



**TUGAS AKHIR - KI141502**

# **Rancang Bangun Modul Perencanaan dan Kontrol Produksi untuk Optimasi Makespan Proses Produksi Manufaktur Menggunakan Metode EDD**

**ALIFA IZZAN AKHSANI  
NRP 05111640000144**

Dosen Pembimbing I  
Prof. Drs. Ec. Ir. Riyanarto Sarno M.Sc., Ph.D

Dosen Pembimbing II  
Dwi Sunaryono S.Kom., M.Kom.

Departemen Informatika  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020









**TUGAS AKHIR - KI141502**

# **Rancang Bangun Modul Perencanaan dan Kontrol Produksi untuk Optimasi Makespan Proses Produksi Manufaktur Menggunakan Metode EDD**

**ALIFA IZZAN AKHSANI  
NRP 05111640000144**

**Dosen Pembimbing I  
Prof. Drs. Ec. Ir. Riyanarto Sarno M.Sc., Ph.D**

**Dosen Pembimbing II  
Dwi Sunaryono S.Kom., M.Kom.**

**Departemen Informatika  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020**

*(Halaman ini sengaja dikosongkan)*



**UNDERGRADUATE THESES - KI141502**

# **Design of Production Planning and Control Modules for Optimizing Makespan of Manufacturing Production Processes Using EDD Method**

**ALIFA IZZAN AKHSANI  
NRP 0511164000144**

**First Advisor**

**Prof. Drs. Ec. Ir. Riyanarto Sarno M.Sc., Ph.D**

**Second Advisor**

**Dwi Sunaryono S.Kom., M.Kom.**

**Department of Informatics  
Faculty of Information Technology and Communication  
Sepuluh Nopember Institute of Technology  
Surabaya 2020**

*(Halaman ini sengaja dikosongkan)*

## LEMBAR PENGESAHAN

### Rancang Bangun Modul Perencanaan dan Kontrol Produksi untuk Optimasi Makespan Proses Produksi Manufaktur Menggunakan Metode EDD

#### TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Bidang Studi Manajemen Informasi  
Program Studi S-1 Departemen Informatika  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember

Oleh:

**ALIFA IZZAN AKHSANI**  
**NRP: 05111640000144**

Disetujui oleh Pembimbing Tugas Akhir:

1. Prof. Drs. Ec. Ir. Riyanarto Sarno M.Sc. Ph.D.  
(NIP. 195908031986011001)

  
(Pembimbing 1)

2. Dwi Sunaryono S.Kom., M.Kom.  
(NIP. 197205281997021001)

  
(Pembimbing 2)

**SURABAYA**  
**JUNI, 2020**

*(Halaman ini sengaja dikosongkan)*

# **RANCANG BANGUN MODUL PERENCANAAN DAN KONTROL PRODUKSI UNTUK OPTIMASI *MAKESPAN* PROSES PRODUKSI MANUFAKTUR MENGGUNAKAN METODE EDD**

**Nama Mahasiswa** : ALIFA IZZAN AKHSANI  
**NRP** : 05111640000144  
**Departemen** : Informatika FTEIC-ITS  
**Dosen Pembimbing 1** : Prof. Drs. Ec. Ir. Riyanarto Sarno  
M.Sc., Ph.D  
**Dosen Pembimbing 2** : Dwi Sunaryono S.Kom., M.Kom.

## **Abstrak**

Kegiatan manufaktur adalah kegiatan yang di lakukan dengan menggunakan tenaga kerja, mesin, dan peralatan lainnya untuk menghasilkan barang. Dalam melakukan kegiatan manufaktur diperlukan perencanaan dan kontrol. Perencanaan disini adalah merencanakan kapasitas produksi, dan bahan baku produksi.

Perencanaan kapasitas produksi disebut CRP (*Capacity Requirement Planning*) di dalamnya merencanakan mesin yang ada dan kapasitasnya, juga merencanakan alur produksi. Selanjutnya ada perencanaan bahan baku yang disebut MRP (*Material Requirement Planning*).

Salah satu yang perlu dikontrol dalam kegiatan produksi adalah penjadwalan dari kegiatan produksi. Penjadwalan adalah pengalokasian sumber daya baik kapasitas maupun bahan baku yang terbatas dari waktu ke waktu untuk digunakan secara optimal.

Tipe perencanaan penjadwalan bergantung pada kondisi tata letak dan alur dari mesin yang ada pada lantai produksi. Setiap mesin dapat memproses satu pekerjaan pada satu waktu. Secara umum permasalahan penjadwalan *flow shop*  $F_m$  dan *job shop*  $J_{nm}$  dengan tujuan *makespan*  $C_{max}$  minimum tergolong *NP-hard* oleh karena itu modul Perencanaan dan Kontrol Produksi yang dihasilkan dari tugas akhir ini diharapkan dapat digunakan untuk mencari metode

yang optimal dalam menyelesaikan permasalahan tersebut. Metode yang di usulkan dalam tugas akhir ini adalah *Earliest Due Date* (EDD). Penjadwalan EDD menghasilkan makespan 3.41% lebih pendek dari pada metode yang digunakan pada PT.P.

***Kata kunci: Job Shop, Flow Shop, Manufaktur, makespan, EDD, MRP, CRP, Penjadwalan***

**DESIGN OF PRODUCTION PLANNING AND CONTROL  
MODULES FOR OPTIMIZING MAKESPAN OF  
MANUFACTURING PRODUCTION PROCESSES USING  
EDD METHOD**

**Student's Name** : ALIFA IZZAN AKHSANI  
**Student's ID** : 05111640000144  
**Department** : Informatics – FTEIC ITS  
**First Advisor** : Prof. Drs. Ec. Ir.Riyanarto Sarno M.Sc.,  
Ph.D  
**Second Advisor** : Dwi Sunaryono S.Kom., M.Kom.

*Abstract*

Manufacturing activities are carried out using labor, machinery, and other equipment to produce goods. In carrying out manufacturing activities, planning and control are needed. Types of planning are production capacity, and raw materials.

The proses of discrening available production is called CRP (Capacity Requirement Planning) in which production capability and production flow are assesed. Then there is the planning of raw materials called MRP (Material Requirement Planning).

One that needs to be controlled in production activities is the scheduling of production activities. Scheduling is the allocation of resources both limited capacity and raw materials from time to time to be used optimally.

The type of scheduling planning depends on the layout and flow conditions of the machines on the production floor. Each machine can process one job at a time. In general, the problem of scheduling flow shop  $F_m$  and job shop  $J_{nm}$  with the aim of making  $C_{max}$  minimum makespan classified as NP-hard, therefore the Production Planning and Control module generated from this final project is expected to be used to find the optimal method for resolving these problems. The method proposed in this thesis is

Earliest Due Date (EDD). EDD scheduling produces makespan 3.41% shorter than the method used at PT.P.

***Keyword: Job Shop, Flow Shop, Manufacture, makespan, EDD, MRP, CRP, Scheduling***

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji syukur kepada Allah Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul **“Rancang Bangun Modul Perencanaan dan Kontrol Produksi untuk Optimasi Makespan Proses Produksi Manufaktur Menggunakan Metode EDD”**.

Harapan penulis semoga apa yang tertulis di dalam buku Tugas Akhir ini dapat bermanfaat bagi pengembangan ilmu pengetahuan saat ini dan ke depannya, serta dapat memberikan kontribusi yang nyata.

Dalam pelaksanaan dan pembuatan Tugas Akhir ini tentunya sangat banyak bantuan yang penulis terima dari berbagai pihak, tanpa mengurangi rasa hormat penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Allah SWT atas segala rahmat yang diberikan sehingga penulis dapat menyelesaikan Tugas Akhir ini.
2. Keluarga penulis terutama Bapak dan Ibu selaku orang tua penulis dan adik-adik penulis atas segala dukungan berupa motivasi, doa, moral, dan material sehingga penulis tetap semangat dan dapat menyelesaikan Tugas Akhir ini.
3. Bapak Prof. Drs. Ec. Ir.Riyanarto Sarno M.Sc., Ph.D., dan Bapak Dwi Sunaryono S.Kom., M.Kom. selaku dosen pembimbing penulis atas nasihat, arahan dan bantuannya sehingga penulis dapat menyelesaikan Tugas Akhir ini.
4. Teman dekat penulis mas anang dan putri yang sudah membantu memberikan dorongan, serta saran dalam menyelesaikan Tugas Akhir ini.

Penulis telah berusaha sebaik-baiknya dalam menyusun Tugas Akhir ini, namun penulis menyadari bahwa masih terdapat kekurangan, kesalahan, maupun kelalaian yang telah penulis lakukan. Oleh karena itu, saran dan kritik yang membangun sangat dibutuhkan untuk penyempurnaan Tugas Akhir ini.

Surabaya, Juni 2020

ALIFA IZZAN AKHSANI

## DAFTAR ISI

Abstrak	vii
<i>Abstract</i>	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xxi
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Permasalahan.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Metodologi.....	3
1.6.1 Penyusunan Proposal Tugas Akhir.....	4
1.6.2 Studi Literatur.....	4
1.6.3 Analisis dan Desain Sistem.....	4
1.6.4 Implementasi Sistem.....	4
1.6.5 Pengujian dan Evaluasi.....	5
1.6.6 Penyusunan Buku.....	5
1.7 Sistematika Penulisan Laporan.....	5
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>7</b>
2.1 <i>Manufacturing</i> .....	7
2.2 <i>Lean Manufacturing</i> .....	8
2.3 <i>Flow Time, Makespan, Tardines</i> .....	8
2.4 Flow Shop Scheduling Problem.....	8
2.5 <i>Job Shop Scheduling Problem</i> .....	10
2.6 <i>ERP (Enterprise Resource Planning)</i> .....	11
2.7 PKP (Perencanaan dan Kontrol Produksi).....	13
2.8 Aggregate Planning.....	14
2.9 Sales & Operations Planning.....	15
2.10 Resource Requirement Planning.....	15
2.11 Master Production Schedules.....	15

2.12	Rough Cut Capacity Planning .....	15
2.13	Capacity Requirements Planning .....	16
2.14	Materials Requirements Planning .....	16
2.15	Production Scheduling .....	18
2.16	Make to Order .....	19
2.17	Gantt Chart.....	19
2.18	Laravel .....	19
2.19	UML .....	20
<b>BAB III</b>	<b>PERANCANGAN .....</b>	<b>20</b>
3.1	Deskripsi Umum .....	21
3.2	Perancangan Optimasi Makespan.....	22
3.3	Perancangan Metode EDD .....	24
3.4	Perancangan Lingkungan Pengembangan .....	24
3.5	Perancangan Modul Perencanaan dan Kontrol Produksi	27
3.6	Perancangan Menu MRP.....	28
3.7	Perancangan Menu CRP .....	29
3.8	Perancangan Menu Product Scheduling .....	31
<b>BAB IV</b>	<b>IMPLEMENTASI .....</b>	<b>32</b>
4.1	Implementasi Optimasi Makespan.....	32
4.1.1	Batasan .....	32
4.1.2	Penjadwalan Manual .....	35
4.1.3	Kalkulasi Makespan.....	36
4.2	Implementasi Metode EDD .....	37
4.2.1	Event .....	37
4.2.2	Handler .....	38
4.3	Implementasi Lingkungan Pengembangan .....	40
4.3.1	Database Model.....	41
4.3.2	Konfigurasi Routing Web Server.....	86
4.4	Implementasi Modul Perencanaan dan Kontrol Produksi	89
4.4.1	Implementasi Daftar Fitur pada Menu MRP.....	90
4.4.2	Implementasi Daftar Fitur Menu CRP.....	91
4.4.3	Implementasi Daftar Menu Production Scheduling	91
4.4.4	Implementasi Daftar Menu Report.....	92

4.5	Implementasi MENU MRP .....	93
4.6	Implementasi Menu CRP .....	100
4.7	Implementasi Menu Production Scheduling .....	103
4.8	Implementasi Menu Report .....	106
<b>BAB V</b>	<b>UJICOBA DAN EVALUASI .....</b>	<b>108</b>
5.1	Lingkungan Uji Coba .....	108
5.2	Hasil Uji Coba.....	109
5.2.1	Hasil Uji Coba Metode Penjadwalan EDD .....	109
5.2.2	Hasil Uji Coba Fungsi Program .....	111
5.2.3	Hasil Uji Coba Performa Sistem .....	116
<b>BAB VI</b>	<b>KESIMPULAN DAN SARAN .....</b>	<b>120</b>
6.1	Kesimpulan .....	120
6.2	Saran.....	120
<b>DAFTAR PUSTAKA</b>	<b>.....</b>	<b>122</b>
<b>LAMPIRAN</b>	<b>.....</b>	<b>124</b>
A.1	Kode Fungsi untuk Konstraint Penjadwalan .....	124
A.2	Kode Fungsi untuk Konstraint Penjadwalan 2 .....	124
A.2	Kode Fungsi untuk Menjadwalkan EDD .....	124
<b>BIODATA PENULIS</b>	<b>.....</b>	<b>126</b>

*(Halaman ini sengaja dikosongkan)*

## DAFTAR GAMBAR

<i>Gambar 3.1</i>	<i>Gambaran Umum Modul PKP</i> .....	21
<i>Gambar 3.2</i>	<i>Desain Algoritma Metode EDD</i> .....	24
<i>Gambar 3.3</i>	<i>Kerangka MVC Laravel</i> .....	25
<i>Gambar 3.4</i>	<i>Menu utama MRP</i> .....	27
<i>Gambar 3.5</i>	<i>Menu Utama CRP</i> .....	27
<i>Gambar 3.6</i>	<i>Menu Utama Production Scheduling</i> .....	28
<i>Gambar 3.7</i>	<i>ERD Menu Material Requirement Planning</i> .....	29
<i>Gambar 3.8</i>	<i>ERD Menu Capacity Requirement Planning</i> .....	30
<i>Gambar 3.9</i>	<i>Menu Utama Production Scheduling</i> .....	31
<i>Gambar 4.1</i>	<i>Potongan Kode Program Batasan 1</i> .....	33
<i>Gambar 4.2</i>	<i>Potongan Kode Program Batasan 2</i> .....	34
<i>Gambar 4.3</i>	<i>Potongan Kode Implementasi Penjadwalan Manual</i>	35
<i>Gambar 4.4</i>	<i>Potongan Kode Kalkulasi Makespan</i> .....	36
<i>Gambar 4.5</i>	<i>Potongan Kode Kelas Event EDDButtonPressEvent</i>	38
<i>Gambar 4.6</i>	<i>Potongan Kode Handler EDDSubscriber</i> .....	39
<i>Gambar 4.7</i>	<i>Penggunaan Event dan Handler untuk Melakukan Metode EDD</i> .....	40
<i>Gambar 4.8</i>	<i>Kode Program untuk Migrasi Tabel Uoms</i> .....	42
<i>Gambar 4.9</i>	<i>Kode Program untuk Migrasi Tabel PurchaseOrders</i> .....	43
<i>Gambar 4.10</i>	<i>Kode Program untuk Migrasi Tabel PurchaseItems</i> .....	44
<i>Gambar 4.11</i>	<i>Kode Program untuk Migrasi Tabel Products</i> .....	45
<i>Gambar 4.12</i>	<i>Kode Program untuk Migrasi Tabel Inventories</i> ....	46
<i>Gambar 4.13</i>	<i>Kode Program untuk Migrasi Tabel BOMS</i> .....	47
<i>Gambar 4.14</i>	<i>Kode Program untuk Migrasi Tabel BOMComponents</i> .....	48
<i>Gambar 4.15</i>	<i>Kode Program untuk Migrasi Tabel Orders</i> .....	49
<i>Gambar 4.16</i>	<i>Kode Program untuk Migrasi Tabel Reserves</i> .....	50
<i>Gambar 4.17</i>	<i>Kode Program untuk Migrasi Tabel WorkCenters</i> .	51
<i>Gambar 4.18</i>	<i>Kode Program untuk Migrasi Tabel Operations</i> ....	52
<i>Gambar 4.19</i>	<i>Kode Program untuk Migrasi Tabel Routings</i> .....	53
<i>Gambar 4.20</i>	<i>Kode Program untuk Migrasi Tabel Capacities</i> .....	54

<b>Gambar 4.21</b>	<i>Kode Program untuk Migrasi Tabel WorkOrders ..</i>	55
Gambar 4.22	Perintah Konsol Artisan Migrate .....	56
<b>Gambar 4.23</b>	<i>Kode Program untuk Model Operations.....</i>	57
<b>Gambar 4.24</b>	<i>Kode Program untuk Kolom Model Operations.....</i>	57
<b>Gambar 4.25</b>	<i>Kode Program untuk Relasi Model Operations.....</i>	58
<b>Gambar 4.26</b>	<i>Kode Program untuk aksi relasi Model Operation.....</i>	59
<b>Gambar 4.27</b>	<i>Kode Program untuk Model BOM .....</i>	59
<b>Gambar 4.28</b>	<i>Kode Program untuk kolom Model BOM .....</i>	60
<b>Gambar 4.29</b>	<i>Kode Program untuk relasi Model BOM.....</i>	60
<b>Gambar 4.30</b>	<i>Kode Program untuk aksi relasi Model BOM .....</i>	61
<b>Gambar 4.31</b>	<i>Kode Program untuk Model Inventory .....</i>	62
<b>Gambar 4.32</b>	<i>Kode Program untuk kolom Model Inventory.....</i>	62
<b>Gambar 4.33</b>	<i>Kode Program untuk relasi Model Inventory .....</i>	63
<b>Gambar 4.34</b>	<i>Kode Program untuk aksi relasi Model Inventory..</i>	64
<b>Gambar 4.35</b>	<i>Kode program untuk Model Order .....</i>	64
<b>Gambar 4.36</b>	<i>Kode program untuk kolom Model Order .....</i>	65
<b>Gambar 4.37</b>	<i>Kode program untuk relasi Model Order .....</i>	66
<b>Gambar 4.38</b>	<i>Kode program untuk aksi relasi Model Order.....</i>	67
<b>Gambar 4.39</b>	<i>Kode program untuk Model Product .....</i>	67
<b>Gambar 4.40</b>	<i>Kode program untuk kolom Model Product .....</i>	68
<b>Gambar 4.41</b>	<i>Kode program untuk relasi Model Product .....</i>	69
<b>Gambar 4.42</b>	<i>Kode Program untuk Aksi Relasi Model Product... </i>	71
<b>Gambar 4.43</b>	<i>Kode Program untuk Model Purchase Order.....</i>	72
<b>Gambar 4.44</b>	<i>Kode Program untuk Kolom Model Purchase Order .....</i>	73
<b>Gambar 4.45</b>	<i>Kode Program untuk Relasi Model Purchase Order .....</i>	74
<b>Gambar 4.46</b>	<i>Kode Program untuk Aksi Relasi Model Purchase Order .....</i>	75
<b>Gambar 4.47</b>	<i>Kode Program untuk Model Routing .....</i>	76
<b>Gambar 4.48</b>	<i>Kode Program untuk Kolom Model Routing .....</i>	76
<b>Gambar 4.49</b>	<i>Kode Program untuk Relasi Model Routing .....</i>	77
<b>Gambar 4.50</b>	<i>Kode Program untuk Aksi Relasi Model Routing ...</i>	78
<b>Gambar 4.51</b>	<i>Kode Program untuk Model UoM .....</i>	79
<b>Gambar 4.52</b>	<i>Kode Program untuk Kolom Model UoM.....</i>	79

<b>Gambar 4.53</b>	<i>Kode Program untuk Relasi Model UoM</i>	80
<b>Gambar 4.54</b>	<i>Kode Program untuk Aksi Relasi Model UoM</i>	80
<b>Gambar 4.55</b>	<i>Kode Program untuk Model Workcenter</i>	81
<b>Gambar 4.56</b>	<i>Kode Program untuk Kolom Model Workcenter</i>	81
<b>Gambar 4.57</b>	<i>Kode Program untuk Relasi Model Workcenter</i>	82
<b>Gambar 4.58</b>	<i>Kode Program untuk Aksi Relasi Model Workcenter</i>	83
<b>Gambar 4.59</b>	<i>Kode Program untuk Model WorkOrders</i>	83
<b>Gambar 4.60</b>	<i>Kode Program untuk Kolom Model WorkOrders</i>	84
<b>Gambar 4.61</b>	<i>Kode Program untuk Relasi Model WorkOrders</i>	84
<b>Gambar 4.62</b>	<i>Hasil Implementasi Database Model</i>	85
Gambar 4.63	Laravel General Routing	86
<b>Gambar 4.64</b>	<i>Laravel MRP Routing</i>	87
Gambar 4.65	Laravel PS Routing	88
<b>Gambar 4.66</b>	<i>Laravel CRP Routing</i>	89
<b>Gambar 4.67</b>	Antar Muka Home	90
<b>Gambar 4.68</b>	Antar Muka Menu MRP	90
<b>Gambar 4.69</b>	Antar Muka Menu CRP	91
<b>Gambar 4.70</b>	Antar Muka Menu PS	92
<b>Gambar 4.71</b>	Antar Muka Menu Report	92
<b>Gambar 4.72</b>	Antar Muka Menu UoM	93
<b>Gambar 4.73</b>	Antar Muka Tambah UoM	94
<b>Gambar 4.74</b>	Antar Muka Produk	94
<b>Gambar 4.75</b>	Antar Muka Tambah Produk	95
<b>Gambar 4.76</b>	Antar Muka BOM	95
<b>Gambar 4.77</b>	Antar Muka Tambah BOM	96
<b>Gambar 4.78</b>	Antar Muka Stock	97
<b>Gambar 4.79</b>	Antar Muka Tambah Stock	97
<b>Gambar 4.80</b>	Antar Muka Purchase	98
<b>Gambar 4.81</b>	Antar Muka Show Purchase	99
<b>Gambar 4.82</b>	Antar Muka Tambah Item Purchase	99
<b>Gambar 4.83</b>	Antar Muka Workcenter	100
<b>Gambar 4.84</b>	Antar Muka Tambah Workcenter	101
<b>Gambar 4.85</b>	Antar Muka Routing	101
<b>Gambar 4.86</b>	Antar Muka Tambah Routing	102

<b>Gambar 4.87</b> Antar Muka Show Routing .....	103
<b>Gambar 4.88</b> Antar Muka Order.....	103
<b>Gambar 4.89</b> Antar Muka Tambah Order.....	104
<b>Gambar 4.90</b> Antar Muka Show Order.....	105
<b>Gambar 4.91</b> Menu Show Makespan .....	106

## DAFTAR TABEL

<b>Tabel 2.2.1</b> Keluaran dan Masukan MRP .....	17
<b>Tabel 2.2.2</b> Aktivitas dari MRP .....	18
<b>Tabel 3.1</b> <i>Daftar Istilah</i> .....	22
Tabel 3.2 Versi kerangka kerja Laravel .....	25
Tabel 3.3 <i>Versi basis data</i> PostgreSQL.....	26
Tabel 3.4 Versi server website .....	<b>Error! Bookmark not defined.</b>
<b>Tabel 5.1</b> <i>Spesifikasi Perangkat yang Digunakan</i> .....	108
<b>Tabel 5.2</b> <i>Data Uji PT.P</i> .....	109
<b>Tabel 5.3</b> Hasil Pengujian Metode EDD .....	110
<b>Tabel 5.4</b> <i>Urutan Pekerjaan Metode EDD</i> .....	111
<b>Tabel 5.5</b> <i>Tabel Hasil Uji Fungsi Menu MRP</i> .....	111
<b>Tabel 5.6</b> <i>Hasil Uji Fungsi Menu CRP</i> .....	113
<b>Tabel 5.7</b> <i>Hasil Uji Fungsi Menu Production Scheduling</i> .....	114
<b>Tabel 5.8</b> <i>Hasil Uji Fungsi Menu Report</i> .....	115
<b>Tabel 5.9</b> <i>Tabel Hasil Uji Coba Performa Menu MRP</i> .....	<b>Error! Bookmark not defined.</b>
<b>Tabel 5.10</b> <i>Tabel Hasil Uji Coba Performa Menu CRP</i> .....	117
<b>Tabel 5.11</b> <i>Tabel Hasil Uji Coba Performa Menu PS</i> .....	118
<b>Tabel 5.12</b> <i>Tabel Hasil Uji Coba Performa Menu Report</i> .....	119

*(Halaman ini sengaja dikosongkan)*

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Kegiatan manufaktur adalah kegiatan produksi yang dilakukan dengan menggunakan tenaga kerja, mesin, dan peralatan lainnya untuk menghasilkan barang. Salah satu perusahaan manufaktur di Indonesia yaitu PT. P merupakan badan usaha dalam bidang industri manufaktur dan alat kesehatan yang didirikan pada tahun 2017. Untuk mencapai produksi yang optimal, sebuah perusahaan harus menerapkan standar proses dalam melakukan kegiatan produksi di samping menerapkan teknologi tercanggih dalam teknologi manufaktur [1].

