

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

**Heri Purwanto<sup>1</sup>, Santa Yusuf Nasution<sup>2</sup>, Sherly Deastuti<sup>3</sup>**

Program Studi Teknik Informatika<sup>1,2,3</sup>

STMIK LPKIA<sup>1,2,3</sup>

[heripurwanto@lpkia.ac.id](mailto:heripurwanto@lpkia.ac.id)<sup>1</sup>, [santayusuf@gmail.com](mailto:santayusuf@gmail.com)<sup>2</sup>, [sherlydeastuti@lpkia.ac.id](mailto:sherlydeastuti@lpkia.ac.id)<sup>3</sup>

## **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*, akan tetapi pada penelitian ini hanya meliputi modul *inspection* saja. 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, Inspection, 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 *inspection* 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 *Inspection*.
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 atau 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, diantaranya:

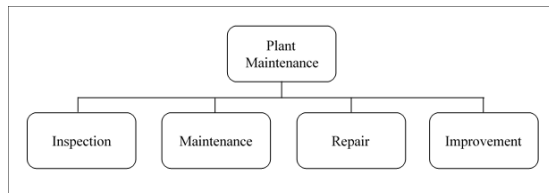
1. Integrasi Bisnis
2. Fleksibel
3. Analisis yang lebih baik dan kapabilitas perencanaan.
4. Penggunaan teknologi terbaru.

## 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. Inspection

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

Untuk memastikan tingginya ketersediaan dan keamanan operasi mesin, sistem teknis, dan peralatan, diperlukan adanya pemeriksaan secara rutin yang dapat menentukan kondisi teknis dari suatu peralatan dan untuk menentukan tugas-tugas pemeliharaan yang diperlukan.

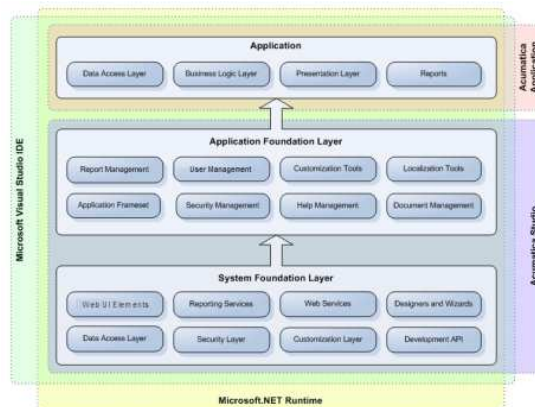
*Inspection* (Inspeksi) terdiri dari beberapa tugas, diantaranya:

1. *Check* (memeriksa)
2. *Measure* (mengukur)
3. *Observe* (mengamati)
4. *Assess* (menilai)

5. *Deduce consequences* (menyimpulkan dampak-dampaknya)

## 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 pemrograman 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*.

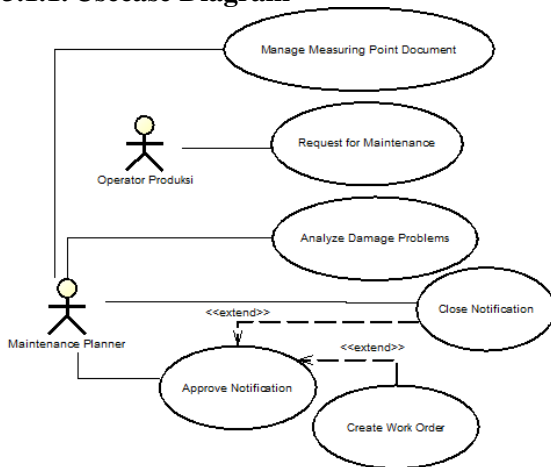
### Presentation Layer

*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 *Inspection*)

