

**PENGEMBANGAN *ENTERPRISE RESOURCE PLANNING*
(*PLANT MAINTENANCE*) BERBASIS *FRAMEWORK ACUMATICA*
(STUDI KASUS: MODUL *MAINTENANCE*)**

Heri Purwanto¹⁾, Santa Yusuf Nasution²⁾, Sherly Deastuti³⁾

Program Studi Teknik Informatika^{1),2),3)}

STMIK LPKIA^{1),2),3)}

heripurwanto@lpkia.ac.id¹⁾, santayusuf@gmail.com²⁾, sherlydeastuti@lpkia.ac.id³⁾

ABSTRAK

Bersamaan dengan begitu pentingnya peralatan/mesin bagi keberlangsungan suatu perusahaan terutama perusahaan industri dalam membuat produk yang sesuai dengan harapan/tujuan perusahaan, maka dibutuhkan pemeliharaan secara khusus agar mesin yang digunakan dapat berjalan dengan performa yang baik sehingga meminimalkan terjadinya kerusakan mesin yang dapat menghambat fungsi bisnis perusahaan. Pada penelitian ini menggunakan modul untuk pemeliharaan mesin yaitu *Plant Maintenance* yang merupakan salah satu modul yang terdapat dalam bagian *Enterprise Resource Planning* (ERP). Modul ini merupakan gabungan dari semua tugas-tugas teknis, administrasi dan manajemen yang dilakukan selama siklus hidup dari suatu *technical object* untuk tetap beroperasi atau untuk dapat mengembalikan ke kondisi semula sehingga peralatan/mesin dapat bekerja sebagaimana yang diharapkan. Pada dasarnya *Plant Maintenance* terdiri dari 4 tugas, yaitu *Inspection*, *maintenance*, *repair* dan *improvement*. Setelah sebelumnya dilakukan penelitian yang memfokuskan pada modul *inspection*, pada penelitian ini akan berlanjut pada modul *maintenance*. Penelitian ini menggunakan *framework* Acumatica yaitu *framework* yang dibangun diatas Microsoft .NET dan Microsoft Visual Studio IDE dan dibuat dengan menggunakan bahasa pemrograman C#. Acumatica merupakan *framework* yang dapat digunakan untuk membuat program berbasis ERP.

Kata Kunci: *Enterprise Resource Planning*, *Plant Maintenance*, *Maintenance*, *Acumatica*.

I. PENDAHULUAN

Proses bisnis suatu perusahaan dapat dibantu dengan adanya teknologi, sama halnya dengan perusahaan industri. Perusahaan ini membutuhkan perhatian khusus pada bagian produksinya. Peralatan/mesin yang digunakan untuk memproduksi setiap produk harus memiliki performa yang baik.

Salah satu sistem informasi manajemen yang dapat membantu permasalahan adalah *Enterprise Resource Planning* (ERP) yaitu sistem informasi yang mengintegrasikan dan mengotomatisasi berbagai praktek bisnis yang berelasi dengan aspek produksi atau operasi perusahaan. [1]

Pada ERP terdapat modul yang digunakan dalam proses pemeliharaan peralatan/mesin yaitu *Plant Maintenance*. Modul ini pada dasarnya terdiri dari 4 tugas, yaitu: *Inspection*, *maintenance*, *repair* dan *improvement*. [3]

Saat ini masih banyak perusahaan yang tidak memiliki perencanaan pemeliharaan mesin yang baik, sehingga sering timbulnya masalah akibat dari kerusakan mesin yang tidak secara cepat ditangani.

Penelitian ini hanya berfokus pada modul *maintenance* dan hanya mengembangkan proses *Corrective Maintenance*, yaitu proses yang menggambarkan adanya kebutuhan pemeliharaan suatu mesin yang disebabkan oleh menurunnya performa mesin tersebut.

Penelitian ini diharapkan dapat membantu mengurangi permasalahan yang sering terjadi, akibat dari:

1. Proses Pemeliharaan yang sering terlambat yang mengganggu kegiatan produksi.
2. Tidak adanya pengecekan fisik mesin secara rutin.
3. Kuantitas dan kualitas produk menurun.

4. Meningkatnya kerugian biaya perawatan mesin terhadap mesin yang rusak.
5. Terjadi kecelakaan kerja akibat dari kerusakan mesin.

Batasan permasalahan dari sistem yang dikembangkan adalah:

1. Sistem yang dikembangkan berfokus pada *Corrective Maintenance*.
2. Modul *Plant Maintenance* yang digunakan hanya berfokus pada area *Maintenance*.
3. Aplikasi dibuat dengan menggunakan *Acumatica Framework* dan *database SqlServer2014*.

Dari identifikasi permasalahan diatas, maka tujuan yang ingin diraih adalah:

1. Proses pemeliharaan dapat dilakukan secara cepat dan tepat.
2. Mesin dapat digunakan dalam jangka waktu yang lebih lama.
3. Meningkatkan kuantitas dan kualitas produk.
4. Memperkecil biaya perawatan.
5. Meningkatkan kemanana operasi.

II. DASAR TEORI

2.1. Enterprise Resource Planning (ERP)

Menurut (Alexis Leon, 2014) mendefinisikan *Enterprise Resource Planning* sebagai berikut ini.

Enterprise Resource Planning is a techniques and concepts for integrated management of businesses or management of business as a whole with the objective of efficient and effective use of management resources and to improve the efficiency of enterprise management.

Berdasarkan pengertian diatas, maka dapat disimpulkan bahwa ERP adalah teknik dan konsep pengelolaan terpadu bisnis / manajemen bisnis secara keseluruhan dengan tujuan penggunaan sumber daya manajemen secara efisien dan efektif dan untuk meningkatkan efisiensi pengelolaan perusahaan.

