BAB III

PEMBAHASAN

3.1 Jadwal Kerja Praktek

Jadwal kerja praktek yang dilakukan penulis dilakukan selama 1 bulan, dimulai dari tanggal 12 Oktober 2009 hingga tanggal 31 Desember 2009, dimana penulis melakukan kerja praktek setiap hari, dari pukul 08.30 sampai dengan pukul 17.30.

3.2 Cara/Teknik Kerja Praktek

Kegiatan yang dilakukan penulis selama kerja praktek adalah penelitian dan pengamatan dengan cara melakukan wawancara dengan pembimbing dan karyawan perusahaan, sebagai bahan untuk penulis melakukan perancangan dan pembuatan program sistem monitoring BBM. Adapun kegiatan yang dilakukan penulis, antara lain :

Tabel 3.1. Uraian Kegiatan Kerja Praktek

Kegiatan	Uraian
Masuk kerja Praktek	Menentukan pembimbing kerja praktek di instansi dan menentukan metoda kerja praktek.
Profil Perusahaan	Membahas kedudukan, tugas pokok, fungsi, serta susunan organisasi dan tata kerja perusahaan.

Tabel 3.1. Lanjutan

Mengenal sistem yang ada di perusahaan	Mengetahui proses bisnis dan proses teknis yang dijalankan di perusahaan.
Membangun program aplikasi Sistem Monitoring BBM	Mempelajari sistem arsitektur, sistem proses dari sistem monitoring BBM, membuat perancangan sistem serta implementasi sistem monitoring BBM.

3.3 Membangun Aplikasi Sistem Monitoring BBM

Sistem Monitoring BBM adalah program aplikasi yang dibangun untuk membantu kepentingan perusahaan dalam melakukan monitoring proses yang terjadi dilapangan sehingga meminimalkan waktu operasi serta mengurangi error yang dilakukan operator. Progam aplikasi Sistem Monitoring BBM dibangun dengan menggunakan *tool programming* dengan pendekatan berorientasi objek.

Program aplikasi ini dibangun dengan menggunakan metode Waterfall, dimana proses-proses yang dilakukan meliputi analisis sistem, perancangan sistem, implementasi sistem serta pengujian sistem.

3.3.1. Analisis Sistem

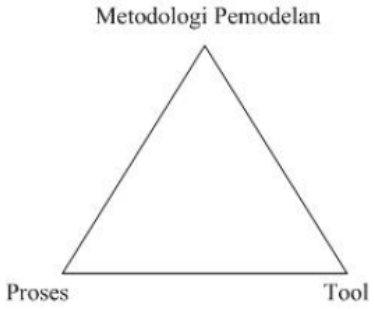
Proses analisis sistem dilakukan dengan cara melakukan observasi dan wawancara yang dilakukan pada jadwal awal kerja praktek, dimana hal ini dilakukan untuk mengetahui proses yang terjadi dilapangan secara langsung serta mengetahui keinginan dari calon pengguna program aplikasi yang akan dibangun. Data tersebut diatas akan sangat membantu dalam memahami permasalahan yang

ada dan sangat membatu dalam menentukan spesifikasi program aplikasi yang diinginkan oleh calon pengguna.

Proses analisis sistem ini dilakukan dengan pendekatan berorientasi objek dimana model yang dibuat menekankan pada apa yang harus dilakukan bukan pada bagaimana melakukannya, sehingga mendapatkan pemahaman sistem seutuhnya sebagai persiapan menuju ke tahap perancangan. Proses ini juga membantu dalam perancangan program aplikasi yang akan dibuat. Perancangan yang matang sangat dipengaruh oleh hasil analisis dari masalah yang dihadapi. Salah satu cara untuk melakukan perencanaan adalah dengan cara pemodelan, dimana pada proses pemodelan tersebut model program aplikasi dianalogikan seperti pembuatan *blueprint* pada pembangunan gedung. Membuat model dari sebuah sistem yang kompleks sangatlah penting karena kita tidak dapat memahami sistem semacam itu secara menyeluruh. Semakin komplek sebuah sistem, semakin penting pula penggunaan teknik pemodelan yang baik.

Kesuksesan suatu pemodelan piranti lunak ditentukan oleh tiga unsur, yang kemudian terkenal dengan sebuan segitiga sukses (*the triangle for success*). Ketiga unsur tersebut adalah metode pemodelan (*notation*), proses (*process*) dan *tool* yang digunakan.

Memahami notasi pemodelan tanpa mengetahui cara pemakaian yang sebenarnya (proses) akan membuat proyek gagal. Dan pemahaman terhadap metode pemodelan dan proses disempurnakan dengan penggunaan *tool* yang tepat.



Gambar 3.1 ***Triangle of success***

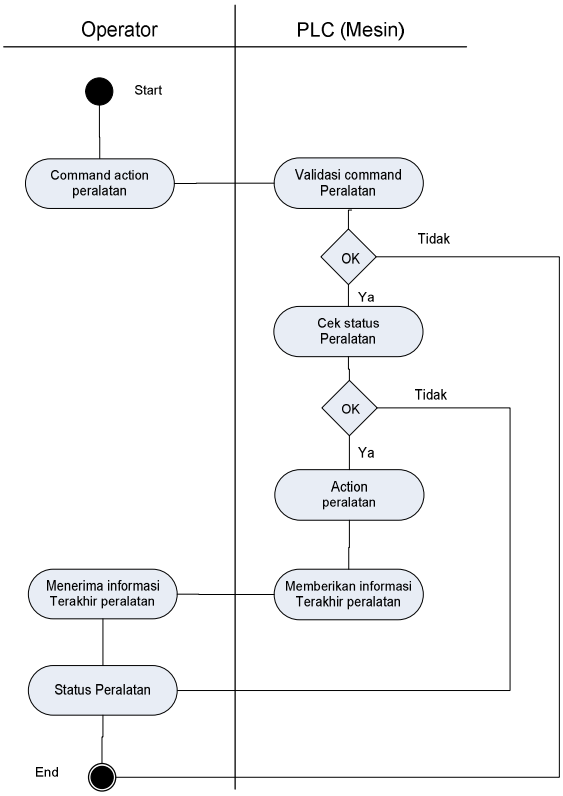
Dalam kegiatan proses perancangan sistem digunakan program Rational Rose yang dapat membantu dalam pemodelan sistem. Berikut ini model analisis pada aplikasi Sistem Monitoring BBM dengan pendekatan program bantu Rational Rose.

a) Proses Bisnis

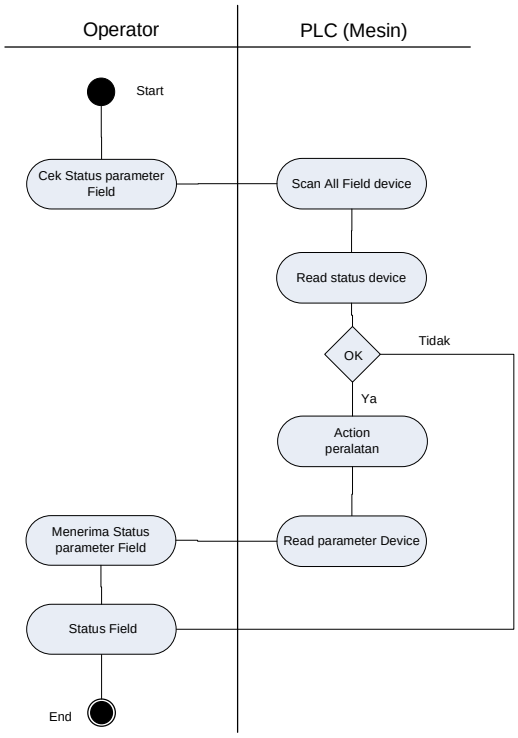
Dalam suatu organisasi mungkin terdapat banyak proses bisnis. Proses bisnis adalah sekumpulan aktivitas yang dirancang untuk menghasilkan keluaran (*output*) tertentu bagi *customer* tertentu. Sebuah proses bisnis menekankan pada bagaimana sebuah organisasi dengan berfokus pada produk yang dihasilkan proses tersebut. Proses disini adalah urutan aktivitas tertentu terhadap waktu dan tempat, dengan sebuah titik awal, sebuah titik akhir dan berbagai *input* dan *output* yang diidentifikasikan dengan baik.

Pemodelan dari sebuah proses bisnis merupakan bagian yang sangat penting dalam rekayasa *sofware* yang akan diterapkan dalam lingkungan organisasi tertentu. Memodelkan proses bisnis dapat membantu menangkap garis besar skema dan prosedur yang menentukan apa yang dikerjakan suatu proses bisnis. Model bisnis didapatkan dengan cara memodelkan proses

bisnis tersebut memberikan gambaran mengenai dimana sebaiknya sistem software yang akan dibuat dianggap cocok untuk diterapkan dalam struktur dan pengujian sistem. Dibawah diterangkan Activity Diagram antara operator dan Mesin PLC.



Gambar 3.2 *Activity Diagram* Operasi Peralatan



Gambar 3.3 *Activity Diagram* Cek Status Lapangan

3.3.2 Perancangan Sistem

Perancangan sistem merupakan langkah selanjutnya jika kita telah menganalisis sistem yang akan di bangun. Perancangan ini mencakup *requirement* dari program aplikasi yang akan dibangun dan mendukung bisnis proses yang telah dibuat pada analisis sistem. *Requirement* merupakan kondisi atau kemampuan yang dipenuhi oleh program aplikasi yang akan dibuat. *Requirement* dalam program aplikasi dibagi menjadi dua kelompok, yaitu :

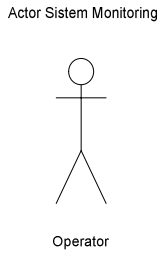
1. ***Fungtional requirement***

   Menentukan tindakan yang harus dapat dimainkan sebuah program aplikasi. *Requirement* ini sering dimodelkan dalam bentuk *use-case* dalam *use-case* model. *Funtional Requirement* yang menentukan masukan dan keluaran dari

sebuah program aplikasi. *Functional Requirement* mencakup fitur kemampuan (*Capabilities),* sifat *(Properties) use-case* tersebut.

a) Actor

Actor menggambarkan pengguna program aplikasi. *Actor* membantu suatu gambaran jelas tentang apa yang harus dikerjakan program aplikasi. Sebagai contoh, sebuah *actor* dapat memberikan masukan ke dalam dan menerima informasi dari program aplikasi. Berikut ini *actor* dari Sistem Monitoring BBM.