##### 3.1.2. Usecase Scenario

Tabel 1. *Usecase Scenario Manage Measuring Point Document*

| <b><i>Usecase Scenario Manage Measuring Point Document</i></b>                                |  |
|---|--|
| <i>Uraian Singkat</i>   | Mengelola data <i>Measuring Point Document</i>                                   |
| <i>Actor</i>  | <i>Maintenance Planner</i>   |
| <i>Trigger Event</i>  |  |
| <i>PreCondition</i>   | <i>Maintenance Planner</i> memilih fungsi <i>Manage Measuring Point Document</i> |
| <i>PostCondition</i>  | Sistem menyimpan data <i>Measuring Point Document</i> ke <i>database</i>         |
| <i>Priority</i>   |  |
| <b><i>SCENARIO</i></b>  |  |
| 1. <i>Maintenance Planner</i> memilih fungsi <i>Manage Measuring Point Document</i> .         |  |
| 2. Sistem menampilkan <i>form Measuring Point Document</i> .                                  |  |
| 3. <i>Maintenance Planner</i> memasukkan dan menyimpan data <i>Measuring Point Document</i> . |  |
| 4. Sistem memvalidasi data yang dimasukkan <i>Maintenance Planner</i> .                       |  |
| 5. Sistem menyimpan data <i>Measuring Point Document</i> ke <i>database</i> .                 |  |

|   |
|---|
| <i>ALTERNATIVES</i><br>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. |
| <i>EXTENSION</i>  |
| <i>NOTE</i>   |

Tabel 2. *Usecase Scenario Request for Maintenance*

| <b><i>Usecase Scenario Request for Maintenance</i></b>   |   |
|--|---|
| Uraian Singkat   | Membuat permintaan pemeliharaan                                 |
| <i>Actor</i>   | Operator Produksi   |
| <i>Trigger Event</i>   |   |
| <i>PreCondition</i>  | Operator produksi memilih fungsi <i>Request for Maintenance</i> |
| <i>PostCondition</i>   | Sistem menyimpan data <i>Notification</i> ke <i>database</i>    |
| <i>Priority</i>  |   |
| <i>SCENARIO</i>  |   |
| <ol style="list-style-type: none"> <li>1. Operator produksi memilih fungsi <i>Request for Maintenance</i>.</li> <li>2. Sistem menampilkan <i>form Notification</i>.</li> <li>3. Sistem mengeset <i>System Status</i> menjadi "OSNO" (<i>Oustanding Notification</i>).</li> <li>4. Sistem mengeset <i>User Status</i> menjadi "NNCR" (<i>Newly Created</i>).</li> <li>5. Operator produksi memasukkan dan menyimpan data mesin produksi/sarana/utilitas yang dirasa adanya kerusakan/ketidaknormalan pada <i>form Notification</i>.</li> <li>6. Sistem memvalidasi data yang dimasukkan Operator produksi.</li> <li>7. Sistem menyimpan data <i>Notification</i> ke <i>database</i>.</li> </ol> |   |
| <i>ALTERNATIVES</i><br>Jika terjadi kesalahan pada saat Operator produksi memasukan data, maka sistem akan menampilkan pesan <i>error</i> / tidak valid dan Operator produksi harus kembali memasukkan data yang valid.  |   |
| <i>EXTENSION</i>   |   |
| <i>NOTE</i>  |   |

Tabel 3. *Usecase Scenario Analyze Damage Problems*

| <b><i>Usecase Scenario Analyze Damage Problems</i></b>   |  |
|--|--|
| <b>Uraian Singkat</b>  | Menganalisis masalah-masalah kerusakan                                   |
| <b>Actor</b>   | <i>Maintenance Planner</i>   |
| <b>Trigger Event</b>   |  |
| <b>PreCondition</b>  | <i>Maintenance Planner</i> memilih fungsi <i>Analyze Damage Problems</i> |
| <b>PostCondition</b>   | Sistem menyimpan data <i>Notification</i> ke <i>database</i> .           |
| <b>Priority</b>  |  |
| <b>SCENARIO</b>  |  |
| <ol style="list-style-type: none"> <li>1. <i>Maintenance Planner</i> memilih fungsi <i>Analyze Damage Problems</i>.</li> <li>2. Sistem menampilkan <i>form Notification</i>.</li> <li>3. <i>Maintenance Planner</i> memilih data <i>Notification</i> yang akan dianalisis.</li> <li>4. Sistem menampilkan data <i>Notification</i> yang dipilih.</li> <li>5. <i>Maintenance Planner</i> memasukkan / mengubah data <i>Notification</i> sesuai dengan hasil analisis.</li> <li>6. <i>Maintenance Planner</i> menyimpan data <i>Notification</i>.</li> <li>7. Sistem memvalidasi data yang dimasukkan <i>Maintenance Planner</i>.</li> <li>8. Sistem menyimpan data <i>Notification</i> ke <i>database</i>.</li> </ol> |  |
| <b>ALTERNATIVES</b>  |  |
| 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.   |  |
| <b>EXTENSION</b>   |  |
| <b>NOTE</b>  |  |

