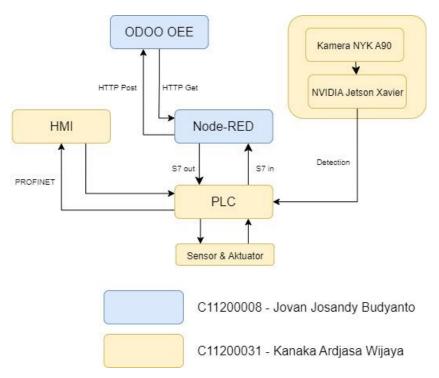
#### 3 PERENCANAAN DAN IMPLEMENTASI SISTEM

Untuk tugas akhir ini akan melakukan kerja sama antara 2 mahasiswa, yaitu Jovan Josandy Budyanto (C11200008) dan Kanaka Ardjasa Wijaya (C11200031). Untuk integrasi sistem produksi dengan OEE dengan memanfaatkan Odoo akan dikerjakan oleh penulis. Sedangkan untuk sistem pendeteksi serbuk gerinda akan dilakukan oleh Kanaka Ardjasa Wijaya (C11200031) dengan menggunakan 1 kamera RGB NYK Nemesis A90 Everest dengan algoritma deep learning YOLO V8.



Gambar 3.1 Blok Diagram Sistem Perhitungan OEE Secara Otomatis

Seperti yang terlihat pada Gambar 3.1, tugas akhir yang dikerjakan oleh penulis memanfaatkan mesin sistem penyortiran otomatis yang terdapat di Laboratorium Sistem Kontrol Universitas Kristen Petra. Sistem penyortiran ini digunakan pada penelitian sebelumnya oleh Eric Harjanto (C11190005), William (C11190011), dan Yoshua Kaleb Purwanto (C11190019). Penulis akan melanjutkan penelitian sebelumnya dengan menambahkan perhitungan OEE dan downtime tracking pada sistem penyortiran yang sudah ada. Tugas akhir akan dibuat untuk memudahkan perusahaan dalam manajemen mesin yang ada sehingga perusahaan bisa meningkatkan produktivitas dan efektivitas produksi.

Produksi tisu pada PT. *Sun Paper Source* melalui beberapa proses mulai dari *jumbo roll* pada mesin *interfold* hingga pengemasan tisu menjadi *bundling*. Mesin *Interfold* memotong dari *jumbo roll* menjadi potongan panjang seperti yang terlihat pada Gambar 3.2, yang kemudian dikirim ke *logsaw cutting machine* untuk dipotong menjadi panjang tisu yang sesuai dengan standar pabrik seperti yang terlihat pada Gambar 3.3. Setelah tisu dipotong menjadi panjang yang sesuai kemudian memasuki mesin *packaging* seperti yang terlihat pada Gambar 3.4 untuk pengemasan potongan tisu, kemudian ke mesin *bundling* untuk menggabungkan 5 kemasan tisu menjadi satu seperti yang terlihat pada Gambar 3.5 untuk dikirim ke pembeli.



Gambar 3.2 Hasil Potongan Panjang Tisu dari Mesin Interfold



Gambar 3.3 Hasil Potongan Tisu Pada Mesin Logsaw Cutting



Gambar 3.4 Mesin Packaging Tisu pada PT. Sun Paper Source



Gambar 3.5 Mesin Bundling Tisu pada PT. Sun Paper Source

Sistem perhitungan OEE otomatis pada tugas akhir ini mengambil salah satu mesin produksi yaitu *logsaw cutting machine* yang sering mengalami kendala dimana seringkali ada serbuk gerinda yang terjatuh pada permukaan lembaran tisu. Pada Gambar 3.6 terlihat mesin

yang sekarang berada pada laboratorium sistem kontrol. Sistem perhitungan OEE otomatis harus dapat terintegrasi dengan PLC yang ada pada mesin untuk bisa mengakses data produksi secara *real-time*. Keseluruhan sistem yang dibuat nanti akan ditampilkan dalam bentuk *visual* yang mudah dimengerti seperti tampilan *gauge meter*, yang akan ditampilkan dengan *dashboard* yang dibuat dalam Odoo. Untuk hal ini, penulis akan merancang *custom modules* yang bisa mencakup semua data dari PLC ke dalam Odoo, melakukan perhitungan OEE, dan melacak *downtime* mesin.