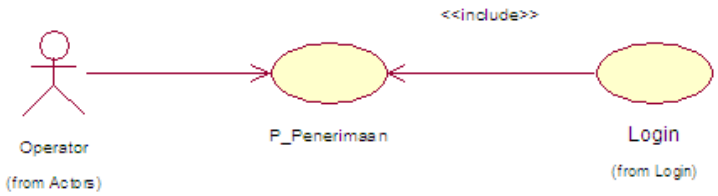
Gambar 3.4 *Actor* Sistem Monitoring BBM

b) Use-Case Model dan Interaksinya dengan Aktor

Use-case Model menggambarkan prilaku program aplikasi, termasuk didalamnya interaksi antara aktor dengan program aplikasi tersebut. Use Case model dibuat untuk mengidentifikasi fungsionalitas yang penting secara arsitektual dari perangkat lunak yang akan dibuat dari lingkungannya.

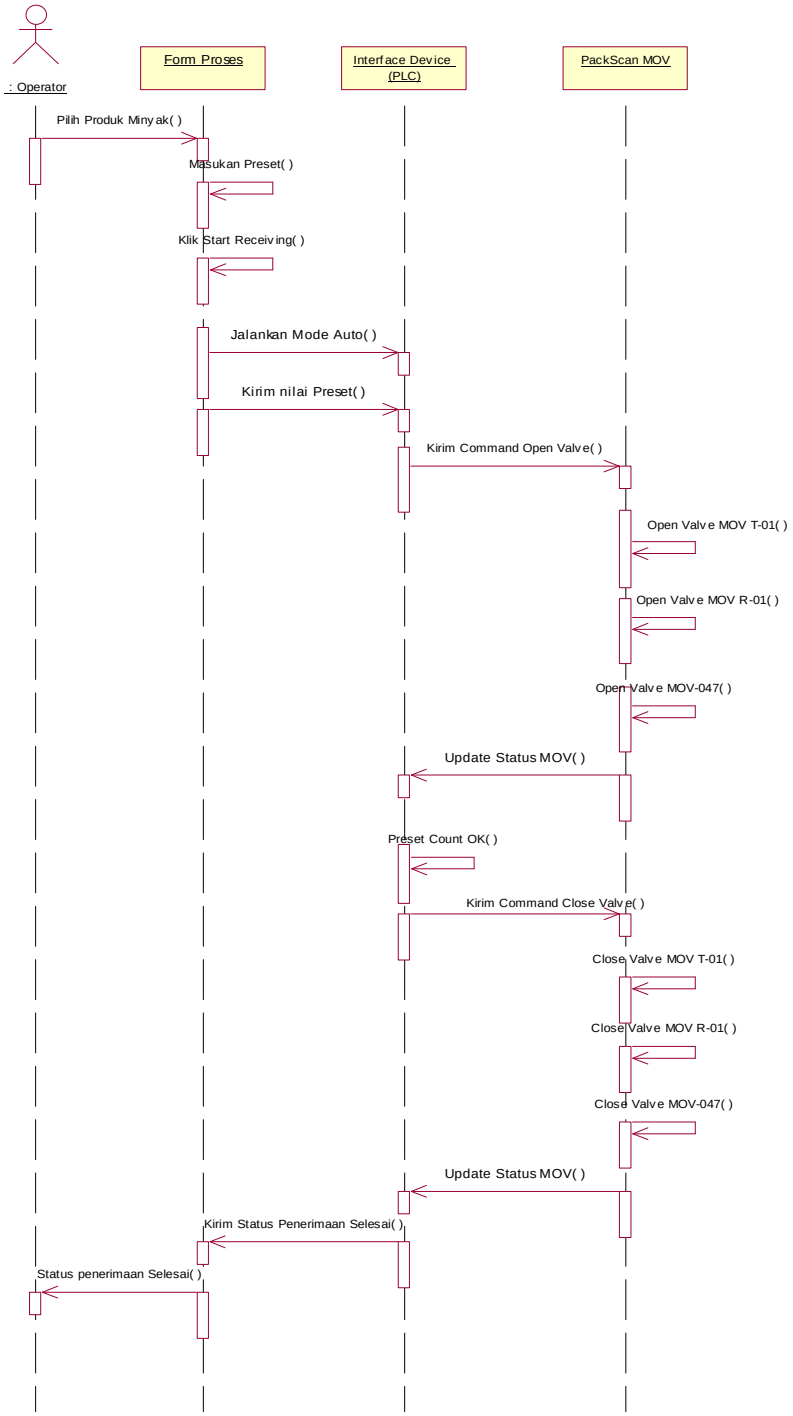
Berikut merupakan Use Case Model dan interaksinya dengan aktor:

a. Diagram Use Case Proses Penerimaan



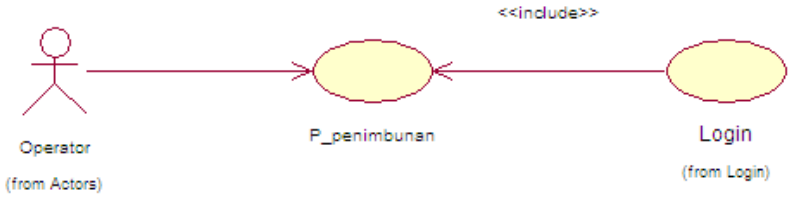
Gambar 3.5 *Use Case Diagram* Proses Penerimaan Minyak

Diagram Use Case Proses Penerimaan digunakan untuk menjelaskan Operator dalam melakukan proses penerimaan minyak Untuk lebih menggambarkan Diagram Use Case proses penerimaan dilakukan dengan membuat Diagram Sequence dari Use Case Diagram diatas, Berikut ini merupakan Diagram Sequence dari Diagram Use Case Proses Penerimaan.



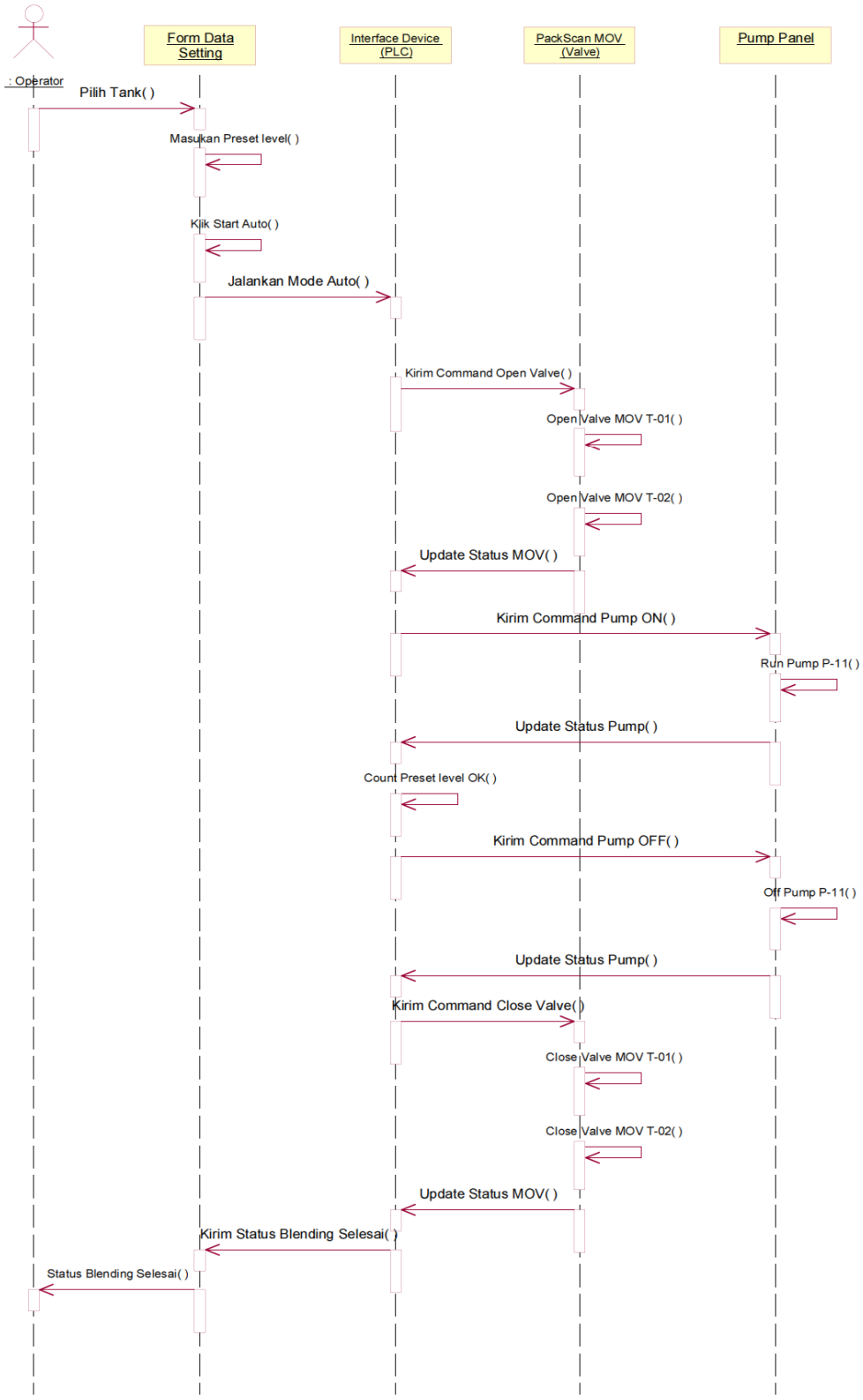
Gambar 3.6 *Sequence Diagram* Proses Penerimaan Minyak

b. Diagram Use Case Proses Penimbunan



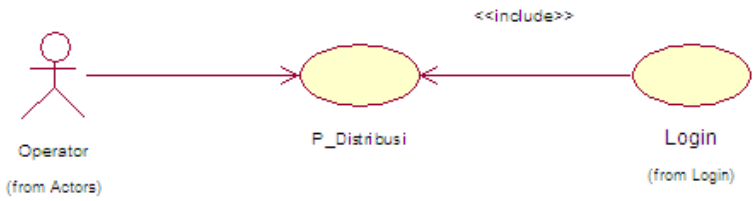
Gambar 3.7 *Use Case Diagram* Proses Penimbunan Minyak

Diagram Use Case Proses Penimbunan digunakan untuk menjelaskan Operator dalam melakukan proses penimbunan (blending tanki) minyak Untuk lebih menggambarkan Diagram Use Case proses penimbunan dilakukan dengan membuat Diagram Sequence dari Use Case Diagram diatas, Berikut ini merupakan Diagram Sequence dari Diagram Use Case Proses Penimbunan.