Tabel 4. *Usecase Scenario Approve Notification*

| <b><i>Usecase Scenario Approve Notification</i></b> |                       |
|---|-----------------------|
| <b>Uraian Singkat</b>                               | Menyetujui notifikasi |

|   |  |
|---|--|
| <i>Actor</i>  | <i>Maintenance Planner</i>   |
| <i>Trigger Event</i>  |  |
| <i>PreCondition</i>   | <i>Maintenance Planner</i><br>memilih fungsi <i>Approve Notification</i> |
| <i>PostCondition</i>  | Sistem menyimpan data <i>Notification</i> ke <i>database</i> .           |
| <i>Priority</i>   |  |
| <b>SCENARIO</b>   |  |
| <ol style="list-style-type: none"> <li>1. <i>Maintenance Planner</i> memilih fungsi <i>Approve Notification</i>.</li> <li>2. Sistem menampilkan <i>form Notification</i>.</li> <li>3. <i>Maintenance Planner</i> memilih data <i>Notification</i> yang akan dilakukan proses persetujuan.</li> <li>4. Sistem menampilkan data <i>Notification</i> yang dipilih.</li> <li>5. <i>Maintenance Planner</i> memilih <i>user status</i> “NAPR” (<i>Approved</i>) / “NPSP” (<i>Postponed</i>) / “NRJT” (<i>Rejected</i>) sesuai dengan hasil keputusan Pihak Produksi.</li> <li>6. <i>Maintenance Planner</i> menyimpan data <i>Notification</i>.</li> <li>7. Sistem memvalidasi data yang dimasukkan <i>Maintenance Planner</i>.</li> <li>8. Sistem menyimpan data <i>Notification</i> ke <i>database</i>.</li> </ol> |  |
| <b>ALTERNATIVES</b>   |  |
| Jika terjadi kesalahan pada saat <i>Maintenance Planner</i> memasukan data, maka sistem akan menampilkan pesan <i>error</i> / tidak valid dan Operator produksi harus kembali memasukan data yang valid.  |  |
| <b>EXTENSION</b>  |  |
| <b>NOTE</b>   |  |

Tabel 5. *Usecase Scenario Close Notification*

| <b><i>Usecase Scenario Close Notification</i></b> |  |
|---|--|
| Uraian Singkat                                    | Menutup Notifikasi   |
| <i>Actor</i>                                      | <i>Maintenance Planner</i>   |
| <i>Trigger Event</i>                              |  |
| <i>PreCondition</i>                               | <i>Maintenance Planner</i><br>memilih fungsi <i>Close Notification</i> |
| <i>PostCondition</i>                              | Sistem menyimpan data  |