Tata letak mesin produksi pada pabrik disusun berdasarkan proses produksi. Mereka menerapkan sistematisa lean manufacturing dalam manajemen produksinya dengan tujuan mengurangi pemborosan sumber daya, sehingga menghasilkan produk dengan kualitas terbaik dengan harga bersaing.

Dalam melakukan kegiatan produksi, PT. P menggunakan alur *flow shop* (alur di mana pekerjaan lintas produk yang berurutan) berdasarkan banyaknya variasi produk yang ditawarkan dan tata letak mesin produksi yang diterapkan. Dalam produksinya PT. P kesulitan dalam menentukan penjadwalan proses produksi. Hal itu dikarenakan PT. P belum memiliki modul perencanaan dan kontrol produksi.

Untuk merencanakan kapasitas dan mengukur tingkat layanan pabrik maka diperlukan sebuah modul perencanaan dan kontrol produksi. Di dalam modul perencanaan dan kontrol produksi terdapat metode pengaturan pekerjaan mana yang di kerjakan terlebih dahulu. Metode EDD (*earliest due date*) atau pengerjaan pekerjaan berdasarkan tenggat waktu pekerjaan paling awal adalah salah satu metode pengaturan tersebut. Untuk mengukur performa dari metode tersebut dapat ditinjau dari tiga

parameter yaitu: *mean flow time*  $F = \frac{1}{n} \sum_{i=1}^n F_i$  [1], *makespan*  $C_{max}$ , dan *mean tardiness*  $T_s = \frac{1}{n} \sum_{i=1}^n T_i$  [1] [2]. Secara umum permasalahan penjadwalan *flow shop* pada kasus  $F_m || C_{max}$  tergolong NP-hard [2] namun solusi dari [3] dapat dijadikan referensi pendekatan penyelesaian permasalahan ini menggunakan metode SPT-LPT, selain itu juga terdapat pendekatan secara *heuristic* seperti pada [4] dengan studi kasus industri manufaktur. Pemilihan metode EDD pada riset ini dikarenakan metode tersebut merupakan salah satu metode yang paling dasar dalam menyelesaikan permasalahan ini.

Untuk dapat membuat rancang bangun modul perencanaan dan kontrol produksi diperlukan pemahaman proses bisnis sektor industri manufaktur secara umum. Proses bisnis ini telah dijabarkan menggunakan notasi BPMN pada referensi [5] dan [6], gambaran umum tersebut juga sesuai dengan definisi sistem ERP pada [7] dan digunakan pada riset ini untuk merancang modul perencanaan dan kontrol produksi.

Pada Tugas Akhir ini, penulis membuat sebuah rancang bangun modul perencanaan dan kontrol produksi untuk optimasi *makespan* proses produksi manufaktur menggunakan EDD. Hasil akhir yang diharapkan adalah menghasilkan sebuah modul perencanaan dan kontrol produksi yang lebih efisien dalam segi waktu proses produksi pada industry manufaktur dengan menggunakan metod EDD.

## 1.2 Rumusan Masalah

Tugas Akhir ini mengangkat beberapa rumusan masalah sebagai berikut:

1. Bagaimana kinerja metode EDD untuk optimasi *makespan* dibandingkan dengan metode lainnya?
2. Bagaimana implementasi modul perencanaan dan kontrol produksi untuk optimasi *makespan*?

### **1.3 Batasan Permasalahan**

Permasalahan yang dibahas pada Tugas Akhir ini memiliki batasan sebagai berikut:

1. Implementasi Modul Perencanaan dan Kontrol Produksi terbatas hanya untuk:
  - a. Materials requirements planning
  - b. Production scheduling
  - c. Capacity requirements planning
  - d. Urutan produksi flow shop
2. Kerangka kerja yang digunakan adalah Laravel.
3. Optimasi yang dilakukan untuk kasus proses industri manufaktur pada PT.P

### **1.4 Tujuan**

Tujuan dari Tugas Akhir ini membuat rancang bangun modul perencanaan dan kontrol produksi guna menguji metode EDD untuk optimasi makespan pada proses produksi manufaktur.

### **1.5 Manfaat**

Manfaat yang diperoleh dari pengerjaan Tugas Akhir ini adalah menghasilkan rancang bangun modul perencanaan dan kontrol produksi guna menguji metode EDD untuk optimasi makespan pada kegiatan produksi pada industri manufaktur.

### **1.6 Metodologi**

Pembuatan Tugas Akhir ini dilakukan dengan menggunakan metodologi sebagai berikut:

### **1.6.1 Penyusunan Proposal Tugas Akhir**

Tahapan awal dari Tugas Akhir ini adalah penyusunan Proposal Tugas Akhir. Proposal Tugas Akhir berisi pendahuluan, deskripsi dan gagasan metode-metode yang dibuat dalam Tugas Akhir ini. Pendahuluan ini terdiri atas hal yang menjadi latar belakang diajukannya Tugas Akhir, rumusan masalah yang diangkat, batasan masalah untuk Tugas Akhir, dan manfaat dari hasil pembuatan Tugas Akhir ini. Selain itu dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan Tugas Akhir.

### **1.6.2 Studi Literatur**

Pada Pada tahap ini akan dilakukan pendalaman studi literatur yang relevan untuk dijadikan referensi dalam melakukan pengerjaan Tugas Akhir. Studi literatur didapatkan dari buku, internet, dan materi-materi kuliah yang berhubungan dengan metode yang akan digunakan yaitu Laravel, ERP, MRP, CRP, PS, JSSP, Flow Shop, Scheduling, Gantt Chart.

### **1.6.3 Analisis dan Desain Sistem**

Pada tahap ini dilakukan analisis dari literatur yang ada. Untuk menghasilkan desain implementasi dari perangkat lunak yang akan dibuat. Untuk analisa menggunakan bantuan ERD yang dibuat berdasarkan definisi sistem/modul terkait sesuai dasar teori dan literatur yang ada. Desain sistem menggunakan bahasa UML sebagai bahasa untuk merancang ERD.

### **1.6.4 Implementasi Sistem**

Implementasi merupakan tahap untuk membangun metode-metode yang sudah diajukan pada proposal Tugas Akhir. Pada tahap ini dilakukan implementasi menggunakan Laravel sebagai kerangka kerja dimana menggunakan fitur Eloquent untuk

memodelkan data. PostgreSQL sebagai basis data, dan bahasa PHP sebagai bahasa pemrograman.

### **1.6.5 Pengujian dan Evaluasi**

Pada tahap ini pengujian dilakukan secara manual, mulai dari kelengkapan fitur, uji performa, dan uji dataset yang didapat dari PT.P untuk menguji apakah metode yang diajukan dapat memberikan hasil yang lebih baik dari metode yang dipakai di dalam PT.P

### **1.6.6 Penyusunan Buku**

Pada tahap ini dilakukan penyusunan buku yang menjelaskan seluruh konsep, teori dasar dari metode yang digunakan, implementasi, serta hasil yang telah dikerjakan sebagai dokumentasi dari pelaksanaan Tugas Akhir.

## **1.7 Sistematika Penulisan Laporan**

Sistematika penulisan laporan Tugas Akhir adalah sebagai berikut:

### **1. Bab I. Pendahuluan**

Bab ini berisikan penjelasan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika penulisan dari pembuatan Tugas Akhir.

### **2. Bab II. Tinjauan Pustaka**

Bab ini berisi kajian teori atau penjelasan dari metode, algoritma, *library*, dan *tools* yang digunakan dalam penyusunan Tugas Akhir ini. Bab ini berisi tentang penjelasan singkat mengenai Laravel, ERP, MRP, CRP, PS, JSSP, Flow Shop, Scheduling, Gantt Chart.

### **3. Bab III. Perancangan**

Bab ini berisi pembahasan mengenai perancangan menu-menu yang ada berdasarkan kajian literatur. Lalu dari menu yang

sudah di rancang dianalisa untuk bentuk entitas serta relasi antar entitasnya dan menghasilkan ERD dari perangkat lunak yang akan dibangun guna menguji metode yang akan diuji.

4. Bab IV. Implementasi

Bab ini menjelaskan implementasi yang berbentuk kode sumber dari perangkat lunak yang telah di desain dan tampilan dari antar muka untuk pengguna

5. Bab V. Uji Coba dan Evaluasi

Bab ini berisikan hasil uji coba dan evaluasi dari implementasi yang telah dilakukan untuk menyelesaikan masalah yang dibahas pada Tugas Akhir.

6. Bab VI. Kesimpulan dan Saran

Bab ini merupakan bab yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan, masalah-masalah yang dialami pada proses pengerjaan Tugas Akhir, dan saran untuk pengembangan solusi ke depannya.

7. Daftar Pustaka

Bab ini berisi daftar pustaka yang dijadikan literatur dalam Tugas Akhir.

8. Lampiran

Dalam lampiran terdapat tabel-tabel data hasil uji coba dan kode sumber program secara keseluruhan.

## **BAB II**

### **TINJAUAN PUSTAKA**

Bab ini berisi pembahasan mengenai teori-teori dasar atau penjelasan dari metode dan *tools* yang digunakan dalam Tugas Akhir.

#### **2.1 *Manufacturing***

Menurut [9] *Manufacturing* adalah kegiatan produksi yang dilakukan dengan menggunakan tenaga kerja, mesin, dan peralatan lainnya untuk menghasilkan barang untuk digunakan atau di jual. Kegiatan ini di jalankan di dalam pabrik, rumah, atau tempat khusus.

Kegiatan ini merupakan kegiatan yang sangat penting bagi manusia. Pada dasarnya manusia membutuhkan satu sama lainnya untuk bertahan hidup sebagai makhluk sosial. Jika dilihat dari perspektif yang lebih luas, kondisi geografis di setiap negara berbeda-beda. Tidak semua negara memiliki kekayaan sumber daya alam untuk memenuhi kebutuhan dari negara tersebut.

Dari keterbatasan tersebut maka kegiatan manufaktur tercipta. Sebagai contoh, untuk membuat perangkat telepon melibatkan banyak perusahaan yang memproduksi bagian-bagian dari perangkat tersebut. Bisa saja kamera di produksi di lokasi A, material dari baterai di tambang dari lokasi B, dan semuanya di rakit di lokasi C.

Dalam menjalankan kegiatan manufaktur tentunya diperlukan kontrol untuk mengatur kegiatan. Ruang lingkup dari kontrol manufaktur sendiri terdapat dua yaitu manajemen dan kualitas. Jika di jabarkan lebih lanjut kontrol manajemen sendiri terdapat SCM(supply chain management), Manajemen Operasi, dan lain sebagainya [8] [10] [11].

Di dalam manajemen operasi sendiri bertujuan untuk memastikan produksi dari produk yang baik seefisien dan seefektif mungkin.

## 2.2 *Lean Manufacturing*

Berdasarkan [11] dan [10] lean manufacturing merupakan sebuah sistematika dalam manajemen manufaktur yang bertujuan mengurangi pemborosan sumber daya. Seperti, waktu, bahan baku, tenaga kerja, tempat penyimpanan, dan lain-lain. Tujuan dari konsep ini adalah merampingkan struktur operasi agar mudah dipahami, dan di kontrol dari lingkungan kerja.

Konsep ini pertama kali di temukan oleh Toyota untuk mengurangi pemborosan sumber daya. Toyota mengalami kesulitan dalam mencari bahan baku untuk memproduksi produk otomotif. Namun, dengan keterbatasan tersebut Toyota berhasil menjadi sala satu produsen otomotif terbesar di dunia. Terdapat beberapa metode pelaksanaan master production schedule seperti yang di jabarkan pada [10].

## 2.3 *Flow Time, Makespan, Tardines*

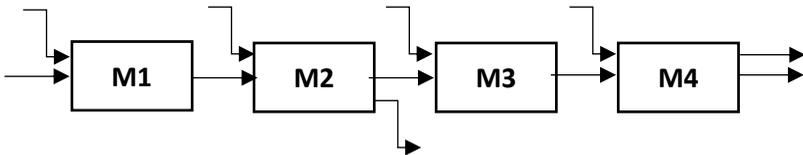
*Flow time* adalah total waktu yang dibutuhkan dalam menyelesaikan pekerjaan.

Makespan adalah indikator lama waktu yang dibutuhkan oleh sistem dalam menyelesaikan himpunan pekerjaan. Makespan yang minimal menandakan sistem bekerja secara efisien secara sumber daya, namun dapat meningkatkan flow time dari pekerjaan yang masuk [3]. Tardiness adalah jumlah satuan waktu pekerjaan yang melebihi tenggat waktu pekerjaan [3].

## 2.4 *Flow Shop Scheduling Problem*

*Flow shop ( $F_m$ ) scheduling problem* merupakan permasalahan yang cukup terkenal di bidang ilmu komputer dan riset operasi. Definisi permasalahan ini berdasarkan [2] pada dasarnya bagaimana cara urutan menyelesaikan pekerjaan dengan sumber daya yang ada. Bentuk umum dari permasalahan ini adalah: “Terdapat  $n$  Pekerjaan  $j_1, j_2, j_3, \dots, j_n$  yang memiliki beberapa  $i$  operasi, tiap operasi perlu di

jalankan pada  $i$  mesin  $m_1, m_2, m_3, \dots, m_n$  secara berurutan dan pada 1 mesin hanya bisa memproses 1 operasi pada satu waktu dengan tujuan total waktu hingga semua pekerjaan selesai paling singkat  $C_{max}$ ”

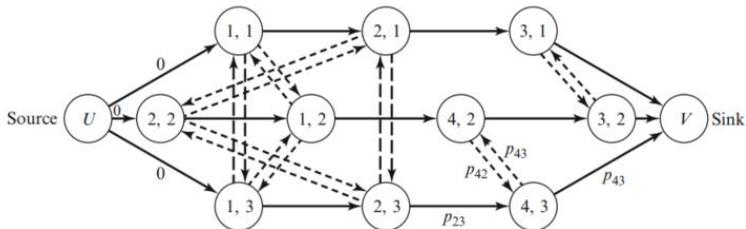


**Gambar 2.1** Model proses flow shop

Beberapa solusi lain untuk *Flow shop scheduling problem*:

- a. *SPT(shortest processing time)*  
Menjadwalkan aktivitas sesuai waktu tersingkat
- b. *LPT(longest processing time)*  
Menjadwalkan aktivitas dengan waktu terpanjang
- c. *SPT-LPT* [3] [2]
- d. *EDD(earliest due date)*  
Menjadwalkan aktivitas sesuai tenggat waktu pekerjaan
- e. *S/RO(slack per remaining operation)*  
Menjadwalkan aktivitas berdasarkan sisa waktu pekerjaan di bagi aktivitas yang tersisa.

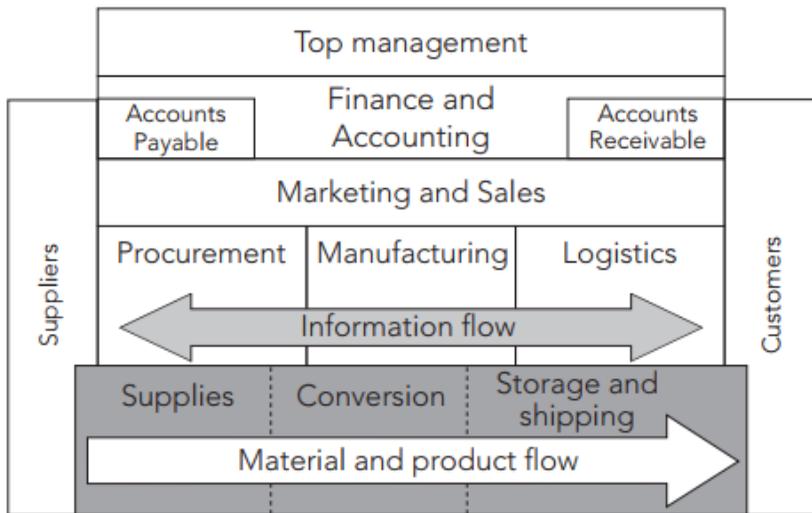
## 2.5 Job Shop Scheduling Problem



**Gambar 2.2** graph jobshop dengan target makespan [2]

Perbedaan *job shop* ( $J_m$ ) *scheduling problem* dengan *flow shop* ( $F_m$ ) *scheduling problem* adalah alur aktivitas pada tiap pekerjaan tidak harus sama, hal tersebut berakibat penjadwalan pada proses *job shop* bisa lebih singkat karena lebih fleksibel [1]. Secara umum permasalahan ini akan memerlukan komputasi yang lebih rumit dari pada kasus  $F_m || C_{max}$  [2]. Hal ini dapat digambarkan pada Bagan 2. Model  $J_m || C_{max}$  ini dapat di selesaikan dengan SPT-LPT [2] [3] dan SBH [4].

## 2.6 ERP (*Enterprise Resource Planning*)



**Gambar 2.2** Gambaran umum sistem ERP [7]

Menurut [7] *Enterprise resource planning* merupakan sebuah sistem informasi. Pada mulanya sistem ini digunakan di dalam pabrik untuk melacak barang pada proses produksi manufaktur.

Manajemen pabrik melihat data penjualan dan prediksi penjualan untuk menentukan penjualan ke depan dan kapasitas produksi yang diperlukan untuk memenuhi kebutuhan tersebut [8].

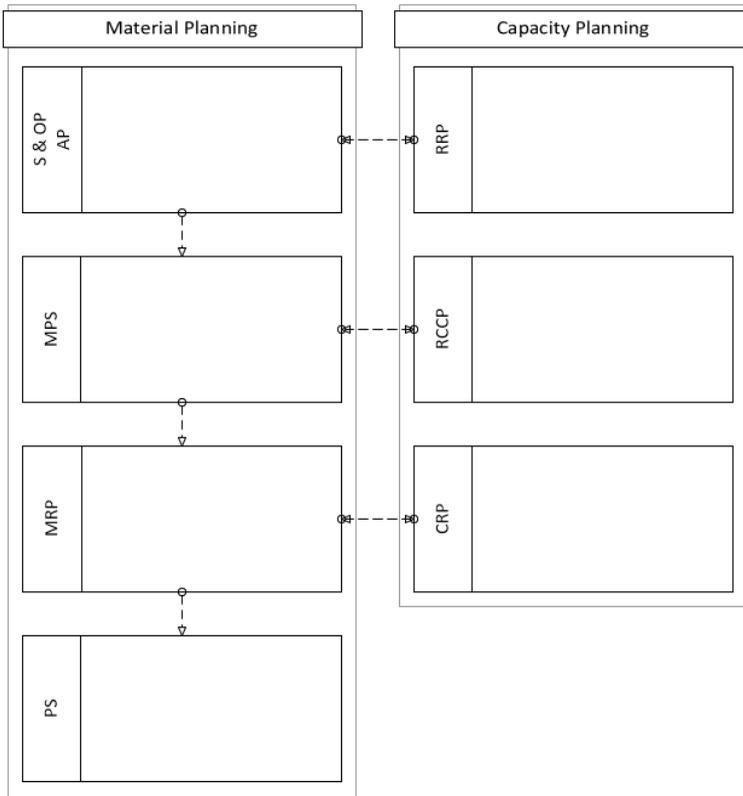
Sistem ini sangat bermanfaat untuk sebuah perusahaan dikarenakan manajemen perusahaan dapat melihat secara keseluruhan proses produksi, kebutuhan pasar, dan kebutuhan sumber daya. Sehingga menghasilkan keputusan yang tepat dalam menentukan strategi perusahaan [9].

Semakin berkembang teknologi, ERP sekarang tidak hanya berurusan dengan pabrik saja namun juga untuk seluruh bagian dari perusahaan seperti sumber daya manusia, hubungan pelanggan dan lain-lain [7].

Beberapa contoh sistem ERP yang ada seperti SAP (*System, Applications and Products in Data Processing*), Oracle, dan Odoo. Masing-masing sistem memiliki pendefinisian yang berbeda dengan satu sama lainnya untuk modul PKP [7].

Sistem ERP dibentuk untuk mendukung proses bisnis yang dilakukan di dalam instansi yang membutuhkan. Tentunya sistem ini harus bisa di menyesuaikan proses bisnis yang ada [7].

## 2.7 PKP (Perencanaan dan Kontrol Produksi)



**Gambar 2.3** BPMN Perencanaan dan Kontrol Produksi [5]

Perencanaan dan Kontrol Produksi merupakan salah satu jenis modul dalam sistem informasi manajemen sumber daya perusahaan. PKP merupakan modul penting dalam penerapan ERP untuk industri manufaktur. Pada dasarnya PKP terbagi menjadi dua kategori yaitu perencanaan material, dan perencanaan kapasitas produksi [7]. Bagan

4 merupakan gambaran proses umum dalam melaksanakan fungsi perencanaan dan kontrol produksi yang terjadi pada industri.

PKP membantu sebuah perusahaan untuk merencanakan tempat produksi serta menentukan sumber daya yang diperlukan untuk memenuhi kapasitas produksi yang dibutuhkan oleh pasar yang dituju [9].

PKP sendiri memiliki beberapa komponen utama yaitu: [6]

1. *Materials requirements planning*
2. *Sales & operations planning*
3. *Aggregate planning*
4. *Master production schedule*
5. *Rough cut capacity planning*
6. *Resource requirement planning*
7. *Capacity requirement planning*
8. *Production scheduling*

## **2.8 Aggregate Planning**

Perencanaan ini digunakan untuk merencanakan berapa banyak dan kapan bahan baku dan tenaga kerja diperlukan untuk membuat suatu produk dalam satu periode perencanaan. Tujuannya untuk menyamakan bahan baku dengan prediksi penjualan suatu produk agar total biaya yang dikeluarkan pada satu periode perencanaan paling minim.