Keuntungan Penggunaan ERP

Adapun keuntungan ERP secara langsung:

1. Integrasi Bisnis

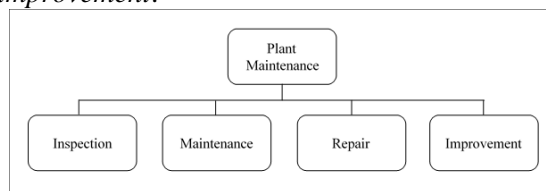
2. Fleksibel
3. Analisis yang lebih baik dan kapabilitas perencanaan.
4. Penggunaan teknologi terbaru.

2.2. Plant Maintenance

Menurut (Karl Liebstückel, 2011) mendefinisikan *Plant Maintenance* sebagai berikut ini.

Plant maintenance is a combination of all technical, administrative, and management tasks carried out during the lifecycle of a technical object to keep it operational or to return it to this condition so it can work as intended. it essentially comprises the following four tasks: inspection, maintenance, repair, and improvement.

Berdasarkan pengertian diatas, maka dapat disimpulkan bahwa *Plant Maintenance* adalah gabungan dari semua tugas teknis, administrasi, dan manajemen yang dilakukan selama siklus hidup dari suatu technical object untuk tetap beroperasi / untuk mengembalikannya ke kondisi semula sehingga dapat bekerja sebagaimana yang diharapkan. Pada dasarnya terdiri dari 4 tugas, yaitu: *Inspection, maintenance, repair, dan improvement.*



Gambar 1. *New Maintenance Terminology*
(Sumber: Karl Liebstückel, 2011)

2.3. Maintenance

Pada penelitian ini Terminologi pemeliharaan baru yang digunakan hanya modul *maintenance*.

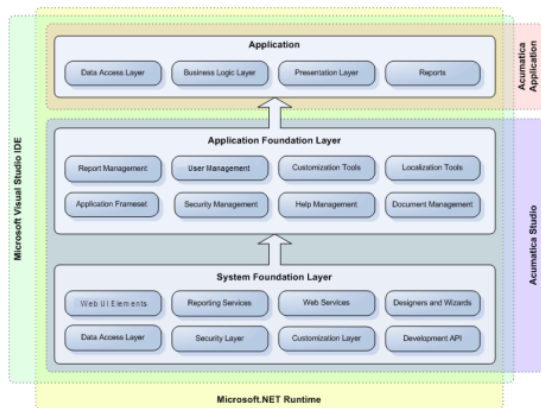
Maintenance sebagai semua tugas untuk menunda pengurangan dari penggunaan alternatif. Secara khusus, *maintenance* terdiri dari beberapa tugas, diantaranya:

1. *Visually inspect* (inspeksi secara visual)
2. *Adjust* (menyesuaikan)
3. *Replace* (menggantikan)
4. *Supplement* (melengkapi)
5. *Lubricate* (memberikan pelumas)
6. *Preserve* (mempertahankan)
7. *Clean* (membersihkan)

8. *Function test* (pemeriksaan fungsi)

2.4. Acumatica Framework

Acumatica *Framework* adalah sebuah platform pengembangan aplikasi web modern yang dirancang untuk mengembangkan aplikasi bisnis. Acumatica Framework dibangun diatas Microsoft.NET dan Microsoft visual studio IDE technologies, acumatica framework dirancang dan dibuat dengan menggunakan bahasa pemograman C#. [8]



Gambar 2. Acumatica Framework Component
(Sumber: Acumatica, Inc., 2013)

System Foundation Layer

System foundation layer adalah pengatur untuk komponen inti dan primitif dengan fungsionalitas yang di perlukan untuk mengembangkan dan menjalankan aplikasi berbasis Acumatica *Framework*.

Application layer foundation

Application layer foundation mengatur *application building blocks* dan mengimplementasikan struktur *database* berada di atas komponen *system foundation layer*.

Data Access Layer

Data Access Layer diimplementasikan sebagai pengatur *data access classes* yang membungkus data dari tabel *database* atau data yang diterima dari sumber eksternal lainnya.

Business Logic Layer

Business Logic adalah pengimplementasian *business logic controller*.

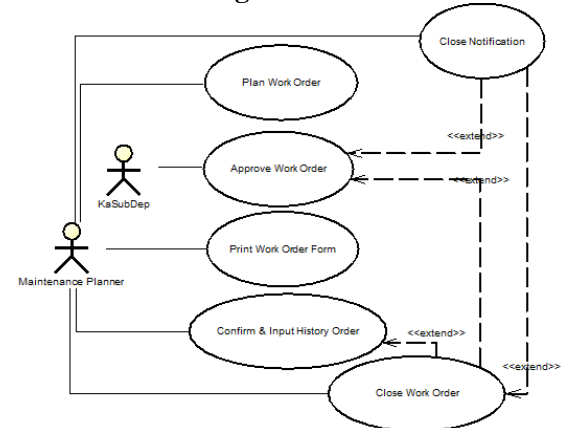
Presentation Layer

Presentation layer bertanggung jawab untuk menyediakan *access* ke *business logic* sebuah aplikasi melalui GUI. Terdiri dari pengaturan pendeklarasian *web forms* untuk *business logic controller* tertentu.