|  |                                  |
|--|----------------------------------|
|  | <i>Notification ke database.</i> |
| <i>Priority</i>  |                                  |
| <b>SCENARIO</b>  |                                  |
| <ol style="list-style-type: none"> <li>1. <i>Maintenance Planner</i> memilih fungsi <i>Close Notification</i>.</li> <li>2. Sistem menampilkan <i>form Notification</i>.</li> <li>3. <i>Maintenance Planner</i> memilih data <i>Notification</i> yang akan ditutup.</li> <li>4. Sistem menampilkan data <i>Notification</i> yang dipilih.</li> <li>5. <i>Maintenance Planner</i> memilih <i>User Status "Close"</i> dan menyimpan ke <i>database</i>.</li> <li>6. Sistem mengeset <i>System Status</i> menjadi "NOCO" (<i>Notification Completed</i>).</li> <li>7. Sistem memvalidasi data yang dimasukkan <i>Maintenance Planner</i>.</li> <li>8. Sistem menyimpan data <i>Notification</i> ke <i>database</i>.</li> </ol> |                                  |
| <b>ALTERNATIVES</b>  |                                  |
| Jika terjadi kesalahan pada saat <i>Maintenance Planner</i> memasukan data, maka sistem akan menampilkan pesan <i>error / tidak valid</i> dan <i>Maintenance Planner</i> harus kembali memasukkan data yang valid.   |                                  |
| <b>EXTENSION</b>   |                                  |
| <b>NOTE</b>  |                                  |

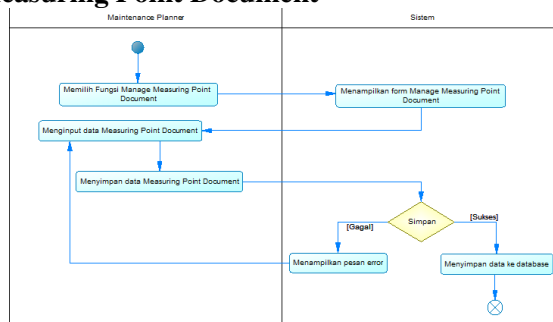
Tabel 6. *Usecase Scenario Create Work Order*

| <b><i>Usecase Scenario Create Work Order</i></b>   |   |
|--|---|
| <i>Uraian Singkat</i>  | Membuat <i>Work Order</i>   |
| <i>Actor</i>   | <i>Maintenance Planner</i>  |
| <i>Trigger Event</i>   | <i>Maintenance Planner Approve Notification</i>   |
| <i>PreCondition</i>  | <i>Maintenance Planner</i> memilih <i>User Status "Approved"</i> pada usecase <i>Approve Notification</i> . |
| <i>PostCondition</i>   | Sistem menyimpan data <i>Work Order</i> ke <i>database</i> .  |
| <i>Priority</i>  |   |
| <b>SCENARIO</b>  |   |
| <ol style="list-style-type: none"> <li>1. <i>Maintenance Planner</i> memilih <i>User Status "Approved"</i> pada usecase <i>Approve Notification</i>.</li> <li>2. Sistem menampilkan <i>form Work Order</i> dan men-generate data <i>notification</i> ke <i>form</i></li> </ol> |   |

|   |
|---|
| <p><i>work order.</i></p> <ol style="list-style-type: none"> <li>3. Sistem mengeset <i>System Status</i> “NOPR ORAS” (<i>Notification in process and Order assigned</i>) pada data <i>Notification</i> dengan nomor yang sama.</li> <li>4. Sistem mengeset <i>System Status</i> “CRTD” (<i>Created</i>) pada data <i>Work Order</i>.</li> <li>5. Sistem mengeset <i>User Status</i> “WNCR” (<i>Newly Created</i>) pada data <i>Work Order</i>.</li> <li>6. Sistem menyimpan data <i>Work Order</i> ke <i>database</i>.</li> </ol> |
| <p><b>ALTERNATIVES</b></p> <p>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.</p>  |
| <p><b>EXTENSION</b></p>   |
| <p><b>NOTE</b></p>  |