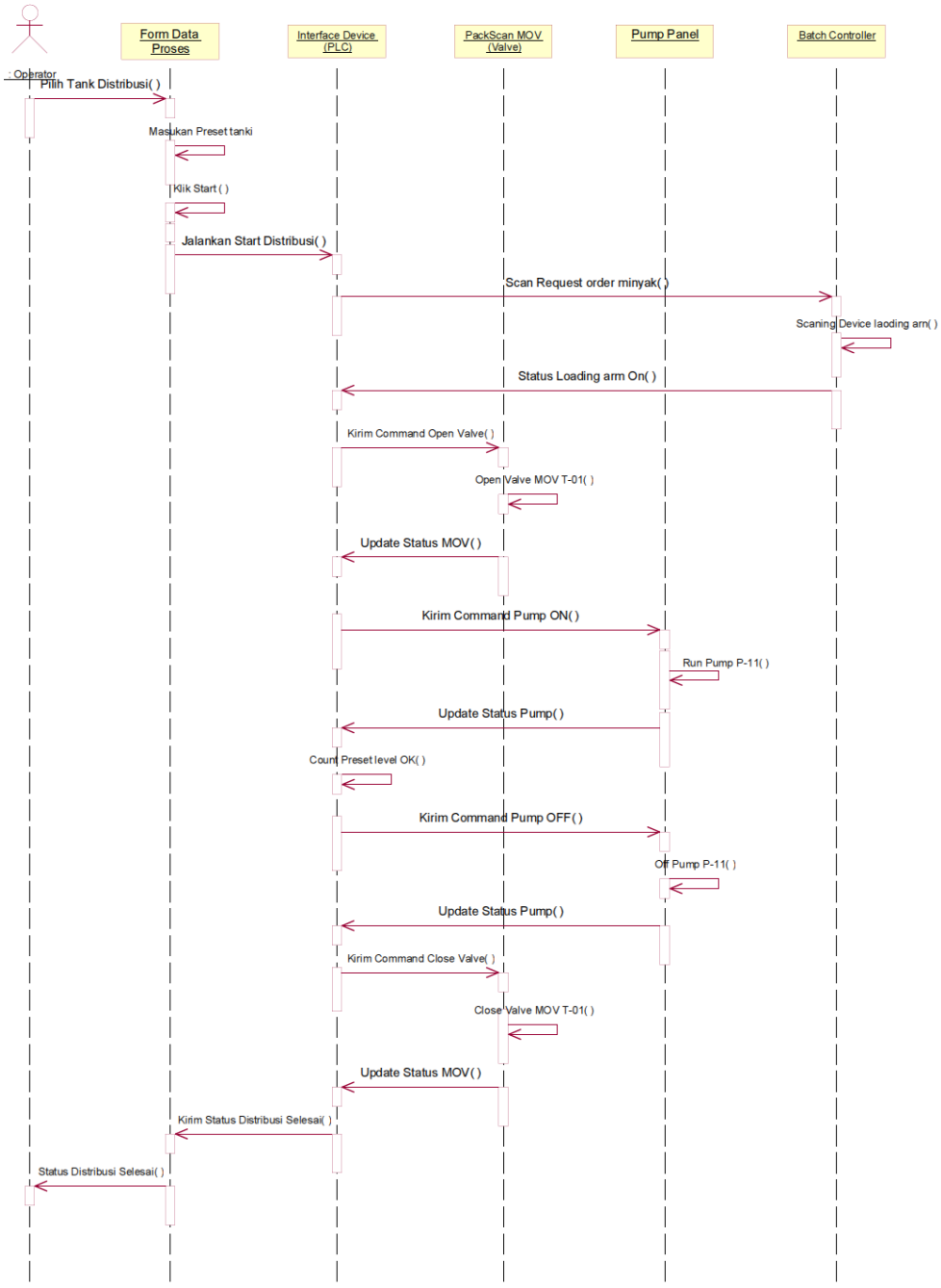
Gambar 3.8 *Sequence Diagram* Proses Penimbunan Minyak

c. Diagram Use Case Proses Distribusi



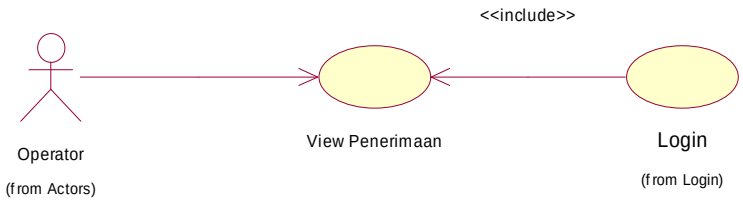
Gambar 3.9 *Use Case Diagram* Proses Distribusi Minyak

Diagram Use Case Proses Distribusi digunakan untuk menjelaskan Operator dalam melakukan proses Distribusi (Penyaluran minyak ke tanki). Untuk lebih menggambarkan Diagram Use Case proses distribusi dilakukan dengan membuat Diagram Sequence dari Use Case Diagram diatas, Berikut ini merupakan Diagram Sequence dari Diagram Use Case Proses distribusi.



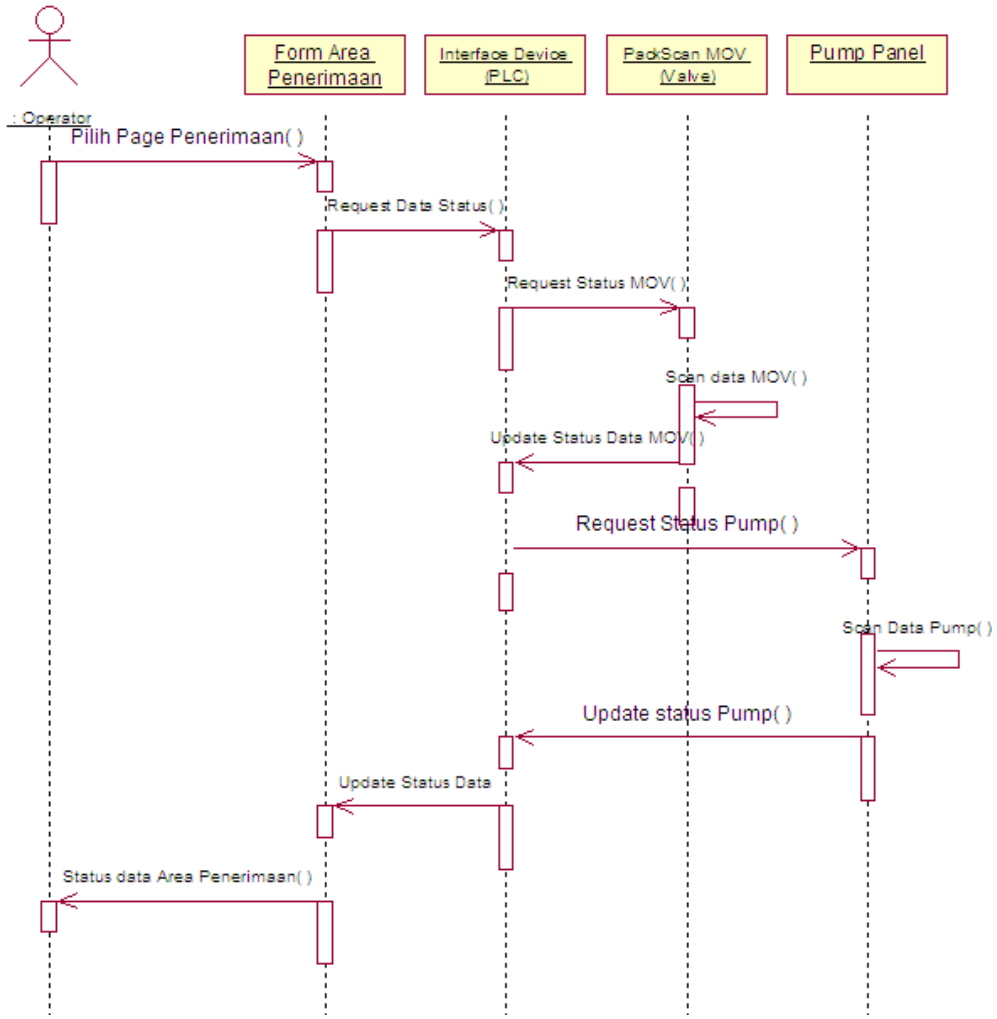
Gambar 3.10 *Sequence Diagram* Proses Distribusi Minyak

d. Diagram Use Case Monitoring Penerimaan



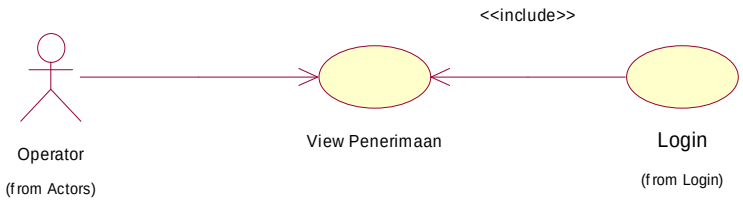
Gambar 3.11 *Use Case Diagram* Monitoring Penerimaan

Diagram Use Case Monitoring Penerimaan digunakan untuk menjelaskan Operator dalam melakukan proses Monitoring Penerimaan. Untuk lebih menggambarkan Diagram Use Case proses monitoring penerimaan dilakukan dengan membuat Diagram Sequence dari Use Case Diagram diatas, Berikut ini merupakan Diagram Sequence dari Diagram Use Case Proses Monitoring Penerimaan.