III. ANALISIS DAN PERANCANGAN

3.1. Aliran Proses

3.1.1. Usecase Diagram



Gambar 3. Usecase Diagram Aplikasi *Plant Maintenance* (Modul *Maintenance*)

3.1.2. Usecase Scenario

Tabel 1. Usecase Scenario *Plan Work Oder*

<i>Usecase Scenario Plan Work Oder</i>	
Uraian Singkat	Merencanakan <i>Work Order</i>
Actor	<i>Maintenance Planner</i>
Trigger Event	
PreCondition	<i>Maintenance Planner</i> memilih fungsi <i>Plan Work Order</i>
PostCondition	Sistem menyimpan data <i>Work Order</i> ke <i>database</i>
Priority	
SCENARIO	
1. <i>Maintenance Planner</i> memilih fungsi <i>Plan Work Order</i> .	
2. Sistem menampilkan <i>form Work Order</i> .	
3. <i>Maintenance Planner</i> memilih nomor <i>Work Order</i> yang akan dilakukan perencanaan.	
4. Sistem menampilkan data <i>Work Order</i> .	
5. <i>Maintenance Planner</i> memasukkan dan menyimpan data <i>Work Center, Activity Type</i> dan <i>Duration</i> .	
6. <i>Maintenance Planner</i> memasukkan dan menyimpan data alokasi aktivitas <i>external</i> .	
7. <i>Maintenance Planner</i> memasukkan dan menyimpan data <i>sparepart</i> dan jumlah yang diperlukan.	
8. <i>Maintenance Planner</i> melakukan penjadwalan <i>work order</i> yang direncanakan dan menyimpan data tersebut ke <i>database</i> .	
9. <i>Maintenance Planner</i> mengeset status <i>Work Order</i> menjadi <i>Order Planned</i> , kemudian menyimpannya ke <i>database</i> .	

10. Sistem memvalidasi data yang dimasukkan <i>Maintenance Planner</i> .
11. Sistem menyimpan data <i>Work Order</i> ke <i>database</i> .
ALTERNATIVES Jika terjadi kesalahan pada saat <i>Maintenance Planner</i> memasukan data, maka sistem akan menampilkan pesan <i>error</i> / tidak valid dan <i>Maintenance Planner</i> harus kembali memasukkan data yang valid.
EXTENSION
NOTE

Tabel 2. Usecase Scenario Approve Work Order

Usecase Scenario Approve Work Order	
Uraian Singkat	Menyetujui <i>Work Order</i>
Actor	KaSubDep
Trigger Event	
PreCondition	KaSubDep memilih fungsi <i>Approve Work Order</i>
PostCondition	Sistem menyimpan data <i>Work Order</i> ke <i>database</i>
Priority	
SCENARIO 1. KaSubDep memilih fungsi <i>Approve Work Order</i> . 2. Sistem menampilkan <i>form Work Order</i> . 3. KaSubDep memilih data <i>Work Order</i> yang akan dilakukan persetujuan. 4. Sistem menampilkan data <i>Work Order</i> berdasarkan yang dipilih KasubDep. 5. KaSubDep memilih <i>User Status</i> “WAPR” (<i>Approve to execute</i>) / “WRJT” (<i>Reject</i>) / “WPSP” (<i>Postpone</i>). 6. KaSubDep menyimpan data <i>Work Order</i> . 7. Sistem memvalidasi data yang dimasukkan KaSubDep. 8. Sistem menyimpan data <i>Work Order</i> ke <i>database</i> .	
ALTERNATIVES Jika terjadi kesalahan pada saat KaSubDep melakukan proses <i>Approve Work Order</i> , maka sistem akan menampilkan pesan <i>error</i> / tidak valid dan KaSubDep harus kembali memasukkan data yang valid.	
EXTENSION	
NOTE	

Tabel 3. Usecase Scenario Print Work Order Form

Usecase Scenario Print Work Order Form	
Uraian Singkat	Mencetak <i>form Work Order</i> untuk Teknisi
Actor	<i>Maintenance Planner</i>
Trigger Event	
PreCondition	<i>Maintenance Planner</i> memilih

	fungsi <i>Print Work Order Form</i>
PostCondition	Printer mencetak <i>form Work Order</i>
Priority	
SCENARIO 1. <i>Maintenance Planner</i> memilih fungsi <i>Print Work Order Form</i> . 2. Sistem menampilkan data <i>Work Order</i> . 3. <i>Maintenance Planner</i> memilih data <i>Work Order</i> yang akan dicetak. 4. <i>Maintenance Planner</i> memilih aksi cetak. 5. Sistem memvalidasi proses yang dilakukan <i>Maintenance Planner</i> . 6. Sistem mengeset <i>System Status</i> menjadi “PRT” (<i>Printed</i>). 7. Sistem mengeset <i>User Status</i> menjadi “WEXE” (<i>Order being executed</i>). 8. Printer mencetak <i>form Work Order</i> .	
ALTERNATIVES Jika terjadi kesalahan pada saat <i>Maintenance Planner</i> melakukan proses <i>Print Form Work Order</i> , maka sistem akan menampilkan pesan <i>error</i> / tidak valid dan <i>Maintenance Planner</i> harus kembali melakukan proses yang valid.	
EXTENSION	
NOTE	

Tabel 4. Usecase Scenario Confirm & Input History Order

Usecase Scenario Confirm & Input History Order	
Uraian Singkat	Melakukan konfirmasi dan input histori pemeliharaan dari data yang dikirim teknisi
Actor	<i>Maintenance Planner</i>
Trigger Event	<i>Form Work Order</i> dari teknisi
PreCondition	<i>Maintenance Planner</i> memilih <i>Confirm & Input History Order</i>
PostCondition	Sistem menyimpan data <i>Work Order</i> ke <i>database</i>
Priority	
SCENARIO 1. <i>Maintenance Planner</i> memilih fungsi <i>Confirm & Input History Order</i> . 2. Sistem menampilkan <i>form Work Order</i> . 3. <i>Maintenance Planner</i> memilih data <i>Work Order</i> yang sesuai dengan <i>form</i> yang diterima dari teknisi. 4. Sistem menampilkan data <i>Work Order</i> . 5. <i>Maintenance Planner</i> melakukan konfirmasi terhadap waktu aktual teknisi saat melakukan penanganan pemeliharaan. 6. <i>Maintenance Planner</i> melengkapi data <i>Work Order</i> dengan menginput histori kerusakan berdasarkan <i>formulir Work Order</i> dari Teknisi. 7. <i>Maintenance Planner</i> menyimpan data. 8. Sistem memvalidasi proses yang dilakukan <i>Maintenance Planner</i> .	