### 3.2. Aliran Kerja (Activity Diagram)

#### III.2.1. Manage Measuring Point Document

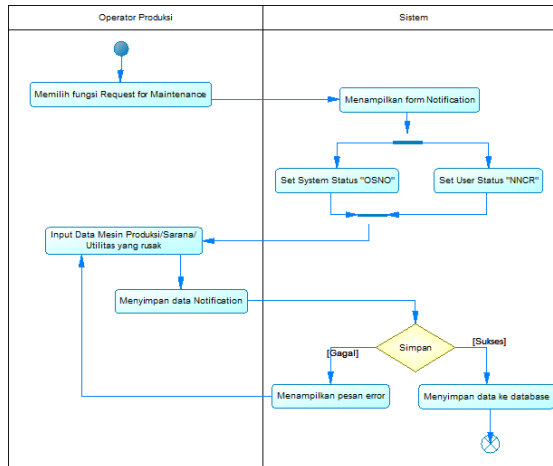


Gambar 4. Activity Diagram Manage Measuring Point Document

#### Uraian Workflow:

1. Dalam gambar 4 dijelaskan bagaimana cara mengelola data *measuring point document*.
2. Langkah pertama, *Maintenance Planner* memilih fungsi *Manage Measuring Point Document*.
3. Sistem akan menampilkan *form Manage Measuring Point Document*.
4. *Maintenance Planner* menginput data *Measuring Point Document* pada *form*, kemudian menyimpan data tersebut.
5. 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.2.2. Request for Maintenance

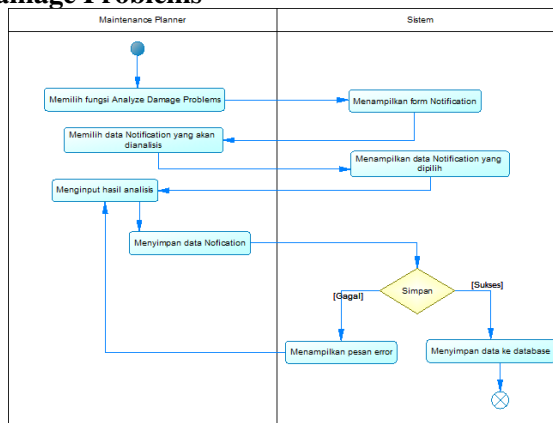


Gambar 5. Activity Diagram Request for Maintenance

#### Uraian Workflow:

1. Dalam gambar 5 dijelaskan bagaimana cara membuat permintaan pemeliharaan.
2. Langkah pertama, Operator Produksi memilih fungsi *Request for Maintenance*.
3. Sistem akan menampilkan *form Notification* dan mengeset *system status* menjadi "OSNO" dan *user status* menjadi "NNCR".
4. Operator Produksi menginput data Mesin Produksi/Sarana/Utilitas yang rusak pada *form*, kemudian menyimpan data tersebut.
5. 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 Operator Produksi harus mengulang proses input data yang valid.

### III.2.3. Analyze Damage Problems

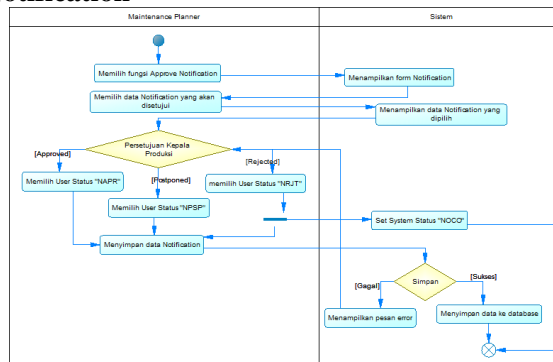


Gambar 6. Activity Diagram Analyze Damage Problems

**Uraian Workflow:**

1. Dalam gambar 6 dijelaskan bagaimana cara menyimpan hasil analisis masalah kerusakan.
2. Langkah pertama, *Maintenance Planner* memilih fungsi *Analyze Damage Problems*.
3. Sistem akan menampilkan *form Notification*.
4. *Maintenance Planner* memilih data *Notification* yang akan dianalisis.
5. Sistem menampilkan data *Notification* yang dipilih.
6. *Maintenance Planner* menginput hasil analisis dan menyimpan data yang telah diinput.
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.2.4. Approve Notification**