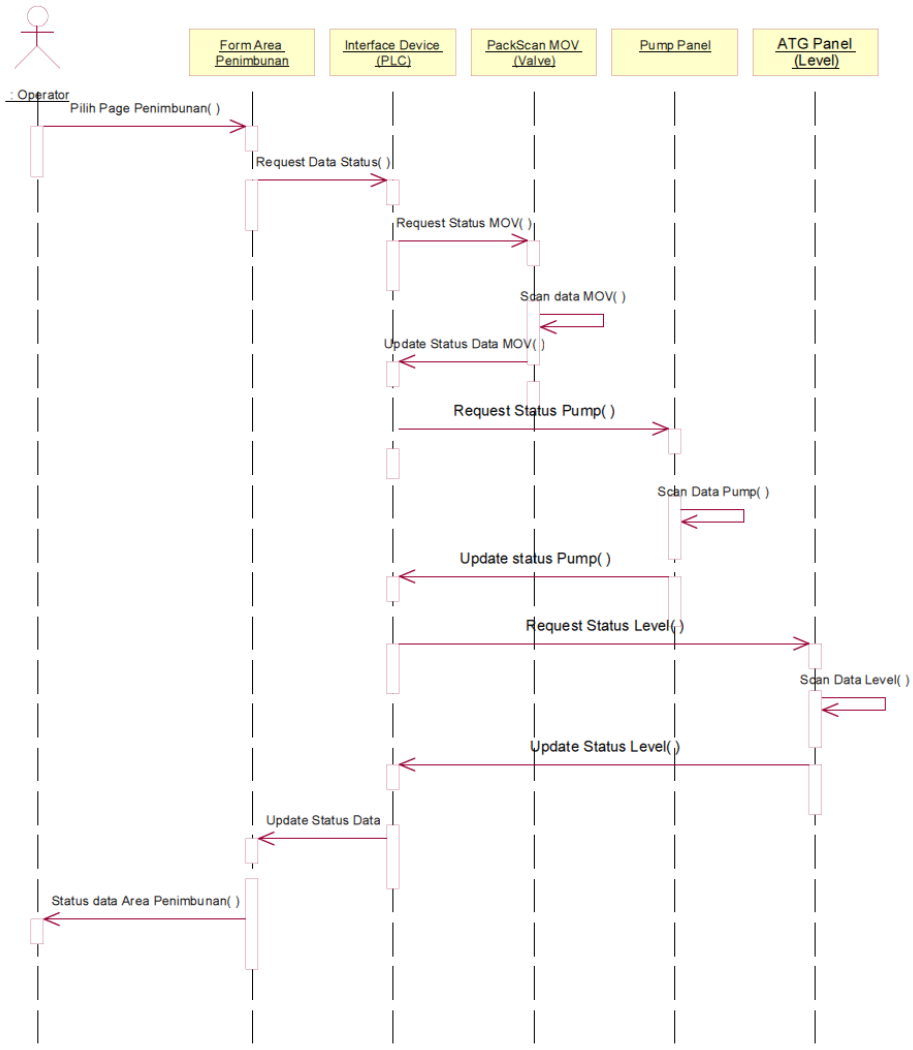
Gambar 3.12 *Sequence Diagram* Monitoring Penerimaan

e. Diagram Use Case Monitoring Penimbunan



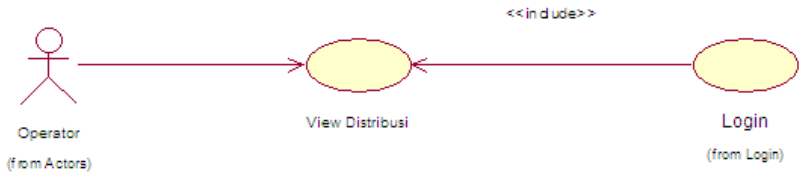
Gambar 3.13 *Use Case Diagram* Monitoring Penimbunan

Diagram Use Case Monitoring Penimbunan digunakan untuk menjelaskan Operator dalam melakukan proses Monitoring Penimbunan. Untuk lebih menggambarkan Diagram Use Case proses monitoring Penimbunan dilakukan dengan membuat Diagram Sequence dari Use Case Diagram diatas, Berikut ini merupakan Diagram Sequence dari Diagram Use Case Proses Monitoring Penimbunan.



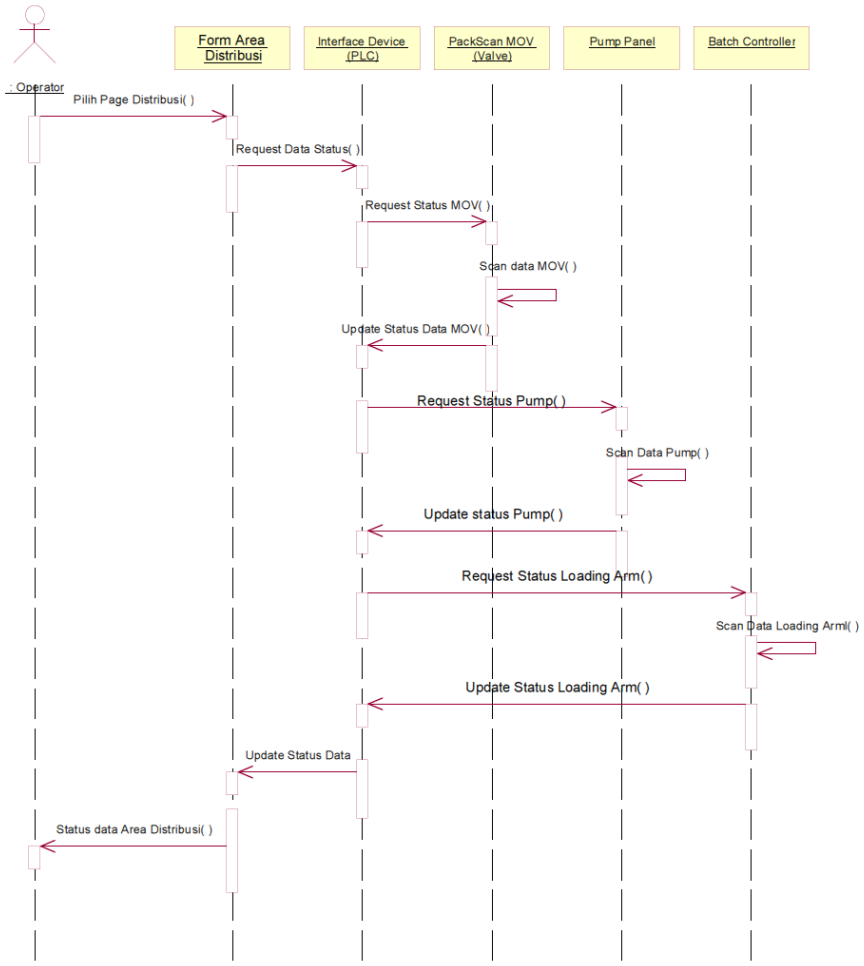
Gambar 3.14 *Sequence Diagram* Monitoring Penimbunan

f. Diagram Use Case Monitoring Distribusi



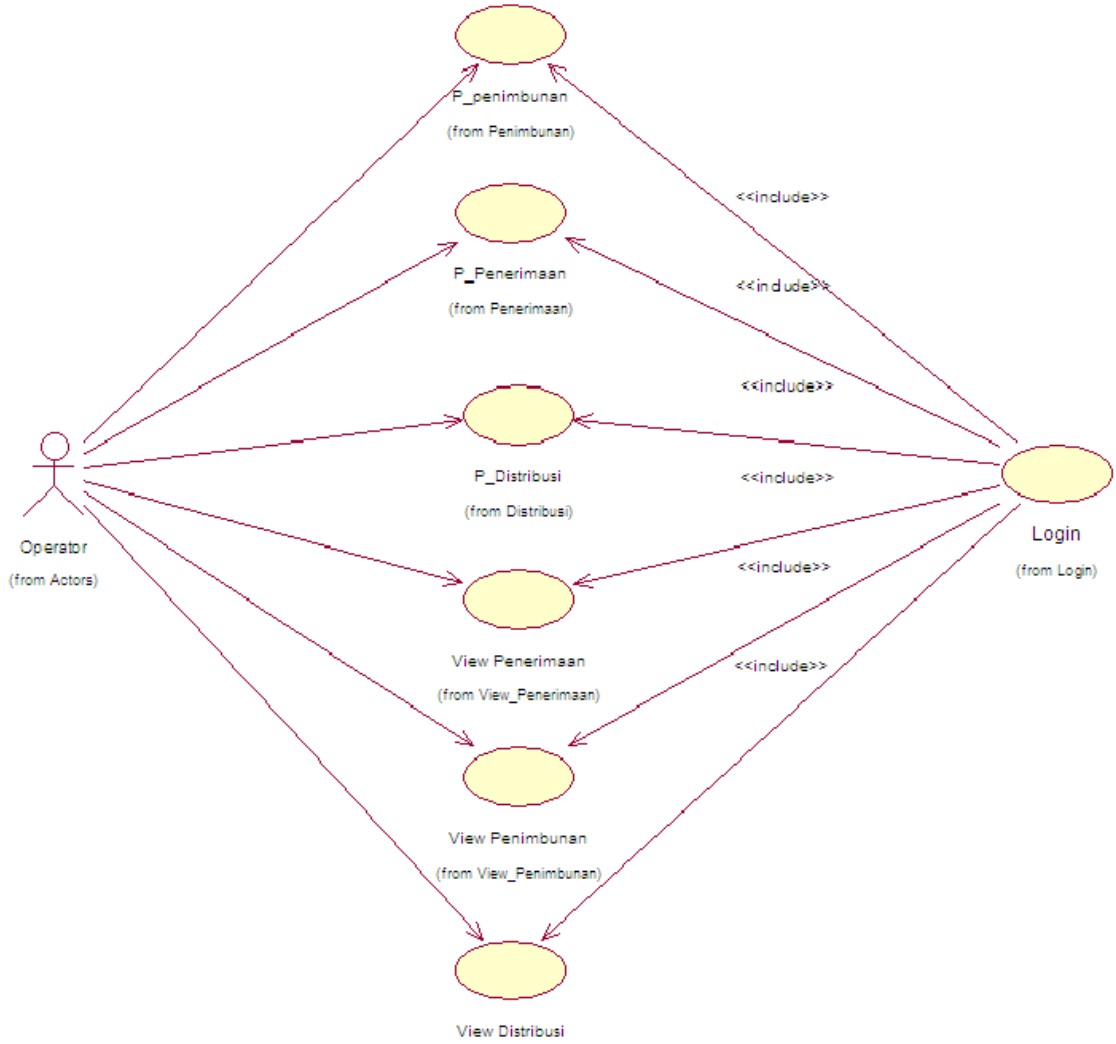
Gambar 3.15 *Use Case Diagram* Monitoring Distribusi

Diagram Use Case Monitoring Distribusi digunakan untuk menjelaskan Operator dalam melakukan proses Monitoring Penimbunan. Untuk lebih menggambarkan Diagram Use Case proses monitoring Penimbunan dilakukan dengan membuat Diagram Sequence dari Use Case Diagram diatas, Berikut ini merupakan Diagram Sequence dari Diagram Use Case Proses Monitoring Penimbunan.