Dalam sistem manajemen perencanaan sumber daya manufaktur, jika dalam definisi piramida dari sistem informasi komponen ini biasanya menjadi bagian dari DSS (decision support system) atau tingkat 3 dalam sistem manajemen informasi [7].

Untuk membuat perencanaan ini haruslah mempertimbangkan beberapa hal, seperti peraturan perusahaan, kapasitas produksi yang ada, produk yang diproduksi, sumber daya manusia, dan lain sebagainya.

Strategi yang dipakai dalam perencanaan ini terdapat tiga macam [9], yaitu:

1. *Chase strategy*
2. *Level strategy*
3. *Mixed strategy*

## **2.9 Sales & Operations Planning**

Komponen ini merupakan salah satu perangkat dalam manajemen bisnis. Biasanya di buat oleh pimpinan dari perusahaan berdasarkan hasil analisis data penjualan di periode sebelumnya. Hasil dari komponen ini adalah perencanaan jumlah bahan baku yang di perlukan dan prediksi penjualan pada periode perencanaan [10]. Analisis ini sangat bermanfaat untuk perusahaan karena dapat memprediksi kapasitas produksi untuk memenuhi kebutuhan [12].

### **2.10 Resource Requirement Planning**

Perencanaan ini berisi kapan dan berapa banyak jumlah bahan baku yang diperlukan untuk membuat produk berdasarkan prediksi penjualan [9].

### **2.11 Master Production Schedules**

Komponen ini berisikan berapa jumlah produk yang harus di produksi dalam suatu periode perencanaan. Data yang diperlukan dari komponen ini didapat dari komponen PKP yaitu S&OP, AP, dan RRP. Di mana data yang dimaksud adalah data kebutuhan pasar sebelumnya serta prediksi kebutuhan pasar pada periode perencanaan. Dari situ dapat dilihat berapa banyak produk yang harus di produksi oleh pabrik dalam satu periode perencanaan [10].

### **2.12 Rough Cut Capacity Planning**

Perencanaan ini bertujuan untuk memberikan gambaran umum apakah kebutuhan dengan kapasitas sudah sesuai. Masukan pada perencanaan ini berupa berapa kebutuhan pada periode sebelumnya dan kapasitas produksi dari fasilitas produksi yang ada. Dari situ, dicari selisih antara kedua masukan tersebut dan menghasilkan gambaran kasar tentang apakah kapasitas produksi sudah sesuai dengan kebutuhan yang ada. Hal ini bermanfaat untuk sebuah perusahaan dalam merencanakan produksi [10].

### **2.13 Capacity Requirements Planning**

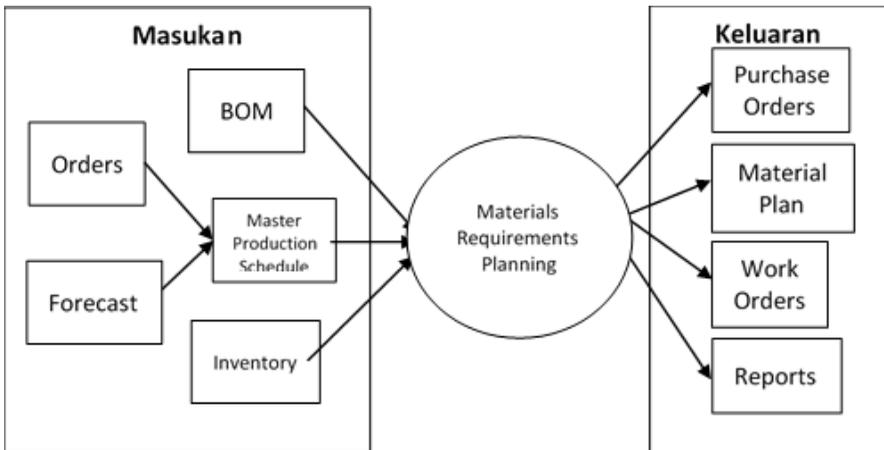
Perencanaan ini bertujuan untuk menentukan kapasitas produksi dalam satu periode. Keluaran yang dibutuhkan oleh manajemen biasanya dalam bentuk profil beban per-kapasitas produksi. Proses pembuatan perencanaan ini dimulai dengan melihat dan mempelajari catatan barang yang di produksi oleh industri. Dengan data tersebut pemilik usaha dapat memprediksi seberapa banyak kebutuhan atas suatu produk ke depan. Sehingga menjawab pertanyaan apakah kapasitas produksi yang ada sudah mencukupi atau belum [8].

Kapasitas produksi sendiri dipengaruhi banyak faktor seperti, mesin, sumber daya manusia, dan ketersediaan bahan baku. Perencanaan kapasitas yang kurang tepat dapat memberikan dampak buruk bagi perusahaan. Jika kapasitas yang direncanakan melebihi kebutuhan kotor dari produk maka akan terjadi kelebihan kapasitas yang berakibat sumber daya yang ada tidak dapat dimanfaatkan sepenuhnya. Sebaliknya, jika kapasitas yang direncanakan kurang dari kebutuhan kotor dari produk maka perusahaan tidak dapat memenuhi kebutuhan pasar yang di target [10].

### **2.14 Materials Requirements Planning**

Komponen ini bertujuan untuk memastikan material yang di perlukan dalam proses produksi sudah terpenuhi dan ada di dalam stok gudang dari suatu perusahaan. Hal ini dikarenakan terkadang pelanggan dari suatu perusahaan ingin barang yang mereka pesan dibuat dalam waktu yang lebih singkat dari waktu produksi biasanya. Untuk memenuhi permintaan tersebut perusahaan harus menerapkan strategi produksi yang dapat di andalkan. Strategi yang dimaksud adalah: kapan memesan bahan baku, berapa banyak bahan baku yang harus disimpan, dan lain sebagainya [7].

Pada dasarnya komponen ini mempunyai beberapa masukan, aktivitas, dan keluaran seperti pada gambar berikut:



**Gambar 2.5** Keluaran dan Masukan MRP [5]

**Tabel 2.2.1** Keluaran dan Masukan MRP

<b>Masukan</b>	<b>Keluaran</b>
<i>Bill of Material (BOM)</i>	Bahan baku kotor yang dibutuhkan
Kebutuhan kotor dari bahan baku	Tanda pemesanan dari bahan baku
Penjadwalan tanda terima bahan baku	Tanda terima bahan baku yang direncanakan
Perencanaan jangka pendek	Proyeksi persediaan dari bahan baku
Persediaan awal bahan baku	Rencana kebutuhan bahan baku
Tingkat aman persediaan ( <i>safety stock</i> )	
Ukuran <i>batch</i> bahan baku	
Aturan ukuran batch untuk tiap bahan baku	

*Tabel 2.2.2 Aktivitas dari MRP*

No.	Aktivitas
1	Mendefinisikan BOM untuk produk final pada master perencanaan produksi
2	Mendefinisikan perencanaan jangka pendek
3	Menghitung kebutuhan kotor
4	Menghitung proyeksi persediaan
5	Menentukan perencanaan tanda terima
6	Menghasilkan perencanaan pemesanan
7	Verifikasi BOM
8	Menghasilkan perencanaan kebutuhan bahan baku
9	Meninjau perencanaan kebutuhan bahan baku

### 2.15 Production Scheduling

Proses penjadwalan ini merupakan salah satu proses penting dalam proses penentuan keputusan dalam perencanaan dan kontrol produksi. Yang di definisikan oleh proses ini adalah: [7]

1. Instruksi pekerjaan
2. Di mana instruksi pekerjaan di laksanakan
3. Kapan instruksi pekerjaan di laksanakan

Penjadwalan produksi bertujuan untuk menemukan urutan dan mengontrol pekerjaan dengan mempertimbangkan pengeluaran dan tenggat waktu pekerjaan sehingga diperoleh hasil yang efisien dengan pengeluaran seminim mungkin. [10]

Sistem yang dihasilkan harus mencakup seri aktivitas serta pemantauan dan pelaksanaan dari aktivitas yang sudah di jadwalkan. Penyajian data pada proses ini biasanya dalam bentuk gantt chart [13].

## 2.16 Make to Order

Make to order merupakan jenis proses manufaktur dimana produksi dimulai ketika pesanan dari konsumen telah diterima oleh perusahaan. Proses ini merupakan alternatif dari proses make to stock. Dimana proses make to stock membutuhkan perencanaan lebih rumit dalam penentuan kapan pembelian material beserta jumlahnya dan berapa banyak produk yang harus di simpan di dalam gudang. Proses ini menawarkan fleksibilitas yang tidak dimiliki oleh proses yang lain. Namun, kelemahan proses ini adalah pelanggan harus menunggu untuk mendapatkan produk yang di pesan. Hal ini jelas berbeda dengan proses yang lain [9].

## 2.17 Gantt Chart

<b>Mesin</b>				
M1				
M2				
<b>Waktu</b>	1	2	3	4

**Gambar 2.4** Ilustrasi Gantt Chart

*Gantt chart* adalah sebuah alat untuk menggambarkan penjadwalan dari suatu aktivitas. Alat ini sangat penting dalam proses penjadwalan karena dapat menyajikan proses kegiatan produksi dalam bentuk visual yang mudah dipahami [10]. Pada *gant chart* terdapat baris dan kolom. Baris berisi aktivitas yang perlu dilakukan. Kolom berisi waktu yang diperlukan untuk menyelesaikan pekerjaan. Selain itu *gant chart* juga dapat berisi informasi lainnya seperti siapa aktor pelaku aktivitas dan lain sebagainya.

## 2.18 Laravel

Laravel adalah kerangka kerja bahasa PHP yang dibuat oleh Taylor Otwell untuk digunakan dalam pengembangan aplikasi berbasis web [11].Laravel menggunakan arsitektur MVC dan dikembangkan menggunakan kerangka kerja symfony. Kerangka kerja Laravel sangat sering digunakan dalam pengembangan aplikasi berbasis web

dikarenakan banyak fitur yang diberikan kepada pengembang aplikasi oleh Laravel yang tidak dimiliki kerangka kerja yang lain. Fitur – fitur yang disediakan oleh kerangka kerja ini antara lain:

1. Modul sistem dengan manajemen ketergantungan Composer
2. Pemetaan objek relasi Eloquent
3. Pembangun pertanyaan untuk basis data
4. Logika aplikasi di dalam tempat khusus
5. Pengarahan tautan dengan logika aplikasi yang dinamis dan otomatis
6. Dan lain sebagainya

Dengan fitur-fitur yang sangat bermanfaat kerangka kerja ini menjadi kerangka kerja yang cukup populer di kalangan pengembang aplikasi berbasis web. Selain itu kerangka kerja ini juga di lisensikan di bawah MIT License. Sehingga semua orang dapat menggunakan, mengubah dan mendistribusikan kerangka kerja ini secara gratis dan terbuka [12].

## **2.19 UML**

UML (*unified modeling language*) adalah bahasa pemodelan umum yang digunakan untuk menstandarkan visualisasi dari desain sebuah sistem. UML telah menjadi standar internasional sejak tahun 2005. Saat ini versi terbaru dari UML sendiri adalah UML 2.5.1 [13]. pemodelan dengan UML menawarkan cara visualisasi terhadap aktivitas, komponen, dan interaksi dari sistem sehingga pembuat sistem dapat menyajikan desain arsitektur dari sistem yang dibangun secara visual dan mudah untuk dipahami.

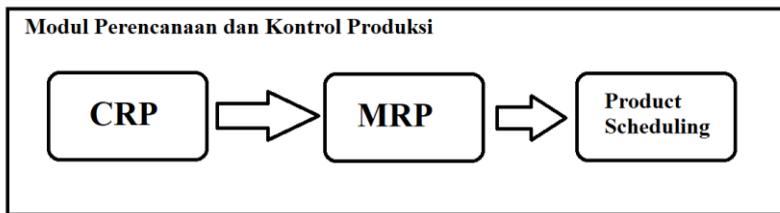
## **BAB III PERANCANGAN**

Perancangan merupakan bagian penting dari pembuatan sistem secara teknis sehingga bab ini secara khusus menjelaskan perancangan sistem yang dibuat dalam Tugas Akhir. Berawal dari

deskripsi umum sistem hingga perancangan skenario, alur dan implementasinya.

### 3.1 Deskripsi Umum

Pada Tugas Akhir ini akan dibuat sebuah rancang bangun modul perencanaan dan kontrol produksi untuk optimasi makespan proses produksi manufaktur menggunakan EDD. Gambaran umum model perencanaan produksi dapat dilihat pada Gambar 3.1.



*Gambar 3.1* Gambaran Umum Modul PKP

Modul PKP diawali dengan melakukan persiapan dengan menjalankan proses CRP yaitu membuat rencana kapasitas produksi berdasarkan jumlah mesin produksi dan kapasitas produksi dari setiap satu set mesin produksi, membuat rencana tahapan proses produksi suatu produk, dan membuat rencana biaya operasional produksi. Kemudian menjalankan proses MRP yaitu membuat rencana komposisi bahan produksi untuk setiap produk, membuat BOM yang merupakan resep dasar pembuatan produk, serta menambahkan data (melakukan pembelian) bahan baku produksi. Selanjutnya akan dijalankan proses Production Scheduling yaitu menerima (input data) pesanan atas produk, melakukan pengecekan jumlah ketersediaan bahan baku, melakukan pemetaan penggunaan bahan baku terhadap peruntukan produksi setiap produk yang dipesan, serta melakukan work order yaitu memberikan (input) perintah operasi proses produksi.

Tabel 3.1 Daftar Istilah

No.	Istilah	Penjelasan
1	MRP	Singkatan dari <i>Material Requirement Planning</i> . MRP merupakan bagian penting dari modul PKP. MRP memiliki fungsi utama untuk merencanakan segala sesuatu yang berhubungan dengan produk dan bahan baku.
2	CRP	Singkatan dari <i>Capacity Requirement Planning</i> . CRP merupakan bagian penting dari modul PKP. CRP memiliki fungsi utama untuk merencanakan kapasitas dan alur produksi.
3	PS	Singkatan dari <i>Production Scheduling</i> , pada bagian ini berisi Manufacture Order dan tempat dimana metode EDD di implementasi.
4	EDD	Singkatan dari <i>Earliest Due Date</i> , adalah salah satu aturan penjadwalan kerja berdasarkan tenggat waktu paling dekat.
5	UoM	Singkatan dari <i>Unit of Measurement</i> . Adalah unit pengukuran jumlah dari produk
6	Makespan	Jumlah total waktu yang diperlukan untuk menyelesaikan semua pekerjaan.

### 3.2 Perancangan Optimasi Makespan

Rancang bangun modul perencanaan dan kontrol produksi yang di hasilkan dari tugas akhir ini harus dapat digunakan sebagai alat untuk mengoptimasi *makespan*  $C_{max}$  pada penjadwalan proses produksi. Berangkat dari definisi permasalahan penjadwalan pada *flow shop* yaitu “Terdapat  $n$  Pekerjaan  $j_1, j_2, j_3, \dots, j_n$  yang memiliki beberapa  $i$  operasi, tiap operasi perlu di jalankan pada  $i$  mesin  $m_1, m_2, m_3, \dots, m_n$  secara berurutan dan pada satu mesin hanya bisa

memproses satu operasi pada satu waktu dengan tujuan total waktu hingga semua pekerjaan selesai paling singkat *makespan*  $C_{max}$ ” didapatkan dua batasan utama yaitu:

1. Operasi  $i$  pada pekerjaan  $j_n$  harus dikerjakan secara berurutan
2. Mesin  $m_n$  hanya bisa memproses satu Operasi  $i$  pada satu waktu

Lalu untuk definisi permasalahan penjadwalan pada *job shop* yaitu “ $n \times m$  dimana terdapat set dari  $n$  Jobs  $\{J_i\} 1 \leq n$  yang akan diproses pada set  $m$  Mesin  $\{M_r\} 1 \leq r \leq m$ . Pada tiap Jobs memiliki urutan produksi yang akan di proses pada mesin. Proses pengerjaan  $J_i$  pada mesin  $M_r$  disebut operasi  $O_{jr}$ . Tiap operasi  $O_{jr}$  yang diproses pada mesin  $M_r$  memakan waktu proses yang eksklusif yang disebut  $P_{jr}$ . Penjadwalan adalah set dari waktu selesai untuk tiap operasi  $\{C_{jr}\} 1 \leq j \leq n, 1 \leq r \leq m$ ” didapatkan dua batasan utama yaitu:

1. Operasi  $O_{jr}$  pada pekerjaan  $J_i$  harus dikerjakan secara berurutan
2. Tiap mesin  $M_r$  hanya bisa memproses satu operasi  $O_{jr}$

Tujuan dari kedua permasalahan penjadwalan baik kasus *job shop* dan kasus *flow shop* untuk mencari nilai *makespan*  $C_{max}$  yang minimum.

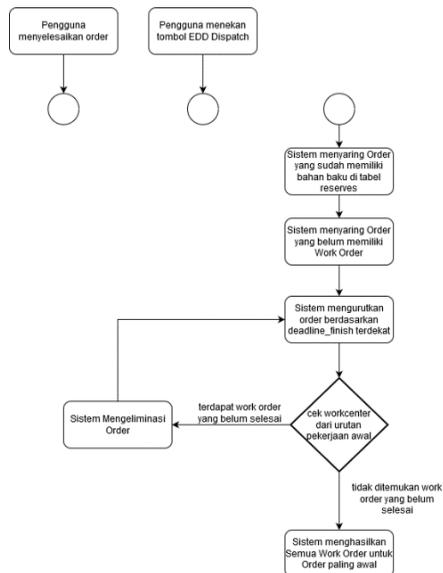
Dari kedua definisi permasalahan *job shop* dan *flow shop* maka dapat dibilang batasan utamanya sama. Masing-masing operasi harus dikerjakan secara berurutan dan tiap mesin hanya bisa memproses satu operasi pada satu waktu. Yang berbeda dari kedua permasalahan tersebut adalah alur produksinya. Pada *flow shop* tidak boleh ada operasi lintas mesin atau alurnya lebih linear. Sementara untuk *job shop* dibolehkan ada operasi lintas mesin.

Oleh karena itu pada Modul Perencanaan dan Kontrol Produksi harus menerapkan desain sistem yang mengakomodir kedua batasan tersebut serta bisa menerima data set yang sesuai dengan deskripsi permasalahan penjadwalan *job shop* dan *flow shop*. Terakhir untuk dapat mengoptimasi nilai  $C_{max}$  yang minimum, Modul Perencanaan dan Kontrol Produksi harus dapat menerima metode lain yang di

masukkan secara manual dan menghitung nilai  $C_{max}$  untuk dibandingkan mana nilai yang lebih optimal.

### 3.3 Perancangan Metode EDD

Perancangan metode EDD diawali dengan membuat diagram algoritma yang akan di implementasikan pada Modul Perencanaan dan Kontrol Produksi



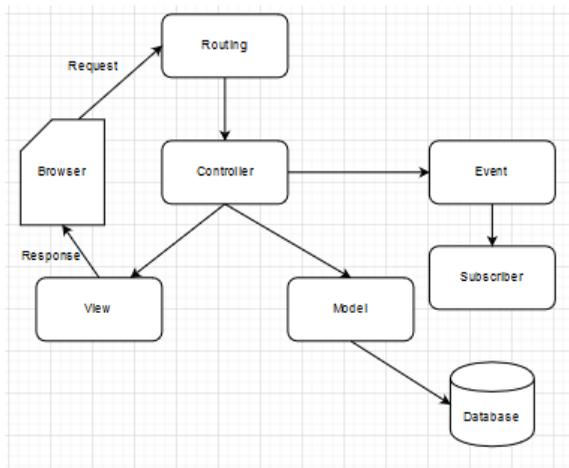
**Gambar 3.2** Desain Algoritma Metode EDD

Pada Gambar 3. 9 adalah desain algoritma dari metode penjadwalan proses produksi menggunakan metode *Earliest Due Date* (EDD) yaitu penjadwalan pengerjaan order berdasarkan *due date* terdekat. Pada rancangan Modul Perencanaan dan Kontrol Produksi ini *due\_date* adalah *deadline\_finish* pada tabel orders.

### 3.4 Perancangan Lingkungan Pengembangan

Perancangan Lingkungan pengembangan dimulai dengan menyusun lingkungan pengembangan perangkat lunak yang terdiri dari kerangka kerja, server basis data, dan server website.

Kerangka kerja pada tugas akhir ini menggunakan kerangka laravel guna memanfaatkan konsep MVC yang merupakan pola arsitektur pada perancangan perangkat lunak berorientasi objek. Kerangka kerja yang diterapkan pada tugas akhir ini tertera pada gambar 3.2



Gambar 3.3 Kerangka MVC Laravel

Pada gambar 3.2 tertera informasi terkait kerangka kerja yang digunakan, dimana ketika user melakukan sebuah permintaan akses akan di arahkan melalui routing sistem menuju suatu kontroller yang di akses pada sistem routing, kemudian kontroller akan mengambil data dari database melalui model dan mengembalikan nilai dan di arahkan ke suatu view yang akan menampilkan sebuah halaman kepada user.

*Tabel 3.2 Versi kerangka kerja Laravel*

Framework	Laravel
Versi	7.13.0

pada tabel 3.2 memberikan informasi terkait spesifikasi kerangka kerja laravel yang digunakan dalam lingkungan pengembangan perangkat lunak pada tugas akhir ini. Laravel yang digunakan merupakan versi 7.13.0 yang dirilis pada tahun 2020.

Basis data pada tugas akhir ini terdapat beberapa tabel yang memiliki relasi yang berkaitan dan satu table tanpa relasi ke table manapun. Server basis data menggunakan PostgreSQL dengan keterangan sebagai berikut:

***Tabel 3.3 Versi basis data PostgreSQL***

Server	127.0.0.1
Jenis server	PostgreSQL
Versi server	postgresql-x64-12
Versi protocol	12
Pengguna	postgres
Kumpulan karakter server	UTF-8 Unicode (utf8)

Pada Tabel 3.3 memberika informasi terkait spesifikasi dari basis data yang digunakan untuk mengelola data pada aplikasi teledermatologi. Server basis data yang digunakan merupakan PostgreSQL versi 12

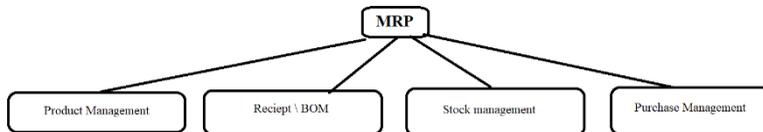
***Tabel 3.4 Versi server website***

Server	PHP development server
Versi klien basis data:	php_pdo_pgsql.dll -7.4.6
Ekstensi PHP :	pdo_pgsql PHP
Versi	7.4.6

Pada Tabel 3.4 tertera informasi spesifikasi dari server website yang digunakan dalam pengembangan modul Perencanaan dan Kontrol Produksi. Server website yang digunakan ialah PHP development server.