9. Sistem mengeset <i>System Status</i> menjadi “TECO” (<i>Technically Completed</i>).
10. Sistem menyimpan data <i>Work Order</i> ke <i>database</i> .
ALTERNATIVES Jika terjadi kesalahan pada saat <i>Maintenance Planner</i> melakukan proses <i>Confirm & Input History Order</i> , maka sistem akan menampilkan pesan <i>error</i> / tidak valid dan <i>Maintenance Planner</i> harus kembali memasukkan data yang valid.
EXTENSION
NOTE

Tabel 5. Usecase Scenario Close Work Order

Usecase Scenario Close Work Order	
Uraian Singkat	Menutup Work Order
Actor	Maintenance Planner
Trigger Event	Approve Work Order, Confirm & Input History Order
PreCondition	Maintenance Planner memilih fungsi Close Work Order
PostCondition	Sistem menyimpan data Work Order ke database
Priority	
SCENARIO 1. <i>Maintenance Planner</i> memilih fungsi <i>Close Work Order</i> . 2. Sistem menampilkan <i>form Work Order</i> . 3. <i>Maintenance Planner</i> memilih data <i>Work Order</i> yang akan diselesaikan. 4. Sistem menampilkan data <i>Work Order</i> . 5. <i>Maintenance Planner</i> memilih <i>User Status</i> “WCOMP” (<i>Completed</i>) <i>Work Order</i> . 6. <i>Maintenance Planner</i> menyimpan data <i>Work Order</i> . 7. Sistem memvalidasi proses yang dilakukan <i>Maintenance Planner</i> . 8. Sistem mengeset <i>system status Notification</i> dengan nomor <i>Work Order</i> yang sama menjadi “NOCO ORAS” (<i>Notification completed and Order assigned</i>). 9. Sistem menyimpan data <i>Work Order</i> ke <i>database</i> .	
ALTERNATIVES Jika terjadi kesalahan pada saat <i>Maintenance Planner</i> melakukan proses <i>Complete Work Order</i> , maka sistem akan menampilkan pesan <i>error</i> / tidak valid dan <i>Maintenance Planner</i> harus kembali memasukkan data yang valid.	
EXTENSION	
NOTE	

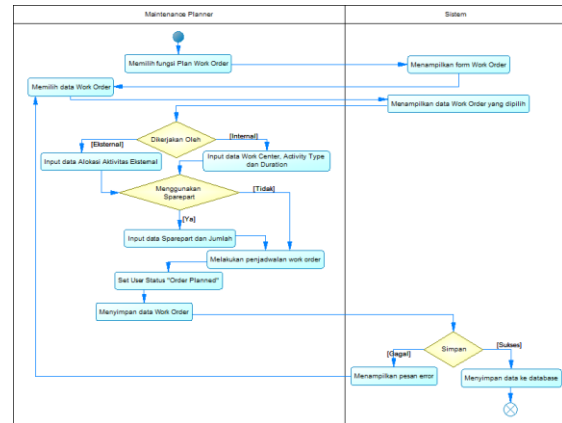
Tabel 6. Usecase Scenario Close Notification

Usecase Scenario Close Notification	
Uraian Singkat	Menutup Notifikasi
Actor	Maintenance Planner
Trigger Event	

PreCondition	Maintenance Planner memilih fungsi Close Notification
PostCondition	Sistem menyimpan data Notification ke database.
Priority	
SCENARIO 1. <i>Maintenance Planner</i> memilih fungsi <i>Close Notification</i> . 2. Sistem menampilkan <i>form Notification</i> . 3. <i>Maintenance Planner</i> memilih data <i>Notification</i> yang akan ditutup. 4. Sistem menampilkan data <i>Notification</i> yang dipilih. 5. <i>Maintenance Planner</i> memilih <i>User Status</i> “Close” dan menyimpan ke <i>database</i> . 6. Sistem mengeset <i>System Status</i> menjadi “NOCO” (<i>Notification Completed</i>). 7. Sistem memvalidasi data yang dimasukkan <i>Maintenance Planner</i> . 8. Sistem menyimpan data <i>Notification</i> ke <i>database</i> .	
ALTERNATIVES Jika terjadi kesalahan pada saat <i>Maintenance Planner</i> memasukan data, maka sistem akan menampilkan pesan <i>error</i> / tidak valid dan <i>Maintenance Planner</i> harus kembali memasukkan data yang valid.	
EXTENSION	
NOTE	

3.2. Aliran Kerja (Activity Diagram)

III.2.1. Plan Work Order



Gambar 4. Activity Diagram Plan Work Order

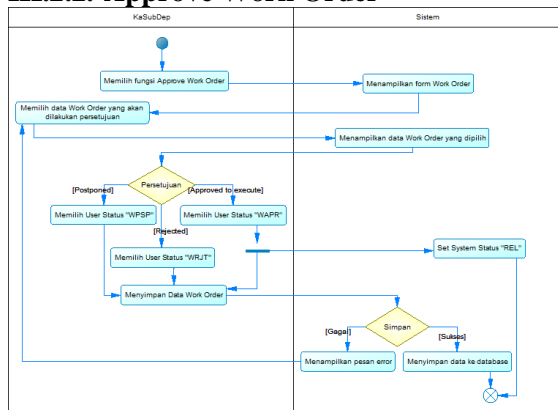
Uraian Workflow:

1. Dalam gambar 4 dijelaskan bagaimana melakukan perencanaan *work order*.
2. Langkah pertama, *Maintenance Planner* memilih fungsi *Plan Work Order*.
3. Sistem akan menampilkan *form Work Order*.
4. *Maintenance Planner* memilih data *Work Order* yang akan dilakukan perencanaan.