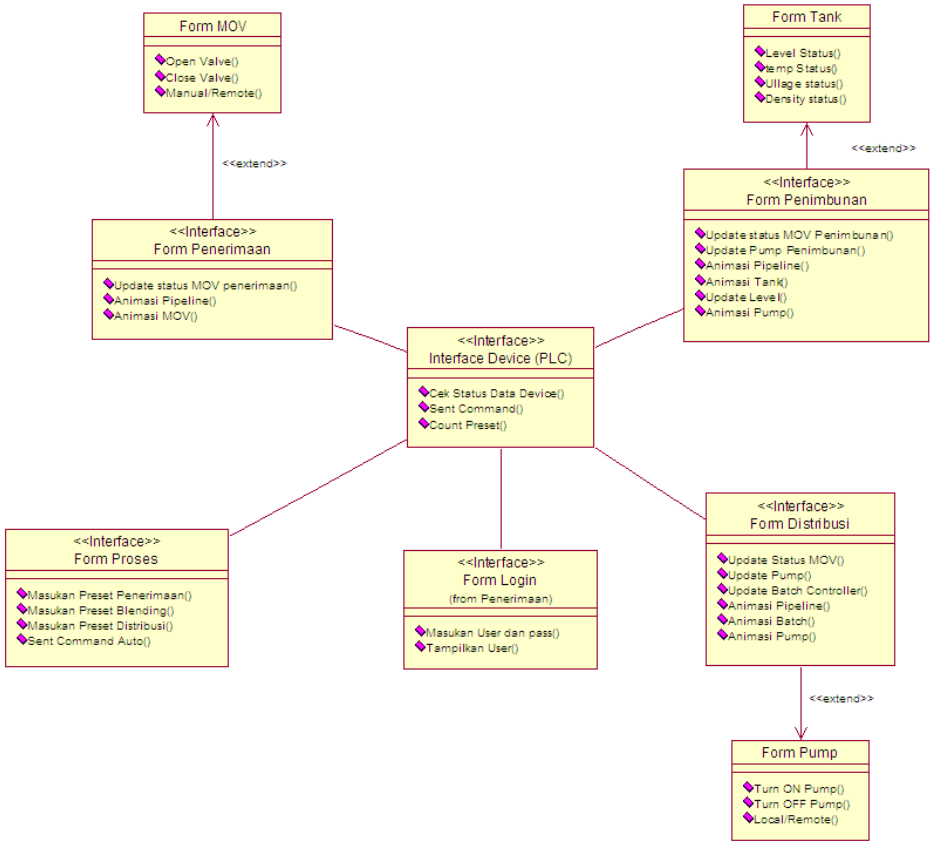
Gambar 3.16 *Sequence Diagram* Monitoring Distribusi

Berdasarkan Diagram-Diagram Use Case Diatas berikut keseluruhan Diagram Use Case Aplikasi Sistem Monitoring BBM :



Gambar 3.17 *Use Case* Sistem Monitoring BBM

Berdasarkan Diagram-diagram Sequence diatas berikut Diagram Class Aplikasi Sistem Monitoring BBM :

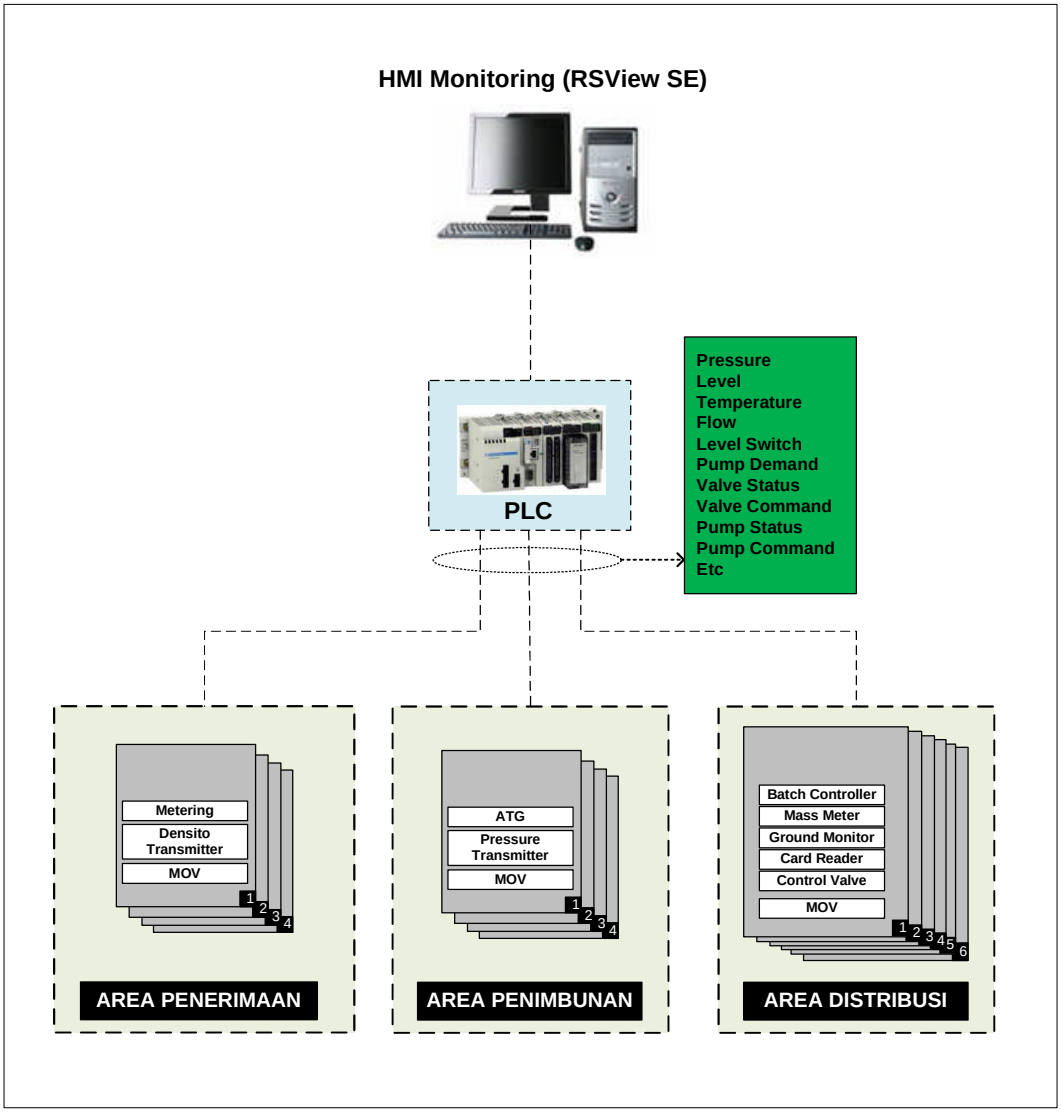


Gambar 3.18 *Class Diagram* Sistem Monitoring BBM

2. Non functional requirment

Non Functional Requirement menggambarkan atribut dari program aplikasi dan lingkungannya. *Requirement* ini biasanya masuk di dalam *use-case* dan tercakup dalam sifat *(property) use-case* tersebut.

3.3.3 Sistem Arsitektur Monitoring BBM



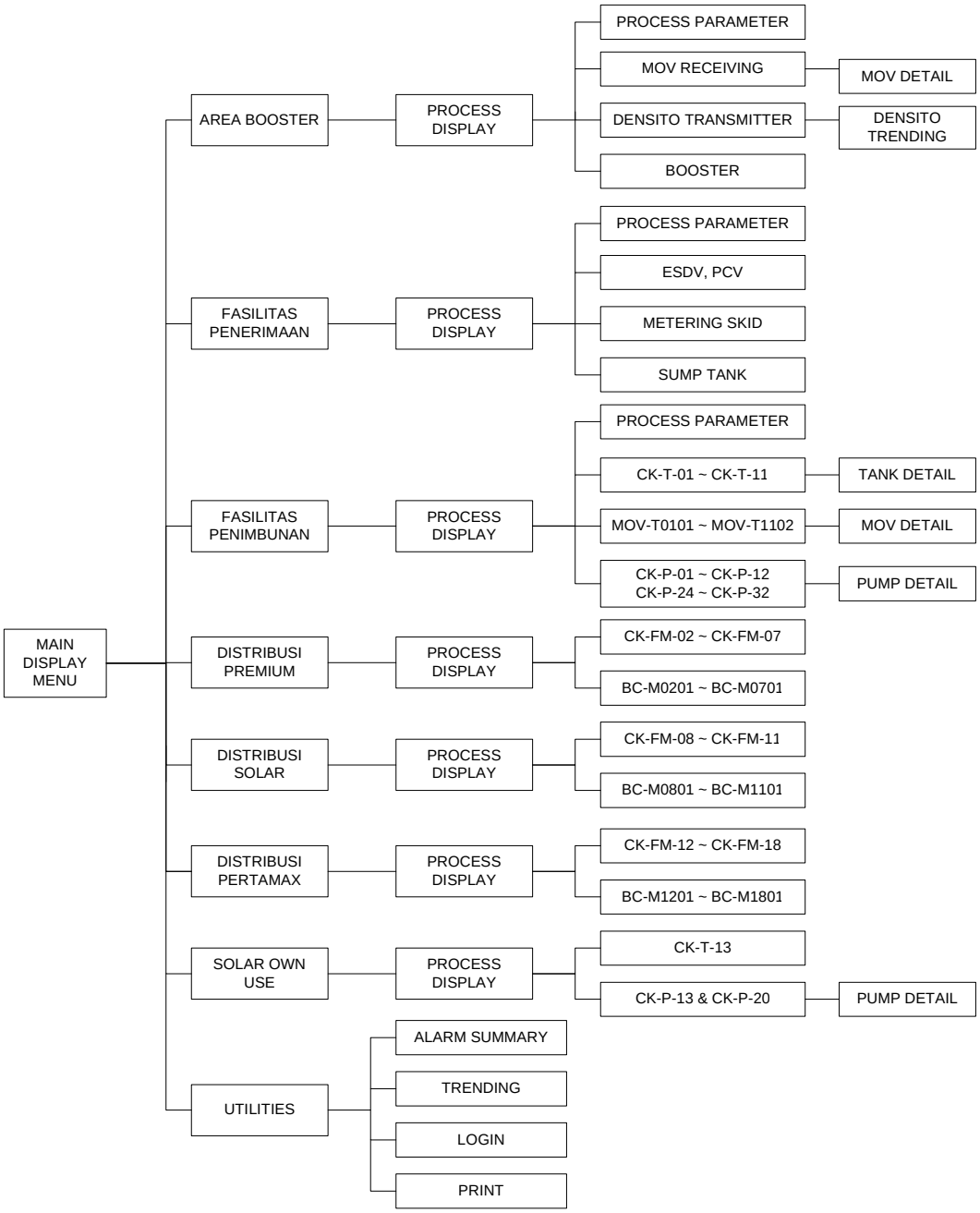
Gambar 3.19 Sistem Arsitektur Monitoring BBM

3.3.4 Perancangan Struktur Menu

Pembuatan aplikasi sistem monitoring BBM menggunakan software HMI RSView SE, berbeda dengan software programming pada umumnya seperti Borland Delphi, C++, Visual Basic dan sebagainya, software HMI RSView SE merupakan sofftware yang dikhususkan untuk aplikasi monitoring proses.