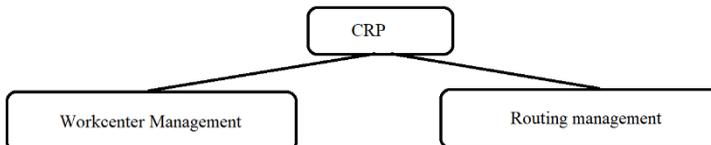
### 3.5 Perancangan Modul Perencanaan dan Kontrol Produksi

Perancangan modul perencanaan dan kontrol produksi diawali dengan membuat daftar kebutuhan fitur untuk mengampu beberapa pekerjaan yang akan dilakukan saat membuat perencanaan dan kontrol produksi. Pada perangkat lunak modul perencanaan dan kontrol produksi terdapat tiga pekerjaan utama yang diterjemahkan kedalam tiga menu utama yaitu *Material Requirement Planning*, *Capacity Requirement Planning*, dan *Production Scheduling*, serta dilengkapi dengan fitur *Report and Analysis*.



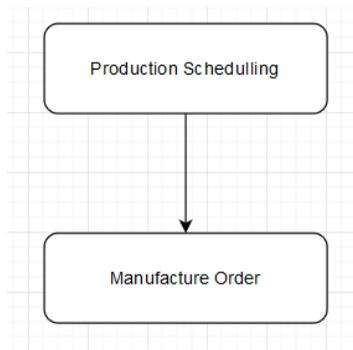
**Gambar 3.4** Menu utama MRP

Pada gambar 3.3 tertera informasi terkait menu utama *Material Requirement Planning* yang terdiri dari fitur *Product Management* guna mengelola data produk pada proses perencanaan dan kontrol produksi, fitur *Recipe \ BOM* guna mengelola data resep yang akan digunakan pada saat proses produksi, fitur *Stock Management* guna mengelola data jumlah produk yang tersedia di dalam tempat penyimpanan produk, dan fitur *Purchase Management* guna mengelola data pembelian bahan baku untuk proses produksi.



**Gambar 3.5** Menu Utama CRP

Pada gambar 3.4 tertera informasi terkait menu utama *Capacity Requirement Planning* yang terdiri dari fitur *Workcenter Management* guna mengelola data mesin/tempat kerja yang tersedia bersedia kapasitasnya dalam mengolah produk dalam satu waktu dan fitur *Routing Management* guna mengelola data tahapan/rute proses pengolahan bahan baku dalam proses produksi.



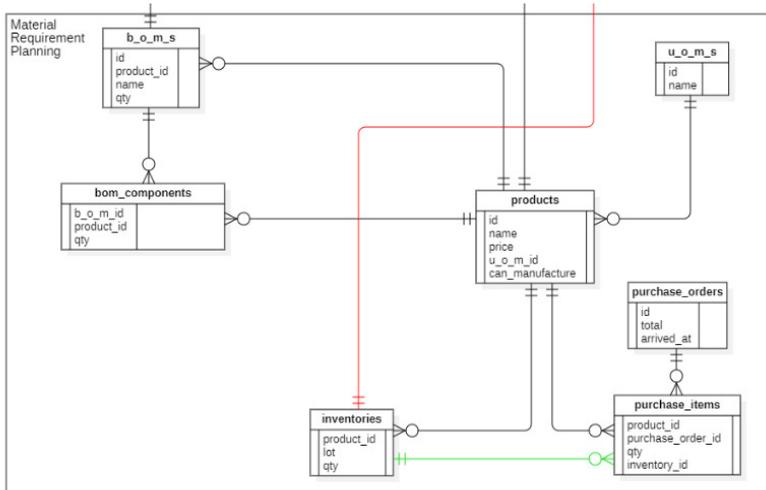
**Gambar 3.6** Menu Utama *Production Scheduling*

Pada gambar 3.5 tertera informasi terkait menu utama *Production Schedule* yang terdiri dari fitur *Manufacture Order* guna mengelola data pesanan yang masuk ke dalam pabrik dan/atau tempat produksi.

### 3.6 Perancangan Menu MRP

Perancangan menu MRP diawali mendefinisikan Entity Relationship Diagram pada menu utama *Material Requirement Planning* yang terdiri dari fitur *Product Management* guna mengelola data produk pada proses perencanaan dan kontrol produksi, fitur *Receipt \ BOM* guna mengelola data resep yang akan digunakan pada saat proses produksi, fitur *Stock Management* guna mengelola data

stok produk yang tersedia, dan fitur *Purchase Management* guna mengelola data pembelian bahan baku untuk proses produksi.

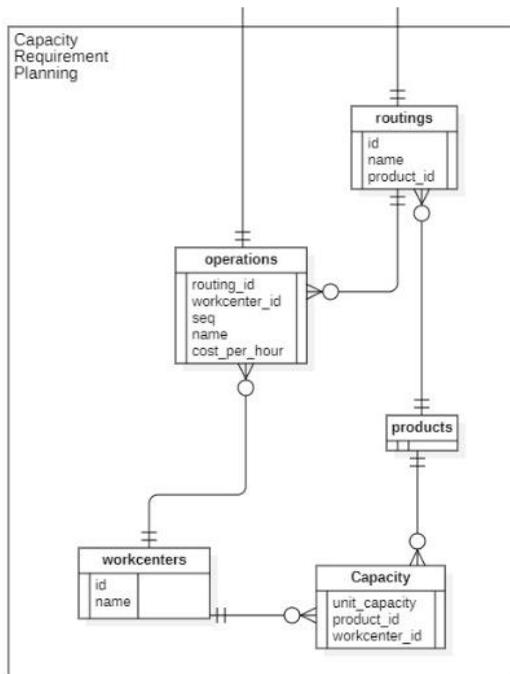


**Gambar 3.7** ERD Menu Material Requirement Planning

Pada Gambar 3.6 tertera informasi terkait entitas produk, bom/recipe, uom, dan inventory. Produk memiliki nama, harga, unit pengukuran, dan can\_manufacture yang menandakan apakah produk tersebut dapat di produksi di dalam lingkungan produksi. Lalu pada bom, menyimpan data resep untuk membuat produk yang memiliki bahan baku pada tabel bom\_components. Lalu inventory menyimpan data kuantitas dari produk serta id LOT tempat produk disimpan. Terakhir purchase\_order merupakan entitas pembelian bahan produk untuk dijadikan bahan baku dari produksi. Purchase\_order memiliki banyak produk dalam satu kali pemesanan dan menyimpan data dimana produk tersebut di simpan.

### 3.7 Perancangan Menu CRP

Perancangan menu MRP diawali mendefinisikan Entity Relationship Diagram pada menu utama *Capacity Requirement Planning* yang terdiri dari fitur *Workcenter Management* guna mengelola data kapasitas produksi berdasarkan kapasitas dan jumlah unit kerja produksi dan fitur *Routing Management* guna mengelola data tahapan proses pengolahan bahan baku dalam proses produksi.



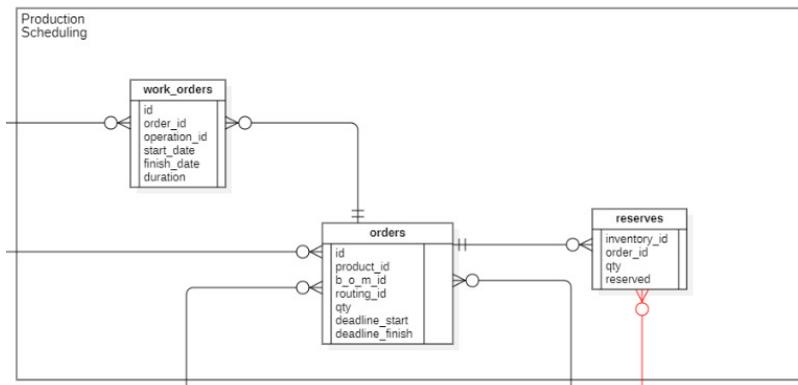
**Gambar 3.8** ERD Menu *Capacity Requirement Planning*

Pada Gambar 3.7 tertera informasi terkait entitas dan relasi dari tabel operation, routing, dan workcenter, dan capacity. Pada bagian ini fungsi utamanya untuk merencanakan kapasitas produksi dari tempat produksi. Mulai dari alur pekerjaan, mesin/tempat kerja, serta

kapasitas dari mesin/tempat kerja. Pada pengaturan alur hal ini ditentukan dari tabel routing yang memiliki banyak operasi. Dalam kasus ini operasi dilakukan pada suatu mesin/tempat kerja dan memiliki variabel seq (urutan), nama, dan biaya per jam.

### 3.8 Perancangan Menu Product Scheduling

Perancangan menu *Production Scheduling* diawali mendefinisikan *Entity Relationship Diagram* pada menu utama *Production Schedule* yang terdiri dari fitur *Manufacture Order* guna mengelola data pesanan yang masuk di dalam lingkungan produksi.



**Gambar 3.9** Menu Utama *Production Scheduling*

Pada Gambar 3.8 tertera informasi terkait entitas dan relasi dari tabel *orders*, *work\_orders*, dan *reserve*. Informasi yang terkait dengan tabel *order* adalah *resep/bom*, *produk*, *routing*, dan *reserve*. Tabel *reserve* memiliki peran untuk memesan bahan baku yang tersedia di dalam *stock* agar tidak digunakan oleh *order* yang lain. Dengan adanya tabel *reserve* maka menu *Production Scheduling* sadar dengan menu *Material Requirement Planning*. Untuk tabel *work\_orders*

sendiri berisi perintah kerja, tabel ini digunakan oleh sistem untuk menghasilkan perintah kerja sesuai rute produksi yang dipilih dan diatur oleh pengguna pada menu *Capacity Requirement Planning*.

## **BAB IV IMPLEMENTASI**

Pada bab ini akan dibahas mengenai implementasi dari perancangan yang sudah dilakukan pada bab sebelumnya. Implementasi berupa kode sumber untuk membangun program.

### **4.1 Implementasi Optimasi Makespan**

Implementasi Optimasi Makespan dibagi menjadi tiga yaitu batasan, penjadwalan manual, dan kalkulasi *makespan*.

#### **4.1.1 Batasan**

Untuk batasan di dalam penjadwalan proses produksi terdapat dua yaitu:

1. Satu mesin hanya dapat memproses satu operasi pada satu waktu.
2. Operasi dari Pekerjaan harus dikerjakan secara berurutan.

Batasan pertama di implementasikan pada kode program yang dipanggil saat pengguna sistem meminta sistem untuk memperbaharui data pada tabel *work\_orders*. potongan kode program dapat dilihat pada gambar 4.1

```
File: App\Http\Controllers\WorkOrderController.php Line: 97-103
...

if ($order
    ->workOrders()
    ->whereRaw("finish_date is null and start_date is not null and id
!= " . $workOrder->id)
    ->exists()) {
    return response()
        ->json(['error' => ['Workcenter is Busy']]);
}
...
```

***Gambar 4.1 Potongan Kode Program Batasan 1***

Penjelasan gambar 4.1 adalah jika terdapat data pada tabel `work_orders` yang data kolom `start_date` nya tidak kosong dan kolom `finish_date` nya masih kosong yang bukan id dari `work_orders` yang dipilih maka kode program akan berhenti untuk mengembalikan respon “Workcenter is Busy” dan tidak akan menjalankan perintah selanjutnya untuk menguba data pada `work_orders` yang dipilih. Terakhir untuk batasan selanjutnya bisa dilihat pada gambar 4.2.

```
File: App\Http\Controllers\WorkOrderController.php Line: 97-103
...

$op = $workOrder->operation;
$prevop = $order->routing->operations()
    ->where('seq', $op->seq - 1)->first('id');

    // Pekerjaan hanya bisa dimulai jika pekerjaan sebelumnya sudah
    // selesai
    if (isset($prevop->id) &&
        $order->workOrders()
            ->whereRaw('finish_date is null and operation_id = ' .
                $prevop->id)
            ->exists()) {
        return response()
            ->json(['error' => ['Previous operation not finished yet']]);
    }
...

```

**Gambar 4.2** Potongan Kode Program Batasan 2

Penjelasan gambar 4.2 adalah jika terdapat data pada tabel `work_orders` yang kolom `finish_date` nya belum terisi dan nilai kolom `operation.seq` nya sama dengan nilai kolom `operation.seq-1` pada data `works_order` yang dipilih maka kode program akan berhenti untuk mengembalikan respon “Previous operation not finished yet” dan tidak akan menjalankan perintah selanjutnya untuk mengubah data pada `work_orders` yang dipilih.

### 4.1.2 Penjadwalan Manual

Pengimplementasian penjadwalan manual dilakukan dengan cara membuat kode program untuk menjadwalkan pekerjaan secara manual. Implementasi dari ini dapat dilihat pada gambar 4.3

```

File: App\Http\Controllers\OrderController.php Line: 183-285
...

public function dispatchWork(Order $order)
{
    if ($order->recipe()->first('id') == null) {
        return response()->json(['error' => ['Recipe Not Set Yet']]);
    } else if ($order->missing_ingredients->count() > 0) {
        return response()->json(['error' => ['Missing ingredients']]);
    } else if ($order->routing()->first('id') == null) {
        return response()->json(['error' => ['Routes not set yet']]);
    } else if ($order->workOrders()->whereNotNull('start_date')-
>exists()) {
        return response()->json(['error' => ['Order already
dispatched']]);
    } else {
        foreach ($order->workOrders as $wo) {
            $wo->delete();
        }
        foreach ($order->routing->operations as $op) {
            $wo = new WorkOrder;
            $wo->order()->associate($order);
            $wo->operation()->associate($op);
            $wo->save();
        }
        return response()->json(['success' => ['Order dispatched']]);
    }
}

```

**Gambar 4.3** Potongan Kode Implementasi Penjadwalan Manual

Penjelasan Gambar 4.3 adalah fungsi yang dipanggil ketika pengguna menekan tombol “Manual Dispatch”. Ketika semua kondisi program untuk data orders terpenuhi maka program akan secara otomatis menghasilkan work\_orders sesuai parameter dari order yang dipilih.

#### 4.1.3 Kalkulasi Makespan

Pengimplementasian kalkulasi makespan dilakukan dengan cara membuat kode program untuk menghitung makespan seperti pada gambar 4.4.

```
File: App\Http\Controllers\ReportController.php Line: 45-57
...
    $from = Carbon::create($request['from']);
    $to = Carbon::create($request['to']);

    $woFirst = WorkOrder::whereBetween('start_date', [$from,
        $to])->orderBy('start_date', 'asc')->first('start_date');
    if (!isset($woFirst->start_date)) {
        return response()->json(['error' => 'Work Order Not Found']);
    }
    $woLast = WorkOrder::whereBetween('start_date', [$from, $to])
        ->orderBy('finish_date', 'desc')->first('finish_date');

    //Kalkulasi Makespan
    $makespan = Carbon::create($woLast->finish_date)
        ->diffInSeconds(Carbon::create($woFirst->start_date)) / 60 / 60;
    $makespan = round($makespan, 2);
    $makespan = $makespan." Hour";
...

```

*Gambar 4.4 Potongan Kode Kalkulasi Makespan*

Penjelasan Gambar 4.4 adalah fungsi yang dipanggil ketika mengirim form report. Sistem akan mengambil semua data `work_orders` yang berada dalam rentang waktu (`from`, `to`) yang dikirim oleh pengguna. Selanjutnya sistem akan menghitung makespan menggunakan fungsi *diff* dari library Carbon.

## **4.2 Implementasi Metode EDD**

Implementasi metode EDD pada modul perencanaan dan kontrol produksi ini terbagi menjadi dua bagian yaitu Event dan Handler. Implementasi sengaja dibuat sedemikian rupa agar penelitian selanjutnya dapat memodifikasi dan/atau menambahkan metode lain selain metode yang diujikan pada tugas akhir ini.

### **4.2.1 Event**

Event merupakan fitur yang disediakan oleh laravel. Untuk mengimplementasikan metode *EDD* pertama yang dilakukan adalah mendeklarasikan kelas Event yang akan memicu Handler yang telah dibuat. Untuk implementasinya dapat dilihat pada gambar 4.5.

```
<?php
namespace App\Events;

class EDDButtonPressEvent
{
    /**
     * Create a new event instance.
     *
     * @return void
     */
    public function __construct()
    {
        //
    }
}
```

**Gambar 4.5** Potongan Kode Kelas Event *EDDButtonPressEvent*

Pada Gambar 4.5 merupakan potongan deklarasi kelas Event pada kerangka kerja Laravel. Kelas ini digunakan untuk mengidentifikasi Event yang terjadi di dalam siklus hidup kode program yang berbasis Laravel.

#### **4.2.2 Handler**

Setelah membuat Event maka perlu diimplementasikan *Handler* yang berguna untuk menangani event yang terjadi seperti pada gambar 4.6.

```

/**
 * Handle Earliest Due Date Dispatch Rule
 *
 * @param object $event
 * @return array
 */
public function handle($event)
{
    $dispatched = collect([]);
    $orders = Order::doesntHave('workOrders')
        ->whereNotNull('routing_id')
        ->orderBy('deadline_finish', 'asc') //EDD Dispatch Rule
        ->get()
        ->append('missing_ingredients');
    $orders->each(function ($order, $key) use ($dispatched) {
        if ($order->missing_ingredients->count() == 0) {
            $dispatched->add($order);
            foreach ($order->routing->operations as $op) {
                $wo = new WorkOrder;
                $wo->order()->associate($order);
                $wo->operation()->associate($op);
                $wo->save();
            }
        }
    });

    return compact('dispatched', 'event');
}

```

**Gambar 4.6** Potongan Kode Handler EDDSubscriber

Pada Gambar 4.6 merupakan potongan kode handler dari kelas EDDSubscriber. Pertama sistem akan mengambil data order yang belum memiliki work\_orders, sudah terisi routing\_id nya dan diurutkan berdasarkan kolom deadline\_finish terkecil. Untuk masing masing order jika tidak memiliki variabel missing\_ingredients maka order tersebut secara otomatis dihasilkan work\_orders nya oleh sistem. Ketika semua sudah terimplementasi pengguna dapat menggunakan potongan kode seperti gambar 4.7 untuk memanggil handler yang dituju.

```

/**
 * EDD Dispatch button
 *
 * @return \Illuminate\Http\JsonResponse
 */
public function EDDDispatch()
{
    $e = event(new EDDButtonPressEvent);
    return response()->json(['success' => ['Orders Processed'],
                            'e' => $e]);
}

```

**Gambar 4.7** Penggunaan Event dan Handler untuk Melakukan Metode EDD

Penjelasan gambar 4.7 adalah jika fungsi EDDDispatch di panggil maka kode program akan memberitahu Laravel bahwa event EDDButtonPressEvent sedang terjadi. Laravel akan memanggil fungsi handler yang sudah diatur berdasarkan kelas event yang sedang terjadi.

### 4.3 Implementasi Lingkungan Pengembangan

Implementasi lingkungan pengembangan dibagi menjadi dua yaitu database model dan konfigurasi web server.

### 4.3.1 Database Model

Dalam meng-implementasikan model basis data, Laravel menyediakan fitur *migrations* guna mendefinisikan tabel yang akan dibuat di dalam database, fitur *console command migrate* untuk membuat tabel pada basis data, dan fitur *Eloquent* sebagai abstraksi pada bahasa PHP untuk mendefinisikan tabel serta relasinya sehingga tidak perlu menggunakan bahasa basis data PostgreSQL dan/atau Maria DB untuk melakukan query data semua sudah disediakan oleh kerangka kerja Laravel. Pada Tugas Akhir ini dibuat 14 tabel guna menangani keseluruhan data yang dikelola dalam perangkat lunak modul Perencanaan dan Kontrol Produksi. Untuk mengimplementasikan rancangan menggunakan fitur migrations dari kerangka kerja Laravel, perlu di definisikan nama tabel dan field datanya. Berikut perintah yang digunakan untuk mendefinisikan nama tabel dan field dalam tabel yang akan dibuat di dalam database:

```
...  
  
class CreateUomsTable extends Migration  
{  
    /**  
     * Run the migrations.  
     *  
     * @return void  
     */  
    public function up()  
    {  
        Schema::create('u_o_m_s', function (Blueprint $table) {  
            $table->id();  
            $table->string('name');  
            $table->timestamps();  
        });  
    }  
    ...  
}
```

**Gambar 4.8** Kode Program untuk Migrasi Tabel Uoms

Pada gambar 4.3 tertera informasi terkait pendefinisian migrasi untuk tabel `u_o_m_s`, yang berisi kolom `id`, `name`, `created_at`, `updated_at`. Tabel ini berperan untuk menyimpan data *unit of measurement* dari produk.

```
...  
  
class CreatePurchaseOrdersTable extends Migration  
{  
    /**  
     * Run the migrations.  
     * @return void  
     */  
    public function up()  
    {  
        Schema::create('purchase_orders', function (Blueprint $table) {  
            $table->id();  
            $table->double('total')->default(0.0);  
            $table->timestamp('arrived_at')->nullable();  
            $table->timestamps();  
        });  
    }  
}  
  
...
```

**Gambar 4.9** Kode Program untuk Migrasi Tabel PurchaseOrders

Pada gambar 4.4 tertera informasi terkait pendefinisian migrasi untuk tabel purchase\_orders, yang berisi kolom id, total, arrived\_at, created\_at, updated\_at. Tabel ini berperan untuk menyimpan data pembelian dari produk.

```

...

class CreatePurchaseItemsTable extends Migration
{
    /**
     * Run the migrations.
     * @return void
     */
    public function up()
    {
        Schema::create('purchase_items', function (Blueprint $table) {
            $table->double('qty');
            $table->unsignedInteger('purchase_order_id');
            $table->unsignedInteger('product_id');
            $table->unsignedInteger('inventory_id')->nullable();
            $table->timestamps();
        });
    }
}

...