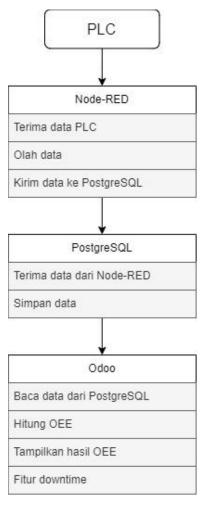


Gambar 3.6 Mesin Penyortir pada Laboratorium Sistem Kontrol

Sistem perhitungan OEE otomatis tidak hanya bisa dipakai pada *logsaw cutting machine*, namun bisa digunakan di mesin-mesin lainnya selama sistem perhitungan bisa berkomunikasi dengan PLC pada mesin produksi. PLC akan mengirim data ke Node-RED untuk dicatat alamatnya untuk data yang diperlukan. Kemudian Node-RED akan mengirimkan data yang diterima PLC menuju *database* PostgreSQL untuk bisa dibaca di Odoo. Odoo sendiri digunakan untuk tampilan dan perhitungan OEE berdasarkan data produksi yang diperoleh termasuk *availability*, *performance*, dan *quality*. Selain itu, terdapat juga fitur *downtime* untuk pencatatan waktu berhenti mesin yang tidak direncanakan agar bisa dianalisa lebih lanjut.

# 3.1 Rancang Bangun Software

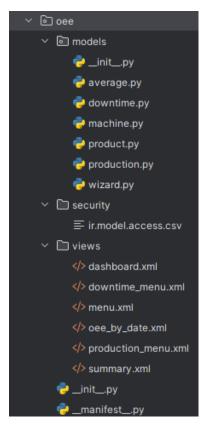
Sistem perhitungan OEE otomatis akan membaca data dari PLC menuju Node-RED yang kemudian akan melakukan pencatatan ke *database* yang memungkinkan Odoo untuk melakukan perhitungan OEE berdasarkan data produksi yang diperoleh. Alur komunikasi sistem secara umum diperlihatkan pada Gambar 3.7 di bawah ini. PLC akan mengirim data produksi ke Node-RED yang mencakup *start\_downtime*, *stop\_downtime*, *good\_product*, *defective\_product*, *start\_downtime*, dan *end\_downtime*, kemudian Node-RED akan menerima data dari PLC serta mengolah dan mencatat data yang diperlukan lalu mengirim data yang diterima ke *database* PostgreSQL yang akan menyimpan data produksi yang diterima dari Node-RED. Odoo kemudian akan membaca data dari database PostgreSQL dan menghitung OEE berdasarkan data yang diperoleh termasuk *downtime* untuk analisa waktu berhenti mesin yang tidak direncanakan. Odoo juga berfungsi untuk menampilkan hasil perhitungan OEE otomatis mesin.



Gambar 3.7 Alur Komunikasi Sistem Secara Umum

## 3.1.1 Program Custom Module Odoo pada PyCharm

Custom module pada Odoo dibuat untuk melakukan akses database PostgreSQL untuk bisa membaca data dari produksi. Modul custom ini dimanfaatkan untuk melakukan perhitungan metrik OEE yang termasuk availability, performance, dan quality berdasarkan data yang diperoleh dari database. Modul juga bertujuan untuk menampilkan hasil perhitungan OEE di antarmuka pengguna yang disesuaikan dengan kebutuhan mesin produksi yang ada. Disediakan juga fitur downtime untuk menganalisa waktu berhenti mesin yang tidak direncanakan. Custom module Odoo berbasis python dan xml, maka untuk memudahkan pembuatan modul digunakan software PyCharm. Pada Gambar 3.8 diperlihatkan folder-folder yang diperlukan untuk pembuatan custom module. Folder models berbasis bahasa python dan berfungsi seperti pembuatan tabel di database. Models juga dimanfaatkan untuk perhitungan dan pencatatan yang disesuaikan dengan keperluan antarmuka pengguna. Folder security digunakan untuk keamanan, dimana bisa dikonfigurasi untuk pengguna mana saja yang bisa mengakses model yang dibuat. Kemudian folder views yang menggunakan bahasa xml digunakan untuk membuat tampilan antarmuka Odoo. File "\_\_manifest\_\_.py" digunakan untuk konfigurasi properti dan informasi modul yang dibuat.



