(id)
img_repo	character varying (null)	-	-
device_code	character varying (null)	-	-
last_status_operator	character varying (null)	-	-
meter_start	integer (null)	-	-
meter_end	integer (null)	-	-
has_vessel_length_limit	boolean (null)	-	-
id_port	integer	foreign key	m_port(id)
coords	jsonb (null)	-	-
id_jetty	integer	foreign key	m_jetty(id)
id_status	integer	foreign key	m_asset_status(id)
id_ownership	integer	foreign key	m_ownership(id)
is_parallelable	boolean (null)	-	-

Tabel 4. 7 Rincian Tabel m_asset (Lanjutan)

Atribut	Tipe Data	Keterangan	Target
wsm	boolean (null)	-	-
secondary_asset	integer (null)	-	-
area	integer (null)	-	-

8. Tabel m_port

Tabel m_port utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Port*. Rincian dari tabel m_port dapat dilihat dari tabel 4.8.

Tabel 4. 8 Rincian Tabel m_port

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	text	-	-
img	text (null)	-	-
height	integer (null)	-	-

9. Tabel m_jetty

Tabel m_jetty utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Jetty*. Rincian dari tabel m_jetty dapat dilihat dari tabel 4.9.

Tabel 4. 9 Rincian Tabel m_jetty

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-
length	double precision [0]	-	-
width	double precision [0]	-	-
capacity	double precision [0]	-	-
max_draught	double precision [0]	-	-
id_port	integer	foreign key	m_port(id)
coords	jsonb (null)	-	-
meter_start	integer [0]	-	-

10. Tabel m_asset_type

Tabel m_asset_type utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Asset Type*. Rincian dari tabel m_asset_type dapat dilihat dari tabel 4.10.

Tabel 4. 10 Rincian Tabel m_asset_type

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

11. Tabel m_asset_status

Tabel m_asset_status digunakan untuk menyimpan data pada kolom Status di halaman *Master Asset*. Rincian dari tabel m_asset_status dapat dilihat dari tabel 4.11.

Tabel 4. 11 Rincian Tabel m_asset_status

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

12. Tabel m_ownership

Tabel m_ownership digunakan untuk menyimpan data pada kolom *Ownership* di halaman *Master Asset* dan *Owner* di halaman *Master Warehouse*. Rincian dari tabel m_ownership dapat dilihat dari tabel 4.12.

Tabel 4. 12 Rincian Tabel m_ownership

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

13. Tabel m_asset_items

Tabel m_asset_items utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Asset Item*. Rincian dari tabel m_asset_items dapat dilihat dari tabel 4.13.

Tabel 4. 13 Rincian Tabel m_asset_items

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
id_asset	integer	foreign key	m_asset(id)
name	character varying	-	-
id_work_center	integer	foreign key	m_work_center(id)
id_area	integer	foreign key	m_area(id)

14. Tabel m_work_center

Tabel m_work_center utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Work Center*. Rincian dari tabel m_work_center dapat dilihat dari tabel 4.14.

Tabel 4. 14 Rincian Tabel m_work_center

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-
description	character varying (null)	-	-

15. Tabel m_asset_item_troubles

Tabel m_asset_item_troubles digunakan untuk menyimpan data *Troubles* di halaman *Master Asset Item*. Rincian dari tabel m_asset_item_troubles dapat dilihat dari tabel 4.15.

Tabel 4. 15 Rincian Tabel m_asset_item_troubles

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
id_trouble	integer	foreign key	m_trouble(id)
id_item	integer	foreign key	m_asset_items(id)

16. Tabel m_asset_type_role

Tabel m_asset_type_role digunakan untuk menyimpan data *Access Role* di halaman *Master Asset Type*. Rincian dari tabel m_asset_type_role dapat dilihat dari tabel 4.16.

Tabel 4. 16 Rincian Tabel m_asset_type_role

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
id_role	integer	foreign key	p_role(id)
id_type	integer	foreign key	m_asset_type(id)

17. Tabel p_role

Tabel p_role digunakan untuk menyimpan keseluruhan data *role user account* pada *website*. Rincian dari tabel p_role dapat dilihat dari tabel 4.17.

Tabel 4. 17 Rincian Tabel p_role

atribut	tipe data	keterangan	target
id	integer	primary key	-
name	character varying (null)	-	-
is_internal_pg	boolean (null)[false]	-	-
description	text (null)	-	-
file_user_guide	character varying (null)	-	-

18. Tabel v_housekeeping_asset

Tabel v_housekeeping_asset utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Asset Housekeeping*. Rincian dari tabel v_housekeeping_asset dapat dilihat dari tabel 4.18.

Tabel 4. 18 Rincian Tabel v_housekeeping_asset

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
display_name	character varying (null)	-	-
total_asset	bigint (null)	-	-

19. Tabel m_vessel_crane_brands

Tabel m_vessel_crane_brands utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Brand Crane*. Rincian dari tabel m_vessel_crane_brands dapat dilihat dari tabel 4.19.

Tabel 4. 19 Rincian Tabel m_vessel_crane_brands

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

20. Tabel m_bwt_exclude

Tabel m_bwt_exclude utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master BWT Exclude*. Rincian dari tabel m_bwt_exclude dapat dilihat dari tabel 4.20.

Tabel 4. 20 Rincian Tabel m_bwt_exclude

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
id_activity	integer	foreign key	m_activity(id)

21. Tabel m_bariksama_note

Tabel m_bariksama_note utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Bariksama Note*. Rincian dari tabel m_bariksama_note dapat dilihat dari tabel 4.21.

Tabel 4. 21 Rincian Tabel m_bariksama_note

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
description	character varying (null)	-	-

22. Tabel m_contract_type

Tabel m_contract_type utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Contract Type*. Rincian dari tabel m_contract_type dapat dilihat dari tabel 4.22.

Tabel 4. 22 Rincian Tabel m_contract_type

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

23. Tabel m_cargo

Tabel m_cargo utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Cargo*. Rincian dari tabel m_cargo dapat dilihat dari tabel 4.23.

Tabel 4. 23 Rincian Tabel m_cargo

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
id_category	integer	foreign key	m_cargo_category(id)
name	character varying	-	-
id_uom	integer	foreign key	m_cargo_uom(id)
id_cargo_orde	integer	foreign key	m_cargo_orde(id)
id_arival_type	integer	foreign key	m_arrival_type(id)
id_procurement	integer	foreign key	m_cargo_procurement(id)
id_status	integer	foreign key	m_cargo_status(id)
visual_color	character varying (null)	-	-
wsm	boolean (null) [false]	-	-
description	character varying (null)	-	-
wsm_input	boolean (null) [false]	-	-
safety_capacity	double precision null	-	-
safety_prosentase	integer (null)	-	-

24. Tabel m_cargo_category

Tabel m_cargo_category utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Cargo Category*. Rincian dari tabel m_cargo_category dapat dilihat dari tabel 4.24.

Tabel 4. 24 Rincian Tabel m_cargo_category

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-
description	text (null)	-	-

25. Tabel m_cargo_uom

Tabel m_cargo_uom utamanya digunakan untuk menyimpan seluruh informasi data *Uom* pada halaman *Master Cargo*. Rincian dari tabel m_cargo_uom dapat dilihat dari tabel 4.25.

Tabel 4. 25 Rincian Tabel m_cargo_uom

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
uom	character varying	-	-
description	character varying	-	-

26. Tabel m_arrival_type

Tabel m_arrival_type utamanya digunakan untuk menyimpan seluruh informasi data *Arrival Type* pada halaman *Master Cargo*. Rincian dari tabel m_arrival_type dapat dilihat dari tabel 4.26.

Tabel 4. 26 Rincian Tabel m_arrival_type

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

27. Tabel m_cargo_status

Tabel m_cargo_status utamanya digunakan untuk menyimpan seluruh informasi data *Status Cargo* pada halaman *Master Cargo*. Rincian dari tabel m_cargo_status dapat dilihat dari tabel 4.27.

Tabel 4. 27 Rincian Tabel m_cargo_status

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

28. Tabel m_cargo_procurement

Tabel m_cargo_procurement utamanya digunakan untuk menyimpan seluruh informasi data *Procurement Cargo* pada halaman *Master Cargo*. Rincian dari tabel m_cargo_procurement dapat dilihat dari tabel 4.28.

Tabel 4. 28 Rincian Tabel m_cargo_procurement

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

29. Tabel m_cargo_color_grade

Tabel m_cargo_color_grade utamanya digunakan untuk menyimpan seluruh informasi data *Color Grade & Material Number Cargo* pada halaman *Master Cargo* dan *Master Warehouse*. Rincian dari tabel m_cargo_color_grade dapat dilihat dari tabel 4.29.

Tabel 4. 29 Rincian Tabel m_cargo_color_grade

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
id_grade	integer	foreign key	m_cargo_grade(id)
id_color	integer	foreign key	m_cargo_color(id)
id_cargo	integer	foreign key	m_cargo(id)
priority	integer (null)	-	-

Tabel 4. 29 Rincian Tabel m_cargo_color_grade (Lanjutan)

Atribut	Tipe Data	Keterangan	Target
material_number	integer (null) double	foreign key	m_wms_material_number(id)
safety_capacity	precision (null) double	-	-
safety_prosentase	precision (null)	-	-

30. Tabel m_cargo_orde

Tabel m_cargo_orde utamanya digunakan untuk menyimpan seluruh informasi data *origin* atau *destination cargo* pada halaman *Master Origin/Destination*. Rincian dari tabel m_cargo_orde dapat dilihat dari tabel 4.30.

Tabel 4. 30 Rincian Tabel m_cargo_orde

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-
id_arrival_type	integer	foreign key	m_arrival_type(id)

31. Tabel m_wms_safety_stock

Tabel m_wms_safety_stock utamanya digunakan untuk menyimpan seluruh informasi data *safety stock* pada halaman *Master Cargo*. Rincian dari tabel m_wms_safety_stock dapat dilihat dari tabel 4.31.

Tabel 4. 31 Rincian Tabel m_wms_safety_stock

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
id_cargo_color_grade	integer	foreign key	m_cargo_color_grade(id)
start_date	timestampz	-	-
end_date	timestampz	-	-

Tabel 4. 31 Rincian Tabel m_wms_safety_stock (Lanjutan)

Atribut	Tipe Data	Keterangan	Target
safety_capacity	double precision	-	-
safety_precentage	integer	-	-
id_cargo	integer(null)	foreign key	m_cargo(id)

32. Tabel m_cargo_owner

Tabel m_cargo_owner utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Cargo Owner*. Rincian dari tabel m_cargo_owner dapat dilihat dari tabel 4.32.

Tabel 4. 32 Rincian Tabel m_cargo_owner

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-
full_name	character varying (null)	-	-
id_user	integer	foreign key	p_user(id)

33. Tabel p_user

Tabel p_user utamanya digunakan untuk menyimpan seluruh informasi *user account* yang ada pada *website*. Rincian dari tabel p_user dapat dilihat dari tabel 4.33.

Tabel 4. 33 Rincian Tabel p_user

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-
username	character varying	-	-
password	integer	-	-
last_login	timestampz (null)	-	-

Tabel 4. 33 Rincian Tabel p_user (Lanjutan)

Atribut	Tipe Data	Keterangan	Target
is_deleted	boolean (null)	-	-
display_name	character varying (null)	-	-
id_role	integer	foreign key	p_role(id)
address	text (null)	-	-
id_kpi_role	integer	foreign key	p_kpi_role(id)
id_wms_authority	integer	foreign key	m_wms_authority(id)
har	text (null)	-	-
scope	integer (null)	foreign key	m_maintenance_scope(id)
alat	text (null)	-	-
om	integer (null)	foreign key	m_maintenance_equipment(id)

34. Tabel m_checksheets_type_ops

Tabel m_checksheets_type_ops utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Checksheet Type OPS*. Rincian dari tabel m_checksheets_type_ops dapat dilihat dari tabel 4.34.

Tabel 4. 34 Rincian Tabel m_checksheets_type_ops

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-
is_shift	boolean (null) [false]	-	-
sequence	smallint (null)	-	-
is_prosedur	boolean (null) [false]	-	-

35. Tabel m_currency_convert

Tabel m_currency_convert utamanya digunakan untuk menyimpan seluruh informasi data *Master Currency Convert* pada halaman Master Currency. Rincian dari tabel m_currency_convert dapat dilihat dari tabel 4.35.

Tabel 4. 35 Rincian Tabel m_currency_convert

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
amount	double precision	-	-
amount_converted	double precision	-	-
from	integer	foreign key	m_currency(id)
to	integer	foreign key	m_currency(id)

36. Tabel m_currency

Tabel m_currency utamanya digunakan untuk menyimpan seluruh informasi data *Currency* pada halaman *Master Currency*. Rincian dari tabel m_currency dapat dilihat dari tabel 4.36.