```

**Gambar 4.10** Kode Program untuk Migrasi Tabel PurchaseItems

Pada gambar 4.5 tertera informasi terkait pendefinisian migrasi untuk tabel purchase\_items, yang berisi kolom qty, purchase\_order\_id, product\_id, inventory\_id, arrived\_at, created\_at, updated\_at. Tabel ini berperan untuk menyimpan data item pembelian dari produk. Pada tabel ini juga menyimpan jumlah pembelian serta data tempat stock untuk menyimpan produk yang sudah sampai.

```
...  
  
class CreateProductsTable extends Migration  
{  
    /**  
     * Run the migrations.  
     * @return void  
     */  
    public function up()  
    {  
        Schema::create('products', function (Blueprint $table) {  
            $table->id();  
            $table->string('name');  
            $table->double('price');  
            $table->boolean('can_manufacture');  
            $table->unsignedInteger('u_o_m_id');  
            $table->timestamps();  
        });  
    }  
}  
  
...
```

***Gambar 4.11*** Kode Program untuk Migrasi Tabel Products

Pada gambar 4.6 tertera informasi terkait pendefinisian migrasi untuk tabel products, yang berisi kolom id, name, price, can\_manufacture, u\_o\_m\_id, created\_at, updated\_at. Tabel ini berperan untuk menyimpan data umum dari produk.

```
...  
  
class CreateInventoriesTable extends Migration  
{  
    /**  
     * Run the migrations.  
     * @return void  
     */  
    public function up()  
    {  
        Schema::create('inventories', function (Blueprint $table) {  
            $table->id();  
            $table->unsignedInteger('product_id');  
            $table->string('lot')->unique();  
            $table->double('qty');  
            $table->timestamps();  
        });  
    }  
}  
  
...
```

**Gambar 4.12** Kode Program untuk Migrasi Tabel Inventories

Pada gambar 4.7 tertera informasi terkait pendefinisian migrasi untuk tabel inventories, yang berisi kolom id, produk\_id, lot, qty, created\_at, updated\_at. Tabel ini berperan untuk menyimpan data stock dari produk. Kolom lot digunakan sebagai pencatatan lokasi stock sesuai kebutuhan pengguna

```
...  
  
class CreateBOMSTable extends Migration  
{  
    /**  
     * Run the migrations.  
     * @return void  
     */  
    public function up()  
    {  
        Schema::create('b_o_m_s', function (Blueprint $table) {  
            $table->id();  
            $table->string('name');  
            $table->unsignedInteger('product_id');  
            $table->double('qty');  
            $table->timestamps();  
        });  
    }  
...  
}
```

**Gambar 4.13** Kode Program untuk Migrasi Tabel BOMS

Pada gambar 4.8 tertera informasi terkait pendefinisian migrasi untuk tabel `b_o_m_s`, yang berisi kolom `id`, `produk_id`, `name`, `qty`, `created_at`, `updated_at`. Tabel ini berperan untuk menyimpan data resep dari produk.

```
...  
  
class CreateBOMComponentsTable extends Migration  
{  
    /**  
     * Run the migrations.  
     *  
     * @return void  
     */  
    public function up()  
    {  
        Schema::create('b_o_m_components', function (Blueprint $  
table) {  
            $table->unsignedInteger('b_o_m_id');  
            $table->unsignedInteger('product_id');  
            $table->double('qty');  
            $table->timestamps();  
        });  
    }  
    ...  
}
```

**Gambar 4.14** Kode Program untuk Migrasi Tabel BOMComponents

Pada gambar 4.9 tertera informasi terkait pendefinisian migrasi untuk tabel `b_o_m_s_components`, yang berisi kolom `b_o_m_id`, `produk_id`, `qty`, `created_at`, `updated_at`. Tabel ini berperan untuk menyimpan data bahan baku / komponen dari resep produk.

```
...  
  
class CreateOrdersTable extends Migration  
{  
    /**  
     * Run the migrations.  
     * @return void  
     */  
    public function up()  
    {  
        Schema::create('orders', function (Blueprint $table) {  
            $table->id();  
            $table->unsignedInteger('product_id');  
            $table->unsignedInteger('b_o_m_id')->nullable();  
            $table->unsignedInteger('routing_id')->nullable();  
            $table->double('qty');  
            $table->timestamp('deadline_finish');  
            $table->timestamps();  
        });  
    }  
    ...  
}
```

**Gambar 4.15** Kode Program untuk Migrasi Tabel Orders

Pada gambar 4.10 tertera informasi terkait pendefinisian migrasi untuk tabel orders, yang berisi kolom id, routing\_id, b\_o\_m\_id, produk\_id, qty, created\_at, updated\_at. Tabel ini berperan untuk menyimpan data manufaktur order.

```
...  
  
class CreateReservesTable extends Migration  
{  
    /**  
     * Run the migrations.    * @return void  
     */  
    public function up()  
    {  
        Schema::create('reserves', function (Blueprint $table) {  
            $table->unsignedInteger('inventory_id')->nullable();  
            $table->unsignedInteger('order_id');  
            $table->double('qty');  
            $table->boolean('reserved')->default(false);  
            $table->timestamps();  
        });  
    }  
...  
}
```

**Gambar 4.16** Kode Program untuk Migrasi Tabel Reserves

Pada gambar 4.11 tertera informasi terkait pendefinisian migrasi untuk tabel reserves, yang berisi kolom inventory\_id, order\_id, qty, reserved, created\_at, updated\_at. Tabel ini berperan untuk menyimpan ketersediaan bahan baku dari order.

```
...  
  
class CreateWorkcentersTable extends Migration  
{  
    /**  
     * Run the migrations.  
     *  
     * @return void  
     */  
    public function up()  
    {  
        Schema::create('workcenters', function (Blueprint $table) {  
            $table->id();  
            $table->string('name');  
            $table->timestamps();  
        });  
    }  
    ...  
}
```

**Gambar 4.17** Kode Program untuk Migrasi Tabel WorkCenters

Pada gambar 4.12 tertera informasi terkait pendefinisian migrasi untuk tabel workcenters, yang berisi kolom id, name, created\_at, updated\_at. Tabel ini berperan untuk menyimpan data tempat kerja/mesin.

```
...  
  
class CreateOperationsTable extends Migration  
{  
    /**  
     * Run the migrations.   * @return void  
     */  
    public function up()  
    {  
        Schema::create('operations', function (Blueprint $table) {  
            $table->id();  
            $table->unsignedInteger('workcenter_id');  
            $table->unsignedInteger('routing_id');  
            $table->double('cost_per_hour');  
            $table->string('name');  
            $table->integer('seq');  
            $table->timestamps();  
        });  
    }  
    ...  
}
```

**Gambar 4.18** Kode Program untuk Migrasi Tabel Operations

Pada gambar 4.13 tertera informasi terkait pendefinisian migrasi untuk tabel operations, yang berisi kolom workcenter\_id, routing\_id, cost\_per\_hour, name, seq created\_at, updated\_at. Tabel ini berperan untuk menyimpan operasi dari rute.

```
...  
  
class CreateRoutingsTable extends Migration  
{  
    /**  
     * Run the migrations.  
     *  
     * @return void  
     */  
    public function up()  
    {  
        Schema::create('routings', function (Blueprint $table) {  
            $table->id();  
            $table->string('name');  
            $table->unsignedInteger('product_id');  
            $table->timestamps();  
        });  
    }  
}  
  
...
```

**Gambar 4.19** Kode Program untuk Migrasi Tabel Routings

Pada gambar 4.14 tertera informasi terkait pendefinisian migrasi untuk tabel routings, yang berisi kolom id, name, product\_id, created\_at, updated\_at. Tabel ini berperan untuk menyimpan rute produksi.

```
...  
  
class CreateCapacitiesTable extends Migration  
{  
    /**  
     * Run the migrations.     * @return void  
     */  
    public function up()  
    {  
        Schema::create('capacities', function (Blueprint $table) {  
            $table->unsignedInteger('workcenter_id');  
            $table->unsignedInteger('product_id');  
            $table->double('unit_capacity');  
            $table->timestamps();  
        });  
    }  
    ...  
}
```

*Gambar 4.20 Kode Program untuk Migrasi Tabel Capacities*

Pada gambar 4.15 tertera informasi terkait pendefinisian migrasi untuk tabel capacities, yang berisi kolom workcenter\_id, unit\_capacity, product\_id, created\_at, updated\_at. Tabel ini berperan untuk menyimpan kapasitas dari mesin produksi.

```
...  
  
class CreateWorkOrdersTable extends Migration  
{  
    /**  
     * Run the migrations.  
     * @return void  
     */  
    public function up()  
    {  
        Schema::create('work_orders', function (Blueprint $table) {  
            $table->id();  
            $table->unsignedInteger('order_id');  
            $table->unsignedInteger('operation_id');  
            $table->timestamp('start_date');  
            $table->timestamp('finish_date');  
            $table->integer('duration');  
            $table->timestamps();  
        });  
    }  
    ...  
}
```

**Gambar 4.21** Kode Program untuk Migrasi Tabel WorkOrders

Pada gambar 4.16 tertera informasi terkait pendefinisian migrasi untuk tabel `work_orders`, yang berisi kolom `id`, `order_id`, `operation_id`, `start_date`, `duration`, `finish_date`, `created_at`, `updated_at`. Tabel ini berperan untuk perintah kerja dari order.

Selanjutnya memanfaatkan console commant migrate pada laravel guna melakukan migrasi dari program menjadi sebuah tabel di dalam database. Berikut potongan kode program guna melakukan migrasi tertera pada gambar 4.15.

```
D:\Tugas Akhir\App\mpkp>php artisan migrate
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (0.01 seconds)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (0 seconds)
Migrating: 2020_06_01_200258_create_u_o_m_s_table
Migrated: 2020_06_01_200258_create_u_o_m_s_table (0 seconds)
Migrating: 2020_06_01_200442_create_purchase_orders_table
Migrated: 2020_06_01_200442_create_purchase_orders_table (0 seconds)
Migrating: 2020_06_01_200908_create_purchase_items_table
Migrated: 2020_06_01_200908_create_purchase_items_table (0 seconds)
Migrating: 2020_06_01_201548_create_products_table
Migrated: 2020_06_01_201548_create_products_table (0 seconds)
Migrating: 2020_06_01_203541_create_inventories_table
Migrated: 2020_06_01_203541_create_inventories_table (0 seconds)
Migrating: 2020_06_01_203929_create_b_o_m_s_table
Migrated: 2020_06_01_203929_create_b_o_m_s_table (0 seconds)
Migrating: 2020_06_01_204642_create_b_o_m_components_table
Migrated: 2020_06_01_204642_create_b_o_m_components_table (0 seconds)
Migrating: 2020_06_01_205635_create_orders_table
Migrated: 2020_06_01_205635_create_orders_table (0 seconds)
Migrating: 2020_06_01_210055_create_reserves_table
Migrated: 2020_06_01_210055_create_reserves_table (0 seconds)
Migrating: 2020_06_01_220117_create_workcenters_table
Migrated: 2020_06_01_220117_create_workcenters_table (0 seconds)
Migrating: 2020_06_01_220601_create_operations_table
Migrated: 2020_06_01_220601_create_operations_table (0 seconds)
Migrating: 2020_06_01_220614_create_routings_table
Migrated: 2020_06_01_220614_create_routings_table (0 seconds)
Migrating: 2020_06_01_221929_create_capacities_table
Migrated: 2020_06_01_221929_create_capacities_table (0 seconds)
Migrating: 2020_06_01_222544_create_work_orders_table
Migrated: 2020_06_01_222544_create_work_orders_table (0 seconds)
```

*Gambar 4.22 Perintah Konsol Artisan Migrate*

Setelah proses migrasi selesai, kemudian langkah selanjutnya ialah melakukan konfigurasi pada tabel dengan memanfaatkan tools Eloquent yang telah disediakan laravel. Berikut kode program yang digunakan untuk membangun keseluruhan konfigurasi tabel dan relasinya pada database model:

```
<?php  
  
namespace App;  
  
use Illuminate\Database\Eloquent\Model;  
  
class Operation extends Model  
{
```

***Gambar 4.23*** Kode Program untuk Model Operations

Pada Gambar 4.16 tertera potongan kode program yang digunakan untuk mendefinisikan nama class yang terhubung dengan tabel operations pada database. Selanjutnya melakukan konfigurasi terhadap field yang boleh diisi di dalam tabel operations dengan menjalankan perintah seperti yang tertera pada Gambar 4.11.

```
protected $fillable = ['seq', 'name', 'cost_per_hour'];
```

***Gambar 4.24*** Kode Program untuk Kolom Model Operations

Kemudian tabel operation harus dihubungkan dengan tabel guna menjalankan komunikasi data dalam tabel operations dengan tabel lain dengan menjalankan perintah seperti pada Gambar 4.18.

```
public function workcenter(){
    return $this->belongsTo('App\Workcenter');
}

public function routing(){
    return $this->belongsTo('App\Routing');
}

public function workOrders(){
    return $this->hasMany('App\WorkOrder');
}
```

**Gambar 4.25** Kode Program untuk Relasi Model Operations

Pada Gambar 4.18 tertera informasi terkait relasi data pada tabel operations yang memiliki relasi dengan tabel Routing, tabel WorkCenter, dan tabel WorkOrder.

```
protected static function booted()
{
    static::deleting(function ($operation) {
        foreach ($operation->workOrders as $ord){
            $ord->delete();
        }
    });
}
```

**Gambar 4.26** Kode Program untuk aksi relasi Model Operation

Pada gambar 4.19 berfungsi untuk memberi tahu kerangka kerja laravel tentang operasi yang perlu dilakukan sebelum menghapus data Operation.

Selanjutnya ialah melakukan konfigurasi pada tabel BOMS dengan memanfaatkan tools Eloquent telah disediakan laravel. Berikut kode program yang digunakan untuk membangun keseluruhan konfigurasi tabel dan relasinya pada database model:

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class BOM extends Model
{
```

**Gambar 4.27** Kode Program untuk Model BOM

Pada Gambar 4.20 tertera kode yang digunakan untuk mendefinisikan nama class yang terhubung dengan tabel BOMS pada database. Selanjutnya melakukan konfigurasi terhadap field yang boleh diisi di dalam tabel BOMS hanya field name dan qty dengan menjalankan perintah seperti yang tertera pada Gambar 4.6

```
protected $fillable = ['name','qty'];

public function product()
{
    return $this->belongsTo('App\Product');
}

public function ingredients()
{
    return $this -> belongsToMany
('App\Product', 'b_o_m_components') -> withPivot('qty');
}

public function orders(){
    return $this->hasMany('App\Order');
}
```

**Gambar 4.29** Kode Program untuk relasi Model BOM

Kemudian tabel BOM harus dihubungkan dengan tabel guna menjalankan komunikasi data dalam tabel BOM dengan tabel lain dengan menjalankan perintah seperti pada Gambar 4.22.

Pada Gambar 4.22 tertera informasi terkait relasi data pada tabel BOMS yang memiliki relasi dengan tabel Products, Ingredients, dan Orders.

```
protected static function booted()
{
    static::deleting(function ($bom) {
        $bom->ingredients()->detach();
        foreach ($bom->orders as $ord){
            $ord->delete();
        }
    });
}
```

**Gambar 4.30** Kode Program untuk aksi relasi Model BOM

Pada gambar 4.23 berfungsi untuk memberi tahu kerangka kerja laravel tentang operasi yang perlu dilakukan sebelum menghapus data BOM.

Selanjutnya ialah melakukan konfigurasi pada tabel Inventory dengan memanfaatkan tools Eloquent telah disediakan laravel. Berikut kode program yang digunakan untuk membangun keseluruhan konfigurasi tabel dan relasinya pada database model:

```
<?php  
  
namespace App;  
  
use Illuminate\Database\Eloquent\Model;  
  
class Inventory extends Model  
{
```

***Gambar 4.31*** Kode Program untuk Model Inventory

Pada Gambar 4.24 tertera kode program yang digunakan untuk mendefinisikan nama class yang terhubung dengan tabel Inventory pada database. Selanjutnya melakukan konfigurasi terhadap field yang boleh diisi di dalam tabel Inventory hanya field lot dan qty dengan menjalankan perintah seperti yang tertera pada Gambar 4.25.

```
protected $fillable = ['lot','qty'];
```

***Gambar 4.32*** Kode Program untuk kolom Model Inventory

Kemudian tabel Inventory harus dihubungkan dengan tabel guna menjalankan komunikasi data dalam tabel Inventory dengan tabel lain dengan menjalankan perintah seperti pada Gambar 4.26

```
public function product(){
    return $this->belongsTo('App\Product');
}

public function reserves(){
    return $this->belongsToMany('App\Order', 'reserves')-
>withPivot('qty','reserved')->with('product');
}

public function purchases(){
    return $this-
>belongsToMany('App\PurchaseOrder', 'purchase_items')-
>withPivot('qty');
}
```

**Gambar 4.33** Kode Program untuk relasi Model Inventory

Pada Gambar 4.26 tertera informasi terkait relasi data pada tabel Inventory yang memiliki relasi dengan tabel Products, tabel Reserve dan tabel Purchase.

```

protected static function booted()
{
    static::deleting(function ($stock) {
        $stock->reserves()->updateExistingPivot($stock-
>id, ['inventory_id' => null]);
        $stock->purchases()->updateExistingPivot($stock-
>id, ['inventory_id' => null]);
    });
}

```

**Gambar 4.34** Kode Program untuk aksi relasi Model Inventory

Pada gambar 4.27 berfungsi untuk memberi tahu kerangka kerja laravel tentang operasi yang perlu dilakukan sebelum menghapus data Inventory.

Selanjutnya ialah melakukan konfigurasi pada tabel Order dengan memanfaatkan tools model yang telah disediakan laravel. Berikut kode program yang digunakan untuk membangun keseluruhan konfigurasi tabel dan relasinya pada database model:

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Order extends Model
{

```

**Gambar 4.35** Kode program untuk Model Order

Pada Gambar 4.28 tertera kode program yang digunakan untuk mendefinisikan nama class yang terhubung dengan tabel Order pada database. Selanjutnya melakukan konfigurasi terhadap field yang boleh diisi di dalam tabel Order hanya field qty, deadline\_start, dan deadline\_finish dengan menjalankan perintah seperti yang tertera pada Gambar 4.29.

```
protected $fillable = ['qty','deadline_start','deadline_finish'];
```

***Gambar 4.36*** Kode program untuk kolom Model Order

Kemudian tabel Inventory harus dihubungkan dengan tabel guna menjalankan komunikasi data dalam tabel Order dengan tabel lain dengan menjalankan perintah seperti pada Gambar 4.30.

```
public function product(){
    return $this->belongsTo('App\Product');
}

public function recipe(){
    return $this->belongsTo('App\BOM');
}

public function ingredients(){
    //include qty on pivot table and take product from related e
nd
    return $this->belongsToMany('App\Inventory', 'reserves')-
>withPivot('qty','reserved')->with('product');
}

public function routing(){
    return $this->belongsTo('App\Routing');
}

public function workOrders(){
    return $this->hasMany('App\WorkOrder');
}
```

**Gambar 4.37** Kode program untuk relasi Model Order

Pada Gambar 4.30 tertera informasi terkait relasi data pada tabel Order yang memiliki relasi dengan tabel Receipt, tabel Ingredient, tabel Routing, dan tabel WorkOrder.

```
protected static function booted()
{
    static::deleting(function ($workCenter) {
        foreach ($workCenter->workOrders as $ord){
            $ord->delete();
        }
        $workCenter->ingredients()->detach();
    });
}
```

**Gambar 4.38** Kode program untuk aksi relasi Model Order

Pada gambar 4.31 berfungsi untuk memberi tahu kerangka kerja laravel tentang operasi yang perlu dilakukan sebelum menghapus data Order.

Selanjutnya ialah melakukan konfigurasi pada tabel Product dengan memanfaatkan tools model yang telah disediakan laravel. Berikut kode program yang digunakan untuk membangun keseluruhan konfigurasi tabel dan relasinya pada database model:

```
<?php

namespace App;

use Carbon\Carbon;
use Illuminate\Database\Eloquent\Model;

class Product extends Model
{
```

**Gambar 4.39** Kode program untuk Model Product

Pada Gambar 4.32 tertera potongan kode program yang digunakan untuk mendefinisikan nama class yang terhubung dengan tabel Product pada database. Selanjutnya melakukan konfigurasi terhadap field yang boleh diisi di dalam tabel Product hanya field name, price, can\_manufacture dengan menjalankan perintah seperti yang tertera pada Gambar 4.33

```
protected $fillable = ['name', 'price', 'can_manufacture'];
```

*Gambar 4.40 Kode program untuk kolom Model Product*

Kemudian tabel Inventory harus dihubungkan dengan tabel guna menjalankan komunikasi data dalam tabel Product dengan tabel lain dengan menjalankan perintah seperti pada Gambar 4.34

```
public function uom()
{
    return $this->belongsTo('App\UOM', 'u_o_m_id');
}

public function purchases()
{
    return $this-
>belongsToMany('App\PurchaseOrder', 'purchase_items')-
>withPivot('qty');
}

public function stocks()
{
    return $this->hasMany('App\Inventory');
}

public function recipes()
{
    return $this->hasMany('App\BOM');
}

public function routings()
{
    return $this->hasMany('App\Routing');
}

public function orders()
{
    return $this->hasMany('App\Order');
}
```

**Gambar 4.41** Kode program untuk relasi Model Product

Pada Gambar 4.34 tertera informasi terkait relasi data pada tabel Product yang memiliki relasi dengan tabel UOM, tabel Purchase, tabel Stock, tabel Reciept, tabel Routing, tabel Order.

```
protected static function booted()
{
    static::updated(function ($product){
        foreach ($product->purchases as $purchase){
            $purchase->updated_at = Carbon::now();
            $purchase->save();
        }
    });
    static::deleting(function ($product) {
        foreach ($product->purchases as $pcs) {
            $pcs->delete();
        }
        $product->purchases()->detach();
        foreach ($product->stocks as $stock) {
            $stock->delete();
        }
        foreach ($product->recipes as $recipe) {
            $recipe->delete();
        }
        foreach ($product->routings as $routing) {
            $routing->delete();
        }
        foreach ($product->orders as $order) {
            $order->delete();
        }
    });
}
```

**Gambar 4.42** Kode Program untuk Aksi Relasi Model Product

Pada gambar 4.35 berfungsi untuk memberi tahu kerangka kerja laravel tentang operasi yang perlu dilakukan sebelum menghapus dan atau mengubah data Product.

Selanjutnya ialah melakukan konfigurasi pada tabel Purchase Order dengan memanfaatkan tools model yang telah disediakan laravel. Berikut kode program yang digunakan untuk membangun keseluruhan konfigurasi tabel dan relasinya pada database model :

```
<?php  
  
namespace App;  
  
use Illuminate\Database\Eloquent\Model;  
  
class PurchaseOrder extends Model  
{
```

**Gambar 4.43** Kode Program untuk Model Purchase Order

Pada Gambar 4.36 tertera potongan kode program yang digunakan untuk mendefinisikan nama class yang terhubung dengan tabel Purchase Order pada database. Selanjutnya melakukan konfigurasi terhadap field yang boleh diisi di dalam tabel Purchase Order hanya field total dan arrived\_at finish dengan menjalankan perintah seperti yang tertera pada Gambar 4.37.

```
protected $fillable = ['total', 'arrived_at'];
```

***Gambar 4.44 Kode Program untuk Kolom Model Purchase Order***

Kemudian tabel Inventory harus dihubungkan dengan tabel guna menjalankan komunikasi data dalam tabel Purchase Order dengan tabel lain dengan menjalankan perintah seperti pada Gambar 4.38.

```
public function items()
{
    return $this-
>belongsToMany('App\Product', 'purchase_items')-
>withPivot(['qty', 'inventory_id']);
}

public function stocks()
{
    return $this-
>belongsToMany('App\Inventory', 'purchase_items')-
>withPivot(['qty', 'product_id']);
}
```

**Gambar 4.45** Kode Program untuk Relasi Model Purchase Order

Pada Gambar 4.38 tertera informasi terkait relasi data pada tabel Purchase Order yang memiliki relasi dengan tabel Item dan tabel Stock.

```

protected static function booted()
{
    static::updating(function ($purchaseOrder) {

        $sum = 0;
        foreach ($purchaseOrder->items as $item) {
            $sum += $item->price * $item->pivot->qty;
        }

        $purchaseOrder->total = $sum;

    });
    static::deleting(function ($purchaseOrder) {
        $purchaseOrder->items()->detach();
    });
}

```

**Gambar 4.46** Kode Program untuk Aksi Relasi Model Purchase Order

Pada gambar 4.39 berfungsi untuk memberi tahu kerangka kerja laravel tentang operasi yang perlu dilakukan sebelum menghapus dan atau mengubah data Purchase Order.