Gambar 3.8 Struktur Modul Odoo

File manifest pada modul Odoo adalah file Python "\_\_manifest\_\_.py" yang mengandung informasi dasar tentang modul, daftar file yang terkait, dependensi, tata letak, tautan menu, dan data awal. Ini memastikan untuk Odoo bisa mengenali dan mengelola modul dengan benar saat diinstal atau diperbarui. Seperti pada Gambar 3.9 yang merupakan isi dari file "\_\_manifest\_\_.py" dimana data yang dibaca adalah folder views atau tampilan yang dibuat dan juga security.

Gambar 3.9 Kode Program file "\_\_manifest\_\_.py"

Model downtime pada Gambar 3.10 berisi isi dari tabel production\_downtime yang meliputi production\_id, machine\_id, start\_datetime, end\_datetime, reason, duration, dan duration\_in\_hours. Start\_datetime dan end\_datetime digunakan untuk mencatat dan melakukan perhitungan durasi downtime secara otomatis pada suatu mesin. Perhitungan durasi downtime secara otomatis seperti terlihat pada Gambar 3.11 dimana end\_datetime akan dikurangi dengan start\_datetime dan menghasilkan durasi total dari downtime.

Gambar 3.10 Kutipan Program Model *Downtime* 

Gambar 3.11 Program Perhitungan pada Model *Downtime* 

Untuk menyimpan data dari mesin produksi dibuat model *machine* seperti pada Gambar 3.12, yang berisi *name, default\_product\_id, product\_ids*. Model ini dibuat untuk menyimpan data nama mesin dan produk yang dihasilkan oleh mesin tersebut sehingga berhubungan dengan model *product* pada Gambar 3.13.

Gambar 3.12 Kutipan Program Model *Machine* 

```
class Product(models.Model):
    _name = 'production.product'
    _description = 'Product Information'

name = fields.Char(string='Product Name', required=True)
    description = fields.Text(string='Product Description')
```

Gambar 3.13 Kutipan Program Model *Product* 

Model *production* merupakan model untuk menyimpan data utama untuk produksi pada *database* dan juga melakukan perhitungan OEE. Pada Gambar 3.14 terlihat isi dari model *production* yang meliputi *planned\_run\_time*, *total\_good\_production*, *total\_defective\_product*, *start\_datetime*, *stop\_datetime*, *actual\_run\_time*, *availability*, *performance*, *quality*, dan oee. *Planned run time* digunakan untuk pengguna bisa mencatat nilai rencana waktu berjalannya suatu mesin dalam satu *shift*. Kemudian *total good production* dan *total defective production* digunakan untuk mencatat hasil produk baik dan cacat yang dihasilkan oleh mesin produksi, yang dimanfaatkan untuk menghitung nilai *performance* dan *quality*. Start datetime dan stop datetime digunakan untuk mencatat waktu berjalannya mesin dan menghitung durasi dari waktu berjalan aktual mesin, yang berguna untuk menghitung *availability* dan *performance* mesin produksi.

```
total good production = fields.Integer(string='Total Good Production',
                     self: fields.Datetime.now())
```