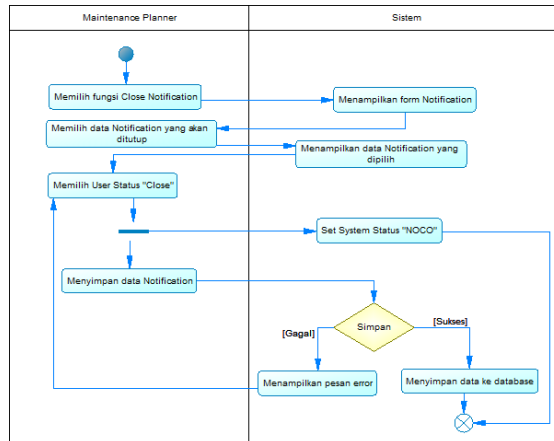


Gambar 7. Activity Diagram Approve Notification

**Uraian Workflow:**

1. Dalam gambar 7 dijelaskan bagaimana cara melakukan persetujuan *notification*.
2. Langkah pertama, *Maintenance Planner* memilih fungsi *Approve Notification*.
3. Sistem akan menampilkan *form Notification*.
4. *Maintenance Planner* memilih data *Notification* yang akan disetujui.
5. Sistem menampilkan data *Notification* yang dipilih.
6. *Maintenance Planner* memilih *user status* berdasarkan persetujuan Kepala Produksi. Apabila di *approve* maka pilih “NAPR”, apabila di *postponed* maka pilih “NPSP”, sedangkan apabila di *rejected* maka pilih “NRJT” 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.