Selanjutnya ialah melakukan konfigurasi pada tabel Routing dengan memanfaatkan tools model yang telah disediakan laravel. Berikut kode program yang digunakan untuk membangun keseluruhan konfigurasi tabel dan relasinya pada database model:

```
<?php  
  
namespace App;  
  
use Illuminate\Database\Eloquent\Model;  
  
class Routing extends Model  
{
```

**Gambar 4.47** Kode Program untuk Model Routing

Pada Gambar 4.40 tertera potongan kode program yang digunakan untuk mendefinisikan nama class yang terhubung dengan tabel Routing pada database. Selanjutnya melakukan konfigurasi terhadap field yang boleh diisi di dalam tabel Routing hanya field name finish dengan menjalankan perintah seperti yang tertera pada Gambar 4.41

```
protected $fillable = ['name'];
```

**Gambar 4.48** Kode Program untuk Kolom Model Routing

Kemudian tabel Routing harus dihubungkan dengan tabel guna menjalankan komunikasi data dalam tabel Routing dengan tabel lain dengan menjalankan perintah seperti pada Gambar 4.42.

```
protected $fillable = ['name'];

public function product(){
    return $this->belongsTo('App\Product');
}

public function operations(){
    return $this->hasMany('App\Operation');
}

public function orders(){
    return $this->hasMany('App\Order');
}
```

**Gambar 4.49** Kode Program untuk Relasi Model Routing

Pada Gambar 4.42 tertera informasi terkait relasi data pada tabel Routing yang memiliki relasi dengan tabel Product, tabel Operation, dan tabel Order.

```
protected static function booted()
{
    static::deleting(function ($routing) {
        foreach ($routing->operations as $operation){
            $operation->delete();
        }
        foreach ($routing->orders as $ord){
            $ord->delete();
        }
    });
}
```

**Gambar 4.50** Kode Program untuk Aksi Relasi Model Routing

Pada gambar 4.43 berfungsi untuk memberi tahu kerangka kerja laravel tentang operasi yang perlu dilakukan sebelum menghapus dan atau mengubah data Routing.

Selanjutnya ialah melakukan konfigurasi pada tabel UOM dengan memanfaatkan tools model yang telah disediakan laravel. Berikut kode program yang digunakan untuk membangun keseluruhan konfigurasi tabel dan relasinya pada database model :

```
<?php  
  
namespace App;  
  
use Illuminate\Database\Eloquent\Model;  
  
class UOM extends Model  
{
```

***Gambar 4.51 Kode Program untuk Model UoM***

Pada Gambar 4.44 tertera potongan kode program yang digunakan untuk mendefinisikan nama class yang terhubung dengan tabel UOM pada database. Selanjutnya melakukan konfigurasi terhadap field yang boleh diisi di dalam tabel UOM hanya field name finish dengan menjalankan perintah seperti yang tertera pada Gambar 4.45

```
protected $fillable = ['name'];
```

***Gambar 4.52 Kode Program untuk Kolom Model UoM***

Kemudian tabel UOM harus dihubungkan dengan tabel guna menjalankan komunikasi data dalam tabel UOM dengan tabel lain dengan menjalankan perintah seperti pada Gambar 4.46

```

public function products()
{
    return $this->hasMany('App\Product');
}

```

**Gambar 4.53** Kode Program untuk Relasi Model UoM

Pada Gambar 4.46 tertera informasi terkait relasi data pada tabel UOM yang memiliki relasi dengan tabel Product.

```

protected static function booted()
{
    static::deleting(function ($uom) {
        foreach ($uom->products as $product){
            $product->delete();
        }
    });
}

```

**Gambar 4.54** Kode Program untuk Aksi Relasi Model UoM

Pada gambar 4.47 berfungsi untuk memberi tahu kerangka kerja laravel tentang operasi yang perlu dilakukan sebelum menghapus dan atau mengubah data UoM.

Selanjutnya ialah melakukan konfigurasi pada tabel Work Center dengan memanfaatkan tools model yang telah disediakan laravel. Berikut kode program yang digunakan untuk membangun keseluruhan konfigurasi tabel dan relasinya pada database model :

```
<?php  
  
namespace App;  
  
use Illuminate\Database\Eloquent\Model;  
  
class Workcenter extends Model  
{
```

**Gambar 4.55** Kode Program untuk Model Workcenter

Pada Gambar 4.48 tertera potongan kode program yang digunakan untuk mendefinisikan nama class yang terhubung dengan tabel Work Center pada database. Selanjutnya melakukan konfigurasi terhadap field yang boleh diisi di dalam tabel Work Center hanya field name dengan menjalankan perintah seperti yang tertera pada Gambar 4.49.

```
protected $fillable = ['name'];
```

**Gambar 4.56** Kode Program untuk Kolom Model Workcenter

Kemudian tabel Inventory harus dihubungkan dengan tabel guna menjalankan komunikasi data dalam tabel Work Center dengan tabel lain dengan menjalankan perintah seperti pada Gambar 4.50.

```
public function capacities()
{
    return $this->belongsToMany('App\Product','capacities')-
>withPivot('unit_capacity');
}

public function workOrders(){
    return $this-
>hasManyThrough('App\WorkOrder','App\Operation');
}

public function operations(){
    return $this->hasMany('App\Operation');
}
```

**Gambar 4.57** Kode Program untuk Relasi Model Workcenter

Pada Gambar 4.50 tertera informasi terkait relasi data pada tabel Work Center yang memiliki relasi dengan tabel Capacities, tabel workOrder, dan tabel Operations

```

protected static function booted()
{
    static::deleting(function ($workCenter) {
        foreach ($workCenter->operations as $operation){
            $operation->delete();
        }
        $workCenter->capacities()->detach();
    });
}

```

**Gambar 4.58** Kode Program untuk Aksi Relasi Model Workcenter

Pada gambar 4.51 berfungsi untuk memberi tahu kerangka kerja laravel tentang operasi yang perlu dilakukan sebelum menghapus dan atau mengubah data Workcenter.

Selanjutnya ialah melakukan konfigurasi pada tabel Work Order dengan memanfaatkan tools model yang telah disediakan laravel. Berikut kode program yang digunakan untuk membangun keseluruhan konfigurasi tabel dan relasinya pada database model :

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Workorder extends Model
{

```

**Gambar 4.59** Kode Program untuk Model WorkOrders

Pada Gambar 4.52 tertera potongan kode program yang digunakan untuk mendefinisikan nama class yang terhubung dengan tabel Work Order pada database. Selanjutnya melakukan konfigurasi terhadap field yang boleh diisi di dalam tabel Work Order hanya field start\_date, finish\_date, dan duration finish dengan menjalankan perintah seperti yang tertera pada Gambar 4.53

```
protected $fillable = ['start_date', 'finish_date', 'duration'];
```

**Gambar 4.60** Kode Program untuk Kolom Model WorkOrders

Kemudian tabel Work Order harus dihubungkan dengan tabel guna menjalankan komunikasi data dalam tabel Work Order dengan tabel lain dengan menjalankan perintah seperti pada Gambar 4.54.

```
public function operation(){  
    return $this->belongsTo('App\operation');  
}  
  
public function order(){  
    return $this->belongsTo('App\Order');  
}
```

**Gambar 4.61** Kode Program untuk Relasi Model WorkOrders

Pada Gambar 4.54 tertera informasi terkait relasi data pada tabel Work Order yang memiliki relasi dengan tabel Operation dan tabel Order.

Setelah semua tabel dan relasinya di buat dan dilakukan migrasu, maka akan didapat sebuah desain database model yang dapat dilihat pada Gambar 4.62.



*Gambar 4.62 Hasil Implementasi Database Model*

### 4.3.2 Konfigurasi Routing Web Server

Dalam mengimplementasikan konfigurasi routing web server memanfaatkan tools routes yang disediakan laravel. Keseluruhan akses halaman perangkat lunak dilakukan melalui url yang dikelola oleh tools routes melalui file routing yang disediakan oleh laravel. Konfigurasi diawali dengan mendefinisikan url routing yang akan diarahkan ke sebuah halaman perangkat lunak melalui controller. Halaman awal yang routing perangkat lunak secara otomatis diarahkan menggunakan perintah pada Gambar 4.56

```
Route::get('/', 'HomeController@index');
```

Gambar 4.63 Laravel General Routing

Kemudian untuk mengakses menu MRP yang terdiri akses fitur product, akses fitur Purchase, akses fitur Stock, akses fitur UOM, dan akses fitur BOM dilakukan melalui beberapa perintah routing pada Gambar 4.57.

```
//MRP Route
Route::resource('products', 'ProductController')->except('create');
Route::resource('uoms', 'UOMController')->except(['show', 'create']);
Route::resource('boms', 'BOMController')->except('create');
Route::resource('stocks', 'InventoryController', [
    'parameters' => [
        'stocks' => 'inventory'
    ]
])->except('create');
Route::resource('boms.ingredients', 'IngredientController', [
    'parameters' => [
        'ingredients' => 'product'
    ]
])->except(['show', 'index', 'create']);

Route::resource('purchases', 'PurchaseOrderController', [
    'parameters' => [
        'purchases' => 'purchase_order'
    ]
])->except('create');

Route::resource('purchases.items', 'PurchaseItemController', [
    'parameters' => [
        'purchases' => 'purchase_order',
        'items' => 'product'
    ]
])->except(['show', 'index', 'create']);
```

**Gambar 4.64** *Laravel MRP Routing*

Kemudian untuk mengakses menu Production Scheduling yang terdiri akses fitur Resource dilakukan melalui beberapa perintah routing pada Gambar 4.58.

```
//Production Scheduling
Route::resource('orders', 'OrderController')
->except('create');

Route::resource('orders.workorders', 'Workko
rderController')-
>except(['show', 'index', 'create']);
```

Gambar 4.65 Laravel PS Routing

Kemudian untuk mengakses menu CRP yang terdiri akses fitur Work Center dan akses fitur Routing dilakukan melalui beberapa perintah routing pada Gambar 4.59

```
//CRP
Route::resource('workcenters', 'WorkcenterC
ontroller')->except('create');

Route::resource('workcenters.capacities', '
CapacityController', [
    'parameters' => [
        'capacities' => 'product'
    ]
])->except(['show', 'index', 'create']);

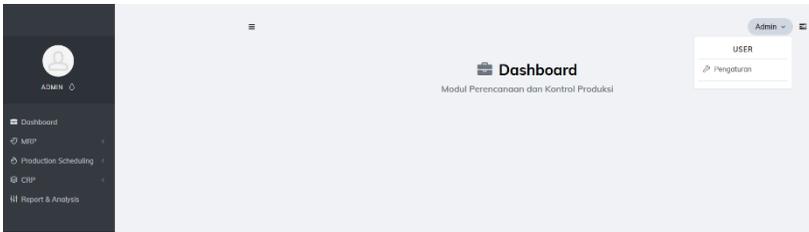
Route::resource('routings', 'RoutingControl
ler')->except('create');