Gambar 3.14 Kutipan Program Model Production

Perhitungan OEE dilakukan juga pada model *production* dengan fungsi "\_compute\_oee" seperti yang terlihat pada Gambar 3.15. Perhitungan akan berdasar dari waktu jalan mesin produksi, rencana waktu jalan mesin, data produk yang baik, dan produk yang cacat, sehingga bisa menghitung nilai *availability*, *performance*, *quality*, dan OEE suatu mesin produksi. Untuk perhitungan *availability* menggunakan rumus pada Persamaan 2.1, *performance* menggunakan rumus pada Persamaan 2.2, *quality* menggunakan rumus pada Persamaan 2.3, serta OEE

menggunakan rumus pada Persamaan 2.4. Untuk perhitungan *cycle time* atau waktu siklus ideal digunakan rumus seperti Persamaan 3.1 berikut ini:

$$cycle\ time = \frac{waktu\ produksi\ aktual}{total\ produk\ yang\ dihasilkan} \tag{3.1}$$

Gambar 3.15 Kutipan Program Perhitungan OEE Pada Model *Production* 

Model *average* seperti yang terlihat pada Gambar 3.16 digunakan untuk tampilan dan digunakan untuk menyimpan nilai rata-rata OEE. Model akan mengambil id mesin dari tabel *production\_machine* dan mencatat nilai *availability, performance, quality,* dan OEE dari tiap mesin yang memiliki id sama. Model kemudian akan menghitung rata-rata data OEE seperti pada Gambar 3.17 dan menyimpannya ke variabel *display* yang ada.

Gambar 3.16 Kutipan Program Model Average

```
def _compute_display_values(self):
    for record in self:
        record.availability_display = self._format_value(record.availability)
        record.performance_display = self._format_value(record.performance)
        record.quality_display = self._format_value(record.quality)
        record.oee_display = self._format_value(record.oee)
```

Gambar 3.17 Program Perhitungan Nilai Rata-rata pada Model Average

Untuk menampilkan nilai total OEE keseluruhan dari semua mesin yang ada, dibuat model *total* seperti pada Gambar 3.18 yang akan menghitung nilai OEE total dari model *average* kemudian disimpan ke variabel *total oee*.

Gambar 3.18 Kutipan Program Model *Total* OEE

Untuk membuat agar memungkinkan pengguna bisa memilih jarak waktu untuk perhitungan, dibuat model wizard seperti yang terlihat pada Gambar 3.19. Model wizard ini digunakan untuk melakukan filter jarak waktu berdasarkan id mesin yang dipilih, dan data yang diperlukan yaitu availability, performance, quality, dan OEE diambil dari model production. Data yang didapatkan kemudian akan dikirim ke model result seperti yang terlihat pada Gambar 3.20, yang digunakan data tampilan.

```
class MachineOEEWizard(models.TransientModel):
                  ('machine_id', '=', self.machine_id.id),
('start_datetime', '>=', self.start_date),
('stop_datetime', '<=', self.end_date)</pre>
                  'performance': total performance,
```

Gambar 3.19 Kutipan Program Model Wizard

```
class MachineOEResult(models.TransientModel):
    _name = 'machine.oee.result'
    _description = 'Machine OEE Result'

    availability = fields.Float(string='Availability')
    performance = fields.Float(string='Performance')
    quality = fields.Float(string='Quality')
    oee = fields.Float(string='Overall Equipment Effectiveness')
```

Gambar 3.20 Kutipan Program Model Result

Folder views pada Odoo berisi file XML yang mendefinisikan tampilan antarmuka pengguna untuk modul tertentu. Ini mencakup tata letak halaman, formulir entri data, daftar record, dan elemen-elemen lain yang diperlukan untuk menampilkan informasi dan berinteraksi dengan pengguna. Seperti terlihat pada Gambar 3.21 yang merupakan tampilan tree view untuk menu detail produksi. Tree view memungkinkan pengguna untuk melihat informasi dalam bentuk list. Saat salah satu item ditekan maka pengguna akan diarahkan ke form view yang dibagi menjadi 2 halaman dengan fitur notebook, seperti pada Gambar 3.22. Pada halaman depan notebook view akan terlihat halaman yang berisi tentang detail informasi mesin secara lengkap. Pada halaman kedua notebook ditampilkan gauge yang memberikan indikator untuk data OEE, yaitu availability, performance, quality, dan OEE.