5. Sistem menampilkan data *Work Order* yang dipilih.
6. *Maintenance Planner* menginput data Alokasi Aktivitas Eksternal apabila pemeliharaan dilakukan oleh pihak eksternal atau menginput data *Work Center, Activity Type* dan *Duration* apabila pemeliharaan dilakukan oleh pihak internal.
7. *Maintenance Planner* menginput data *sparepart* dan jumlahnya, serta melakukan penjadwalan *work order* apabila pemeliharaan menggunakan *sparepart* atau melakukan penjadwalan *work order* saja apabila pemeliharaan tidak menggunakan *sparepart*.
8. *Maintenance Planner* mengeset *user status* menjadi "*Order Planned*" dan menyimpan data *work order*.
9. Sistem akan mengecek data yang diinput, apabila data yang diinput valid dan dinyatakan sukses maka sistem akan menyimpan data tersebut ke *database*. Jika gagal maka sistem akan menampilkan pesan *error* dan *Maintenance Planner* harus mengulang proses input data yang valid.

6. KaSubDep memilih user status "WSPSP" apabila menunda (postponed) persetujuan, memilih user status "WRJT" apabila menolak (rejected) persetujuan, atau memilih user status "WAPR" apabila menyetujui (approved to execute) persetujuan yang secara otomatis sistem mengeset system status menjadi "REL".
7. KaSubDep menyimpan data work order.
8. Sistem akan mengecek data yang diinput, apabila data yang diinput valid dan dinyatakan sukses maka sistem akan menyimpan data tersebut ke *database*. Jika gagal maka sistem akan menampilkan pesan error dan KaSubDep harus mengulang proses input data yang valid.

III.2.2. Approve Work Order

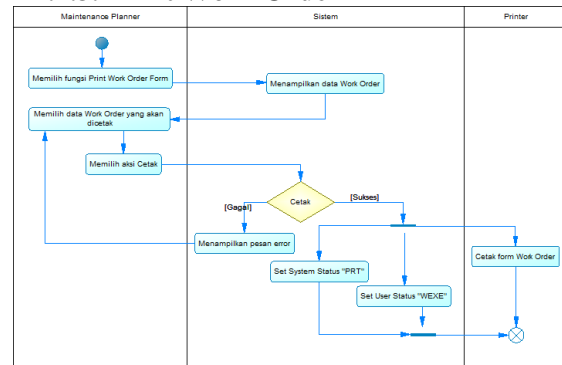


Gambar 5. Activity Diagram Approve Work Order

Uraian Workflow:

1. Dalam gambar 5 dijelaskan bagaimana melakukan persetujuan work order.
2. Langkah pertama, KaSubDep memilih fungsi Approve Work Order.
3. Sistem menampilkan *form* Work Order.
4. KaSubDep memilih data Work Order yang akan dilakukan persetujuan.
5. Sistem menampilkan data Work Order yang dipilih.

III.2.3. Print Work Order

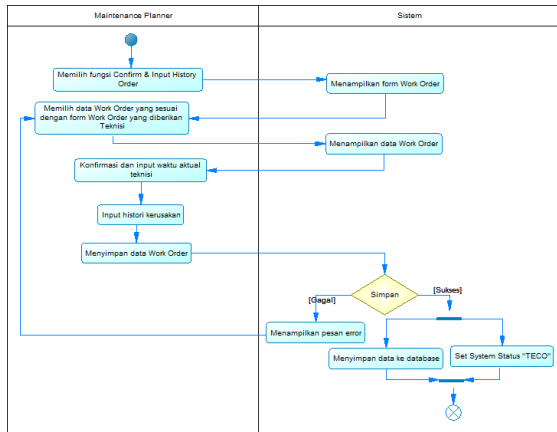


Gambar 6. Activity Diagram Print Work Order

Uraian Workflow:

1. Dalam gambar 6 dijelaskan bagaimana melakukan proses cetak *work order*.
2. Langkah pertama, *Maintenance Planner* memilih fungsi *Print Work Order*.
3. Sistem akan menampilkan data *Work Order*.
4. *Maintenance Planner* memilih data *Work Order* yang akan dicetak dan memilih aksi cetak.
5. Sistem akan mengecek data yang diinput, apabila data yang diinput valid.