Tabel 4. 36 Rincian Tabel m_currency

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

37. Tabel m_document_type

Tabel m_document_type utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Document Type*. Rincian dari tabel m_document_type dapat dilihat dari tabel 4.37.

Tabel 4. 37 Rincian Tabel m_document_type

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

Tabel 4. 3 Rincian Tabel m_document_type (Lanjutan)

Atribut	Tipe Data	Keterangan	Target
is_required	boolean (null)	-	-
id_uploader_role	integer	foreign key	p_role(id)
id_document_region	integer	foreign key	m_document_region (id)
required_on_berthing	boolean (null) [false]	-	-
required_on_unberthing	boolean (null) [false]	-	-
id_approver_role	integer	foreign key	p_role(id)
required_on_project_management	boolean (null) [false]	-	-
required_on_wsm	boolean (null) [false]	-	-

38. Tabel m_document_region

Tabel m_document_region utamanya digunakan untuk menyimpan seluruh informasi data *Region Requirement* pada halaman *Master Document Type*. Rincian dari tabel m_document_region dapat dilihat dari tabel 4.38.

Tabel 4. 38 Rincian Tabel m_document_region

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

39. Tabel t_wms_forecast_authority

Tabel t_wms_forecast_authority utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Forecast Authority*. Rincian dari tabel t_wms_forecast_authority dapat dilihat dari tabel 4.39.

Tabel 4. 39 Rincian Tabel t_wms_forecast_authority

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
id_wms_authority	integer	foreign key	m_wms_authority

40. Tabel m_wms_authority

Tabel m_wms_authority utamanya digunakan untuk menyimpan seluruh informasi data *Authority* pada halaman *Master Forecast Authority*. Rincian dari tabel m_wms_authority dapat dilihat dari tabel 4.40.

Tabel 4. 40 Rincian Tabel m_wms_authority

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

41. Tabel m_housekeeping_group

Tabel m_housekeeping_group utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Housekeeping Group*. Rincian dari tabel m_housekeeping_group dapat dilihat dari tabel 4.41.

Tabel 4. 41 Rincian Tabel m_housekeeping_group

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

42. Tabel m_target_cargo

Tabel m_target_cargo utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Target Cargo*. Rincian dari tabel m_target_cargo dapat dilihat dari tabel 4.42.

Tabel 4. 42 Rincian Tabel m_target_cargo

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
id_cargo	integer	foreign key	m_cargo(id)
target	integer	-	-
year	integer	-	-
id_asset	integer	foreign key	m_asset(id)

43. Tabel m_holiday

Tabel m_holiday utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Great Day*. Rincian dari tabel m_holiday dapat dilihat dari tabel 4.43.

Tabel 4. 43 Rincian Tabel m_holiday

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-
date	date	-	-
multiplier	double precision (null)	-	-
count_as_holiday	boolean[false]	-	-

44. Tabel m_vessel_hatch_type

Tabel m_vessel_hatch_type utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Hatch Type*. Rincian dari tabel m_vessel_hatch_type dapat dilihat dari tabel 4.44.

Tabel 4. 44 Rincian Tabel m_vessel_hatch_type

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

45. Tabel m_isps

Tabel m_isps utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master ISPS Code*. Rincian dari tabel m_isps dapat dilihat dari tabel 4.45.

Tabel 4. 45 Rincian Tabel m_isps

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-
no	integer (null)	-	-

46. Tabel m_laytime

Tabel m_laytime utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Laytime*. Rincian dari tabel m_laytime dapat dilihat dari tabel 4.46.

Tabel 4. 46 Rincian Tabel m_laytime

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-
Is_exclude_ltu	boolean (null) [false]	-	-

47. Tabel m_activity_laytime

Tabel m_activity_laytime utamanya digunakan untuk menyimpan seluruh informasi data *Activity* pada halaman *Master Laytime*. Rincian dari tabel m_activity_laytime dapat dilihat dari tabel 4.47.

Tabel 4. 47 Rincian Tabel m_activity_laytime

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
id_activity	integer	foreign key	m_activity(id)
id_laytime	integer	foreign key	m_laytime(id)

48. Tabel m_laytime_start_count

Tabel m_laytime_start_count utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Laytime Start Count*. Rincian dari tabel m_laytime_start_count dapat dilihat dari tabel 4.48.

Tabel 4. 48 Rincian Tabel m_laytime_start_count

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
laytime	character varying	-	-
description	character varying (null)	-	-

49. Tabel m_liquid_logsheets

Tabel m_liquid_logsheets utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Liquid Logsheets*. Rincian dari tabel m_liquid_logsheets dapat dilihat dari tabel 4.49.

Tabel 4. 49 Rincian Tabel m_liquid_logsheets

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

50. Tabel m_location

Tabel m_location utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Location* (Op). Rincian dari tabel m_location dapat dilihat dari tabel 4.50.

Tabel 4. 50 Rincian Tabel m_location

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-
description	text	-	-

51. Tabel m_wms_material_number

Tabel m_wms_material_number utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Material Number*. Rincian dari tabel m_wms_material_number dapat dilihat dari tabel 4.51.

Tabel 4. 51 Rincian Tabel m_wms_material_number

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-
code	integer	-	-

52. Tabel m_wms_nopol

Tabel m_wms_nopol utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master NOPOL*. Rincian dari tabel m_wms_nopol dapat dilihat dari tabel 4.52.

Tabel 4. 52 Rincian Tabel m_wms_nopol

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
transporter_id	integer	foreign key	m_wms_transportir(id)
nopol	character varying	-	-
driver	character varying	-	-
no_kb	character varying	-	-

53. Tabel m_wms_transportir

Tabel m_wms_transportir utamanya digunakan untuk menyimpan seluruh informasi data *Transportir* pada halaman *Master NOPOL*. Rincian dari tabel m_wms_transportir dapat dilihat dari tabel 4.53.

Tabel 4. 53 Rincian Tabel m_wms_transportir

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-

Tabel 4. 53 Rincian Tabel m_wms_transportir (Lanjutan)

Atribut	Tipe Data	Keterangan	Target
name	character varying	-	-

54. Tabel m_origin

Tabel m_origin utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Origin Vessel*. Rincian dari tabel m_origin dapat dilihat dari tabel 4.54.

Tabel 4. 54 Rincian Tabel m_origin

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

55. Tabel m_pg_personel

Tabel m_pg_personel utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master PG Personel*. Rincian dari tabel m_pg_personel dapat dilihat dari tabel 4.55.

Tabel 4. 55 Rincian Tabel m_pg_personel

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-
id_position	integer	foreign key	m_pg_personel_position(id)
id_unit	integer	foreign key	m_pg_personel_unit(id)
nik	character carying	-	-
phone	character carying	-	-
email	character carying	-	-
id_warehouse_authority	integer	foreign key	m_wms_authority(id)
id_pg_personel_scope	integer	foreign key	m_pg_personel_scope(id)

56. Tabel m_pg_personel_scope

Tabel m_pg_personel_scope utamanya digunakan untuk menyimpan seluruh informasi data *Scope* pada halaman *Master PG Personel*. Rincian dari tabel m_pg_personel_scope dapat dilihat dari tabel 4.56.

Tabel 4. 56 Rincian Tabel m_pg_personel_scope

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

57. Tabel m_pg_personel_position

Tabel m_pg_personel_position utamanya digunakan untuk menyimpan seluruh informasi data *Position* pada halaman *Master PG Personel*. Rincian dari tabel m_pg_personel_position dapat dilihat dari tabel 4.57.

Tabel 4. 57 Rincian Tabel m_pg_personel_position

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-
description	text	-	-

58. Tabel m_pg_personel_unit

Tabel m_pg_personel_unit utamanya digunakan untuk menyimpan seluruh informasi data *Unit* pada halaman *Master PG Personel*. Rincian dari tabel m_pg_personel_unit dapat dilihat dari tabel 4.58.

Tabel 4. 58 Rincian Tabel m_pg_personel_unit

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

59. Tabel m_housekeeping_payrate

Tabel m_housekeeping_payrate utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Payrate*. Rincian dari tabel m_housekeeping_payrate dapat dilihat dari tabel 4.59.

Tabel 4. 59 Rincian Tabel m_housekeeping_payrate

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-
price	numeric (20,2)	-	-

60. Tabel m_housekeeping_payrate_asset

Tabel m_housekeeping_payrate_asset utamanya digunakan untuk menyimpan seluruh informasi data *Asset* pada halaman *Master Payrate*. Rincian dari tabel m_housekeeping_payrate_asset dapat dilihat dari tabel 4.60.

Tabel 4. 60 Rincian Tabel m_housekeeping_payrate_asset

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
id_payrate	integer	foreign key	m_housekeeping_payrate(id)
id_asset	integer	foreign key	m_asset(id)

61. Tabel m_project_aspek_penilaian

Tabel m_project_aspek_penilaian utamanya digunakan untuk menyimpan seluruh informasi pada halaman Master Penilaian Kinerja Rekanan. Rincian dari tabel m_project_aspek_penilaian dapat dilihat dari tabel 4.61.

Tabel 4. 61 Rincian Tabel m_project_aspek_penilaian

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
aspek_penilaian	character varying	-	-
prosentase	integer	-	-

62. Tabel m_project_permit

Tabel m_project_permit utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Permit*. Rincian dari tabel m_project_permit dapat dilihat dari tabel 4.62.

Tabel 4. 62 Rincian Tabel m_project_permit

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
aspek_penilaian	character varying	-	-
prosentase	integer	-	-

63. Tabel m_wms_plant

Tabel m_wms_plant utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Plant Code*. Rincian dari tabel m_wms_plant dapat dilihat dari tabel 4.63.

Tabel 4. 63 Rincian Tabel m_wms_plant

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
plant_code	character varying	-	-
name	character varying	-	-

64. Tabel m_wms_plant_warehouse

Tabel m_wms_plant_warehouse utamanya digunakan untuk menyimpan seluruh informasi data *Plant Warehouse* pada halaman *Master Plant Code*. Rincian dari tabel m_wms_plant_warehouse dapat dilihat dari tabel 4.64.

Tabel 4. 64 Rincian Tabel m_wms_plant_warehouse

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
id_wms_plant	integer	foreign key	m_wms_plant(id)
id_warehouse	integer	foreign key	m_warehouse(id)
id_sloc	integer	foreign key	m_wms_sloc(id)

65. Tabel m_wms_sloc

Tabel m_wms_sloc utamanya digunakan untuk menyimpan seluruh informasi data *Sloc* pada halaman *Master Plant Code*. Rincian dari tabel m_wms_sloc dapat dilihat dari tabel 4.65.

Tabel 4. 65 Rincian Tabel m_wms_sloc

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
sloc_code	character varying	-	-
description	character varying	-	-

66. Tabel m_project_location

Tabel m_project_location utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Project Location*. Rincian dari tabel m_project_location dapat dilihat dari tabel 4.66.

Tabel 4. 66 Rincian Tabel m_project_location

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
sloc_code	character varying	-	-
description	character varying	-	-

67. Tabel m_route

Tabel m_route utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Route*. Rincian dari tabel m_route dapat dilihat dari tabel 4.67.

Tabel 4. 67 Rincian Tabel m_route

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
code	character varying	-	-
id_asset	integer	foreign key	m_asset(id)
id_warehouse	integer	foreign key	m_warehouse(id)
pbm_cost	double precision	-	-
emkl_cost	double precision	-	-
other_cost	double precision	-	-
total_cost	double precision	-	-
id_asset_end	integer	foreign key	m_asset(id)
id_port	integer	-	-
id_arrival_type	integer	-	-

Tabel 4. 67 Rincian Tabel m_route (Lanjutan)

Atribut	Tipe Data	Keterangan	Target
transport_cost	double precision	-	-
id_vessel_type	integer	-	-

68. Tabel m_warehouse

Tabel m_warehouse utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Warehouse*. Rincian dari tabel m_warehouse dapat dilihat dari tabel 4.68.

Tabel 4. 68 Rincian Tabel m_warehouse

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-
description	text (null)	-	-
created_date	timestampz (null)	-	-
created_by	integer	foreign key	p_user(id)
img_repo	character varying (null)	-	-
updated_by	integer	foreign key	p_user(id)
updated_date	timestampz (null)	-	-
convert_rule	bpchar (null)	-	-
owner	text (null)	-	-
id_ownership	integer	foreign key	m_ownership(id)
id_wms_authority	integer	foreign key	m_wms_authority(id)
visualization_name	character varying (null)	-	-
capacity	double precision (null)	-	-
id_warehouse_type	integer	foreign key	m_warehouse_type(id)

Tabel 4. 68 Rincian Tabel m_warehouse (Lanjutan)

Atribut	Tipe Data	Keterangan	Target
output	boolean (null) [true]	-	-
is_approved	boolean (null) [false]	-	-
alias	character varying (null)	-	-
sloc	integer	foreign key	m_wms_sloc(id)

69. Tabel m_route_compatible_cargos

Tabel m_route_compatible_cargos utamanya digunakan untuk menyimpan seluruh informasi data *Specific Cargos* pada halaman *Master Route*. Rincian dari tabel m_route_compatible_cargos dapat dilihat dari tabel 4.69.