### III.2.5. Close Notification

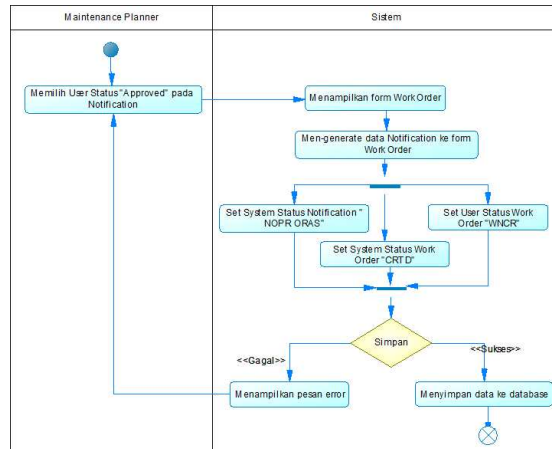


Gambar 8. Activity Diagram Close Notification

#### Uraian Workflow:

1. Dalam gambar 8 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.

### III.2.6. Create Work Order

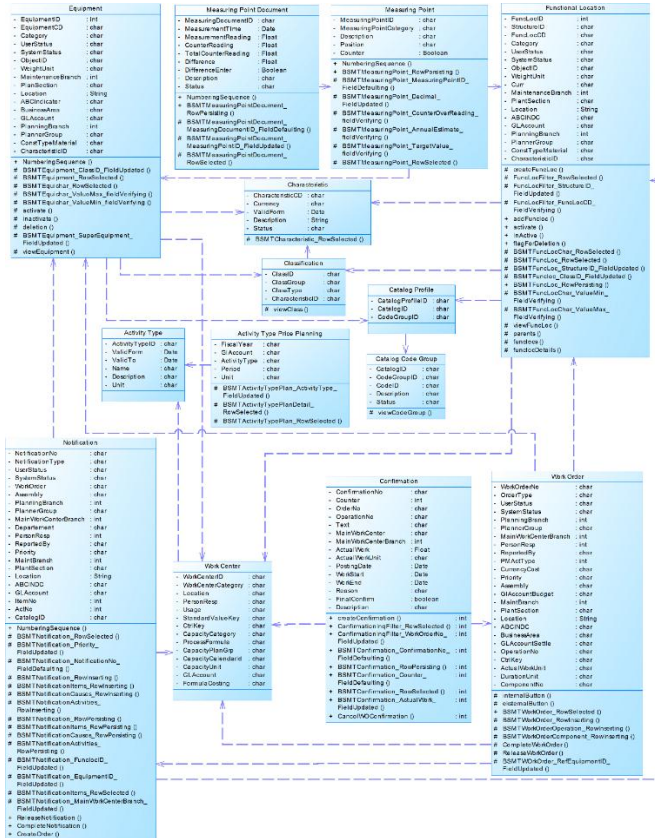


Gambar 9. Activity Diagram Create Work Order

### Uraian Workflow:

1. Dalam gambar 9 dijelaskan bagaimana cara membuat *work order*.
2. Langkah pertama, *Maintenance Planner* dapat membuat *work order* apabila telah memilih *user status* “*Approve*” pada *Notification*.
3. Sistem akan menampilkan *form Work Order* dan Men-generate data *Notification* ke *form Work Order* dan secara otomatis sistem akan mengeset *system status Notification* menjadi “*NOPR*”, *system status Work Order* menjadi “*CRTD*” dan *user status Work Order* menjadi “*WNCR*”.
4. 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.

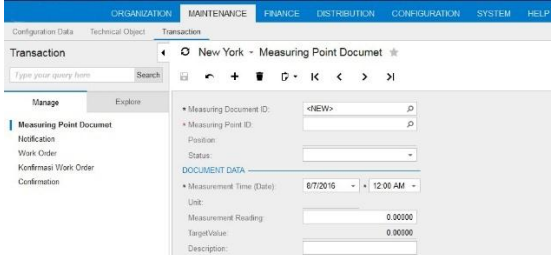
### 3.3. Pemodelan Data (Class Diagram)



Gambar 10. Class Diagram Plant Maintenance

## IV. IMPLEMENTASI

### IV.1. Implementasi Antarmuka Measuring Point Document

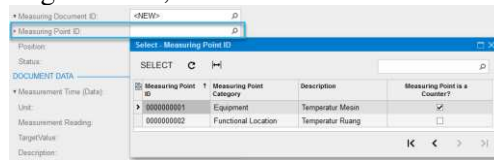


Gambar 11. Dialog Screen Measuring Point Document

Penjelasan Gambar 11 *Dialog Screen Measuring Point Document*

- Halaman ini akan muncul saat pengguna memilih *Measuring Point Document* pada menu *Maintenance* → *Transaction* → *Manage*.

2. Untuk menambah data baru klik tombol **Add New Record**(+). *Field* Measuring Document ID tidak perlu di isi karena *field* tersebut AutoNumber.
3. Pilih *Selector* measuring Point ID,



Perhatikan pada *field* Measuring Point is a Counter?

- a. Jika *value True/Check* maka *form* akan seperti berikut,  
Jika **Diference Enter**? Di *uncheck* maka Total Conter Reading = Counter Reading.

Tetapi jika **Diference Enter**? Di *check* maka Total Conter Reading = Counter Reading + Difference.

- b. Jika *value False/Uncheck* maka *form* akan seperti berikut,

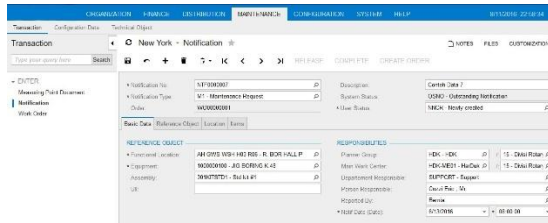
4. Untuk menyimpan data *Measuring Point Document* pilih tombol **Save**(💾).
5. Untuk mencari data *Measuring Point Document*, pilih *selector* Measuring Document ID, cari dan pilih data yang diinginkan, maka data akan otomatis muncul pada *screen* ini.
6. Untuk mengedit data *Measuring Point Document*, pilih *selector* Measuring Document ID, cari dan pilih data yang akan di edit, maka data akan otomatis



muncul pada *screen* ini. Setelah mengedit data, pilih tombol **Save** (💾) agar data yang di edit tersimpan di *database*.

7. Untuk menghapus data *Measuring Point Document*, pilih *selector* Measuring Document ID, cari dan pilih data yang akan di hapus, maka data akan otomatis muncul pada *screen* ini. Kemudian pilih tombol **Delete** (🗑️).