III.2.4. Confirm & Input History Order

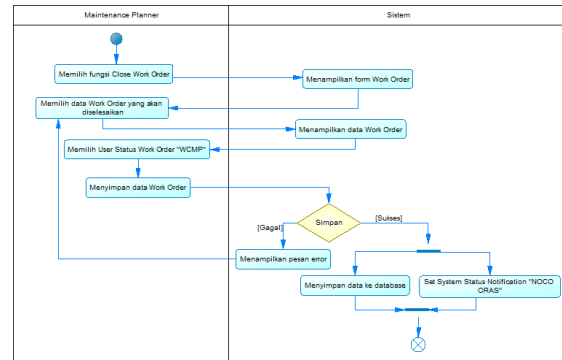


Gambar 7. Activity Diagram Confirm & Input History Order

Uraian Workflow:

1. Dalam gambar 7 dijelaskan bagaimana melakukan konfirmasi dan input *history order*.
2. Langkah pertama, *Maintenance Planner* memilih fungsi *Confirm & Input History Order*.
3. Sistem akan menampilkan *form Work Order*.
4. *Maintenance Planner* memilih data *Work Order* yang sesuai dengan *formulir work order* yang diberikan oleh teknisi.
5. Sistem menampilkan data *Work Order* yang dipilih.
6. *Maintenance Planner* melakukan konfirmasi dan input waktu aktual teknisi, input *history* kerusakan dan menyimpan data yang diinput tersebut.
7. Sistem akan mengecek data yang diinput, apabila data yang diinput valid dan dinyatakan sukses maka sistem akan menyimpan data tersebut ke *database*. Jika gagal maka sistem akan menampilkan pesan *error* dan *Maintenance Planner* harus mengulang proses input data yang valid.

III.2.5. Close Work Order

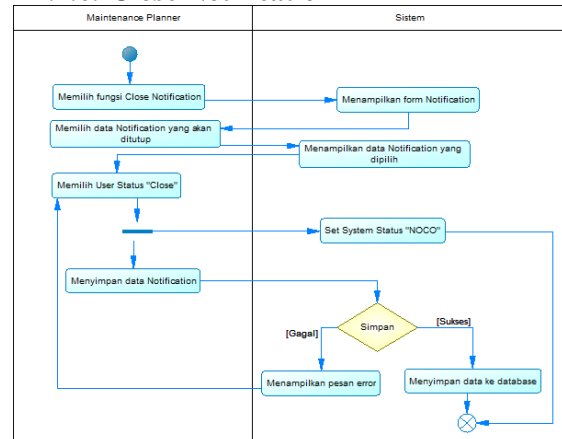


Gambar 8. Activity Diagram Close Work Order

Uraian Workflow:

1. Dalam gambar 8 dijelaskan bagaimana cara melakukan penutupan *work order*.
2. Langkah pertama, *Maintenance Planner* memilih fungsi *Close Work Order*.
3. Sistem akan menampilkan *form Work Order*.
4. *Maintenance Planner* memilih data *Work Order* yang akan diselesaikan.
5. Sistem menampilkan data *Work Order* yang dipilih.
6. *Maintenance Planner* memilih *user status work order* “WCMP” dan menyimpan data yang diinput tersebut.
7. Sistem akan mengecek data yang diinput, apabila data yang diinput valid dan dinyatakan sukses maka sistem akan menyimpan data tersebut ke *database* dan secara otomatis akan mengeset *system status notification* menjadi “NOCO ORAS”. Jika gagal maka sistem akan menampilkan pesan *error* dan *Maintenance Planner* harus mengulang proses input data yang valid.

III.2.6. Close Notification



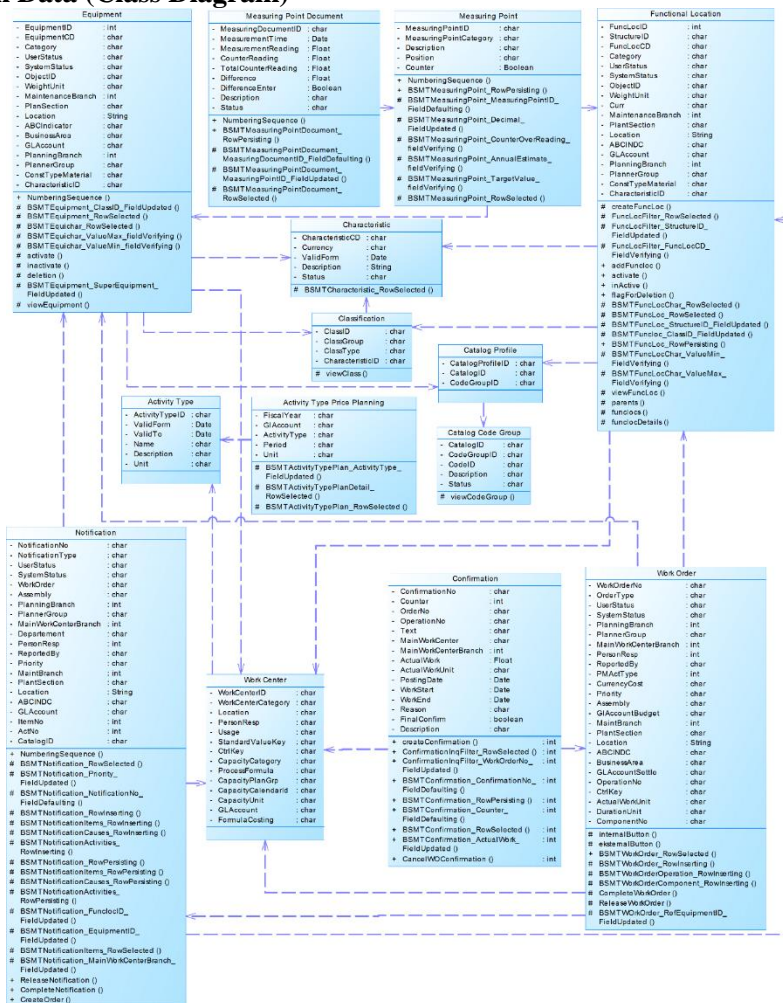
Gambar 9. Activity Diagram Close Notification

Uraian Workflow:

1. Dalam gambar 9 dijelaskan bagaimana cara melakukan penutupan *notification*.
2. Pertama, *Maintenance Planner* memilih fungsi *Close Notification*.
3. Sistem akan menampilkan *form Notification*.
4. *Maintenance Planner* memilih data *Notification* yang akan ditutup.
5. Sistem menampilkan data *Notification* yang dipilih.