Tabel 4. 69 Rincian Tabel m_route_compatible_cargos

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
id_route	integer	foreign key	m_route(id)
id_cargo	integer	foreign key	m_cargo(id)

70. Tabel m_route_conveyors

Tabel m_route_conveyors utamanya digunakan untuk menyimpan seluruh informasi data *Conveyors* pada halaman *Master Route*. Rincian dari tabel m_route_conveyors dapat dilihat dari tabel 4.70.

Tabel 4. 70 Rincian Tabel m_route_conveyors

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
id_route	integer	foreign key	m_route(id)
id_asset	integer	foreign key	m_asset (id)
sequence	integer (null)	-	-

71. Tabel m_route_cargo_specifier

Tabel m_route_cargo_specifier utamanya digunakan untuk menyimpan seluruh informasi data *Specifier Route* pada halaman *Master Route*. Rincian dari tabel m_route_cargo_specifier dapat dilihat dari tabel 4.71.

Tabel 4. 71 Rincian Tabel m_route_cargo_specifier

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
id_route	integer	foreign key	m_route(id)
specifier	json	-	-
pbm_cost	double precision (null)	-	-
emkl_cost	double precision (null)	-	-
transport_cost	double precision (null)	-	-
other_cost	double precision (null)	-	-
rule_name	character varying (null)	-	-

72. Tabel m_signature

Tabel m_signature utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Signature*. Rincian dari tabel m_signature dapat dilihat dari tabel 4.72.

Tabel 4. 72 Rincian Tabel m_signature

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
id_user	integer	foreign key	p_user(id)
is_active	boolean [false]	-	-
file	character varying (null)	-	-
id_role	integer	foreign key	p_role(id)

73. Tabel m_standby_cause

Tabel m_standby_cause utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Stand by Cause*. Rincian dari tabel m_standby_cause dapat dilihat dari tabel 4.73.

Tabel 4. 73 Rincian Tabel m_standby_cause

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	text	-	-
order_num	integer (null)	-	-
type	integer	foreign key	m_asset_type(id)

74. Tabel m_surveyor_service

Tabel m_surveyor_service utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Surveyor Service*. Rincian dari tabel m_surveyor_service dapat dilihat dari tabel 4.74.

Tabel 4. 74 Rincian Tabel m_surveyor_service

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-
id_unit	integer	foreign key	m_surveyor_service_unit(id)
price	double precision	-	-
id_category	integer	foreign key	m_surveyor_service_category(id)

75. Tabel m_surveyor_service_unit

Tabel m_surveyor_service_unit utamanya digunakan untuk menyimpan seluruh informasi data *Unit* pada halaman *Master Surveyor Service*. Rincian dari tabel m_surveyor_service_unit dapat dilihat dari tabel 4.75.

Tabel 4. 75 Rincian Tabel m_surveyor_service_unit

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

76. Tabel m_surveyor_service_category

Tabel m_surveyor_service_category utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Surveyor Category*. Rincian dari tabel m_surveyor_service_category dapat dilihat dari tabel 4.76.

Tabel 4. 76 Rincian Tabel m_surveyor_service_category

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

77. Tabel m_temp_pressure_indicator

Tabel m_temp_pressure_indicator utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Temperature & Pressure Indicator*. Rincian dari tabel m_temp_pressure_indicator dapat dilihat dari tabel 4.77.

Tabel 4. 77 Rincian Tabel m_temp_pressure_indicator

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

78. Tabel m_trouble

Tabel m_trouble utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Trouble (Op)*. Rincian dari tabel m_trouble dapat dilihat dari tabel 4.78.

Tabel 4. 78 Rincian Tabel m_trouble

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

79. Tabel m_productivity_tw

Tabel m_productivity_tw utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Truck Weighing*. Rincian dari tabel m_productivity_tw dapat dilihat dari tabel 4.79.

Tabel 4. 79 Rincian Tabel m_productivity_tw

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

80. Tabel m_vendor_personel

Tabel m_vendor_personel utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Vendor Personel*. Rincian dari tabel m_vendor_personel dapat dilihat dari tabel 4.80.

Tabel 4. 80 Rincian Tabel m_vendor_personel

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-
no_telp	character varying (null)	-	-
id_vendor	integer	foreign key	p_user(id)
description	character varying (null)	-	-
kib	character varying (null)	-	-
id_vendor_role	integer	foreign key	m_vendor_personel_role (id)
jabatan	character varying (null)	-	-

81. Tabel m_vendor_personel_role

Tabel m_vendor_personel_role utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Vendor Personel Role*. Rincian dari tabel m_vendor_personel_role dapat dilihat dari tabel 4.81.

Tabel 4. 81 Rincian Tabel m_vendor_personel_role

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

82. Tabel m_vessel

Tabel m_vessel utamanya digunakan untuk menyimpan seluruh informasi pada halaman *Master Vendor Vessel*. Rincian dari tabel m_vessel dapat dilihat dari tabel 4.82.

Tabel 4. 82 Rincian Tabel m_vessel

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-
imo	character varying (null)	-	-
built_on	integer (null)	-	-
loa	numeric (null)	-	-
grt	numeric (null)	-	-
nt	numeric (null)	-	-
dwt	numeric (null)	-	-
draught_max	numeric (null)	-	-
created_by	integer	-	-
created_date	timestamp (null)	-	-
updated_by	integer (null)	-	-
updated_date	timestamp (null)	-	-
manifold_diameter	numeric (null)	-	-

Tabel 4. 82 Rincian Tabel m_vessel (Lanjutan)

Atribut	Tipe Data	Keterangan	Target
max_pressure	numeric (null)	-	-
number_of_tank	numeric (null)	-	-
id_vessel_pulledby	integer	foreign key	m_vessel(id)
grab_used_by_jetty	boolean (null)	-	-
	daouble		
draught_before_ld	precision (null)	-	-
id_vessel_type	integer	foreign key	m_vessel_type(id)
coords	jsonb (null)		-
id_origin	integer	foreign key	m_origin(id)
mmsi	character varying (null)	-	-
	double		
draught_summer	precision(null)	-	-
	double		
draught_winter	precision(null)	-	-
	character		
barge_pulledby	varying (null)	-	-
id_region	integer	foreign key	m_vessel_region(id)

83. Tabel m_vessel_region

Tabel m_vessel_region utamanya digunakan untuk menyimpan seluruh informasi data *Region* pada halaman *Master Vendor Vessel*. Rincian dari tabel m_vessel_region dapat dilihat dari tabel 4.83.

Tabel 4. 83 Tabel m_vessel_region

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

84. Tabel m_vessel_type

Tabel m_vessel_type utamanya digunakan untuk menyimpan seluruh informasi data *Vessel Type* pada halaman *Master Vendor Vessel*. Rincian dari tabel m_vessel_type dapat dilihat dari tabel 4.84.

Tabel 4. 84 Rincian Tabel m_vessel_type

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

85. Tabel m_vessel_hatch

Tabel m_vessel_hatch utamanya digunakan untuk menyimpan seluruh informasi data *Vessel Hatches* pada halaman *Master Vendor Vessel*. Rincian dari tabel m_vessel_hatch dapat dilihat dari tabel 4.85.

Tabel 4. 85 Rincian Tabel m_vessel_hatch

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character	-	-
description	character varying (null)	-	-
id_vessel	integer	foreign key	m_vessel(id)
id_type	integer	foreign key	m_vessel_hatch_type(id)

86. Tabel m_vessel_grab

Tabel m_vessel_grab utamanya digunakan untuk menyimpan seluruh informasi data *Vessel Grabs* pada halaman *Master Vendor Vessel*. Rincian dari tabel m_vessel_grab dapat dilihat dari tabel 4.86.

Tabel 4. 86 Rincian Tabel m_vessel_grab

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character	-	-
used_by_jetty	boolean (null)	-	-

Tabel 4. 86 Rincian Tabel m_vessel_grab (Lanjutan)

Atribut	Tipe Data	Keterangan	Target
id_vessel	integer	foreign key	m_vessel(id)
id_type	integer	foreign key	m_vessel_hatch_type(id)
	double		
capacity	precision (null)	-	-

87. Tabel m_vessel_crane

Tabel m_vessel_crane utamanya digunakan untuk menyimpan seluruh informasi data *Vessel Cranes* pada halaman *Master Vendor Vessel*. Rincian dari tabel m_vessel_crane dapat dilihat dari tabel 4.87.

Tabel 4. 87 Rincian Tabel m_vessel_crane

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
crane_number	numeric (null)	-	-
swl_of_crane	numeric (null)	-	-
cycle_time_of_crane	numeric (null)	-	-
id_vessel	integer	foreign key	m_vessel(id)
id_brand	integer	foreign key	m_vessel_crane_brands (id)
created_by	integer (null)	-	-
created_on	timestampz (null)	-	-
updated_by	integer (null)	-	-
updated_on	timestampz (null)	-	-
id_type	integer	foreign key	m_vessel_crane_type (id)

88. Tabel m_wms_stock_maximum

Tabel m_wms_stock_maximum utamanya digunakan untuk menyimpan seluruh informasi data Kajian pada halaman *Master Warehouse*. Rincian dari tabel m_wms_stock_maximum dapat dilihat dari tabel 4.88.

Tabel 4. 88 Rincian Tabel m_wms_stock_maximum

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-
created_date	timestampz	-	-
created_by	integer	-	-
updated_date	timestampz (null)	-	-
updated_by	integer (null)	-	-
id_warehouse	integer	foreign key	m_warehouse(id)
start_date	timestampz (null)	-	-
end_date	timestampz (null)	-	-

89. Tabel m_wms_maximum_cargo

Tabel m_wms_maximum_cargo utamanya digunakan untuk menyimpan seluruh informasi data *Maximum Stock* pada halaman *Master Warehouse*. Rincian dari tabel m_wms_maximum_cargo dapat dilihat dari tabel 4.89.

Tabel 4. 89 Rincian Tabel m_wms_maximum_cargo

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

90. Tabel m_warehouse_type

Tabel m_warehouse_type utamanya digunakan untuk menyimpan seluruh informasi data *Type* pada halaman *Master Warehouse*. Rincian dari tabel m_warehouse_type dapat dilihat dari tabel 4.90.

Tabel 4. 90 Rincian Tabel m_warehouse_type

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character varying	-	-

91. Tabel m_warehouse_cargo

Tabel m_warehouse_cargo utamanya digunakan untuk menyimpan seluruh informasi data *Warehouse Compatible Cargo* pada halaman *Master Warehouse*. Rincian dari tabel m_warehouse_cargo dapat dilihat dari tabel 4.91.

Tabel 4. 91 Rincian Tabel m_warehouse_cargo

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
id_warehouse	integer	foreign key	m_warehouse(id)
id_cargo	integer	foreign key	m_cargo(id)

92. Tabel m_wms_pillars

Tabel m_wms_pillars utamanya digunakan untuk menyimpan seluruh informasi data *Pillars* pada halaman *Master Warehouse*. Rincian dari tabel m_wms_pillars dapat dilihat dari tabel 4.92.

Tabel 4. 92 Rincian Tabel m_wms_pillars

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
code	integer	-	-
id_warehouse	integer	foreign key	m_warehouse(id)
capacity	double precision	-	-
description	text (null)	-	-
current_id_cargo	integer	foreign key	m_cargo(id)

Tabel 4. 92 Rincian Tabel m_wms_pillars (Lanjutan)

Atribut	Tipe Data	Keterangan	Target
tonnage	double	-	-
	precision(null)		
tonnage_convert	double	-	-
	precision(null)		
created_by	integer	foreign key	p_user(id)
created_date	timestamp	-	-
updated_by	integer	foreign key	p_user(id)
updated_date	timestamp	-	-

93. Tabel m_training

Tabel m_training digunakan untuk mengambil data Training yang akan ditampilkan pada halaman *Summary Training Realization*. Rincian dari tabel m_training dapat dilihat dari tabel 4.93.

Tabel 4. 93 Rincian Tabel m_training

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
name	character	-	-
	varying		
vendor	character	-	-
	varying		
year	integer (null)	-	-
competency_issues	character	-	-
	varying		
category	integer	foreign key	m_training_category(id)

94. Tabel t_training

Tabel t_training merupakan tabel utama yang digunakan untuk mengambil data dan ditampilkan pada halaman *Summary Training Realization*. Rincian dari tabel t_training dapat dilihat dari tabel 4.94.

Tabel 4. 94 Rincian Tabel t_training

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
id_training	integer	foreign key	m_training(id)
start_date	timestampz	-	-
end_date	timestampz	-	-
duration_per_day	double precision	-	-
except_day	double precision (null)	-	-
participant_capacity	integer	-	-
total_manhour	double precision	-	-
vendor	character varying	-	-

95. Tabel t_training_participant

Tabel t_training_participant merupakan tabel yang digunakan untuk mengambil data *Personel* dan ditampilkan pada halaman *Summary Training Realization*. Rincian dari tabel t_training_participant dapat dilihat dari tabel 4.95.