Gambar 3.21 Program Tree View pada Menu Produksi

```
<field name="performance"/>
<field name="quality"/>
  </div>
```

Gambar 3.22 Program Notebook View pada Menu Produksi

Diperlihatkan pada Gambar 3.23 yang merupakan kutipan program untuk menu downtime. Digunakan jenis tampilan tree view untuk melihat detail dari downtime yaitu nama mesin, waktu mulai downtime, waktu selesai downtime, alasan, dan durasi downtime. Kemudian ada form view seperti pada gambar 3.24 untuk melihat atau membuat record baru untuk downtime mesin.

Gambar 3.23 Program Tree View pada Menu Downtime

Gambar 3.24 Program Form View pada Menu Downtime

Pada menu OEE *by Date* yang diperlihatkan pada Gambar 3.25, digunakan *form view* untuk menampilkan jendela *pop-up* dimana user bisa member *input* pada *field* nama mesin, dan memilih tanggal *start* dan *stop*. Lalu tersedia tombol "*Get OEE Data*" yang akan pindah ke tampilan *form view* untuk menampilkan hasil *filter* yang sudah diterapkan seperti yang terlihat pada Gambar 3.266, yang akan menampilkan data *availability, performance, quality,* dan OEE yang sudah di *filter*.

Gambar 3.25 Program Form View pada Menu OEE by Date

Gambar 3.26 Program Form View pada Menu OEE Result

Pada Gambar 3.27 terlihat cuplikan program untuk membuat tampilan *dashboard*. Tampilan ini memanfaatkan modul bawaan Odoo yaitu *Dashboards*, yang dipanggil dan di *custom* sehingga bisa menampilkan data yang disesuaikan.

Gambar 3.27 Program Menu Dashboard

Untuk membuat tampilan yang menunjukkan hasil total rata-rata data OEE, dibuat menu *Summary* seperti pada gambar 3.28. Menu ini dibuat untuk menampilkan tampilan *kanban* yang terlihat seperti kartu untuk tiap data mesin yang ada. Tampilan ini kemudian diberi isi dengan *widget gauge*, yang memudahkan pengguna untuk membaca informasi penting mesin dengan cepat.

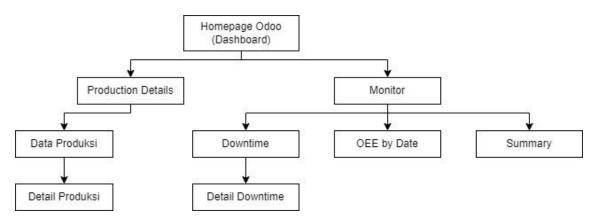
```
kanban>
   <field name="oee_display"/>
<field name="max_rate"/>
                            <field name="availability display" widget="gauge"</pre>
                                    options="{'max field':'max rate'}"/>
                  </div>
```

Gambar 3.28 Program Kanban View pada Menu Summary

### 3.1.2 Desain Tampilan Odoo

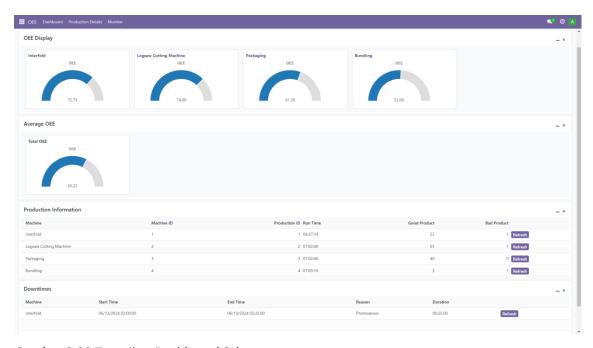
Tampilan Odoo digunakan untuk memudahkan pengguna dalam melakukan pemantauan mesin produksi. Dengan desain tampilan yang baik, pengguna dapat memantau kinerja mesin secara *real-time* dan memungkinkan pengguna untuk menganalisis data secara mendalam. Pada Gambar 3.29 merupakan hirarki dari tampilan Odoo dimana *dashboard* merupakan *homepage* yang pertama kali terlihat saat membuka modul Odoo. Pada *dashboard* ini akan menampilkan *gauge* untuk nilai OEE tiap mesin, nilai OEE total seluruh mesin, *list* data produksi, dan *list downtime* yang terjadi. Kemudian ada *submenu production details* yang berisi tentang data produksi dan detail dari masing-masing produksi. Pada *submenu monitor* dibagi menjadi tiga menu yaitu *downtime* untuk melihat data *downtime* yang terjadi serta detailnya,