6. *Maintenance Planner* memilih *user status* “Close” yang secara otomatis sistem akan mengeset *system status* menjadi “NOCO”.
7. *Maintenance Planner* menyimpan data *Notification*.
8. Sistem akan mengecek data yang diinput, apabila data yang diinput valid dan dinyatakan sukses maka sistem akan menyimpan data tersebut ke *database*. Jika gagal maka sistem akan menampilkan pesan *error* dan *Maintenance Planner* harus mengulang proses input data yang valid.

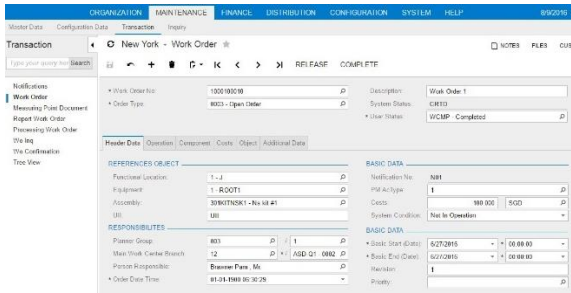
3.3. Pemodelan Data (Class Diagram)



Gambar 10. Class Diagram Plant Maintenance

IV. IMPLEMENTASI

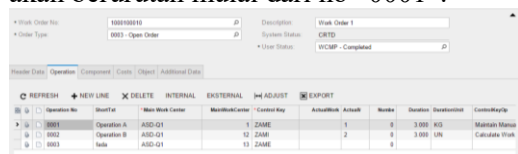
IV.1. Implementasi Antarmuka Work Order



Gambar 11. Dialog Screen Work Order

Penjelasan Gambar 11 *Dialog Screen Work Order*

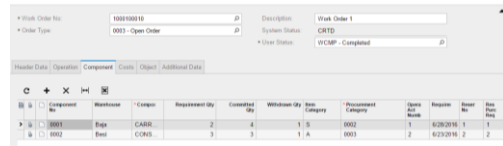
1. Halaman ini akan muncul saat pengguna memilih *Work Order* pada menu *Maintenance* → *Transaction* → *Manage*.
2. Pilih *field* *Work Order No* yang sudah tersedia berdasarkan hasil *generate* dari *Notification*.
3. Input data pada *field* yang harus diisi.
4. Pada saat akan memilih *User Status*, pilih terlebih dahulu *Order Type* karena *value* dari *User Status* yang muncul berdasarkan dari konfigurasi *Order Type*.
5. Jika memilih data *Equipment* terlebih dulu, maka data *Functional Location* otomatis muncul sesuai dengan *Equipment* yang di pilih. Tetapi jika memilih data *Functional Location* terlebih dulu, maka isi *selector* *Equipment* akan mengacu kepada data *Functional Location* yang dipilih.
6. Pada tab *Operation* akan muncul *grid*. Isi *grid* tersebut dengan cara klik tombol **New Line**. *Field* *Operation No* akan muncul secara otomatis dari sistem yang akan berurutan mulai dari no “0001”.



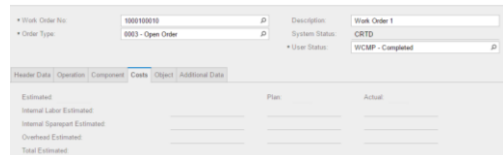
Terdapat 2 *button* *INTERNAL* dan *EKSTERNAL*, *button* tersebut berfungsi untuk memasukkan data suatu *operation* apakah di kerjakan oleh pihak internal atau eksternal.

7. Pada tab *Component* berfungsi untuk mengisi komponen apa saja yang digunakan sebagai komponen dalam pengajuan *Work Order*. Cara menggunakan tab ini dengan cara mengklik tombol **Add Row** (+) untuk menambahkan komponen baru, *field* *Component No* akan secara otomatis

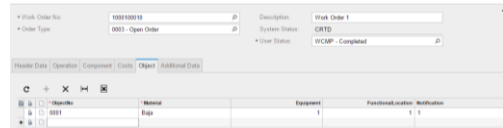
muncul dengan nomor dimulai dengan no “0001” dan seterusnya.



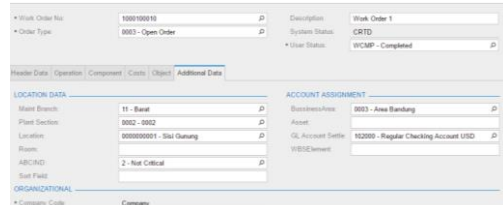
8. Tab *Cost* berfungsi untuk menampilkan hasil dari *estimated plan* dan *actual* suatu *Work Order*. *Field* tersebut terisi secara otomatis.



9. Tab *Object* digunakan untuk mengisi objek apa saja yang terlibat di dalam suatu *Work Order*.



10. Tab *Additional Data* berfungsi untuk menginputkan *additional data* di dalam *Work Order*.



11. Untuk menyimpan data *Work Order* pilih tombol **Save** (S).
12. Untuk mengedit data *Work Order*, pilih *selector* *Work Order No*, cari dan pilih data yang akan di edit, maka data akan otomatis muncul pada *screen* ini. Setelah mengedit data, pilih tombol **Save** (S) agar data yang di edit tersimpan di *database*.
13. Untuk menghapus data *Work Order*, pilih *selector* *Work Order No*, cari dan pilih data yang akan di hapus, maka data akan otomatis muncul pada *screen* ini. Kemudian pilih tombol **Delete** (D).