Tabel 4. 95 Rincian Tabel t_training_participant

Atribut	Tipe Data	Keterangan	Target
id	integer	primary key	-
id_t_training	integer	foreign key	t_training(id)
id_pd_personel	integer	foreign key	m_pg_personel(id)
sertifikat	character varying (null)	-	-

4.4 Implementasi Sistem

Tahapan implementasi sistem merupakan tahap pengembangan *Website* Sistem Informasi Manajemen Pelabuhan Pada PT Patrakim Timur dengan cara menggunakan *framework* Royal. Basis dari *framework* ini menggunakan bahasa pemrograman TypeScript. *Framework* Royal memiliki struktur yang mencakup implementasi React.js, Node.js, Tailwind CSS, Prisma, dan PostgreSQL. Berikut implementasi dari pengembangan *website* sistem informasi PT Patrakim Timur.

4. 4. 1. Halaman baru Summary Training Realization

1. Koding

Pada *framework* Royal untuk menerapkan fungsi CRUD dapat menggunakan *tag* Admin. Struktur untuk penggunaan *tag* Admin dapat dilihat pada gambar 4.9

```
<Admin
  content={{
    t_training: {
      table: 't_training',
      list: {
        action: { ...
      },
      table: { ...
    },
    params: { ...
  },
}}
/>
```

Gambar 4. 9 Struktur *tag* Admin pada *framework* Royal

Properti *table* digunakan untuk menentukan tabel mana yang akan dipakai. Properti *action* digunakan untuk menambah aksi seperti *create*, *export*, *delete* dan bisa juga untuk membuat *custom action*. Properti *params* digunakan untuk memfilter data yang akan dipakai.

Halaman *Summary Training Realization* menggunakan tabel dari `t_training`. Penambahan fitur *export* PDF dan Excel dilakukan dengan cara penambahan *tag Button* pada properti *action*.

Berikut adalah koding untuk properti *table*.

```
columns: [
  [
    'm_training',
    {
      title: 'Training Name',
      width: 300,
      value: (item) => {
        return <>{item.m_training.name}</>
      },
    },
  ],
  [
    'm_training',
    {
      title: 'Competency Issues',
      width: 150,
      value: (item) => {
        if (item.m_training.competency_issues !== null)
        {
          return <>{item.m_training.competency_issues}</>
        } else {
          return (
            <div className={`text-xs text-gray-400`}>-
</div>
          )
        }
      },
    },
  ],
  [
    'start_date',
    {
      title: 'Start Date',
      width: 150,
    },
  ],
  [
    'end_date',
    {
      title: 'End Date',
      width: 150,
    },
  ],
  [
    'vendor',
    {
      title: 'Vendor',
      width: 150,
    },
  ],
  [
```

```

        't_training_participant',
        {
            title: 'Personel',
            width: 150,
            value: (item) => {
                return <>{item.t_training_participant.length}</>
            },
        },
    ],
],

```

Selanjutnya adalah koding untuk pembuatan fitur *export* PDF, koding dapat dilihat dibawah ini.

```

<Button
  type={'custom'}
  key={'runall'}
  iconProps={{ iconName: 'EntryView' }}
  onClick={async () => {
    const data = await db.t_training.findMany({
      include: {
        m_training: true,
        t_training_participant: {
          include: {
            m_pg_personel: true,
          },
        },
      },
      where: {
        start_date: {
          gte: selectedYear === -
1 ? undefined : new Date(selectedYear, 0, 1),
          lte: selectedYear === -
1 ? undefined : new Date(selectedYear, 11, 31),
        },
      },
      orderBy: {
        id: 'asc',
      },
    })
    let tempStock = [] as any
    if (data.length > 0) {
      meta.data_pdf = data.map((item) => {
        let obj = {}
        ;(obj['Training Name'] =
          get(item, 'm_training.name') !== undefined
            ? get(item, 'm_training.name')
            : ''),
          (obj['Competency Issues'] =
            get(item, 'm_training.competency_issues') !==
            null
              ? get(item, 'm_training.competency_issues')

```

```

        : '-'),
        (obj['Start Date'] = format(
            new Date(get(item, 'start_date')),
            'dd MMM yyyy - hh:mm'
        )),
        (obj['End Date'] = format(
            new Date(get(item, 'end_date')),
            'dd MMM yyyy - hh:mm'
        )),
        (obj['Vendor'] = get(item, 'vendor')),
        (obj['Personel'] = get(
            item,
            't_training_participant.length'
        ))
        tempStock.push(obj)
    })
}
meta.tabelPdf = tempStock
meta.headerPdf = [
    'Training Name',
    'Competency Issues',
    'Start Date',
    'End Date',
    'Vendor',
    'Personel',
]
meta.action.openModalPdf(true)
meta.render()
}}
>
    Export PDF
</Button>

```

Yang terakhir adalah koding untuk pembuatan fitur *export* Excel, koding dapat dilihat dibawah ini.

```

<PrimaryButton
  css={css`
    margin-right: 6px;
    height: 30px;
    width: fit-content;
  `}
  className="save px-0"
  onClick={async () => {
    const data = await db.t_training.findMany({
      include: {
        m_training: true,
        t_training_participant: {
          include: {
            m_pg_personel: true,
          },
        },
      },
    },
  },
)

```

```

        where:{
            start_date:{
                gte: selectedYear === -
1 ? undefined : new Date(selectedYear, 0, 1),
                lte: selectedYear === -
1 ? undefined : new Date(selectedYear, 11, 31),
            }
        },
        orderBy: {
            id: 'asc',
        },
    })
    if (data.length > 0) {
        meta.dataExcel = data.map((item) => {
            return {
                'Training Name':
                    get(item, 'm_training.name') !== undefined
                    ? get(item, 'm_training.name')
                    : '',
                'Competency Issues': get(
                    item,
                    'm_training.competency_issues'
                ),
                'Start Date': format(
                    new Date(get(item, 'start_date')),
                    'dd MMM yyyy - hh:mm'
                ),
                'End Date': format(
                    new Date(get(item, 'end_date')),
                    'dd MMM yyyy - hh:mm'
                ),
                Vendor: get(item, 'vendor'),
                Personel: get(
                    item,
                    't_training_participant.length'
                ),
            }
        })
    }
    meta.render()

```

```

let xlsx = await import('xlsx')
const table =
    document.getElementById('TableInputxlsx')
const sheet = xlsx.utils.table_to_sheet(table)
let js1 = xlsx.utils.sheet_to_json(sheet, {
    blankrows: true,
    header: 'A',
    raw: true,
    rawNumbers: true,
}) as any
await exportExcelWithDate(
    js1,

```

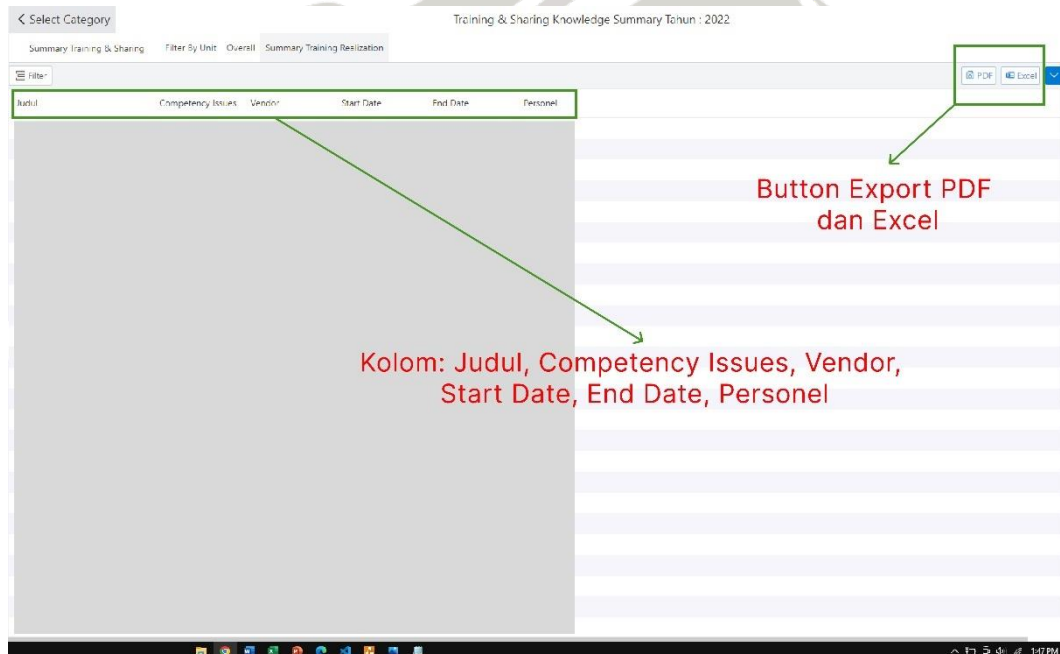
```

        sheet,
        `Training Summary ${new Date().toLocaleDateString()}
    }`
    )
    }}
    iconProps={{
        iconName: 'ExcelLogo',
    }}
    >
    Export Excel
</PrimaryButton>

```

2. Hasil

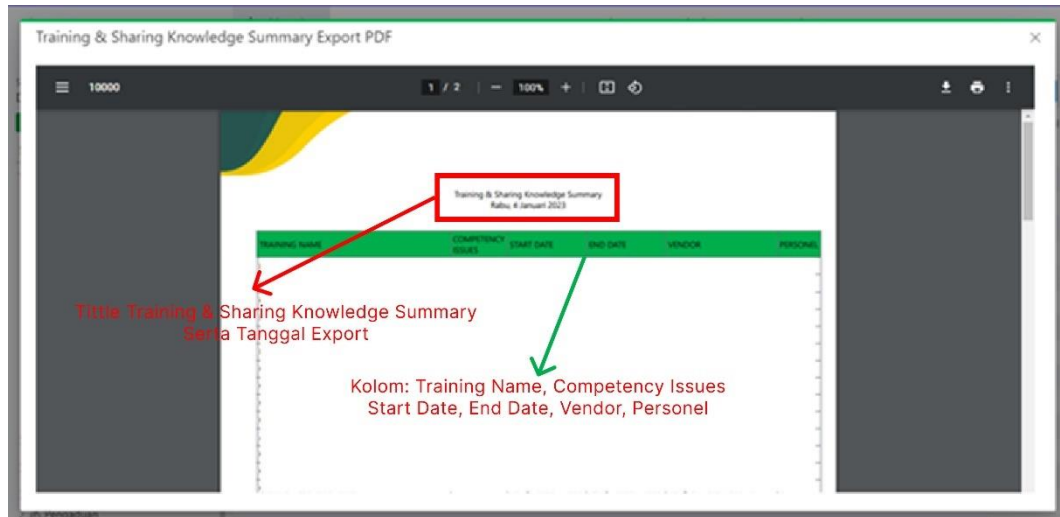
Gambar 4. 10 menunjukkan hasil pembuatan halaman baru *Training Realization Summary* beserta fitur *export* PDF & Excel.



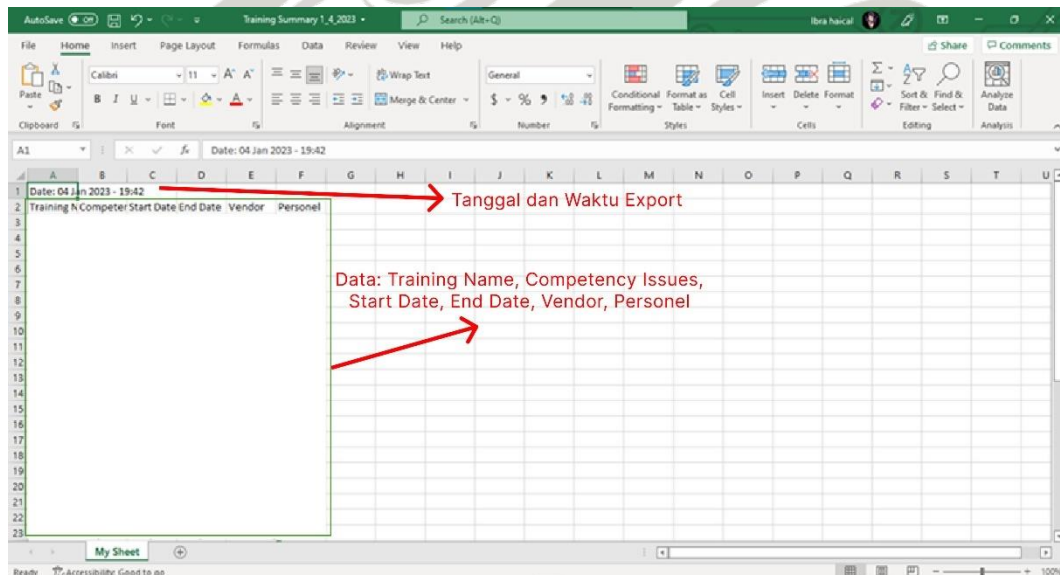
Gambar 4. 10 Halaman *Summary Training Realization*

Pada halaman *Summary Training Realization* terdapat informasi *Training Name* yang di ambil dari tabel *m_training field name*, *Competency Issues* dari tabel *m_training field competency_issues*, *Start Date & End Date* dari tabel *t_training field start_date & end_date*, *Vendor* dari tabel *t_training field vendor*, dan *Personel* dari jumal data di *t_training_participant*.