Software ini lebih mudah digunakan karena lebih banyak menggunakan objek yang telah tersedia dalam library yang telah dilengkapi dengan animasi objek serta penggunaan script yang sederhana sehingga listing program hampir tidak ada dikarenakan aplikasi ini tidak berjalan sendiri tetapi harus terpasang dengan software HMI RSView itu sendiri dengan istilah software runtime, berbeda dengan software programming umumnya yang lebih banyak menggunakan listing program fungsi atau procedure dalam membuat aplikasi software sehingga untuk membuat tampilan proses sangat sulit.

Sebelum membuat program aplikasi sistem monitoring BBM rancangan program sebelumnya harus kita buat terlebih dahulu. Dalam software HMI RSView sendiri kita dituntut membuat rancangan struktur menu yaitu halaman apa saja yang akan ditampilkan dalam program aplikasi dikarenakan *license* HMI RSView berdasarkan banyaknya halaman yang akan tampil dalam sebuah program aplikasi. Berikut rancangan struktur menu Sistem Monitoring BBM.



Gambar 3.20 Struktur Menu Sistem Monitoring BBM

Perancangan Sistem Monitoring dibagi berdasarkan area lokasi, ada 7 area proses yang akan dimonitoring, yaitu:

1. Area Booster, di area ini merupakan awal dari sisi penerimaan BBM melalui pipa dari kilang Balongan, ada 2 jalur yang akan dikontrol

diarea ini yaitu jalur penerimaan minyak ke Depot Cikampek serta jalur pipeline minyak BBM ke Depot Plumpang (Jakarta). Instrument yang akan dimonitoring serta dikontrol diarea ini adalah nilai besaran Densito transmitter, MOV (Motor Operating Valve), dan status Booster.

2. Fasilitas Penerimaan, di area ini merupakan area pengukuran minyak BBM yang akan diterima oleh Depot Cikampek. Instrument yang akan dimonitoring serta di kontrol adalah MOV (Motor Operating Valve) jalur manifold ke tanki timbun, nilai besaran Densito transmitter dan metering skid.
3. Fasilitas Penimbunan, di area ini merupakan area penimbunan minyak BBM dalam tanki, terdapat 9 tanki timbun BBM serta 2 tanki minyak interface, selain tanki yang dimonitoring juga ada peralatan MOV dan Pompa yang bisa dikontrol lewat aplikasi HMI.
4. Area Distribusi Premium, merupakan area pendistribusian minyak BBM jenis Premium ke mobil tanki. Instrument yang dimonitoring diarea ini adalah MOV (Motor Operating Valve) outlet tank timbun premium, Pressure transmitter, pompa, FCV (Flow Control Valve) serta Batch Controller.
5. Area Distribusi Solar, merupakan area pendistribusian minyak BBM jenis Solar ke mobil tanki. Instrument yang dimonitoring diarea ini adalah MOV (Motor Operating Valve) outlet tank timbun solar, Pressure transmitter, pompa, FCV (Flow Control Valve) serta Batch Controller.

6. Area Distribusi Pertamax, merupakan area pendistribusian minyak BBM jenis Pertamax ke mobil tanki. Instrument yang dimonitoring diarea ini adalah MOV (Motor Operating Valve) outlet tank timbun pertamax, Pressure transmitter, pompa, FCV (Flow Control Valve) serta Batch Controller.
7. Area Solar Own Use, adalah area pendistribusian Solar yang dipakai untuk kebutuhan Depot seperti untuk penyalaan Genset, instrument yang dimonitoring di area ini antara lain: Pompa, Pressure transmitter.

Selain area tersebut diatas yang dimonitoring, rancangan yang diharapkan di sistem monitoring adalah adanya tampilan trending yang merekam perubahan data terutama data analog (densito trasmitter, level minyak) serta alarm summary yang akan menampilkan alarm – alarm yang terjadi di lapangan. Untuk kepentingan keamanan juga dilakukan proteksi terhadap operator yang akan mengoperasikan aplikasi monitoring ini sehingga tidak sembarang orang yang bisa mengoperasikan aplikasi monitoring.

3.3.5 Implementasi Sistem

A. Membuat Program HMI Dengan Software RSView SE

Seperti yang telah disampaikan dalam pembahasan sebelumnya, bahwa RSView SE adalah salah satu software HMI diantara sekian banyak software yang sejenis seperti In-Touch, Citect, Nematron dan banyak lainnya.

RSView SE memiliki feature yang cukup lengkap seperti Add-On Architecture, penggunaan COM, ActiveX dan VBA, support OPC & DDE dan bemacam-

macam feature lainnya, akan tetapi hal yang paling menarik bagi penulis dalam penggunaan RSView SE adalah kemudahan dalam pembuatan programnya.

Untuk membuat program dengan RSView SE dapat dengan mengikuti langkah-langkah berikut :

1. Membuat project

   Seperti halnya pada pembuatan program komputer pada umumnya sebelum kita mulai membuat form/graphic, dll, maka terlebih dahulu kita harus membuat project. Project ini akan membentuk satu folder utama, namanya sesuai dengan nama projectnya. Didalam folder ini akan dibuat pula beberapa sub folder sesuai dengan bagian-bagian project dan sebuah file ber-ekxtensi .rsv sebagai file projectnya.

2. Setup komunikasi

   Setup ini dibutuhkan untuk memperkenalkan atau menghubungan perangkat keras/device yang akan dikontrol / dimonitor dengan program RSView SE. Untuk komunikasi dengan device PLC buatan Allen Bradley seperti PLC-5, SLC atau Micrologix dapat menggunakan koneksi direct driver dengan program driver RSLinx. Sedangkan untuk komunikasi dengan device lainnya RSView SE menggunakan OPC atau DDE sebagai penghubungnya. OPC memungkinkan RSView SE bertindak sebagai server maupun sebagai client, atau komunikasi peer to peer dengan program/project RSView SE lain dan atau OPC server lainnya. RSView SE dapat pula di set untuk komunikasi dengan DDE server atau DDE client untuk bertukar data dengan aplikasi lain seperti Microsoft Excel dan sejenisnya yang support DDE.

3. Membuat graphic display

Graphic display adalah gambaran dari suatu proses, sehingga gambar yang ditampilkan didalam graphic ini harus mencerminkan objek yang sedang dikontrol/dimonitor, misalnya menggambar motor dengan simbol bulatan dan huruf M didalamnya serta simbol-simbol untuk objek lain yang mudah dimengerti oleh operator. Sebaiknya juga menggunakan kombinasi warna untuk membedakan status objek tersebut (misalnya dalam keadaan operasi, stop, alarm, dll) hal ini penting dilakukan agar operator dapat dengan mudah menerjemahkan status peralatan tanpa harus melihat status summary maupun alarm reportnya.

Untuk membuat graphic ada beberapa cara yang dapat digunakan seperti :

a) Menggunakan Drawing Tool untuk membuat gambar dan text. Dengan fasilitas ini kita dapat membuat gambar bulat, segitiga, segiempat dan bentuk gambar lainnya serta membuat gambar yang lebih kompleks seperti trend atau alarm summary atau dapat juga langsung memasukkan objek ActiveX.
b) Graphic juga dapat dibentuk dengan meng-drag and drop library yang disiapkan oleh RSView SE maupun library yang dibuat sendiri.
c) RSView SE juga memungkinkan untuk meng-import graphic dari gambar yang telah dibuat terlebih dahulu dengan program lain seperti Corel Draw dan sejenisnya.

4. Setup Tag

Sebelum membuat tag, kita harus memahami terlebih dahulu apa dimaksud dengan tag di RSView SE, jenis dan kegunaannya. Tag dapat diartikan sebagai Nama Variabel, baik untuk variabel device maupun memory (RAM). Tag yang menerima data/nilai dari device controller seperti PLC, DCS dll atau dari server OPC disebut Device Tags sedangkan yang menerima data/nilai dari internal RSView SE disebut sebagai Memory Tags. Berdasarkan tipe data, tag dapat dibedakan menjadi 3 tipe tag, yaitu :

1. Analog tag, menyimpan data numeric dalam range tertentu
2. Digital tag, menyimpan data dengan nilai 0 atau 1
3. String tag, menyimpan data string yang panjangnya maksimal 82 karakter

Untuk membuat tag kita dapat melakukannya dengan beberapa cara, diantaranya:

a) Membuat tag berdasarkan kebutuhan (as needed), pembuatan dengan cara ini biasanya dilakukan pada saat kita akan membuat animasi atau event dari sebuah graphic. Cara ini dapat dilakukan dengan menggunakan fasilitas tag browser.
b) Tag juga dapat dibuat terlebih dahulu / sebelum membuat graphic. Dengan menggunakan fasilitas Tag Database Editor, kita dapat membuat tag langsung dalam jumlah banyak.

c) Untuk tag device dari controller buatan Allen Bradley, kita bisa langsung mengimport database device tersebut menjadi tag dengan menggunakan fasilitas PLC Database Editor.

d) Import juga dapat digunakan untuk membuat tag yang berasal dari server OPC.

5. Setup logging data

Logging sangat dibutuhkan untuk merekam seluruh aktifitas program, baik yang dilakukan oleh operator, maupun kejadian-kejadian yang ditimbulkan oleh device atau perlatan dilokasi. Logging yang dimaksud disini dapat berupa Alarm Log, Activity Log maupun Data Log, ketiga hal ini dapat dilakukan dengan menggunakan fasilitas masing-masing Alarm Log Setup Editor, Activity Log Setup Editor dan Data Log Setup Editor. Semua informasi log akan disimpan dalam format dBASE IV (.DBF) dan dapat ditampilkan dengan program lain seperti Microsoft Excel, Crystal Report, dll. Khusus untuk Data Log, kita dapat menyimpannya dalam media database yang lain dengan menggunakan fasilitas ODBC.