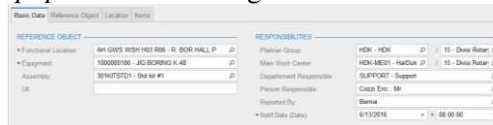
## Notification



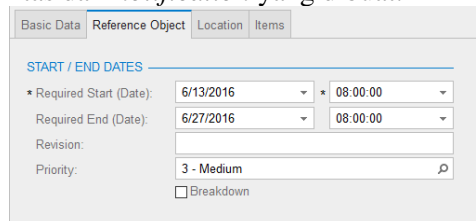
Gambar 12. Dialog Screen Notification

### Penjelasan Gambar 11 *Dialog Screen Notification*

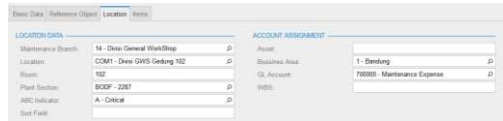
1. Halaman ini akan muncul saat pengguna memilih *Notification* pada menu *Maintenance* → *Transaction* → *Manage*.
2. Untuk menambah data baru klik tombol **Add New Record** (+). *Field* Notification No tidak perlu di isi karena *field* tersebut AutoNumber dan *field* System Status secara otomatis terisi “OSNO” (*Outstanding Notification*).
3. Pilih *selector* Notification Type untuk menentukan apakah tipe notifikasi tersebut **Corrective**, **Preventive** atau **Emergency**.
4. Pada tab Basic Data, pilih satu data *Functional Location* dan sistem akan menampilkan data *Equipment* sesuai dengan *Functional Location* yang dipilih.



5. Apabila *Functional Location* dalam keadaan kosong dan user memilih satu data *Equipment* maka sistem akan menampilkan *Functional Location* yang sesuai dengan *Equipment* yang dipilih.
6. Inputkan data pada semua *field* yang harus diisi.
7. Pada tab Reference Object, isi tanggal mulai dan akhir pengerjaan perbaikan dan tentukan tingkat prioritas dari *notification* yang dibuat.



8. Pada tab Location, isi data lokasi dan tentukan akun GL untuk menghitung biaya perbaikan.



9. Pada tab Item, terdapat 3 *grid* antara lain:
  - a. *Grid items* = untuk menyimpan data *object part* dan *damage*
  - b. *Grid causes* = untuk menyimpan data *cause* berdasarkan item yang dipilih.



- c. *Grid activities* = untuk menyimpan data aktifitas dari item yang dipilih.



10. Untuk menyimpan data *Notification* pilih tombol **Save** (💾).
11. Untuk mencari data *Notification*, pilih *selector* Notification No, cari dan pilih data yang diinginkan, maka data akan otomatis muncul pada *screen* ini.
12. Untuk mengedit data *Notification*, pilih *selector* Notification No, cari dan pilih data yang akan di edit, maka data akan otomatis muncul pada *screen* ini. Setelah mengedit data, pilih tombol **Save** (💾) agar data yang di edit tersimpan di *database*.
13. Untuk menghapus data *Notification*, pilih *selector* Notification No, cari dan pilih data yang akan di hapus, maka data akan otomatis muncul pada *screen* ini. Kemudian pilih tombol **Delete** (🗑).

## V. KESIMPULAN DAN SARAN

### V.1. Kesimpulan

Setelah melakukan penelitian terhadap permasalahan yang ada, maka dapat disimpulkan bahwa penerapan hasil penelitian mengenai *Plant Maintenance* ini sangat akurat untuk digunakan di perusahaan, khususnya untuk perusahaan industri yang memerlukan pemeliharaan peralatan/mesin dengan baik.

Dengan adanya perbaikan dalam proses pemeliharaan mesin, 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.

## V.2. Saran

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 bukan hanya untuk modul *inspection* dan jenis Corrective Maintenance saja tetapi juga dikembangkan lebih lanjut untuk modul dan jenis-jenis pemeliharaan yang lain.

## 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. Raum 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-AI-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] R. S. Pressman, Rekayasa Perangkat Lunak, vol. 7, Yogyakarta: Andi, 2012.
- [12] G. D. Gupta Mukul, Research Methodology, New Delhi: PHI Learning Pvt., 2011.