Selanjutnya dapat dilihat pada gambar 4. 11 yang merupakan hasil dari pembuatan fitur *export* PDF dan gambar 4. 12 hasil fitur *export* Excel.



Gambar 4. 11 Hasil Export PDF



Gambar 4. 12 Hasil Export Excel

4. 4. 2. Perbaikan halaman Training Realization

1. Koding

Pada halaman *Training Realization* admin dapat menginput data *Training*, *Start Date*, *End Date*, *Except Day*, *Duration Per Day*, *Participants Quantity*, *Total Manhour*, dan *Vendor*. Sebelumnya terdapat permasalahan pada input *Start Date* karena admin dapat menginput tahun yang berbeda dengan data tahun *Master Data Training*.

Gambar 4. 13 *Form Training Realization*

Berdasarkan *Master Data Training*, data *Training* mempunyai tahun sehingga input tahun di *Start Date* Harus sesuai dengan tahun yang dimiliki data *Training*. Gambar 4. 13 menunjukkan *form* apabila admin ingin menambah atau merubah data di halaman *Training Realization*.

Berikut adalah koding untuk perbaikan input *Start Date* di halaman *Training Realization*.

1) Pembuatan *meta* untuk menyimpan data.

```
const declare_meta = () => {
  const meta = {
    yearAll: [] as any[],
    selectedYearTraining: null as any,
  }
  return meta
}

const init_meta = () => {
  return async () => {
    const start_date = await db.query(
      'select start_date from t_training order by start_date asc'
    )
    const setYear = new Set()
    start_date.forEach((item: any) => {
      setYear.add(new Date(item.start_date).getFullYear())
    })
    meta.yearAll = Array.from(setYear)
    meta.render()
  }
}

const meta = useLocal(declare_meta(), init_meta())
const [selectedYear, setSelectedYear] = useState < number | null > (-1)
const [selectedCategory, setSelectedCategory] =
  useState<Category | null>({
    id: -1,
    name: '', })
```

2) Property Form framework Royal

```
form: {  
  onLoad: async (data, state) => { ...  
  },  
  onSave: async (state) => { ...  
  },  
  tabs: ({ tabs }) => { ...  
  },  
  alter: { ...  
  },  
  layout: ({ layout }) => {  
    return layout([  
      ['m_training', 'start_date', 'end_date'],  
      ['except_day', 'duration_per_day', 'participant_capacity'],  
      ['total_manhour', ['vendor'], ['']],  
      't_training_participant',  
    ])  
  },  
},  
},
```

Gambar 4. 14 Property Form Framework Royal

onLoad berjalan ketika sebuah halaman dibuka. *onSave* dieksekusi ketika admin menyimpan perubahan. *alter* digunakan untuk memodifikasi bagian *form* pada data tertentu seperti merubah judul, tipe data, dan data pada kolom. *layout* digunakan sebagai data apa saja yang akan dipakai pada *form*. Tampilan halaman *form* akan terlihat seperti gambar 4. 14.

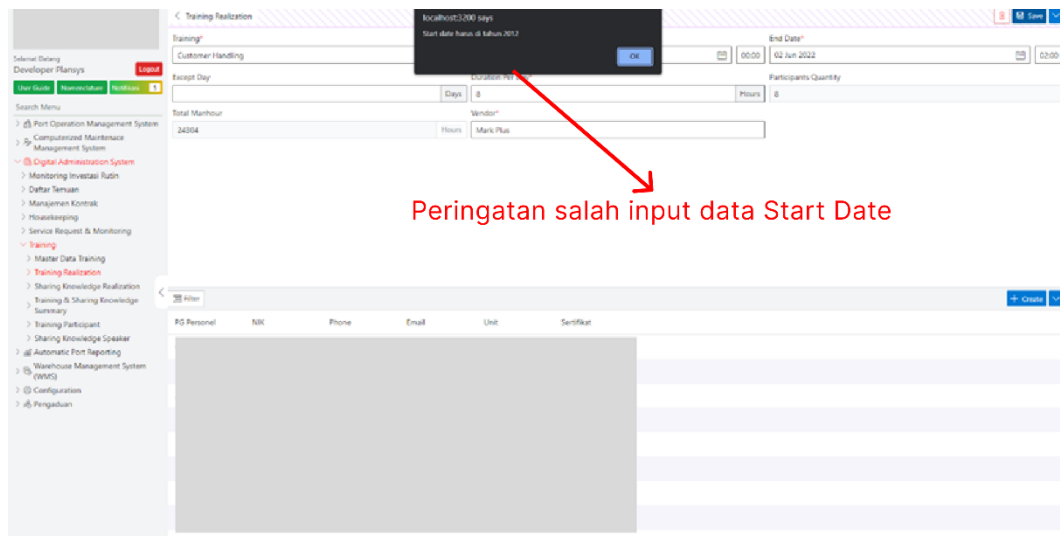
Pada bagian *onSave* diberi kode sebagai berikut:

```
if (meta.selectedYearTraining && (new  
Date(state.data.start_date).getFullYear() !=  
meta.selectedYearTraining)) {  
  return alert("Start date harus di tahun " +  
meta.selectedYearTraining)  
}  
state.save()
```

Pemberian bagian kode di atas bertujuan untuk mengecek apakah input *Start Date* sudah sesuai dengan data yang ada di *Master Data Training*. Jika data yang diinput tidak sesuai maka sistem akan menampilkan peringatan yang berisi informasi tahun data *Training*.

2. Hasil

Gambar 4. 15 menunjukkan peringatan pada admin bahwa *Start Date* yang diinputkan tidak sesuai dengan tahun di *Master Data Training*.



Gambar 4. 15 Peringatan salah input Start Date

4. 4. 3. Penambahan fitur mass update & delete pada halaman Master Data

Selanjutnya merupakan penambahan fitur *mass update & delete* di seluruh halaman *Master Data*. Penambahan fitur *mass update & delete* pada halaman *Master Data* berjumlahkan 57 halaman. Tujuan dari pembuatan fitur ini adalah untuk mempermudah admin dalam mengelola setiap data yang ada di halaman *Master Data*. Terdapat *template* Excel yang diperoleh dari fitur *mass update* sehingga dapat digunakan untuk merubah data sesuai data yang telah dipilih sebelumnya. Beberapa *template* Excel memiliki beberapa *sheet* yang berisi detail data berdasarkan ID data yang tampil di *main sheet* Excel. Fungsi dari ID yang ada pada *main sheet* Excel adalah untuk sinkronisasi dengan *database* sehingga *sheet* lain merupakan keterangan data berdasarkan ID di *main sheet* yang berfungsi untuk mempermudah admin dalam mengelola data. Langkah-langkah yang dapat dilakukan dalam menggunakan fungsi ini yaitu:

1. Admin menekan tombol biru pojok kanan atas yang bersimbol *down arrow*.
2. Admin memilih fitur *mass update* atau *mass delete*.
 - Jika pilihanya *mass update* maka akan tampil sebuah kotak dialog yang berisi dokumen *template* Excel untuk merubah data dan bisa di *upload* kembali setelah dokumen selesai dirubah.
 - Jika pilihanya *mass delete* maka admin bisa langsung menghapus data yang diinginkan.

Adapun halaman *Master Data* yang telah ditambahkan fitur *mass update & delete*, antara lain:

1. Halaman *Master Cargo Color* terhubung dengan tabel `m_cargo_color`.
2. Halaman *Master Cargo Grade* terhubung dengan tabel `m_cargo_grade`.
3. Halaman *Master Hard File* terhubung dengan tabel `m_pg_personel_unit`.
4. Halaman *Master Unit* terhubung dengan tabel `m_unit`.
5. Halaman *Master Activity* terhubung dengan tabel `m_activity`.
6. Halaman *Master Area (Op)* terhubung dengan tabel `m_area`.
7. Halaman *Master Asset* terhubung dengan tabel `m_asset`, `m_port`, `m_jetty`, `m_asset_type`, `m_asset_status`, `m_ownership`, dan `m_asset_items`.
8. Halaman *Master Asset Item* terhubung dengan tabel `m_asset_items`, `m_area`, `m_work_center`, `m_asset`, dan `m_asset_item_troubles`.
9. Halaman *Master Asset Type* terhubung dengan tabel `m_asset_type`, `m_asset_type_role`, dan `p_role`.
10. Halaman *Master Asset Housekeeping* terhubung dengan tabel `v_housekeeping_asset`.
11. Halaman *Master Brand Crane* terhubung dengan tabel `m_vessel_crane_brands`.
12. Halaman *Master BWT Exclude* terhubung dengan tabel `m_bwt_exclude` dan `m_activity`.
13. Halaman *Master Bariksama Note* terhubung dengan tabel `m_bariksama_note`.
14. Halaman *Master Contract Type* terhubung dengan tabel `m_contract_type`.
15. Halaman *Master Cargo* terhubung dengan tabel `m_cargo`, `m_cargo_category`, `m_cargo_uom`, `m_arrival_type`, `m_cargo_status`, `m_cargo_procurement`, `m_cargo_color_grade`, `m_cargo_orde`, dan `m_wms_safety_stock`.
16. Halaman *Master Cargo Category* terhubung dengan tabel `m_cargo_category`.
17. Halaman *Master Cargo Owner* terhubung dengan tabel `m_cargo_owner` dan `p_user`.
18. Halaman *Master Checkcheet Type OPS* terhubung dengan tabel `m_checksheets_type_ops`.
19. Halaman *Master Currency* terhubung dengan tabel `m_currency_convert`, `m_currency`, dan `m_currency_by_to`.

20. Halaman *Master Document Type* terhubung dengan tabel `m_document_type`, `p_role_by_id_uploader_role`, `m_document_region`, dan `p_role`,
21. Halaman *Master Forecast Authority* terhubung dengan tabel `t_wms_forecast_authority` dan `m_wms_authority`.
22. Halaman *Master Housekeeping Group* terhubung dengan tabel `m_housekeeping_group`.
23. Halaman *Master Target Cargo* terhubung dengan tabel `m_target_cargo`, `m_cargo_orde`, dan `m_cargo`.
24. Halaman *Master Great Day* terhubung dengan tabel `m_holiday`.
25. Halaman *Master Hatch Type* terhubung dengan tabel `m_vessel_hatch_type`.
26. Halaman *Master ISPS Code* terhubung dengan tabel `m_isps`.
27. Halaman *Master Jetty* terhubung dengan tabel `m_jetty` dan `m_port`.
28. Halaman *Master Laytime* terhubung dengan tabel `m_laytime` dan `m_activity_laytime`.
29. Halaman *Master Laytime Start Count* terhubung dengan tabel `m_laytime_start_count`.
30. Halaman *Master Liquid Logsheets* terhubung dengan tabel `m_liquid_logsheets`.
31. Halaman *Master Location (Op)* terhubung dengan tabel `m_location`.
32. Halaman *Master Material Number* terhubung dengan tabel `m_wms_material_number`.
33. Halaman *Master NOPOL* terhubung dengan tabel `m_wms_nopol` dan `m_wms_transportir`.
34. Halaman *Master Origin/Destination Cargo* terhubung dengan tabel `m_cargo_orde` dan `m_arrival_type`.
35. Halaman *Master Origin Vessel* terhubung dengan tabel `m_origin`.
36. Halaman *Master PG Personel* terhubung dengan tabel `m_pg_personel`, `m_pg_personel_scope`, `m_pg_personel_position`, `m_pg_personel_unit`, dan `m_wms_authority`.
37. Halaman *Master Payrate* terhubung dengan tabel `m_housekeeping_payrate` dan `m_housekeeping_payrate_asset`.
38. Halaman *Master Penilaian Kinerja Rekanan* terhubung dengan tabel `m_project_aspek_penilaian`.