Confirmation



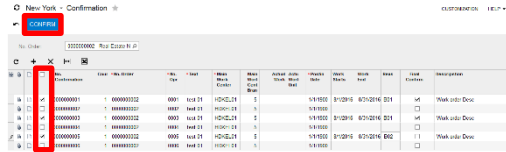
Gambar 12. Dialog Screen Confirmation (Operator Inquiry)

agar data yang di edit tersimpan di database.

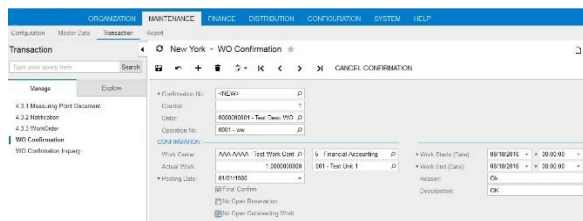
Penjelasan Gambar 12 *Dialog Screen Confirmation (Operation Inquiry)*

1. Halaman ini akan muncul saat pengguna memilih *Confirmation Inquiry* pada menu *Maintenance* → *Transaction* → *Manage*.
2. Pilih No order dari *Selector*

No. Order: , data yang tampil pada No. Order *selector* hanya order nomor yang memiliki *operation* dengan *control key internal processing*.



3. Data yang telah di konfirmasi tidak akan muncul kembali di dalam *grid*.



Gambar 13. Dialog Screen Confirmation (Work Order Confirmation)

V. PENUTUP

Setelah melakukan penelitian terhadap permasalahan yang ada, maka dapat disimpulkan bahwa penerapan hasil penelitian tim pengembang mengenai Plant Maintenance ini sangat akurat untuk digunakan di perusahaan, khususnya untuk perusahaan industri yang memiliki mesin-mesin yang harus dilakukan pemeliharaan dengan baik.

Dengan adanya perbaikan dalam proses pemeliharaan mesin dan dengan menggunakan teknologi berbasis cloud computing yang telah disesuaikan, maka akan sangat bermanfaat antara lain:

1. Proses pemeliharaan mesin menjadi semakin cepat dan teratur.
2. Mesin tidak mudah rusak dan dapat digunakan dalam jangka waktu yang lebih lama karena adanya pemeliharaan yang terencana.
3. Meningkatnya kuantitas dan kualitas produk perusahaan.
4. Biaya perawatan mesin menjadi lebih kecil karena kemungkinan kerusakan mesin lebih kecil selama mesin tersebut terpelihara dengan baik.
5. Meningkatnya keamanan operasi dari kesalahan yang terjadi akibat dari kerusakan mesin.

Penjelasan Gambar 13 *Dialog Screen Confirmation (Work Order Confirmation)*

1. Halaman ini akan muncul saat pengguna memilih *Confirmation* pada menu *Maintenance* → *Transaction* → *Manage*.
2. Isi data *Confirmation* pada setiap *field*.
3. Untuk membatalkan konfirmasi, maka pilih tombol **CANCEL CONFIRMATION**. Dengan menekan tombol tersebut pada satu data *confirmation*, maka *field* Counter akan ditambah 1 dan *field* Actual Work Confirmation tersebut akan memiliki nilai minus (-).
4. Untuk menyimpan data *Confirmation* pilih tombol *Save* (💾).
5. Untuk mengedit data *Confirmation*, pilih *selector* Confirmation No, cari dan pilih data yang akan di edit, maka data akan otomatis muncul pada *screen* ini. Setelah mengedit data, pilih tombol *Save* (💾)

Sehubungan dengan hasil penelitian yang telah dilakukan oleh tim pengembang, maka saran-saran yang dapat disampaikan untuk pengembangan lebih lanjut terhadap sistem *Plant Maintenance* ini adalah sebagai berikut:

Sistem dikembangkan lebih lanjut ke modul *repair* dan *improvement*.

DAFTAR PUSTAKA

[1] W. Siswoutomo, Panduan Karir di Dunia Teknologi Informasi, Jakarta: Elex Media Komputindo, 2005.
[2] A. Leon, Enterprise Resource Planning, New Delhi: McGraw Hill Education (India) Private Limited, 2014.
[3] K. Liebstückel, Plant Maintenance with SAP, Boston: Galileo Press, 2011.
[4] A. Nugroho, Perancangan dan

- Implementasi Sistem Basis Data, vol. 1, Yogyakarta: Andi, 2011.
- [5] M. Fowler, UML Distilled, Yogyakarta: Andi, 2005.
- [6] A. Nugroho, Algoritma dan Struktur Data dengan C#, Yogyakarta: Andi, 2009.
- [7] P. K. Raouf A. Mata-Toledo, Schaum's Outline Dasar-dasar Database Relational, Jakarta: Erlangga, 2007.
- [8] Acumatica Inc., "Developer Guide Acumatica Framework 5.0," *Development Guide*, pp. 7-18, 2013.
- [9] B. A. Masood-Al-Farooq, SQL Server 2014 Development Essentials, Birmingham: Packt Publishing, 2014.
- [10] A. Nugroho, Rekayasa Perangkat Lunak Berorientasi Objek dengan Metode USDP, vol. 1, Yogyakarta: Andi, 2010.
- [11] Heri Purwanto, Santa YN, Sherly D Pengembangan ERP (Plant Maintenance) Berbasis Framework Accumatica (Studi Kasus Modul Inspection), STMIK LPKIA, Bandung, 2016
- [12] R. S. Pressman, Rekayasa Perangkat Lunak, vol. 7, Yogyakarta: Andi, 2012.
- [13] G. D. Gupta Mukul, Research Methodology, New Delhi: PHI Learning Pvt., 2011.