6. Customize dan integrasi project

Salah satu fasilitas yang cukup menarik dari RSView SE adalah tersedianya fasilitas VBA (Visual Basic for Application), dengan fasilitas ini kita dapat meningkatkan kapabilitas program yang dihasilkan oleh RSView SE. Selain itu dengan fasilitas ini kita juga dengan mudah dapat mengintegrasikan program RSView SE dengan program-program lain yang mempunyai fasilitas application interface.

7. Pengamanan project

Pengamanan project dapat dilakukan dengan membuat pembatasan user / operator dengan membuat User Account, dan mengaktifkan/meng-setup security project melalui Security Code Editor. Dengan fasilitas ini kita dapat membatasi aktifitas user untuk menjalankan perintah, menampilan graphic atau pembatasan tag yang dapat diakses. Setelah melakukan step-step diatas maka kita akan memiliki sebuah project yang siap untuk digunakan untuk mengontrol, memonitor dan merekam seluruh aktifitasnya.

B. Instalasi Program Aplikasi

Untuk menjalankan aplikasi yang telah dibuat kita harus menginstall software RSView SE pada komputer yang akan dijadikan monitoring proses , berikut ini proses instalasi RSView SE yang umum digunakan untuk windows 9x hingga windows XP:

1. Tutup semua aplikasi windows yang sedang terbuka
2. Taruh CD Installer RSView SE didalam CD-ROM Drive hingga CD-ROM start secara otomatis. Atau jalankan Setup.exe melalui windows explorer jika CD-ROM tidak dapat jalan secara otomatis.
3. Pilih opsi instalasi RSView SE (Development atau Run Time)
4. Ikuti petunjuk yang ditampilkan dalam proses instalasi.
5. Setelah proses instalasi akan ditampilkan permintaan untuk proses aktivasi.

Aktivasi RSView SE dibedakan berdasarkan jenis instalasi (Development atau Runtime) serta jumlah tag yang dibeli. Bagaimana dengan RSView SE yang

tidak di-aktivasi? jawabnya; tetap bisa digunakan, tetapi dengan trial mode tentunya. Dalam mode ini, RSView SE dapat digunakan untuk latihan karena kita tetap dapat membuat project dengan tag memory sebagai sumber nilainya.

C. Penjelasan Aplikasi

Latar belakang warna untuk semua tampilan proses adalah warna hitam, alasan digunakan warna hitam sebagai background tampilan dikarenakan karena sistem monitoring dilakukan terus menerus supaya tidak membuat lelah mata operator yang menjalankan monitoring tersebut sehingga merasa nyaman dalam mengoperasikan aplikasi tersebut.

Hal-hal yang sangat penting dalam pengoperasian aplikasi sistem monitoring BBM , antara lain :

a. Tampilan proses

   Kondisi jika terjadi kondisi alarm dilapangan, maka aplikasi akan menunjukkan warna merah yang berkedip. pada objek yang terjadi alarm. Kondisi abnormal equipment ditunjukkan oleh warna kuning/merah berkedip pada equipment bersangkutan. Warna pipa tergantung pada jenis fluida yang mengalir, jika tidak ada aliran maka pipa akan berwarna putih.

   Setiap tampilan proses dirancang sederhana tapi menampilkan semua informasi yang diperlukan. Warna yang dipilih adalah yang warna-warna yang cerah dan jelas.

b. Tampilan Popup

Tampilan popup memungkinkan operator untuk mengontrol atau melihat kondisi detail equipment tertentu dari HMI.

c. Tag names data

Semua data analog dari lapangan akan ditampilkan pada HMI. Untuk menghindari kesalahan, setiap tampilan data analog di HMI dilengkapi dengan nama tag yang berkaitan.

d. Engineering Unit

Berikut adalah tabel engineering unit pada HMI.

Tabel 3.2 Tabel Engineering

Variable	**Engineering Units**	**Abbreviations**
Level	Meter	M
Suhu	Derajat celcius	DEG C
Tekanan	Kilogram per centimeter kuadrat	Kg/cm2
Aliran	Liter per jam	L/Hr
Densitas	Gram per centimeter kubik	Gr/cm3

e. Detail Tampilan

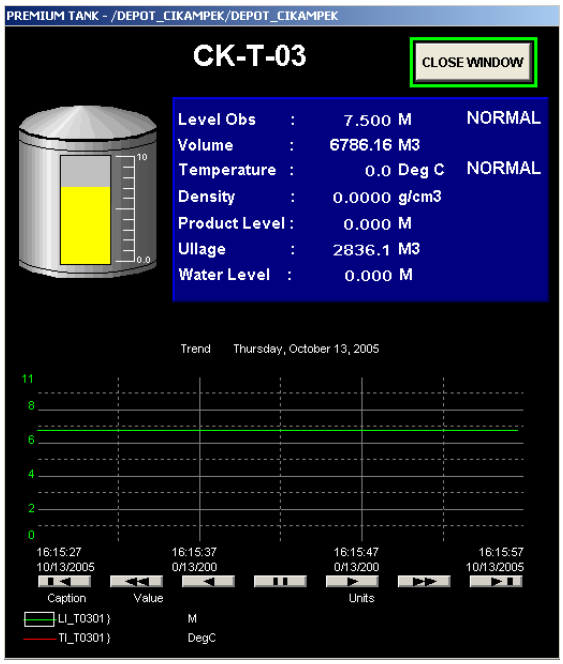
Indikasi alarm berupa teks ditempatkan berdekatan dengan equipment yang berkaitan. Khusus untuk pompa, indikasi alarm untuk kondisi fault ditunjukkan dengan perubahan warna pompa menjadi merah berkedip.

Popup window dapat ditampilkan dengan mengklik pada equipment yang menampilkan detail dari equipment yang bersangkutan. Masing-masing popup dijelaskan di bawah ini.

1) Tank

Popup window untuk tank menampilkan informasi sebagai berikut:

1. Level, alarm status dan level data trending
2. Volume
3. Temperature, alarm status dan temperature data trending
4. Density
5. Product level
6. Ullage
7. Water Level



Gambar 3.20 Tampilan *Popup Tank*

2) Pompa

Popup window untuk pompa menampilkan informasi sebagai berikut:

a) Tekanan

b) Alarm status

c) Control status

d) Operation status

e) Running hours

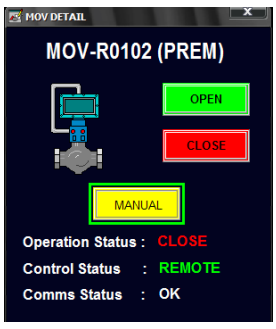


Gambar 3.21 Tampilan *Popup* Pompa

3) MOV (Motor Operating Valve)

Window popup MOV menampilkan informasi sebagai berikut::

1. Operation status
2. Control Status
3. Communication Status



Gambar 3.22 Tampilan Popup MOV

Tampilan popup MOV diatas berfungsi untuk melakukan pengoperasian MOV secara manual, HMI hanya bisa memerintahkan MOV untuk melakukan operasi buka (open) dan tutup (close) serta melakukan override supaya MOV membuka atau menutup.

Prosedure pengoperasian untuk tampilan popup MOV:

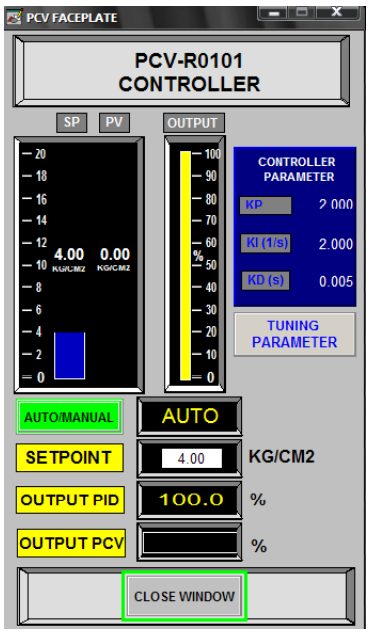
a) Untuk menampilkan button OPEN dan CLOSE, button MANUAL terlebih dahulu harus ditekan.

b) Jika salah satu button OPEN dan CLOSE ditekan maka button MANUAL akan menghilang.

c) Jika kondisi button CLOSE dan button OPEN dalam kondisi ditekan,kondisi ini tidak diperbolehkan dalam system, maka akan muncul button RESET.

4) PCV (Pressure Control Valve)

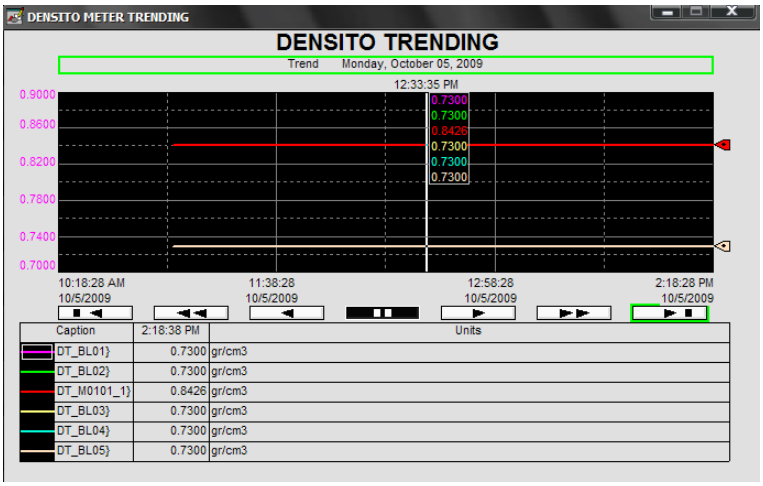
Popup window PCV (Pressure Control Valve) berfungsi untuk memonitor dan mengontrol valve menggunakan kontrol PID. PCV controller bisa dioperasikan dalam mode AUTO/MANUAL Untuk melakukan tuning parameter hanya bisa dilakukan oleh pihak yang telah diberi akses.



Gambar 3.23 Tampilan *Popup Pressure Control Valve*

5) Densito trending

Popup Densito trending berfungsi untuk melihat trending dari semua densito transmitter yang terhubung ke PLC, dalam trending tersebut kita bisa melihat record dari nilai trending sebelumnya.



Gambar 3.24 Tampilan *Popup Densito Trending*

1. Trending dan alarm summary dapat diakses operator dengan menekan tombol navigasi yang terdapat pada bagian atas tampilan.
2. Alarm summary akan menampilkan semua alarm yang pernah muncul.
3. Trending menampilkan data secara real time dan historical. Untuk data historical disimpan dalam hardisk untuk periode 3 bulan dan setelah melewati batas 3 bulan data akan diupdate secara otomatis.

f. Sekuriti level dan user login

Untuk dapat menggunakan HMI diperlukan Login user. Terdapat 2 tipe login user yaitu engineer level dan operator level. Login HMI akan secara otomatis sama dengan user login dalam windows.