39. Halaman *Master Permit* terhubung dengan tabel m_project_permit.
40. Halaman *Master Plant Code* terhubung dengan tabel m_wms_plant, m_wms_plant_warehouse, dan m_wms_sloc.
41. Halaman *Master Project Location* terhubung dengan tabel m_project_location.
42. Halaman *Master Port* terhubung dengan tabel m_port.
43. Halaman *Master Route* terhubung dengan tabel m_route, m_asset, m_warehouse, m_route_compatible_cargos, m_route_conveyors, dan m_route_cargo_specifier.
44. Halaman *Master Signature* terhubung dengan tabel m_signature, p_user, dan p_role.
45. Halaman *Master Sloc* terhubung dengan tabel m_wms_sloc.
46. Halaman *Master Stand by Cause* terhubung dengan tabel m_standby_cause.
47. Halaman *Master Surveyor Service* terhubung dengan tabel m_surveyor_service, m_surveyor_service_unit, dan m_surveyor_service_category.
48. Halaman *Master Surveyor Category* terhubung dengan tabel m_surveyor_service_category.
49. Halaman *Master Temperature & Pressure Indicator* terhubung dengan tabel m_temp_pressure_indicator.
50. Halaman *Master Trouble (Op)* terhubung dengan tabel m_trouble.
51. Halaman *Master Truck Weighing* terhubung dengan tabel m_productivity_tw.
52. Halaman *Master Transportir* terhubung dengan tabel m_wms_transportir.
53. Halaman *Master Vendor Personel* terhubung dengan tabel m_vendor_personel, m_vendor_personel_role, dan p_user.
54. Halaman *Master Vendor Personel Role* terhubung dengan tabel m_vendor_personel_role.
55. Halaman *Master Vessel* terhubung dengan tabel m_vessel, m_origin, m_vessel_region, m_vessel_type, m_vessel_hatch, m_vessel_grab, dan m_vessel_crane.
56. Halaman *Master Warehouse* terhubung dengan tabel m_warehouse, m_wms_stock_maximum, m_cargo, m_cargo_grade, m_cargo_color, m_wms_maximum_cargo, m_ownership, m_cargo_color_grade, p_user,

m_wms_authority, m_warehouse_type, m_warehouse_cargo, m_cargo, m_cargo_orde, m_arrival_type, m_cargo_category, dan m_wms_pillars.

57. Halaman *Master Work Center* (Op) terhubung dengan tabel m_work_center.

Penambahan fitur *mass update & delete* dilakukan dengan menambah kode pada bagian *ActionOptions* di *framework* Royal. *ActionOptions* merupakan dokumen yang berisi kumpulan aksi yang nantinya dapat digunakan secara global. Berikut adalah kode yang ditambahkan.

```
'mass-delete': {
  label: 'Mass Delete',
  icon: 'Trash',
  onClick: async () => {
    let s = state as IBaseListContext
    s.table.web.checkbox = true
    s.table.web.massDelete = true
    s.component.render()
  },
},
'mass-update': {
  label: 'Mass Update',
  icon: 'EditMirrored',
  onClick: async () => {
    let s = state as IBaseListContext
    s.table.web.checkbox = true
    s.table.web.massUpdate = true
    s.component.render()
  },
},
```

Selanjutnya penambahan kode *mass update* dan *delete* bagian *Record* Typescript *IAction*. Bagian ini berhubungan dengan *tag* Admin dibagian properti *action*.

```
export type IAction = Record<string, IActionSingle> & {
  create?: IActionSingle
  save?: IActionSingle
  copy?: IActionSingle
  mv?: IActionSingle
  extra?: IActionSingle
  massDelete?: IActionSingle
  massUpdate?: IActionSingle
  paste?: IActionSingle
  export?: IActionSingle
  custom?: Record<string, IActionSingle> | IActionSingle
}
```

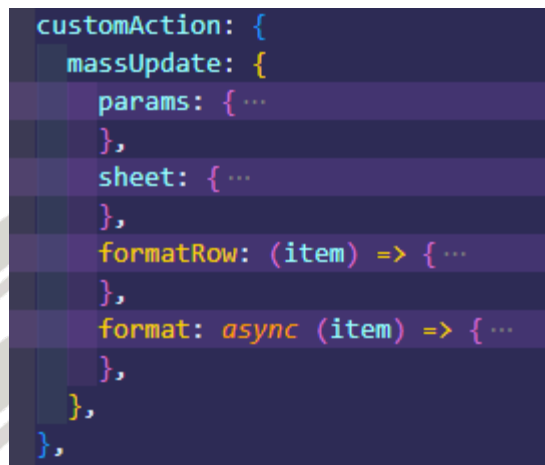
Fitur *mass update* dan *delete* diaktifkan dengan cara menambahkan *action* *massUpdate* dan *massDelete* pada *tag* Admin.

```

action: {
  massDelete: true,
  massUpdate: true,
}

```

Kode diatas dapat ditambahkan pada setiap halaman *Master Data*. Namun, pada bagian *mass update* masih dalam keadaan *default* sehingga belum ada *sheet* tambahan pada *template* Excel yang berfungsi untuk memberikan informasi lebih detail di Excel. *Mass update* yang telah dibuat dapat di *custom* pada bagian tag *Admin*. *Custom action mass update* dapat dilihat pada gambar 4. 16



```

customAction: {
  massUpdate: {
    params: { ...
  },
  sheet: { ...
  },
  formatRow: (item) => { ...
  },
  format: async (item) => { ...
  },
}

```

Gambar 4. 16 Custom Action Mass Update

Gambar 4. 16 menunjukkan struktur kode dari properti *customAction* fitur *mass update*. *params* berfungsi untuk menghubungkan relasi antar tabel. *sheet* digunakan untuk menampilkan data dalam bentuk *sheet* di Excel. *formatRow* berfungsi untuk mengatur format kolom pada Excel. Kemudian *format async* yang digunakan untuk mensinkronisasi data dari Excel ke *database*.

Bagian terakhir adalah pembuatan modal ketika menjalankan fitur *mass update*. Admin dapat mengunduh *template* Excel yang diperoleh dari fitur *mass update*. Kemudian *template* Excel yang diunduh digunakan untuk merubah data dan diunggah kembali setelah data selesai dirubah.

Gambar 4. 17 menunjukkan *template* Excel yang digunakan untuk merubah data. Kemudian bagian kolom yang menggunakan inputan ID data dapat melihat detail data pada *sheet* yang tersedia.

ID	Name	ID Vessel	ID Origin	ID Region	IMO	MMSI	LOA	Built On	Depth	Draught B	Draught S	Draught W	GRT	NRT	DWT	Manifold C Max Pressure
933																
934																
935																

Diatur berdasarkan banyaknya data yang dipakai

Banyaknya sheet menyesuaikan data yang mempunyai relasi

Sheets: Data, Vessel Type, Origin, Region

Gambar 4. 17 Template Excel Fitur *Mass Update*

Berikut adalah kode pembuatan komponen untuk modal *mass update* halaman *Master Data*.

```
export const showImportExcelMassUpdatePopup = (s:
IBaseListContext, data) => {
  s.table.web.popup = {
    loading: false,
    show: true,
    text: (
      <div
        className="border flex flex-col border-blue-400
rounded-md p-2 shadow-md select-none"
        css={css`
          margin: -20px;
          width: 350px;
          height: 200px;
          z-index: 99;
          background: white;
          position: absolute;
          top: 50%;
          left: 50%;
          transform: translate(-50%, -50%);
        `}
      >
        <div className="flex justify-between">
          <div className="flex flex-row items-center space-
x-2">
            <Icon iconName="ExcelDocument" />
            <div> Update By Excel</div>
          </div>
          <Icon
            iconName="ChromeClose"
            onClick={ (e) => {
              if (s.table.web.popup) {
                s.table.web.popup.show = false
              }
            }}
          />
        </div>
      </div>
    )
  }
}
```

```

    }
    s.component.render()
    e.preventDefault()
    e.stopPropagation()
  }}
  className="cursor-pointer m-1"
/>
</div>

<div className="flex-1 flex flex-col justify-center
space-y-2">
  <div
    className="border-2 border-blue-400 bg-blue-50
p-2 rounded-md cursor-pointer items-center flex flex-row
space-x-2"
    onClick={async (e) => {
      e.preventDefault()
      e.stopPropagation()
      generateExcelTemplate(s)
    }}
  >
    <span>1.</span>
    <Icon iconName="ExcelLogo" />
    <span>
      Download <span className="underline">Template
Excel Import</span>
    </span>
  </div>

  <div className="flex flex-col border-2 border-
blue-400 bg-blue-50 p-2 rounded-md space-y-1 relative ">
    <div>2. Upload Edited Excel Template</div>
    <div className="flex flex-row items-center">
      <input
        type="file"
        onChange={async (e) => {
          let xlsx = await import('xlsx')
          const files = e.target.files
          const sel = s.table.web.selection
          const def = s.db.definition as any
          if (files) {
            const reader = new FileReader()
            reader.onload = async (e) => {
              const data = e.target?.result
              const wb = xlsx.read(data, { type:
'binary' })

              const list = xlsx.utils.sheet_to_json(
                wb.Sheets[wb.SheetNames[0]]
              )
              let fieldMap = get(s,
'db.definition.columns')
              const custom = get(s,
'header.customAction.massUpdate')

```

```

let result = [] as any
let table = s.db.tableName || ''
const customRow: any = get(custom,
'format')

const customTable: any = get(custom,
'table')

if (customTable) {
  table = customTable
}
let fin = [] as any
console.log({ list })
if (customRow) {
  await Promise.all(
    list.map(async (e, idx) => {
      const row: any = e
      let res = await custom.format(e)
      if (!res.id) {
        res = { ...res }
      }
      e = res
      fin.push(e)
    })
  )
} else {
  fin = list
}
let validation = fin.filter((l) =>
!l['id']) || []

if (validation.length > 0) {
  alert('Error! Harus ada ID ')
} else {
  if (custom) {
    const customImportRow: any =
get(custom, 'importRow')
    console.log(customImportRow)
    if (customImportRow) {
      let message = ''
      showLoadingImport(message, s)
      let closeModal = async () => {
        if (s.table.web.popup) {
          s.table.web.popup.show =
false

          s.table.web.selection =
undefined

          s.table.web.massUpdate = false
          s.table.web.checkbox = false
          s.component.render()
          s.db.query()
        }
      }
      await custom.importRow(fin,
closeModal)
    }
  }
}

```

```

    } else {
      let message = ''
      showLoadingImport(message, s)
      let count = 0
      let max = fin.length
      message = `Your import is still
running ${count}/${max}`

      await Promise.all(
        fin.map(async (e, idx) => {
          let row = e as any
          // s.table.web.popup?.text =
          await db[table]
            .update({
              where: {
                id: row.id,
              },
              data: row,
            })
            .then(() => {
              count++
              message = `Your import
is still running ${count}/${max}`
              showLoadingImport(message, s)

              if (count >= max) {
                if (s.table.web.popup)
                {
                  s.table.web.popup.show = false

                  s.table.web.selection
                    = undefined
                  s.table.web.massUpdate
                    = false
                  s.table.web.checkbox =
                    false
                  s.component.render()
                  s.db.query()
                }
              })
            })
          })
      )
    } else {
      let message = ''
      console.log('masuk sini')
      showLoadingImport(message, s)
      let count = 0
      let max = fin.length
      message = `Your import is still
running ${count}/${max}`

      console.log({ fin })
      await Promise.all(

```

```

    fin.map(async (e, idx) => {
      let row = e as any
      for (let [k, value] of
Object.entries(row)) {
        let cols = fieldMap[k]
        let v: any = value
        if (cols && value) {
          if (
            cols.type === 'date' ||
            cols.type === 'datetime'
          ||
            cols instanceof Date
          ) {
            try {
              const date = new
Date(v)

              if (date.getFullYear()
=== 1970) {
                row[k] =
ExcelDateToJSDate(v)
              } else {
                row[k] =
date.toISOString()
              }
            } catch (e) {
              console.log(
                `failed to parse
                date (${k})`,
                value
              )
            }
          }
          if (cols.type ===
'number') {
            row[k] = v ? Number(v) :
null
          }
        }
      }
      await db[table]
        .update({
          where: {
            id: row.id,
          },
          data: row,
        })
        .then(() => {
          count++
          message = `Your import is
still running ${count}/${max}`
          showLoadingImport(message,
s)

```

```

        if (count >= max) {
            if (s.table.web.popup) {
                s.table.web.popup.show()
            }
            s.table.web.selection = s.table.web.selection || {}
            s.table.web.massUpdate = s.table.web.massUpdate || {}
            s.table.web.checkbox = s.table.web.checkbox || {}
            s.component.render()
            s.db.query()
        }
    })
})
}
},
),
},
s.component.render()
}

const generateExcelTemplate = async (s) => {
    const custom = get(s, 'header.customAction.massUpdate')
    const first = {} as any
    let data = [] as any[]
    let table = s.db.tableName || ''
    const customTable: any = get(custom, 'table')
    if (customTable) {
        table = customTable
    }
    const sheets = {} as Record<string, Record<string, any>[]>
    let fieldMap = get(s, 'db.definition.columns')
    const parent = null
    const imconf = null
    let paramsParent: any = {}
    if (custom)
        if (custom.params) {
            if (
                !(custom.params instanceof Date) &&
                typeof custom.params === 'object'
            ) {