OEE *by date* yang digunakan untuk mencari nilai OEE pada rentang waktu tertentu, dan *summary* untuk melihat nilai rata-rata OEE masing-masing mesin.



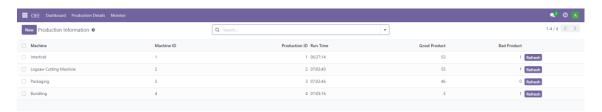
Gambar 3.29 Hirarki *User Interface* pada Odoo

Untuk desain tampilan awal yaitu *dashboard* bisa terlihat pada Gambar 3.30 yang merupakan tampilan awal Odoo saat modul OEE dibuka. *Dashboard* ini akan menunjukkan informasi lengkap tentang mesin yang harus diakses dengan cepat, meliputi OEE Tiap mesin, OEE Total mesin, Detail Produksi, dan *Downtime*.

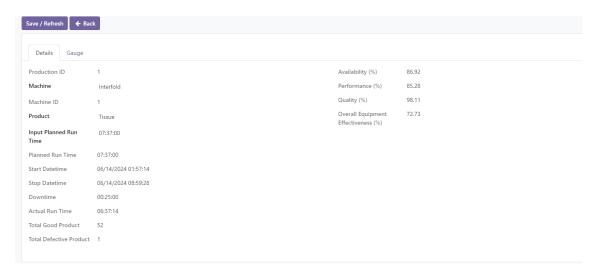


Gambar 3.30 Tampilan Dashboard Odoo

Pada menu *Production Details* akan tertampil informasi produksi yang bisa terlihat secara lengkap seperti pada Gambar 3.31. Pada menu ini terlihat catatan mesin mana yang sedang memproduksi dan terlihat *production id*. Jika salah satu *item* dipilih maka akan tampil tampilan seperti pada Gambar 3.32 yaitu tampilan *notebook* dengan halaman *Details*. Untuk melihat nilai OEE dari mesin yang dipilih dengan tampilan *gauge*, pilih halaman *Gauge* maka tampilan akan berubah seperti yang terlihat pada Gambar 3.33.



Gambar 3.31 Tampilan Utama Menu Production Details



Gambar 3.32 Tampilan Notebook Menu Production Details

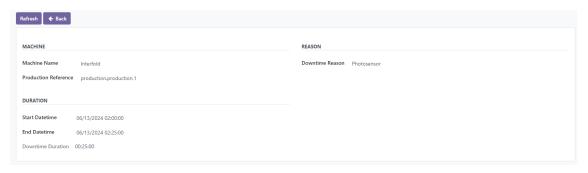


Gambar 3.33 Tampilan Gauge Menu Production Details

Menu *Downtime* memiliki tampilan awal *tree view* seperti pada Gambar 3.34, dimana pengguna bisa melihat sedikit detail dari mesin yang mengalami *downtime* seperti waktu mulai, waktu selesai, alasan, dan durasi dari *downtime*. User juga bisa membuat *record* baru untuk *downtime*. Ketika salah satu *item* ditekan atau membuat *record* baru, maka akan muncul detail dalam tampilan *form* yang bisa diubah oleh pengguna seperti yang terlihat pada Gambar 3.35.