Route::resource('routings.operations', 'Ope
rationController')-
>except(['show', 'index', 'create']);
```

*Gambar 4.66 Laravel CRP Routing*

#### **4.4 Implementasi Modul Perencanaan dan Kontrol Produksi**

Pada Tugas Akhir ini dibuat sebuah modul perencanaan dan kontrol produksi yang dibagi menjadi empat menu utama yaitu *Maaterial Requirement Planning*, *Capacity Requirement Planning*, dan *Production Scheduling*, serta dilengkapi dengan fitur *Report and Analysis*.

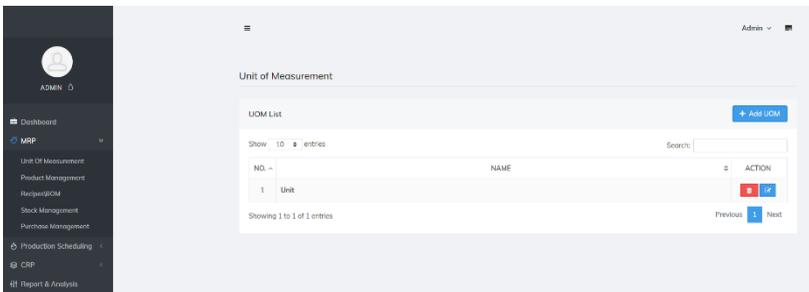


**Gambar 4.67** Antar Muka Home

Pada gambar 4.60 merupakan tampilan halaman utama yang menampilkan seluruh daftar menu pada perangkat lunak Perencanaan dan kontrol produksi dibagian sidebar. User dapat dengan mudah mengakses keseluruhan menu melalui sidebar yang dibuat tampil static dengan posisi fixed.

#### 4.4.1 Implementasi Daftar Fitur pada Menu MRP

Menu utama MRP terdiri dari empat fitur yaitu Product, fitur Receipt \ BOM, fitur stock management, dan fitur purchase management. Tampilan daftar fitur dibuat static position fixed guna memudahkan user dalam mengakses fitur-fitur MRP secara langsung melalui sidebar dropdown.



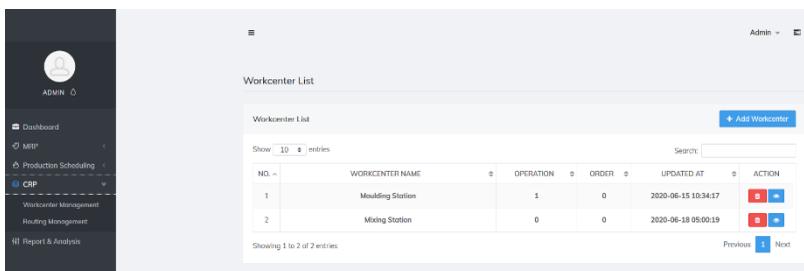
**Gambar 4.68** Antar Muka Menu MRP

Pada gambar 4.61 tertera informasi terkait menu utama Material Requirement Planning yang terdiri dari fitur Product management

guna mengelola data produk pada proses perencanaan dan kontrol produksi, fitur Receipt \ BOM guna mengelola data resep yang akan digunakan pada saat proses produksi, fitur stock management guna mengelola data stok produk yang tersedia, dan fitur purchase management guna mengelola data pembelian untuk proses produksi.

#### 4.4.2 Implementasi Daftar Fitur Menu CRP

Menu utama CRP terdiri dari dua fitur yaitu fitur Work Center Management dan fitur Routing. Tampilan daftar fitur dibuat static position fixed guna memudahkan user dalam mengakses fitur-fitur MRP secara langsung melalui sidebar dropdown.



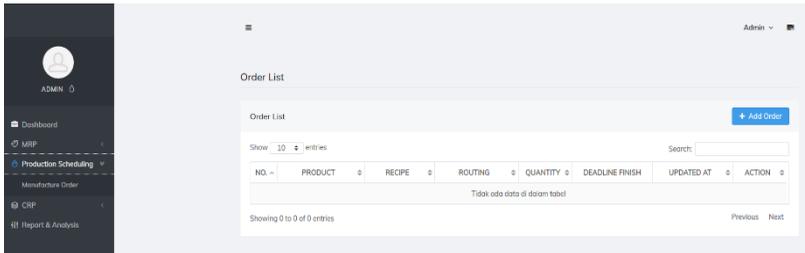
**Gambar 4.69** Antar Muka Menu CRP

Pada gambar 4.62 tertera informasi terkait menu utama Capacity Requirement Planning yang terdiri dari fitur Workcenter Management guna mengelola data kapasitas produksi berdasarkan kapasitas dan jumlah unit kerja produksi dan fitur Routing Management guna mengelola data tahapan proses pengolahan bahan baku dalam proses produksi.

#### 4.4.3 Implementasi Daftar Menu Production Scheduling

Menu utama Production Scheduling terdiri dari dua fitur yaitu fitur Production Schedule dan fitur Work Order. Tampilan daftar fitur dibuat static position fixed guna memudahkan user dalam

mengakses fitur-fitur MRP secara langsung melalui sidebar dropdown.

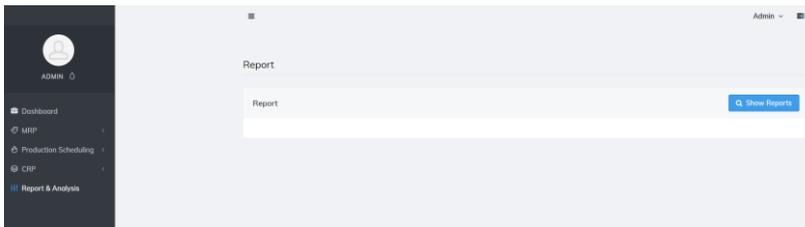


**Gambar 4.70** Antar Muka Menu PS

Pada gambar 4.63 tertera informasi terkait menu utama Production Schedule yang terdiri dari fitur Manufacture Order guna mengelola data pesanan yang masuk dari pelanggan dan fitur Work Order guna mengelola data daftar tahapan proses produksi.

#### 4.4.4 Implementasi Daftar Menu Report

Menu utama Report terdiri dari satu fitur utama yaitu Report&Analysis. Tampilan daftar fitur dibuat static position fixed guna memudahkan user dalam mengakses fitur-fitur Report secara langsung melalui sidebar dropdown.



**Gambar 4.71** Antar Muka Menu Report

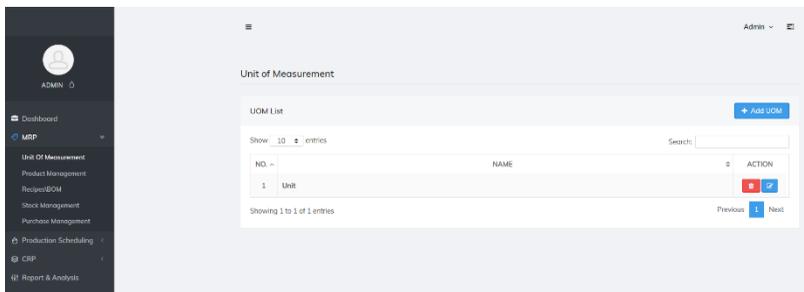
Pada gambar 4.64 tertera informasi terkait menu report yang terdiri dari tombol Show Report. Pada menu ini pengguna dapat meminta

sistem untuk menampilkan hasil kalkulasi makespan pada rentang waktu tertentu.

## 4.5 Implementasi Menu MRP

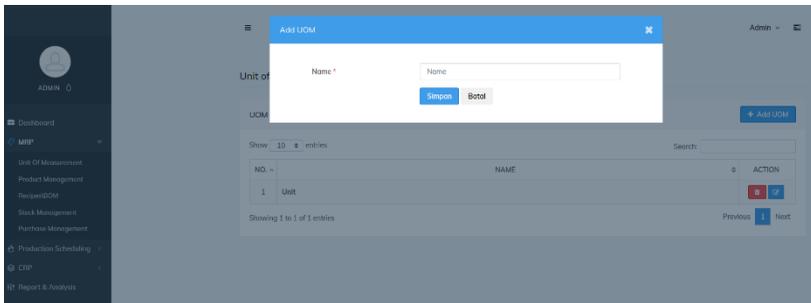
Pada menu utama Material Requirement Planning yang terdiri dari fitur Product management guna mengelola data produk pada proses perencanaan dan kontrol produksi, fitur Receipt \ BOM guna mengelola data resep yang akan digunakan pada saat proses produksi, fitur stock management guna mengelola data stok produk yang tersedia, dan fitur purchase management guna mengelola data pembelian untuk proses produksi.

Implementasi diawali dengan membuat halaman CRUD untuk pengelolaan data sumber seperti UoM dan product yang akan banyak direlasikan ke data tabel lain.



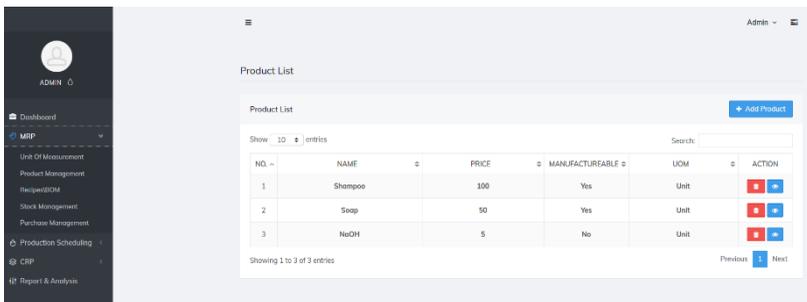
**Gambar 4.72** Antar Muka Menu UoM

Pada Gambar 4.65 tertera halaman data tabel daftar UoM yang menampilkan semua data UoM yang sudah dibuat seta fungsi hapus dan edit pada setiap baris data. Pada data tabel juga dibuat fungsi pencarian dan navigasi halaman untuk mempercepat waktu pencarian data oleh user. Pada bagian kanan atas tabel disematkan tombol yang difungsikan untuk menambah data UoM. Formulir tambah data UoM dapat dilihat pada Gambar 4.66.



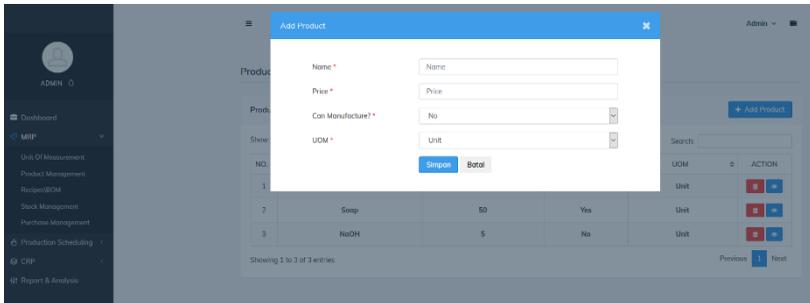
**Gambar 4.73** Antar Muka Tambah UoM

Pada Gambar 4.66 tertera tampilan formulir tambah data UoM yang dibuat dalam bentuk pop-up guna memberi kemudahan bagi user untuk tetap berada pada halaman yang sama dan juga guna mendapatkan efisiensi waktu proses memuat formulir.



**Gambar 4.74** Antar Muka Produk

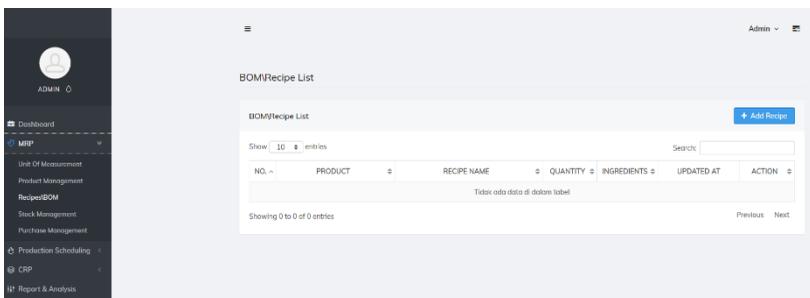
Pada Gambar 4.67 tertera halaman data tabel daftar Product yang menampilkan semua data Product yang sudah dibuat seta fungsi hapus dan edit pada setiap baris data. Pada data tabel juga dibuat fungsi pencarian dan navigasi halaman untuk mempercepat waktu pencarian data oleh user. Pada bagian kanan atas tabel disematkan tombol yang difungsikan untuk menambah data Product. Formulir tambah data Product dapat dilihat pada Gambar 4.68



**Gambar 4.75** Antar Muka Tambah Produk

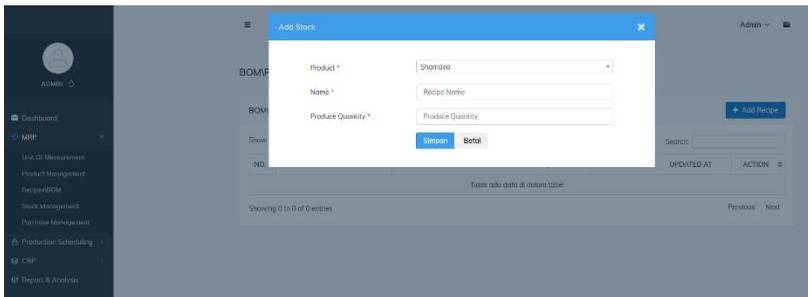
Pada Gambar 4.68 tertera tampilan formulir tambah data Product yang dibuat dalam bentuk pop-up guna memberi kemudahan bagi user untuk tetap berada pada halaman yang sama dan juga guna mendapatkan efisiensi waktu proses memuat formulir. Di dalam formulir terdapat kolom input UoM bertipe pilihan yang direlasikan dengan tabel UoM sehingga semua data UoM dapat langsung ditampilkan sebagai pilihan nilai input pada formulir.

Selanjutnya dibuat fitur BOM/Recipe guna mengelola data resep yang akan digunakan pada saat proses produksi, fungsi dari menu recipe ini untuk menentukan bahan baku yang digunakan di dalam kegiatan produksi.



**Gambar 4.76** Antar Muka BOM

Pada Gambar 4.69 tertera halaman data tabel daftar Recipe yang menampilkan semua data Product yang sudah dibuat seta fungsi hapus dan edit pada setiap baris data. Pada data tabel juga dibuat fungsi pencarian dan navigasi halaman untuk mempercepat waktu pencarian data oleh user. Pada bagian kanan atas tabel disematkan tombol yang difungsikan untuk menambah data Product. Formulir tambah data Recipe dapat dilihat pada Gambar 4.70



**Gambar 4.77** Antar Muka Tambah BOM

Pada Gambar 4.70 tertera tampilan formulir tambah data Recipe yang dibuat dalam bentuk pop-up guna memberi kemudahan bagi user untuk tetap berada pada halaman yang sama dan juga guna mendapatkan efisiensi waktu proses memuat formulir. Di dalam formulir terdapat kolom input Product bertipe pilihan yang direlasikan dengan tabel Product sehingga semua data Product dapat langsung ditampilkan sebagai pilihan nilai input pada formulir.

Selanjutnya dibuat fitur Stock Management guna mengelola data stok produk yang tersedia, fungsi dari menu stock ini untuk mengelola stok dari produk.

Stock List

Stock List + Add Stock

Show 10 entries Search:

NO. #	PRODUCT	LOT #	QTY #	UPDATED AT #	ACTION
1	Shampoo	AA01	150	2020-06-17 15:32:25	<span style="color: red;">-</span> <span style="color: blue;">+</span>
2	Shampoo	AA01	0	2020-06-18 23:43:18	<span style="color: red;">-</span> <span style="color: blue;">+</span>

Showing 1 to 2 of 2 entries Previous Next

**Gambar 4.78** Antar Muka Stock

Pada Gambar 4.71 tertera halaman data tabel daftar Stock yang menampilkan semua data Product yang sudah dibuat seta fungsi hapus dan edit pada setiap baris data. Pada data tabel juga dibuat fungsi pencarian dan navigasi halaman untuk mempercepat waktu pencarian data oleh user. Pada bagian kanan atas tabel disematkan tombol yang difungsikan untuk menambah data Product. Formulir tambah data Stock dapat dilihat pada Gambar 4.72.

Add Stock

Stock

Product \*

LOT \*

Quantity \*

Stock

Show

NO.	PRODUCT	LOT	QTY	UPDATED AT	ACTION
1	Shampoo	AA01	150	2020-06-17 15:32:25	<span style="color: red;">-</span> <span style="color: blue;">+</span>
2	Shampoo	AA01	0	2020-06-18 23:43:18	<span style="color: red;">-</span> <span style="color: blue;">+</span>

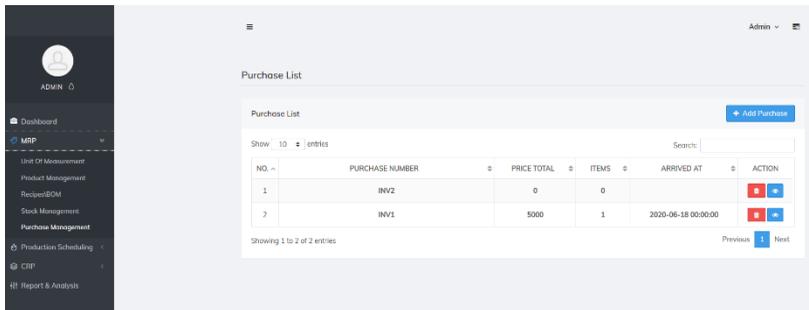
Showing 1 to 2 of 2 entries Previous Next

**Gambar 4.79** Antar Muka Tambah Stock

Pada Gambar 4.72 tertera tampilan formulir tambah data Stock yang dibuat dalam bentuk pop-up guna memberi kemudahan bagi user untuk tetap berada pada halaman yang sama dan juga guna mendapatkan efisiensi waktu proses memuat formulir. Di dalam formulir terdapat kolom input Product bertipe pilihan yang direlasikan

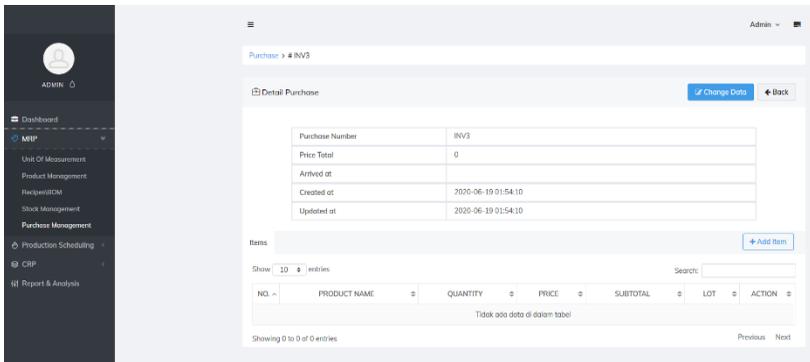
dengan tabel Product sehingga semua data Product dapat langsung ditampilkan sebagai pilihan nilai input pada formulir.

Selanjutnya dibuat fitur Purchase Management guna mengelola data pengadaan stok produk, fungsi dari menu Purchase Management ini untuk mengelola data pembelian bahan baku.



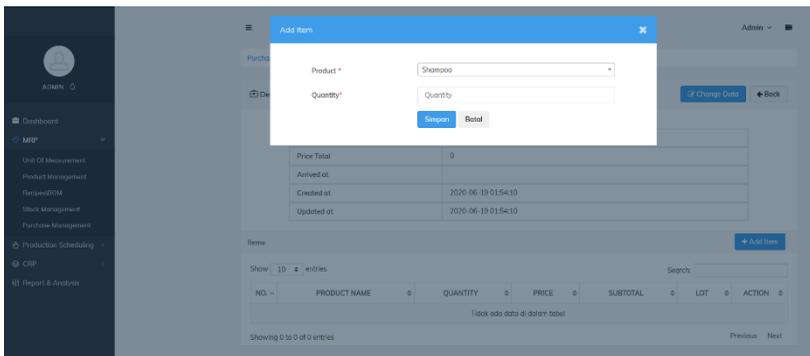
**Gambar 4.80** Antar Muka Purchase

Pada Gambar 4.73 tertera halaman data tabel daftar Purchase yang menampilkan semua data Product yang sudah dibuat seta fungsi hapus dan edit pada setiap baris data. Pada data tabel juga dibuat fungsi pencarian dan navigasi halaman untuk mempercepat waktu pencarian data oleh user. Pada bagian kanan atas tabel disematkan tombol yang difungsikan untuk menambah data Purchase. Formulir tambah data Stock dapat dilihat pada Gambar 4.74



**Gambar 4.81** Antar Muka Show Purchase

Pada Gambar 4.74 tertera tampilan formulir tambah data Purchase yang dibuat dalam dua bagian yaitu dibagian atas untuk input data purchase dan di bagian bawah untuk menampilkan preview data produk guna memberi kemudahan bagi user untuk tetap berada pada halaman yang sama dan juga guna mendapatkan efisiensi waktu proses memuat formulir.



**Gambar 4.82** Antar Muka Tambah Item Purchase

Pada Gambar 4.75 tertera tampilan formulir tambah data Item Purchase yang dibuat dalam bentuk pop-up guna memberi kemudahan bagi user untuk tetap berada pada halaman yang sama dan juga guna

mendapatkan efisiensi waktu proses memuat formulir. Formulir terdapat kolom input Product bertipe pilihan yang direlasikan dengan tabel Product sehingga semua data Product dapat langsung ditampilkan sebagai pilihan nilai input pada formulir. Jika data arrive\_at pada purchase sudah diisi maka pengguna dapat memasukkan data kedalam stock yang dipilih. Di bagian Purchase ini nilai total dihitung secara otomatis oleh sistem.

## 4.6 Implementasi Menu CRP

Pada menu utama Capacity Requirement Planning yang terdiri dari fitur Workcenter Management guna mengelola data kapasitas produksi berdasarkan kapasitas dan jumlah unit kerja produksi dan fitur Routing Management guna mengelola data tahapan proses pengolahan bahan baku dalam proses produksi.

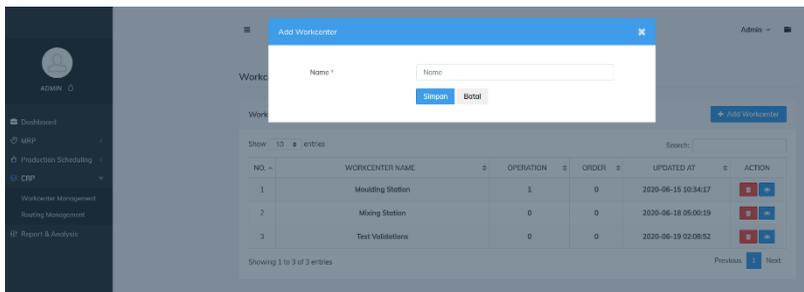
Implementasi diawali dengan membuat fitur work center, pada menu ini pengguna dari sistem dapat memasukkan data tempat kerja/mesin beserta kapasitasnya.

NO.	WORKCENTER NAME	OPERATION	ORDER	UPDATED AT	ACTION
1	Moulding Station	1	0	2020-06-15 10:34:17	<span style="color: red;">✖</span> <span style="color: blue;">✎</span>
2	Mixing Station	0	0	2020-06-19 05:00:19	<span style="color: red;">✖</span> <span style="color: blue;">✎</span>
3	Test Validations	0	0	2020-06-19 02:08:52	<span style="color: red;">✖</span> <span style="color: blue;">✎</span>

**Gambar 4.83** Antar Muka Workcenter

Pada Gambar 4.76 tertera halaman data tabel daftar Work Center yang menampilkan semua data Product yang sudah dibuat seta fungsi hapus dan edit pada setiap baris data. Pada data tabel juga dibuat fungsi pencarian dan navigasi halaman untuk mempercepat waktu pencarian

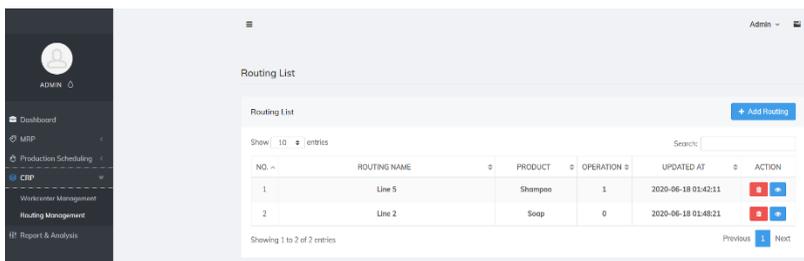
data oleh user. Pada bagian kanan atas tabel disematkan tombol yang difungsikan untuk menambah data Work Center. Formulir tambah data Stock dapat dilihat pada Gambar 4.77



**Gambar 4.84** Antar Muka Tambah Workcenter

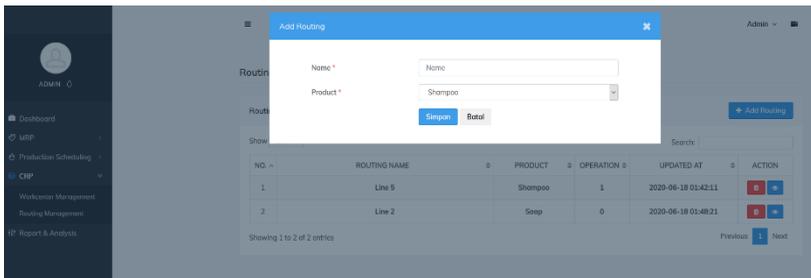
Pada Gambar 4.77 tertera tampilan formulir tambah data Work Center yang dibuat dalam bentuk pop-up guna memberi kemudahan bagi user untuk tetap berada pada halaman yang sama dan juga guna mendapatkan efisiensi waktu proses memuat formulir. Di dalam formulir terdapat kolom input name saja. Dan pengguna dapat memasukkan data kapasitas di halaman Show Workcenter.

Selanjutnya dibuat fitur Routing Management guna mengelola data rute. Yang dimaksud rute adalah alur produksi untuk memproduksi suatu barang, hal ini terdiri dari operasi apa yang dilakukan dan pada mesin/tempat kerja apa itu dilakukan.



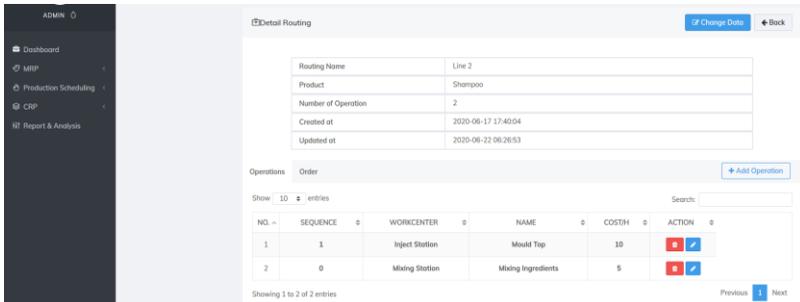
**Gambar 4.85** Antar Muka Routing

Pada Gambar 4.78 tertera halaman data tabel daftar Routing yang menampilkan semua data Product yang sudah dibuat seta fungsi hapus dan edit pada setiap baris data. Pada data tabel juga dibuat fungsi pencarian dan navigasi halaman untuk mempercepat waktu pencarian data oleh user. Pada bagian kanan atas tabel disematkan tombol yang difungsikan untuk menambah data Routing. Formulir tambah data Routing dapat dilihat pada Gambar 4.79



**Gambar 4.86** Antar Muka Tambah Routing

Pada Gambar 4.79 tertera tampilan formulir tambah Routing yang dibuat dalam bentuk pop-up guna memberi kemudahan bagi user untuk tetap berada pada halaman yang sama dan juga guna mendapatkan efisiensi waktu proses memuat formulir. Di dalam formulir terdapat kolom input name dan product saja, selanjutnya pengguna dapat memasukkan data Operasi apa yang dilakukan di dalam rute ini seperti pada gambar 4.80

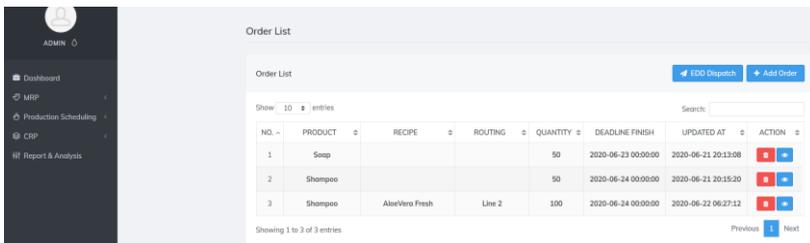


**Gambar 4.87** Antar Muka Show Routing

## 4.7 Implementasi Menu Production Scheduling

Pada menu utama Production Schedule yang terdiri dari fitur Manufacture Order guna mengelola data pesanan yang masuk dari pelanggan dan fitur Work Order guna mengelola data daftar tahapan proses produksi.

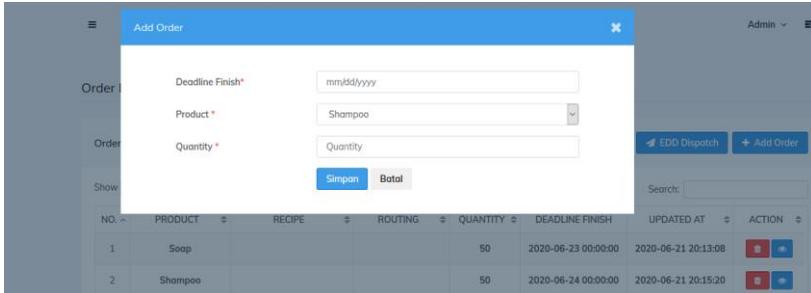
Implementasi diawali dengan membuat fitur manufacture order, fitur ini berguna untuk mencatat pesanan yang masuk kedalam lingkungan produksi.



**Gambar 4.88** Antar Muka Order

Pada Gambar 4.81 tertera halaman data tabel daftar Manufacture Order yang menampilkan semua data Order yang sudah dibuat seta fungsi hapus dan edit pada setiap baris data. Pada data tabel juga dibuat fungsi pencarian dan navigasi halaman untuk mempercepat

waktu pencarian data oleh user. Pada bagian kanan atas tabel disematkan tombol yang difungsikan untuk menambah data Manufacture Order. Selain itu terdapat juga terdapat tombol EDD Dispatch guna memerintahkan sistem untuk melaksanakan penjadwalan EDD secara manual.



**Gambar 4.89** Antar Muka Tambah Order

Pada Gambar 4.82 tertera tampilan formulir tambah data Manufacture Order yang dibuat dalam bentuk pop-up guna memberi kemudahan bagi user untuk tetap berada pada halaman yang sama dan juga guna mendapatkan efisiensi waktu proses memuat formulir. Di dalam formulir terdapat kolom input produk, deadline finish dan quantity, selanjutnya pengguna diarahkan ke halaman show order untuk melihat data dari order.

Detail Order Change Data Back

Order ID	ORD3
Product	Shampoo
Amount	100
Recipe	AloeVera Fresh
Routing	Line 2
Deadline Finish	2020-06-24 00:00:00
Created at	2020-06-21 19:56:58
Updated at	2020-06-22 06:27:12

Work Orders Ingredients Reserve Missing Ingredients Dispatch Order Reserve Stock

Show 8 entries Search:

NO. ^	OPERATION SEQ ^	OPERATION NAME ^	WORKCENTER ^	START DATE ^	FINISH DATE ^	ACTION ^
1	0	Mixing Ingredients	Mixing Station	2020-06-22 08:44:00	2020-06-22 10:44:00	<span style="color: red;">✖</span> <span style="color: blue;">↗</span>
2	1	Mould Top	Inject Station	2020-06-22 10:44:00	2020-06-22 11:45:00	<span style="color: red;">✖</span> <span style="color: blue;">↗</span>

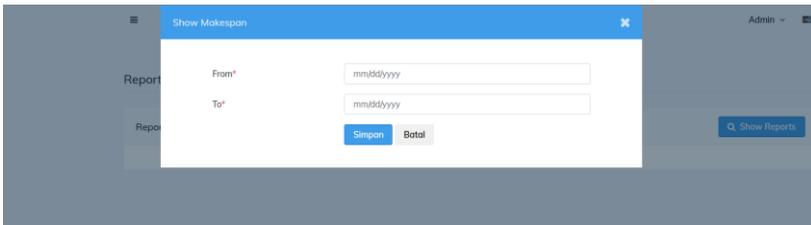
Showing 1 to 2 of 2 entries Previous 1 Next

**Gambar 4.90** Antar Muka Show Order

Pada Gambar 4.83 tertera tampilan detil dari order. Pada halaman ini pengguna dapat melihat dan mengubah detil dari order. Selain itu juga terdapat tombol Reserve Stock, jika pengguna menekan tombol ini maka sistem secara otomatis mencari stock yang tersedia untuk di gunakan dalam order ini. Selain itu pengguna dapat melihat jumlah bahan baku yang diperlukan di dalam tab Missing Ingredients, sistem secara otomatis menghasilkan jumlah bahan baku yang diperlukan. Dan terdapat juga tombol Dispatch Order, tombol ini berguna untuk mendispatch order yang dipilih. Jika tombol Dispatch Order ditekan maka sistem akan otomatis menghasilkan perintah kerja dari order yang dipilih. Terakhir jika order selesai sistem akan secara otomatis menjadwalkan Order selanjutnya sesuai metode yang dimplemntasikan.

## 4.8 Implementasi Menu Report

Pada Menu Report, yang diimplementasikan adalah fungsi Show Makespan. Fitur ini diperlukan untuk pengujian metode pada tugas akhir ini. Pada Gambar 4.84 pengguna cukup memasukkan rentang tanggal yang di pilih. Setelah itu sistem akan menampilkan makespan pada rentang tanggal tersebut.



The image shows a web application interface with a modal window titled "Show Makespan". The modal has a white background and a blue header. It contains two input fields: "From\*" and "To\*", both with the placeholder text "mm/dd/yyyy". Below the input fields are two buttons: "Simpan" (Save) and "Batal" (Cancel). To the right of the modal, there is a "Show Reports" button with a magnifying glass icon. The background of the application is dark grey with a sidebar on the left containing "Report" and "Repor" menu items, and an "Admin" user profile in the top right corner.

**Gambar 4.91** Menu Show Makespan

(halaman ini sengaja dikosongkan)

## BAB V UJICOBA DAN EVALUASI

Pada bab ini akan dilakukan tahap ujicoba dan evaluasi sesuai dengan rancangan dan implementasi. Dari hasil yang didapatkan setelah melakukan uji coba, akan dilakukan evaluasi sehingga dapat ditarik kesimpulan pada bab selanjutnya.

### 5.1 Lingkungan Uji Coba

Uji coba dilakukan pada perangkat dengan spesifikasi seperti yang tertera pada Tabel 5.1.

*Tabel 5.1 Spesifikasi Perangkat yang Digunakan*

<b>Komponen</b>	<b>Spesifikasi</b>
<b>CPU</b>	Ryzen 5 3600 X CPU @ 3.79 GHz
<b>Sistem Operasi</b>	Windows 10 19041.329
<b>Memori</b>	2x8 GB 3000Mhz
<b>Storage</b>	Samsung SATA SSD 850 Pro (Laravel) XPG Spextrix pci3.0x4 NVME S40G (PgSql)

Adapun versi perangkat lunak yang digunakan dalam Tugas Akhir ini adalah sebagai berikut:

- Mozilla Firefox 77.0.1 (64-bit) untuk berinteraksi dengan website.
- Navicat Premium 15.0.13 (64-bit) untuk berinteraksi dengan server basis data.
- JetBrains - PHPSTORM 2020.1 sebagai IDE untuk mengembangkan aplikasi.
- barryvdh/laravel-debugbar 3.3.3 untuk melakukan uji performa dan men-debug aplikasi.
- barryvdh/laravel-ide-helper 2.7.0 untuk membantu IDE dalam memahami kode pada kerangka kerja Laravel.

Uji coba metode menggunakan data manufaktur yang diperoleh dari PT. P. Untuk uji coba fungsi sistem dilakukan secara manual, dan di cek satu persatu apakah fungsi sistem sudah bekerja sebagaimana mestinya, untuk uji coba performa menggunakan data dummy sebagai beban sintetik dan diukur dengan library laravel-debugbar.

## 5.2 Hasil Uji Coba

Hasil uji coba dari implementasi modul perancangan dan kontrol produksi, dan uji coba metode EDD dapat dilihat sebagai berikut:

### 5.2.1 Hasil Uji Coba Metode Penjadwalan EDD

Pengujian metode EDD digunakan untuk melihat perbandingan makespan dari metode yang digunakan pada PT.P dengan metode EDD yang diimplementasikan pada modul Perencanaan dan Kontrol Produksi.

Pengambilan data uji pada metode EDD dilakukan dengan survei langsung ke PT.P. Data uji kemudian diolah agar kerahasiaan PT.P terjaga. Data tersebut dapat dilihat Tabel 5.2.

*Tabel 5.2 Data Uji PT.P*

<b>Job</b>	<b>W 1</b>	<b>W 2</b>	<b>W 3</b>	<b>Due</b>
<b>1</b>	7	7	3	60
<b>2</b>		6	1	9
<b>3</b>			2	10
<b>4</b>			1	10
<b>5</b>	5	5	2	76
<b>6</b>		6	2	28
<b>7</b>			3	12
<b>8</b>	5	5	1	86
<b>9</b>			2	12
<b>10</b>	8	6	2	81

<b>11</b>			3	34
<b>12</b>		6	2	52
<b>13</b>	6	8	2	65
<b>14</b>			3	46
<b>15</b>			3	51
<b>16</b>	8	7	2	71
<b>17</b>		7	3	21
<b>18</b>	7	7	3	62
<b>19</b>		7	3	23
<b>20</b>		7	3	42

Data pada Tabel 5.2 dapat di deskripsikan sebagai  $n \times m$  dimana terdapat set dari  $n$  *Jobs*  $\{J_i\} 1 \leq n$  yang akan diproses pada set  $m$  Mesin  $\{M_r\} 1 \leq r \leq m$ . Pada tiap *Jobs* memiliki urutan produksi yang akan di proses pada mesin. Proses pengerjaan  $J_i$  pada mesin  $M_r$  disebut operasi  $O_{jr}$ . Tiap operasi  $O_{jr}$  yang diproses pada mesin  $M_r$  memakan waktu proses yang eksklusif yang disebut  $P_{jr}$ . Penjadwalan adalah set dari waktu selesai untuk tiap operasi  $\{C_{jr}\} 1 \leq j \leq n, 1 \leq r \leq m$  yang tidak melanggar batasan diatas. Waktu total untuk menyelesaikan semua pekerjaan disebut *makespan*  $L$  atau  $C_{max}$ . Tujuan dari metode EDD untuk meminimalkan *makespan*  $L$ . Urutan penjadwalan pada PT.Purut sesuai nomor *job*.

**Tabel 5.3 Hasil Pengujian Metode EDD**

Metode	$C_{max}$	Perbedaan
Orisinal	88	0
EDD	85	3

Pada tabel 5.3 ditemukan metode EDD 3,41% lebih baik daripada metode orisinal yang digunakan pada PT.P. untuk urutan penjadwalan pada metode EDD dapat dilihat pada tabel 5.4.

**Tabel 5.4** *Urutan Pekerjaan Metode EDD*

Job	W 1	W 2	W 3	Due
2		6	1	9
3			2	10
4			1	10
7			3	12
9			2	12
17		7	3	21
19		7	3	23
6		6	2	28
11			3	34
20		7	3	42
14			3	46
15			3	51
12		6	2	52
1	7	7	3	60
18	7	7	3	62
13	6	8	2	65
16	8	7	2	71
5	5	5	2	76
10	8	6	2	81
8	5	5	1	86

Tabel 5.4 merupakan hasil penjadwalan pekerjaan yang dihasilkan oleh modul Perencanaan dan Kontrol Produksi. Penjadwalan yang dilakukan oleh sistem menggunakan metode EDD.

### 5.2.2 Hasil Uji Coba Fungsi Program

**Tabel 5.5** *Tabel Hasil Uji Fungsi Menu MRP*

Fungsi	Fitur	Berhasil
--------	-------	----------

UoM	Pengguna dapat menambahkan UoM	Ya
	Pengguna dapat menghapus UoM	Ya
	Pengguna dapat mengubah UoM	Ya
	Pengguna dapat melihat semua UoM	Ya
Product	Pengguna dapat menambahkan Product	Ya
	Pengguna dapat menghapus Product	Ya
	Pengguna dapat mengubah Product	Ya
	Pengguna dapat melihat detail Product	Ya
	Pengguna dapat melihat semua Product	Ya
Bom/Recipe	Pengguna dapat menambahkan Bom/Recipe	Ya
	Pengguna dapat menghapus Bom/Recipe	Ya
	Pengguna dapat mengubah Bom/Recipe	Ya
	Pengguna dapat melihat detail Bom/Recipe	Ya
	Pengguna dapat melihat semua Bom/Recipe	Ya
	Pengguna dapat menambahkan Ingredien dari Bom/Recipe	Ya
	Pengguna dapat mengubah Ingredien dari Bom/Recipe	Ya
	Pengguna dapat menghapus Ingredien dari Bom/Recipe	Ya
Stock	Pengguna dapat menambahkan Stock	Ya
	Pengguna dapat menghapus Stock	Ya
	Pengguna dapat mengubah Stock	Ya
	Pengguna dapat melihat detail Stock	Ya
	Pengguna dapat melihat semua Stock	Ya
	Sistem dapat menghitung jumlah dari Stock	Ya

Purchase	Pengguna dapat menambahkan Purchase	Ya
	Pengguna dapat menghapus Purchase	Ya
	Pengguna dapat mengubah Purchase	Ya
	Pengguna dapat melihat detil Purchase	Ya
	Pengguna dapat melihat semua Purchase	Ya
	Pengguna dapat menambahkan Item Purchase	Ya
	Pengguna dapat mengubah Item Purchase	Ya
	Pengguna dapat menghapus Item Purchase	Ya
	Sistem dapat menghitung total dari Purchase	Ya

Pada Tabel 5.5 berisi fungsi serta nama fitur yang ada di dalam menu MRP pada Modul Perencanaan dan Kontrol Produksi. Dalam uji ini menu MRP memperoleh hasil 32/32 fitur terpenuhi atau 100%.

*Tabel 5.6 Hasil Uji Fungsi Menu CRP*

<b>Fungsi</b>	<b>Fitur</b>	<b>Berhasil</b>
Workcenter	Pengguna dapat menambahkan Workcenter	Ya
	Pengguna dapat menghapus Workcenter	Ya
	Pengguna dapat mengubah Workcenter	Ya
	Pengguna dapat melihat detil Workcenter	Ya
	Pengguna dapat melihat semua Workcenter	Ya
	Pengguna dapat menambahkan Kapasitas dari Workcenter	Ya

	Pengguna dapat mengubah Kapasitas Workcenter	Ya
	Pengguna dapat menghapus Kapasitas Workcenter	Ya
Routing	Pengguna dapat menambahkan Routing	Ya
	Pengguna dapat menghapus Routing	Ya
	Pengguna dapat mengubah Routing	Ya
	Pengguna dapat melihat detail Routing	Ya
	Pengguna dapat melihat semua Routing	Ya
	Pengguna dapat menambahkan Operation dari Routing	Ya
	Pengguna dapat mengubah Operation Routing	Ya
	Pengguna dapat menghapus Operation Routing	Ya

Pada Tabel 5.6 berisi fungsi serta nama fitur yang ada di dalam menu CRP pada Modul Perencanaan dan Kontrol Produksi. Dalam uji ini menu CRP memperoleh hasil 16/16 fitur terpenuhi atau 100%.

*Tabel 5.7 Hasil Uji Fungsi Menu Production Scheduling*

<b>Fungsi</b>	<b>Fitur</b>	<b>Berhasil</b>
Order	Pengguna dapat membuat Order	Ya
	Pengguna dapat mengubah Order	Ya
	Pengguna dapat menghapus Order	Ya
	Pengguna dapat menampilkan detail Order	Ya
	Pengguna dapat melakukan operasi cek bahan baku secara otomatis	Ya
	Pengguna dapat melakukan operasi Dispatch untuk order secara manual	Ya

	Pengguna dapat melakukan operasi Dispatch untuk order dengan menggunakan metode EDD	Ya
	Pengguna dapat mengubah Work Order	Ya
	Pengguna dapat menghapus Work Order	Ya

Pada Tabel 5.7 berisi fungsi serta nama fitur yang ada di dalam menu PS pada Modul Perencanaan dan Kontrol Produksi. Dalam uji ini menu PS memperoleh hasil 9/9 fitur terpenuhi atau 100%.

*Tabel 5.8 Hasil Uji Fungsi Menu Report*

<b>Fungsi</b>	<b>Fitur</b>	<b>Berhasil</b>
Report	Pengguna dapat menampilkan Report yang berisi Makespan dari Order yang berdasarkan jendela waktu tertentu	Ya

Pada Tabel 5.8 berisi fungsi serta nama fitur yang ada di dalam menu Report pada Modul Perencanaan dan Kontrol Produksi. Dalam uji ini menu Report memperoleh hasil 1/1 fitur terpenuhi atau 100%. Rata rata keberhasilan dari 4 menu yang diuji sebesar 100%.

### 5.2.3 Hasil Uji Coba Performa Sistem

*Tabel 5.9 Tabel Hasil Uji Coba Performa Menu MRP*

Fungsi	Fitur	Rata-Rata Waktu
UoM	Pengguna dapat menambahkan UoM	20 ms
	Pengguna dapat menghapus UoM	20 ms
	Pengguna dapat mengubah UoM	23 ms
	Pengguna dapat melihat semua UoM	27 ms
Product	Pengguna dapat menambahkan Product	25 ms
	Pengguna dapat menghapus Product	21 ms
	Pengguna dapat mengubah Product	23 ms
	Pengguna dapat melihat detail Product	27 ms
	Pengguna dapat melihat semua Product	26 ms
Bom/Recipe	Pengguna dapat menambahkan Bom/Recipe	24 ms
	Pengguna dapat menghapus Bom/Recipe	25 ms
	Pengguna dapat mengubah Bom/Recipe	26 ms
	Pengguna dapat melihat detail Bom/Recipe	50 ms
	Pengguna dapat melihat semua Bom/Recipe	32 ms
	Pengguna dapat menambahkan Ingredients dari Bom/Recipe	19 ms
	Pengguna dapat mengubah Ingredients dari Bom/Recipe	20 ms
	Pengguna dapat menghapus Ingredients dari Bom/Recipe	17 ms
Stock	Pengguna dapat menambahkan Stock	16 ms
	Pengguna dapat menghapus Stock	19 ms
	Pengguna dapat mengubah Stock	20 ms
	Pengguna dapat melihat detail Stock	27 ms

	Pengguna dapat melihat semua Stock	31 ms
	Sistem dapat menghitung jumlah dari Stock	6.5 ms
Purchase	Pengguna dapat menambahkan Purchase	21 ms
	Pengguna dapat menghapus Purchase	27 ms
	Pengguna dapat mengubah Purchase	28 ms
	Pengguna dapat melihat detil Purchase	31 ms
	Pengguna dapat melihat semua Purchase	24 ms
	Pengguna dapat menambahkan Item Purchase	21 ms
	Pengguna dapat mengubah Item Purchase	20 ms
	Pengguna dapat menghapus Item Purchase	19 ms
	Sistem dapat menghitung total dari Purchase	4.3 ms

Pada Tabel 5.9 berisi hasil pengujian performa terhadap fitur-fitur pada Menu MRP. Hasil yang di dapat dari 32 fitur yang di uji memiliki rata-rata waktu respon 23.12 ms dengan waktu tertinggi 50 ms dan waktu paling rendah 4.3 ms.

**Tabel 5.10** Tabel Hasil Uji Coba Performa Menu CRP

Fungsi	Fitur	Rata-Rata Waktu
IWorkcenter	Pengguna dapat menambahkan Workcenter	17 ms
	Pengguna dapat menghapus Workcenter	18 ms
	Pengguna dapat mengubah Workcenter	19 ms
	Pengguna dapat melihat detil Workcenter	26 ms

	Pengguna dapat melihat semua Workcenter	21 ms
	Pengguna dapat menambahkan Kapasitas dari Workcenter	17 ms
	Pengguna dapat mengubah Kapasitas Workcenter	19 ms
	Pengguna dapat menghapus Kapasitas Workcenter	17 ms
Routing	Pengguna dapat menambahkan Routing	12 ms
	Pengguna dapat menghapus Routing	14 ms
	Pengguna dapat mengubah Routing	17 ms
	Pengguna dapat melihat detail Routing	26 ms
	Pengguna dapat melihat semua Routing	21 ms
	Pengguna dapat menambahkan Operation dari Routing	18 ms
	Pengguna dapat mengubah Operation Routing	19 ms
	Pengguna dapat menghapus Operation Routing	22 ms

Pada Tabel 5.10 berisi hasil pengujian performa terhadap fitur-fitur pada Menu CRP. Hasil yang di dapat dari 16 fitur yang di uji memiliki rata-rata waktu respon 18,93 ms dengan waktu tertinggi 26 ms dan waktu paling rendah 12 ms.

**Tabel 5.11** Tabel Hasil Uji Coba Performa Menu PS

Fungsi	Fitur	Rata-Rata Waktu
Order	Pengguna dapat membuat Order	17 ms
	Pengguna dapat mengubah Order	18 ms
	Pengguna dapat menghapus Order	21 ms

	Pengguna dapat menampilkan detail Order	27 ms
	Pengguna dapat melakukan operasi cek bahan baku secara otomatis	19 ms
	Pengguna dapat melakukan operasi Dispatch untuk order secara manual	22 ms
	Pengguna dapat melakukan operasi Dispatch untuk order dengan menggunakan metode EDD	23 ms
	Pengguna dapat mengubah Work Order	16 ms
	Pengguna dapat menghapus Work Order	18 ms

Pada Tabel 5.11 berisi hasil pengujian performa terhadap fitur-fitur pada Menu CRP. Hasil yang di dapat dari 9 fitur yang di uji memiliki rata-rata waktu respon 20.11 ms dengan waktu tertinggi 27 ms dan waktu paling rendah 16 ms.

***Tabel 5.12** Tabel Hasil Uji Coba Performa Menu Report*

Fungsi	Fitur	Rata-Rata Waktu
Report	Pengguna dapat menampilkan Report yang berisi Makespan dari Order yang berdasarkan jendela waktu tertentu	22 ms

Pada Tabel 5.12 berisi hasil pengujian performa terhadap fitur report sebesar 22 ms. Rata rata performa sistem keseluruhan sebesar 21.04 ms.

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Pada Bab ini akan diberikan kesimpulan yang diperoleh dari Tugas Akhir yang telah dikerjakan dan saran tentang pengembangan dari Tugas Akhir ini yang dapat dilakukan di masa yang akan datang.

#### **6.1 Kesimpulan**

Kesimpulan yang diperoleh pada uji coba dan evaluasi Tugas Akhir ini adalah sebagai berikut:

1. Dampak metode EDD terhadap makespan sebesar 3.4% lebih pendek dari metode yang digunakan pada pabrik PT.P.
2. Implementasi Modul Perencanaan dan Kontrol Produksi terdiri dari *Material Requiriement Planning*, *Production Scheduling*, *Capacity Requirement Planning* dapat digunakan pada kondisi nyata pada pabrik PT.P.

#### **6.2 Saran**

Saran yang dapat diberikan dari hasil uji coba dan evaluasi adalah sebagai berikut:

1. Menambahkan aspek lain pengujian untuk menguji metode penjadwalan seperti *max. Tardiness*, *Total Flow Time*, dan *Late Job*
2. Melakukan lebih banyak pengujian sebagai pembanding metode yang diuji.

*(Halaman ini sengaja dikosongkan)*

## DAFTAR PUSTAKA

- [1] R. Sarno, B. Meidyani dan A. L. Nurlaili, "Time and cost optimization using scheduling job shop and linear goal programming model," dalam *2018 International Conference on Information and Communications Technology (ICOIACT)*, Yogyakarta, 2018.
- [2] M. L. Pinedo, *Scheduling, Theory, Algorithms, and Systems*, 5 penyunt., New York City, New York: Springer International Publishing, 2016.
- [3] S. M. Johnson, "Optimal Two Stage and Three Stage Production Schedules with Setup Time included," *Naval Research Logistics Quarterly*, vol. 1, no. 1, pp. 61-68, 1954.
- [4] T. Bijen, "Developing a scheduling model for a cable manufacturer," University of Twente, Deurningen, 2015.
- [5] T. C. Entringer dan A. d. S. Ferreira, "Proposal of a Reference Model in BPMN Notation for an MRP System," *International Journal of Advanced Engineering Research and Science*, vol. 5, no. 11, pp. 39-51, 2018.
- [6] J. E. Hernández dan J. Mula, "A reference model for conceptual modelling of production planning processes," *Production, Planning & Control*, vol. 19, no. 8, pp. 725-734, 2008.
- [7] E. Monk, *Concepts in Enterprise Resource Planning*, 4 penyunt., Boston, Massachusetts: Course Technology, Cengage Learning, 2012.
- [8] P. Dennis, *Lean Production Simplified*, 3 penyunt., Boca Raton, Florida: CRC Press, 2015.

- [9] B. Kerber dan B. J. Dreccckshage, *Lean Supply Chain Management Essentials A Framework for Material Managers*, 1st penyunt., Boca Raton, Florida: CRC Press, 2011, p. 184.
- [10] W. Clark, *The Gantt Chart: A Working Tool of Management*, New York: The Ronald Press Company, 1992.
- [11] T. Otwell, "Documentation - Laravel - The PHP Framework For Web Artisans," Laravel LLC, 26 Desember 2019. [Online]. Available: <https://laravel.com/docs/7.x>. [Diakses 27 Desember 2019].
- [12] T. Otwell, "laravel/README.md at master · laravel/laravel," GitHub, Inc., 12 November 2019. [Online]. Available: <https://github.com/laravel/laravel/blob/master/README.md>. [Diakses 27 Desember 2019].
- [13] G. Booch, J. Rumbaugh dan I. Jacobson, *Unified Modeling Language User Guide, The, 2nd Edition*, Boston: Addison-Wesley, 2005.

# LAMPIRAN

## A.1 Kode Fungsi untuk Konstrait Penjadwalan