```

```

        paramsParent = { ...paramsParent, ...custom.params }
    }
}
if (custom) {
    const sel = s.table.web.selection
    const def = s.db.definition
    let form = {} as any
    if (get(s, 'tree.parent.crud.content.form.alter'))
        form = get(s, 'tree.parent.crud.content.form.alter')
    let result = {} as any
    if (def && sel && sel.getSelectedCount() > 0) {
        let id = sel.getSelection().map((e) => e['id'])
        paramsParent = {
            ...paramsParent,
            where: {
                ...paramsParent.where,
                [def.pk]: {
                    in: sel.getSelection().map((e) => e[def.pk]),
                },
            },
        }
        let res = await db[table].findMany(paramsParent)
        data = [...res]
    }
    const promises = {} as Record<string, Promise<any>>
    const isheets = {} as Record<string, Record<string,
any>[]>

    for (let [k, v] of Object.entries(custom.sheet)) {
        let col = v as any
        if (typeof col === 'object' && !(col instanceof Date)
&& col.table) {
            promises[k] = new Promise(async (e) => {
                const table = db[col.table]
                if (table) {
                    const def = await table.definition()
                    if (def) {
                        let name = ''
                        for (let col of Object.values(def.columns) as
any) {
                            if (!col.name.includes('id') && col.type ===
'string') {
                                if (!name) {
                                    name = col.name
                                }
                            }
                        }
                        if (col.name.includes('name') ||
col.name.includes('nama')) {
                            name = col.name
                            break
                        }
                    }
                    if (!name) name = Object.keys(def.columns)[0]
                }
            })
        }
    }
}

```

```

        let params = {}
        if (col.params)
            if (typeof v === 'object' && !(v instanceof
Date)) {
                params = col.params
            }
            isheets[k] = await table.findMany(params)
            if (typeof col === 'object' && !(v instanceof
Date)) {
                if (col.format) {
                    for (let [r, s] of
Object.entries(isheets[k])) {
                        isheets[k][r] = col.format(s)
                    }
                }
                if (isheets[k].length > 0) {
                    e(isheets[k][0][def.pk])
                } else {
                    e('')
                }
            }
        })
    } else {
        promises[k] = new Promise((e) => e(v))
    }
}
await Promise.all(Object.values(promises))
for (let [k, v] of Object.entries(promises)) {
    if (isheets[k]) {
        sheets[k] = isheets[k]
    }
}
} else {
    const promises = {} as Record<string, Promise<any>>
    const isheets = {} as Record<string, Record<string,
any>[]>
    const sel = s.table.web.selection
    const def = s.db.definition
    let form = {} as any
    if (get(s, 'tree.parent.crud.content.form.alter'))
        form = get(s, 'tree.parent.crud.content.form.alter')
    let result = {} as any
    if (def && sel && sel.getSelectedCount() > 0) {
        let id = sel.getSelection().map((e) => e['id'])
        let res = await db[table].findMany({
            where: {
                [def.pk]: {
                    in: sel.getSelection().map((e) => e[def.pk]),
                },
            },
        })
    }
}

```

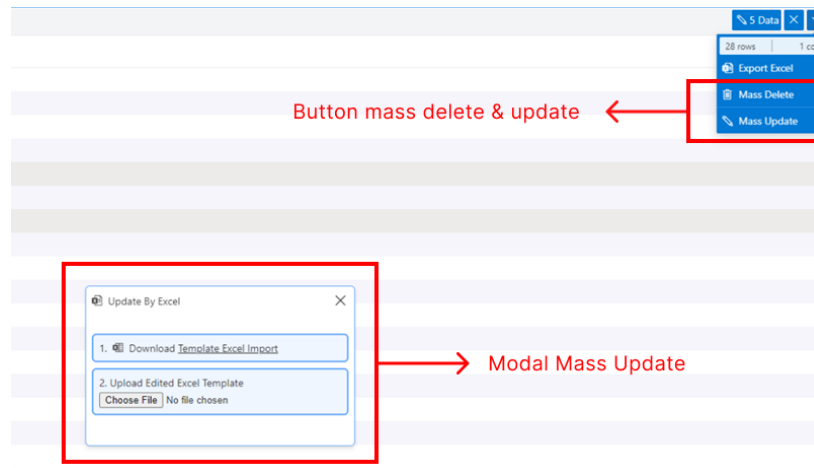
```

    res.map((e) => {
      for (let [k, value] of Object.entries(e)) {
        let cols = fieldMap[k]
        if (cols.name === 'updated_by') {
          e[k] = window.user.id
          value = window.user.id
        }
        if (cols.name === 'updated_date') {
          e[k] = new Date()
          value = new Date()
        }
        if (cols && value) {
          if (
            cols.type === 'date' ||
            cols.type === 'datetime' ||
            cols instanceof Date
          ) {
            let v: any = value
            e[k] = new Date(v)
          }
          if (cols.name === 'updated_by') {
            e[k] = window.user.id
          }
          if (cols.name === 'updated_date') {
            e[k] = new Date()
          }
        }
      }
    })

    data = [...res]
  }
}
if (custom)
  if (custom.formatRow) {
    if (!(custom.formatRow instanceof Date)) {
      await Promise.all(
        data.map(async (e, idx) => {
          data[idx] = custom.formatRow(e)
        })
      )
    }
  }
}
ExportExcel({
  data: data,
  sheets,
  filename: `Import ${s.db.tableName?.toLowerCase()} ${new
Date().toLocaleDateString()}`,
})
}

```

Gambar 4. 18 adalah hasil dari pembuatan fitur *mass update & delete* yang akan digunakan pada halaman *Master Data*.



Gambar 4. 18 Hasil pembuatan fitur mass update & delete

4.5 Testing

Tahapan ini dilakukan setelah pengkodean selesai. Pada tahapan ini dilakukan pengujian terhadap halaman baru *Summary Training Realization* beserta fitur *export PDF & Excel*, pengujian terhadap halaman *Training Realization* pada input *Start Date* apakah sudah sesuai dengan data tahun di *Master Data Training*, dan pengujian terhadap fitur *mass update & delete* di halaman *Master Data*. Pengujian bertujuan untuk mengetahui kesalahan apa saja yang timbul saat *website* sedang berjalan serta mengetahui apakah fitur yang dibangun sudah sesuai dengan kebutuhan pengguna. Pengujian ini dilakukan oleh mentor dan Tim *Business Analyst* mulai dari kode yang dibuat hingga hasil fitur yang selesai dibangun. Berikut ini adalah tabel rangkuman dari hasil uji coba bagian *website* yang telah dikerjakan oleh penulis.

Tabel 4. 96 Hasil Pengujian

No.	Fitur yang telah diuji	Berhasil	Tidak Berhasil	Time Load (s)
1.	<i>Mass update & delete</i> Halaman Master Cargo Color	✓		0.598
2.	<i>Mass update & delete</i> Halaman Master Cargo Grade	✓		0.603
3.	<i>Mass update & delete</i> Halaman Master Hard File	✓		0.588
4.	<i>Mass update & delete</i> Halaman Master Unit	✓		0.589

Tabel 4. 96 Hasil Pengujian (Lanjutan 1)

No.	Fitur yang telah diuji	Berhasil	Tidak Berhasil	Time Load (s)
5.	<i>Mass update & delete</i> Halaman Master Activity	✓		0.611
6.	<i>Mass update & delete</i> Halaman Master Area (Op)	✓		0.630
7.	<i>Mass update & delete</i> Halaman Master Asset	✓		1.63
8.	<i>Mass update & delete</i> Halaman Master Asset Item	✓		0.995
9.	<i>Mass update & delete</i> Halaman Master Asset Type	✓		0.606
10.	<i>Mass update & delete</i> Halaman Master Asset Housekeeping	✓		0.597
11.	<i>Mass update & delete</i> Halaman Master Brand Crane	✓		0.605
12.	<i>Mass update & delete</i> Halaman Master BWT Exclude	✓		0.607
13.	<i>Mass update & delete</i> Halaman Master Bariksama Note	✓		0.610
14.	<i>Mass update & delete</i> Halaman Master Contract Type	✓		0.594
15.	<i>Mass update & delete</i> Halaman Master Cargo	✓		1.18
16.	<i>Mass update & delete</i> Halaman Master Cargo Category	✓		0.595
17.	<i>Mass update & delete</i> Halaman Master Cargo Owner	✓		0.610
18.	<i>Mass update & delete</i> Halaman Master Checkcheet Type OPS	✓		0.611
19.	<i>Mass update & delete</i> Halaman Master Currency	✓		0.612

Tabel 4. 96 Hasil Pengujian (Lanjutan 2)

No.	Fitur yang telah diuji	Berhasil	Tidak Berhasil	Time Load (s)
20.	<i>Mass update & delete</i> Halaman Master Document Type	✓		1.18
21.	<i>Mass update & delete</i> Halaman Master Forecast Authority	✓		0.604
22.	<i>Mass update & delete</i> Halaman Master Housekeeping Group	✓		0.605
23.	<i>Mass update & delete</i> Halaman Master Target Cargo	✓		0.780
24.	<i>Mass update & delete</i> Halaman Master Great Day	✓		0.632
25.	<i>Mass update & delete</i> Halaman Master Hatch Type	✓		0.608
26.	<i>Mass update & delete</i> Halaman Master ISPS Code	✓		0.605
27.	<i>Mass update & delete</i> Halaman Master Jetty	✓		0.601
28.	<i>Mass update & delete</i> Halaman Master Laytime	✓		0.618
29.	<i>Mass update & delete</i> Halaman Master Laytime Start Count	✓		0.588
30.	<i>Mass update & delete</i> Halaman Master Liquid Logsheet	✓		0.596
31.	<i>Mass update & delete</i> Halaman Master Location (Op)	✓		0.603
32.	<i>Mass update & delete</i> Halaman Master Material Number	✓		0.608
33.	<i>Mass update & delete</i> Halaman Master NOPOL	✓		0.610
34.	<i>Mass update & delete</i> Halaman Master Origin/Destination Cargo	✓		0.636

Tabel 4. 96 Hasil Pengujian (Lanjutan 3)

No.	Fitur yang telah diuji	Berhasil	Tidak Berhasil	Time Load (s)
35.	<i>Mass update & delete</i> Halaman Master Origin Vessel	✓		0.603
36.	<i>Mass update & delete</i> Halaman Master PG Personel	✓		0.834
37.	<i>Mass update & delete</i> Halaman Master Payrate	✓		0.602
38.	<i>Mass update & delete</i> Halaman Master Penilaian Kinerja Rekanan	✓		0.597
39.	<i>Mass update & delete</i> Halaman Master Permit	✓		0.603
40.	<i>Mass update & delete</i> Halaman Master Plant Code	✓		0.602
41.	<i>Mass update & delete</i> Halaman Master Project Location	✓		0.602
42.	<i>Mass update & delete</i> Halaman Master Port	✓		0.604
43.	<i>Mass update & delete</i> Halaman Master Route	✓		1.23
44.	<i>Mass update & delete</i> Halaman Master Signature	✓		0.613
45.	<i>Mass update & delete</i> Halaman Master Sloc	✓		0.612
46.	<i>Mass update & delete</i> Halaman Master Stand by Cause	✓		0.658
47.	<i>Mass update & delete</i> Halaman Master Surveyor Service	✓		1.18
48.	<i>Mass update & delete</i> Halaman Master Surveyor Category	✓		0.629
49.	<i>Mass update & delete</i> Halaman Master Temperature & Pressure Indicator	✓		0.604

Tabel 4. 96 Hasil Pengujian (Lanjutan 4)

No.	Fitur yang telah diuji	Berhasil	Tidak Berhasil	Time Load (s)
50.	<i>Mass update & delete</i> Halaman Master Trouble (Op)	✓		0.618
51.	<i>Mass update & delete</i> Halaman Master Truck Weighing	✓		0.605
52.	<i>Mass update & delete</i> Halaman Master Transpotir	✓		0.593
53.	<i>Mass update & delete</i> Halaman Master Vendor Personel	✓		0.797
54.	<i>Mass update & delete</i> Halaman Master Vendor Personel Role	✓		0.597
55.	<i>Mass update & delete</i> Halaman Master Vessel	✓		0.889
56.	<i>Mass update & delete</i> Halaman Master Warehouse	✓		7.68
57.	<i>Mass update & delete</i> Halaman Master Work Center (Op)	✓		0.595
58.	Validasi tanggal di halaman <i>Training Realization</i>	✓		0.965
59.	Pembuatan halaman <i>Summary Training Realization</i>	✓		0.666
60.	Export PDF di halaman <i>Summary Training Realization</i>	✓		3.33
61.	Export Excel di halaman <i>Summary Training Realization</i>	✓		0.683
Total Time Load (s)				51.936
Rata-rata Time Load (s)				0.8514

4.6 Perawatan Sistem

Setelah dilakukan pengujian sistem, maka akan masuk pada tahap pemakaian *website* oleh pengguna. Untuk proses pemeliharaan, memungkinkan pengembang untuk melakukan perbaikan terhadap kesalahan yang ditemukan pada *website* setelah digunakan oleh pengguna.