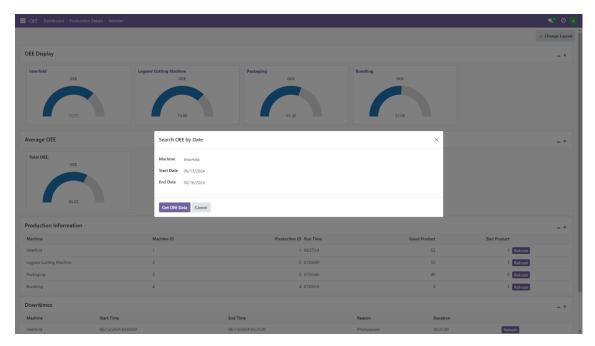


Gambar 3.34 Tampilan Tree View Menu Downtime

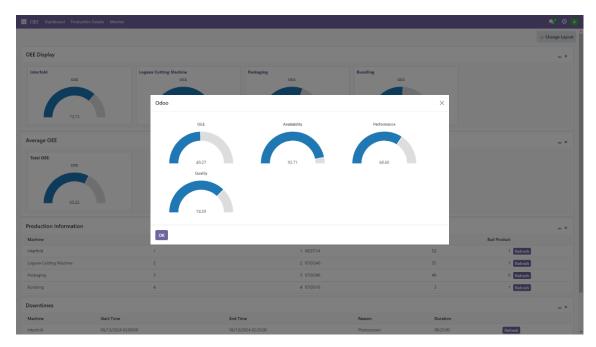


Gambar 3.35 Tampilan Form View Menu Downtime

Untuk menu OEE *by Date*, akan keluar jendela *pop-up* dimana pengguna akan diminta untuk memasukkan nama mesin dan jarak waktu seperti pada Gambar 3.36. Jika ketentuan sudah terisi maka saat pengguna menekan tombol "*Get OEE Data*" akan dibawa ke tampilan *form* yang menunjukkan data OEE dari mesin yang dipilih seperti yang terlihat pada gambar 3.37.

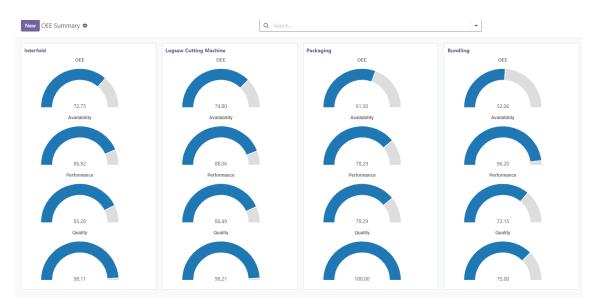


Gambar 3.36 Tampilan Pop-up Window Menu OEE by Date



Gambar 3.37 Tampilan Data OEE Menu OEE by Date

Menu *Summary* digunakan untuk menunjukkan data OEE berdasarkan *record* mesin yang terdaftar. Tampilan akan terlihat seperti pada Gambar 3.38, yang merupakan *kanban views* dengan *gauge* yang disesuaikan dengan data OEE yang ditampilkan.



Gambar 3.38 Tampilan Gauge Menu Summary

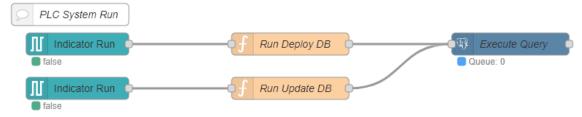
### 3.1.3 Program Node-Red

Node-RED sendiri digunakan sebagai penghubung antar PLC dengan *database* sistem. Node-RED akan menerima data dari PLC untuk diolah datanya dan mengirim data yang sudah di format ke *database* PostgreSQL. Node-RED mengirimkan data berupa *query* SQL ke PostgreSQL untuk menyimpan data di database. Pada Gambar 3.39 terlihat *flow* pada Node-RED yang digunakan untuk meng-*input* id mesin yang akan digunakan untuk pencatatan nilai ke *database* berdasarkan *machine\_id*.



Gambar 3.39 Flow Node-RED untuk Pencatatan ID Mesin

Agar Node-RED bisa mencatat data PLC pada saat mesin menyala digunakan alamat PLC Indicator Run, seperti pada Gambar 3.40. PLC kemudian menyalakan node query ke node database PostgreSQL untuk mencatat saat Indicator Run menyala dan mengirimkan waktu menyala mesin ke tabel production\_production seperti yang terlihat pada Gambar 3.41. Untuk memastikan data waktu nyala mesin ter-update tiap saat, digunakan query untuk memperbarui data selama mesin berjalan seperti yang terlihat pada Gambar 3.42. Nilai dari data stop\_datetime mesin akan terus tercatat sampai mesin berada dalam kondisi system stop.