Tabel 3.3 Security Level dan User Login

User	**Operasi yang dapat dilakukan**	**Login**
Engineer	- Memonitor proses - Melakukan kontrol dari HMI - Merubah parameter kontrol	CIKAMPEK-ENG
Operator	- Memonitor proses - Melakukan kontrol dari HMI	CIKAMPEK-OPRT

g. Layout tampilan

Berikut ini adalah layout untuk setiap tampilan proses:

1. Banner Informasi

Bagian ini muncul pada setiap tampilan. Terdiri dari informasi tanggal dan waktu, Login user aktif, status alarm terakhir dan tombol acknowledge buzzer.

2. Tombol navigasi

Bagian ini muncul pada setiap tampilan. Terdiri atas tombol untuk berpindah ke tampilan proses, alarm summary atau trending sesuai dengan teks yang terdapat pada tombol. Selain itu juga terdapat fasilitas untuk berganti login user dan mencetak tampilan window yang sedang aktif.

3. Tampilan proses

Berbeda pada setiap tampilan. Operator dapat memonitor kondisi plant berdasarkan tombol navigasi yang dipilih.

D. Pelatihan

Proses pengenalan program aplikasi ini juga dilakukan dengan tujuan untuk mengenalkan pengguna pada program aplikasi yang nanti akan digunakan, meliputi mengetahui cara menjalankan, mematikan, pengenalan antar muka, mengenai warna dan objek , serta fungsi-fungsi yang terdapat pada program aplikasi.

3.3.6 Pengujian Sistem

Proses pengujian dilakukan dengan cara mencoba program aplikasi yang sudah dibuat pada aplikasi Sistem Monitoring BBM yang telah terinstall. Dengan interval waktu yang cukup sehingga pemakai mengetahui fungsi-fungsi dalam program aplikasi yang baru dan dapat memberi tahukan fungsi yang masih kurang atau mungkin dirasa tidak perlu.

Proses pengujian dibagi menjadi beberapa kategori, yaitu :

1. *Functional*

   Pengujian dilakukan untuk memastikan fungsional setiap menu yang ada pada program aplikasi, data-data dapat ditampilkan dan sesuai dengan keinginan, yakni sesuai dengan filter data yang diinputkan.

2. *Performance*

   Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi persyaratan kinerja yang sesuai, meliputi kecepatan dan keakuratan dalam menampilkan data yang diinginkan. Dalam kategori ini, harus dirancang dengan baik sebab sangat berpengaruh pada jalannya sistem.

Myers membedakan jenis program testing menjadi beberapa macam, meliputi :

1. *Module test* : tes per modul dalam lingkungan yang tertutup.
2. *Integration test* : verifikasi antar muka antar modul dan bagian dalam suatu sistem.

3. *External function test* : verifikasi fungsi eksternal.
4. *System test* : verifikasi dan validasi kerja sistem dalam lingkungan yang dibuat secara khusus.
5. *Acceptance test* : validasi sistem dan persyaratan pengguna.
6. *Installation test* : tes untuk menemukan kesalahan pada proses instalasi.
7. *Simulation test* : tes yang mencoba untuk meniru keadaan pada lingkungan pada saat sistem digunakan.
8. *Field test* : tes yang dilakukan pada lingkungan pemakaian pada kondisi sebenarnya.

Steve McConnel mengusulkan bahwa sistem/perangkat lunak seharusnya memiliki karakteristik-karakteristik luar dan dalam sebagai berikut :

Karakteristik luar yang meluputi karakteristik-karakteristik yang dapat diamati secara langsung oleh pengguna, seperti :

1. Ketetapan

   Tingkat kebebasan sistem dari kesalahan analisis, perancangan, serta implementasi. Ketepatan disini maksudnya adalah sedekat (atau sejauh) mana sistem/perangkat lunak yang kita kembangkan memenuhi harapan serta kebutuhan pengguna

2. Daya guna

   Tingkat kemudahan untuk belajar serta menggunakan sistem. Sebuah sistem/perangkat lunak seharusnya mencoba memahami karakteristik semua pengguna; baik yang mahir, berkembang

sedang, maupun yang awam tentang komputer (meskipun seharusnya ia memahami apa yang dapat dilakukan sistem baginya), Karena pengguna bukanlah seseorang yang mau belajar temtang komputer melainkan menggunakannya untuk melakukan suatu pekerjaan tertentu, maka sistem/perangkat lunak seharusnya dikembangkan untuk dapat digunakan oleh semua pengguna dengan berbagai tingkat kemahiran.

3. Efisiensi

   Pemakai minimal dari sumber daya sistem. Sistem/perangkat lunak seharusnya dapat meminimalisasi penggunaan memori, penggunaan sumber daya CPU, penggunaan ruang penyimpanan (hardisk) dan sebagainya sehingga sistem/perangkat lunak dapat digunakan dalam spesifikasi yang tidak terlalu tinggi yang dapat dipenuhi oleh kebanyakan calon pengguna.

4. Keandalan

   Mencakup kemampuan sistem/perangkat lunak untuk menjalankan semua fungsi yang dibutuhkan dan diharapkan.

5. Integritas

   Tingkat kemampuan sistem/perangkat lunak dalam mencegah akses yang tidak sah atau tidak pada tempatnya ke program atau data didalamnya. Yang termasuk dengan integritas adalah menjaga sistem dari akses yang dilakukan oleh orang-orang yang tidak berhak pada sumberdaya yang penting bagi organisasi,

misalnya basis data serta berkas-berkas (*file-file*) lain yang penting bagi organisasi dimana sistem/perangkat lunak kita gunakan. Selain itu, integritas juga berarti menjaga konsistensi data saat terjadi manipulasi-manipulasi penambahan, pembaharuan, serta penghapusan data pada basis data.

6. Kemampuan adaptasi

   Bagaimana sistem/perangkat lunak dapat dipakai dalam aplikasi atau lingkungan yang lain dari lingkungan saat sistem/perangkat lunak itu diciptakan. Salah satu kelemahan umum dari pengembang yang tidak berpengalaman adakah merasa cukup puas ketika sistem/perangkat lunak berjalan dengan baik di lingkungannya namun saat di instal di lingkungan pengguana, sistem/perangkat lunak itu kemudian bermasalah. Hal ini sesungguhnya tidak boleh terjadi.

7. Keakuratan

   Tingkat kebebasan sistem/perangkat lunak dari kesalahan-kesalahan (*bugs*), khususnya ditinjau dari keluaran yang dihasilkan. Keakuratan berbeda dengan ketepatan. Keakuratan menentukan seberapa baik sistem melakukan pekerjaan yang diberikan; tekanannya bukan pada apakah sistem dibuat dengan baik (ketepatan).

8. Kekuatan

   Tingkat kemampuan sistem melanjutkan fungsinya bila terdapat masukan yang tidak sah atau kondisi lingkungan yang menekan. Sebuah sistem yang baik, misal untuk ATM, seharusnya dilengkapi dengan piranti-piranti yang mencegah perlakuan yang tidak wajar dari pengguna, misal dengan menambahkan alarm, pesan-pesan peringatan saat pengguna melakukan hal-hal yang tidak wajar.

Karakteristik Dalam. Karakteristik dalam hal ini mencakup karekteristik karakteristik yang dipikirkan oleh pengembang; baik itu analisis, perancangan, ataupun pemrograman. Karakteristik dalam hal ini mencakup :

1. Daya tahan

   Kemudahan mengubah sistem/perangkat lunak untuk mengubah atau menambah kemampuan , memperbaiki unjuk kerja atau memperbaiki kerusakan. Ini dapat dicapai dengan dokumentasi yang baik saat perencanaan , analis, perancangan serta implementasi.

2. Keluwesan

   Tingkat seberapa jauh kita dapat menempatkan sistem/perangkaat lunak dalam lingkungan saat sistem/perangkat lunak diciptakan. Dalam hal ini, seorang pengembang yang baik seharusnya

mengadaptasi kemungkinan-kemungkinan jika sistem/perangkat lunak diinstal dilingkungan pengguna.

3. Kemudahan diinstal

   Sejauh mana sistem/perangkat lunak yang dikembagkan mudah diterapkan dilingkungan pengguna. Suatu sistem/perangkat lunak betapapun canggihnya, akan ditinggalkan pengguna kalau sukar diterapkan dilingkungan pengguna.

4. Daya guna ulang

   Sampai dimana kemudahan kita dapat menggunakan bagian sebuah sistem dalam sistem-sistem yang lain. Bagian ini sangat penting sebab seringkali kita mengembangkan sistem/perangkat lunak dimana bagiannya diperlukan sistem/perangkat lunak lain. Modularisasi serta karakteristik-karakteristik yang dijanjikan bahasa-bahasa pemrograman berorientasi objek memungkinkan penggunaan ulang bagian dari sistem/perangkat lunak yang diciptakan dalam suatu sistem dalam sistem yang lain.

5. Kemudahan dipahami

   Jika kemudahan dipahami berbicara tentang hal yang makro sifatnya, maka kemudahan dibaca lebih bersifat mikro, yaitu kemudahan membaca serta mamahami kode sumber sebuah sistem/perangkat lunak, khususnya pada tingkat pernyataan yang rinci.

6. Daya uji

   Tingkat dimana kita sebagai pengembang dapat memeriksa setiap unit dan sistem keseluruhan; tingkat dimana kita dapat menyatakan bahwa sistem/ perangkat lunak sesuai dengan spesifikasi yang ditentukan dalam SRS (*System/Software Requirement Spesification*).

Metode pengujian yang digunakan adalah *White-box Testing.* Pengujian jenis ini mengasumsikan bahwa spesifikasi logika adalah penting dan perlu dilakukan pengujian untuk menjamin apakah sistem/perangkat lunak berfungsi dengan baik. Tujuan utama dari strategi ini adalah pengujian berbasis kesalahan. Dalam hal sistem berbasis objek, kita harus memeriksa semua objek yang terlibat dalam sistem. Ada beberapa cara yang dilakukan untuk melakukan pengujian ini, yaitu :

a. Memeriksa semua fungsi dalam setiap objek dengan cara mengeksekusi satu persatu. Tentu saja tidak realistis untuk memeriksa objek dengan semua masukan yang mungkin. Salah satu cara yang dapat dilakukan adalah dengan memilih-memilih masukan yang akan diujikan.
b. Memeriksa setiap alternatif yang mungkin pada setiap objek.