BAB V

PENUTUP

5.1 Simpulan

Berdasarkan hasil dan pembahasan terkait sistem, pengujian sistem dan analisa kebutuhan dari *website* PT Patrakim Timur, maka penulis mengambil kesimpulan sebagai berikut:

1. Terciptanya halaman baru yaitu *Summary Training Realization* beserta fitur *export* PDF & Excel pada *website* sistem informasi pelabuhan kapal PT Patrakim Timur.
2. Telah selesai melakukan perbaikan pada halaman *Training Realization* karena sebelumnya input tahun *Start Date* tidak sesuai dengan data tahun di *Master Data Training*.
3. Terciptanya fitur *mass update & delete* pada halaman *Master Data website* sistem informasi pelabuhan kapal PT Patrakim Timur.
4. Berhasil melewati 61 tahap pengujian yang di uji oleh mentor beserta tim *business analyst* dengan memperoleh presentase keberhasilan sebesar 100% dan rata-rata *time load* 0.8514 detik sehingga dapat disimpulkan bahwa tugas yang dikerjakan sudah sesuai dengan spesifikasi dan bekerja dengan baik.

5.2 Saran

Berdasarkan hasil pengembangan *website* sistem informasi pelabuhan kapal PT Patrakim Timur yang telah dibuat, diharapkan untuk melakukan pengecekan dan percobaan *website* secara berkala agar mengetahui apakah ada *bug* pada *website* tersebut. Diharapkan *website* dapat dikembangkan lagi dengan melakukan perbaikan pada UI (*User Interface*) atau UX (*User Experience*) agar dapat berkembang lebih baik lagi. Hal ini bertujuan agar *website* dapat digunakan dengan mudah dan nyaman.

DAFTAR PUSTAKA

- Agnes, D. S. (2020, Januari 13). *Wripolinema*. Retrieved from Medium: <https://medium.com/wripolinema/sudah-kenal-sama-orm-34712e85c6fa>
- Anitasari, N. (2021, November 4). *Zahir Blog*. Retrieved from Zahir: <https://zahiraccounting.com/id/blog/teknologi-informasi-perusahaan/>
- Aprilia, P. (2022, Mei 7). *Niagahoster Blog*. Retrieved from Niagahoster: https://www.niagahoster.co.id/blog/javascript-adalah/#Apa_itu_Javascript
- Bauroziq. (2022, Februari 28). *Caraguna blogging*. Retrieved from Caraguna: <https://caraguna.com/apa-itu-typescript/>
- Iswara, B. D. (2022, Desember 12). *DosenIT*. Retrieved from DosenIT: <https://dosenit.com/software/dbms/tutorial-postgresql-pengenalan-lengkap-dan-sejarahny>
- Juliarto, R. (2021, Mei 12). *Dicoding Blog*. Retrieved from Dicofing: <https://www.dicoding.com/blog/apa-itu-uml/>
- Mahatmavidya, P. A. (2022, Oktober 7). *Mekari Blog*. Retrieved from Mekari: https://mekari.com/blog/pengertian-erp/#Pengertian_Sistem_ERP
- McCue, I. (2022, Agustus 12). *Oracle netsuite*. Retrieved from netsuite: <https://www.netsuite.com/portal/resource/articles/erp/what-is-erp.shtml>
- Muhardian, A. (2017, Februari 8). *petanikode*. Retrieved from petanikode: <https://www.petanikode.com/git-untuk-pemula/>
- Nayoan, A. (2022, Juni 10). *Niagahoster Blog*. Retrieved from Niagahoster: https://www.niagahoster.co.id/blog/pengertian-css/#Apa_Itu_CSS
- Oktriwina, A. S. (2020, Desember 15). *glints blog*. Retrieved from glints: <https://glints.com/id/lowongan/typescript-adalah/#.Y7WjSHZBzDe>
- Pamungkas, R. B. (2020, Juni 12). *Niagahoster Blog*. Retrieved from Niagahoster: https://www.niagahoster.co.id/blog/react-js-adalah/?gclid=CjwKCAiA2LdBhACEiwAu8Q9YG9vn31BWoNrD1QygSyMFh3cf9ltkGTCdh4aZZdEFZ7oYweARatW3hoCvPAQAvD_BwE
- Prasad, M. (2021, Desember 18). *Medium*. Retrieved from Medium: <https://medium.com/sliit-foss/a-next-generation-orm-prisma-3c3f2b46bd5b>
- Priatna, G. G. (2022, Februari 4). *qadrlabs explore*. Retrieved from qadrlabs: <https://qadrlabs.com/post/tutorial-install-adminer-tool-alternatif-phpmyadmin-di-hp-android#>

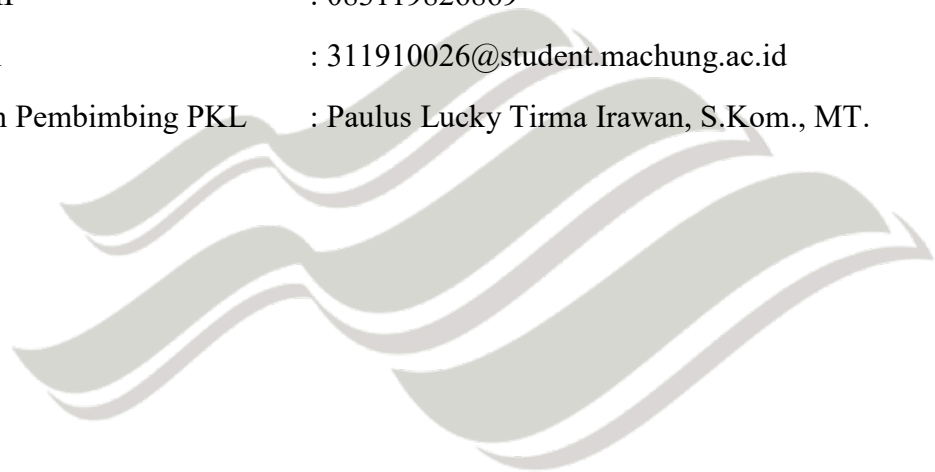
- Puspa, R. (2021, Juli 26). *Alterra Academy Blog*. Retrieved from Alterra Academ: <https://academy.alterra.id/blog/pengertian-dan-fungsi-javascript/>
- Puspasari, E. K. (2022, September 7). *Alterra Academy Blog*. Retrieved from Alterra Academy: <https://academy.alterra.id/blog/node-js-adalah-pengertian-sejarah-fitur%E2%99%A2%E2%99%A3/>
- Raharja, A. D. (2022, Januari 18). *Ekrut Media*. Retrieved from Ekrut: <https://www.ekrut.com/media/css-adalah>
- Roy, S. D. (2022, September 12). *FreeCodeCamp News*. Retrieved from FreeCodeCamp: <https://www.freecodecamp.org/news/what-is-tailwind-css-a-beginners-guide/>



UNIVERSITAS
MA CHUNG

LAMPIRAN A
BIODATA MAHASISWA

Nama : Falachul Akhadihima Ibrahaical
NIM : 311910026
Program Studi : Teknik Informatika
Tempat, Tanggal Lahir : Malang, 13 Mei 2001
Alamat : Jl. Trisula, Arjowilangun, Kalipare, Malang
No. HP : 083119826869
Email : 311910026@student.machung.ac.id
Dosen Pembimbing PKL : Paulus Lucky Tirma Irawan, S.Kom., MT.



UNIVERSITAS
MA CHUNG

LAMPIRAN B

FORM BIMBINGAN DOSEN PEMBIMBING



FAKULTAS
SAINS & TEKNOLOGI
UNIVERSITAS MA CHUNG

Villa Puncak Tidar N-01 Malang 65151
fakultas.sains.teknologi@machung.ac.id
+62 341 550171 www.machung.ac.id

FORM PKL-FST09

LEMBAR BIMBINGAN PRAKTIK KERJA LAPANGAN DARING (PKL)

Nama Mahasiswa	:	Falachul Akhadihima Ibrahaical
NIM	:	311910026
Program Studi	:	Teknik Informatika
Judul Laporan PKL	:	Pengembangan Sistem Informasi Pelabuhan Kapal PT Patrakim Timur Berbasis Website

No	Hari, tanggal	Topik Bimbingan	TTD Dosen Pembimbing
1	15 Agustus 2022	Konsultasi Program Magang dan Studi Independen Bersertifikat	
2	16 November 2022	Progres Kegiatan Selama PKL	
3	02 Desember 2022	Konsultasi Topik Laporan PKL	
4	13 Desember 2022	Konsultasi Judul Laporan dan Ketentuan Format Laporan PKL	
5	06 Januari 2023	Bimbingan Laporan dan Revisi Laporan PKL	
6			
7			
8			
9			
10			



LAMPIRAN C

FORM PARTISIPASI UJIAN PKL



FAKULTAS
SAINS & TEKNOLOGI
UNIVERSITAS MA CHUNG

Villa Puncak Tidar N-01 Malang 65151
fakultas.sains.teknologi@machung.ac.id
+62 341 550171 www.machung.ac.id

FORM PKL-FST11

LEMBAR PARTISIPASI UJIAN PRAKTIK KERJA LAPANGAN DARING (PKL)

Nama Mahasiswa	:	Falachul Akhadihima Ibrahaical
NIM	:	311910026
Program Studi	:	Teknik Informatika

No	Hari, tanggal	Judul PKL	Pemateri PKL (Nama dan NIM)	Nama Dosen Pembimbing Pemateri
1	21 Juli 2022	Pengembangan Fitur Direct Debit dan Home Delivery pada Aplikasi Mobile Customer di PT. Sayap Mas Utama	Natasya Liemena (311910013)	Hendry Setiawan, ST., M.Kom.
2	2 Agustus 2022	Rancang Bangun Website Sistem Informasi Pelabuhan Kapal PT. Petrokimia Gresik	Karisna Rachmazaki (311910017)	Paulus Lucky Tirma Irawan, S.Kom., MT.
3	20 Juni 2022	Rancang Bangun Sistem Gaji Berdasarkan Absensi dengan Menggunakan PHP di CV. Girisa Teknologi	Christopher (311810010)	Ir. Oesman Hendra Kelana, M.Cs
4	20 Desember 2022	Rancang Bangun List Aplikasi IT Support Berbasis Web Menggunakan Framework Laravel di PT. Pegadaian	David Try Setyo Wicaksono (311810011)	Dr.Eng. Romy Budhi, ST., MT.
5	30 Juni 2022	Pengembangan Sistem Informasi Web Point of Sales Koperasi Konsumen Ma Chung Mandiri Sejahtera	Adam Lazuardi Haris (311510001)	Mochamad Subianto, S.Kom., M.Cs.



LAMPIRAN D

FORM PENILAIAN PEMBIMBING LAPANGAN



FAKULTAS
SAINS & TEKNOLOGI
UNIVERSITAS MA CHUNG

Villa Puncak Tidar N-01 Malang 65151
✉ fakultas.sains.teknologi@machung.ac.id
☎ +62 341 550171 🌐 www.machung.ac.id

FORM PKL-FST04

LEMBAR PENILAIAN PRAKTIK KERJA LAPANGAN (PKL) PIHAK EKSTERNAL

Nama Mahasiswa	: Falachul Akhadihima Ibrahaical
NIM	: 311910026
Program Studi	: Teknik Informatika
Pembimbing Lapangan	: M. Miftahul Nizam, S.Kom.
Tanggal Pelaksanaan	: 18 Agustus – 31 Desember 2022
Tempat PKL	: PT Andromedia
Topik PKL	: Pengembangan Sistem Informasi Pelabuhan Kapal Berbasis Website

Komponen Penilaian :

No	Macam Kegiatan yang dinilai	Nilai Angka (0-100)
1	Disiplin dalam kehadiran (presensi)	100
2	Kesungguhan dalam melakukan praktik kerja	88
3	Disiplin dalam pekerjaan	88
4	Kemampuan memecahkan masalah	82
5	Tanggung jawab	85
6	Kemauan untuk mengetahui hal-hal yang ada di tempat praktik kerja	90
7	Pengetahuan tentang ilmu yang dilaksanakan	82
8	Keterampilan	83
9	Kemampuan menyampaikan pendapat	84
10	Kemampuan dalam bekerja sama	82
Rerata (Jumlah Nilai : 10 komponen)		86.4

MUTU	RENTANG	BOBOT NILAI
A	95,00 s/d 100	4,00
AB	90,00 s/d 94,99	3,70
BA	82,00 s/d 89,99	3,30
B	73,00 s/d 81,99	3,00
BC	65,00 s/d 72,99	2,70
CB	60,00 s/d 64,99	2,30
C	56,00 s/d 59,99	2,00
CD	50,00 s/d 55,99	1,50
D	40,00 s/d 49,99	1,00
E	00,00 s/d 39,99	0,00

Surabaya, 15 Desember 2022

Pembimbing Lapangan*


(M. Miftahul Nizam, S.Kom.)
(Technical Manager)

*dilengkapi dengan stempel perusahaan dan tanda tangan