Gambar 3.40 Flow Node-RED untuk Pencatatan Waktu

Gambar 3.41 Kutipan Program Query Nyala Mesin

Gambar 3.42 Kutipan Program Query Update Nyala Mesin

Untuk pencatatan produk yang baik dan yang cacat, dibuat *flow* seperti pada Gambar 3.43, dimana PLC akan diambil alamatnya untuk melakukan pencatatan *current good product* dan *current reject product. Node query* untuk pencatatan produk baik menggunakan *update* untuk memperbarui tabel *production\_production* kolom *total\_good\_production*. Untuk fungsi *current reject product* juga sama dimana *node* digunakan untuk memperbarui kolom *total defective production*.



Gambar 3.43 Flow Node-RED untuk Pencatatan Jumlah Produk

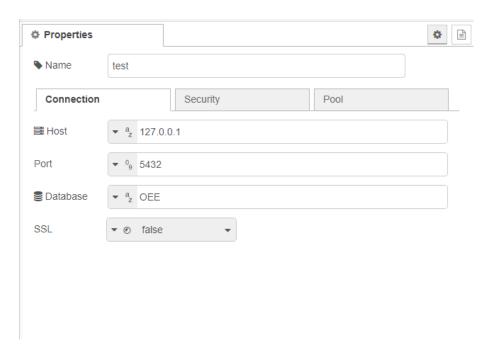
Agar downtime mesin bisa tercatat, digunakan alamat PLC untuk mengambil status saat mesin terjadi berhenti yang tidak direncanakan. Seperti pada Gambar 3.44, alamat PLC untuk no detection akan mengirim query untuk mencatat ke tabel production\_downtime yang kemudian menginput ke database PostgreSQL. Query yang digunakan seperti yang terlihat pada Gambar 3.45, akan mencatat ke kolom start\_datetime pada saat no detection menyala kemudian pada saat no detection kembali mati maka akan update pada kolom end\_datetime.



Gambar 3.44 Flow Node-RED untuk Pencatatan Downtime

Gambar 3.45 Kutipan Program Query Insert Downtime

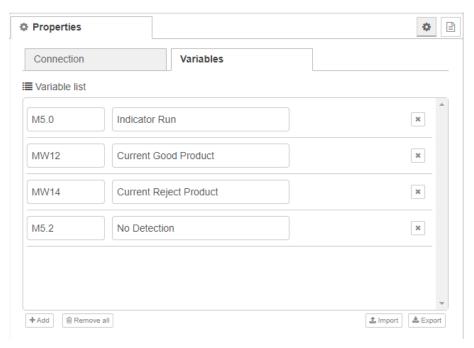
Untuk menyimpan ke *database* digunakan *node* PostgreSQL yang dikonfigurasi seperti pada Gambar 3.46. *Database* yang dituju adalah "OEE" dan IP *Address* dari database lokal yang digunakan adalah 127.0.0.1 dengan *port default* yaitu 5432. Untuk pengaturan PLC, digunakan node *s7 in* dengan konfigurasi seperti yang terlihat pada Gambar 3.47. Untuk *transport* menggunakan *Ethernet* dengan *address* yang sudah dikonfigurasi di PLC yaitu 192.168.0.201 dengan port 102. Alamat-alamat yang digunakan seperti pada Gambar 3.48 dimana digunakan alamat "M5.0" untuk *indicator run*, "MW12" untuk *current good product*, "MW14" untuk *current reject product*, dan "M5.2" untuk *no detection*.



Gambar 3.46 Konfigurasi Node PostgreSQL pada Node-RED



Gambar 3.47 Konfigurasi Node S7 in pada Node-RED



Gambar 3.48 Konfigurasi Variabel pada Node *S7 in*