```
// Hanya 1 pekerjaan yang berjalan pada 1 Workcenter
if ($order->workOrders()->whereRaw('sql:finish_date is null and start_date is not null and id != ' . $workOrder->id->exists())) {
    return response()->json(['error' => ['Workcenter is Busy']]);
}
```

## A.2 Kode Fungsi untuk Konstrait Penjadwalan 2

```
$op = $workOrder->operation;
$prevop = $order->routing->operations()->where('seq', $op->seq - 1)->first('id');

// Pekerjaan hanya bisa dimulai jika pekerjaan sebelumnya sudah selesai
if (isset($prevop->id) && $order->workOrders()->whereRaw('sql:finish_date is null and operation_id = ' . $prevop->id->exists())) {
    return response()->json(['error' => ['Previous operation not finished yet']]);
}
```

## A.2 Kode Fungsi untuk Menjadwalkan EDD

```
/**
 * Handle Earliest Due Date Dispatch Rule
 *
 * @param object $event
 * @return array
 */
public function handle($event)
{
    $dispatched = collect([]);
    $orders = Order::doesntHave('workOrders')
        ->whereNotNull('routing_id')
        ->orderBy('deadline_finish', 'asc') //Earliest Due Date Dispatch Rule
        ->get()
        ->append('missing_ingredients');
    $orders->each(function ($order, $key) use ($dispatched) {
        if ($order->missing_ingredients->count() == 0) {
            $wk = $order->routing->operations()->orderBy('seq', 'asc')->first()->workcenter;

            if ($wk->workOrders()->where(['finish_date', '=', null], ['start_date', '!=', null])->doesntExist()) {
                $dispatched->add($order);
                foreach ($order->routing->operations as $op) {
                    $wo = new WorkOrder;
                    $wo->order()->associate($order);
                    $wo->operation()->associate($op);
                    $wo->save();
                }
            }
        }
    });
}
```

*(Halaman ini sengaja dikosongkan)*

## BIODATA PENULIS



**ALIFA IZZAN AKHSANI**, lahir di Surabaya, 23 July 1998. Penulis adalah anak pertama dari 7 bersaudara. Penulis pernah menempuh pendidikan sekolah dasar di SD Angkasa 1 Jakarta dan SD Muhammadiyah 15 Surabaya lalu melanjutkan pendidikan sekolah menengah pertama di SMP Negeri 1 Surabaya. dan penulis menempuh pendidikan menengah atas di SMA Negeri 5 Surabaya. Selanjutnya penulis melanjutkan pendidikan sarjana di

Departemen Informatika, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember Surabaya. Selama kuliah, penulis aktif dalam berbagai organisasi baik tingkat jurusan maupun universitas.

Dalam menyelesaikan pendidikan S1, penulis mengambil bidang minat Manajemen Informasi (MI). Sebagai mahasiswa, penulis berperan aktif dalam beberapa organisasi kampus seperti kepala biro Kajian Strategis Himpunan Mahasiswa Teknik-Computer (HMTC) ITS, staf Media Informasi Badan Eksekutif Mahasiswa FTIf ITS, staf WEBKES pada acara ITS EXPO 2016, dan *Committee* dalam acara *International Conference on Information & Communication Technology and System (ICTS) 2019*. Selain itu, penulis juga pernah mengisi stand start-up pada Schematics 2018-2019. Penulis pernah melakukan kerja praktik di PT Poros Kolektif Nusantara pada Mei 2020 dan membuat aplikasi web dan mobile untuk monitoring penyebaran penyakit yang menular lewat udara. Penulis dapat dihubungi melalui nomor *handphone*: 083 830 5050 98 atau *email*: <mailto:alfizzanakhsani@gmail.com>.