

## RANCANGAN PROTYPE PERANGKAT LUNAK PENJADWALAN TINDAK PERAWATAN PENCEGAHAN BERBASIS KEANDALAN

Rispianda<sup>1)</sup>

Adrian<sup>2)</sup>

Jurusan Teknik Industri, Fakultas Teknologi Industri, Institut Teknologi Nasional<sup>1,2)</sup>

Jl. P.H.H. Mustofa No 23 Bandung

Telepon (022) 7272215 ekst 137

E-mail: rispianda@gmail.com<sup>1)</sup>

### Abstrak

Perusahaan telah menerapkan kebijakan tindak perawatan pencegahan (*preventive maintenance*) pada seluruh mesin yang dimilikinya. Adapun sejumlah besar kejadian kerusakan pada selang interval PM masih ditemukan pada level subsistem. Pada penelitian ini, penjadwalan PM diurai menjadi level subsistem. Interval penjadwalan ditentukan menggunakan metode *Least-cost Preventive Maintenance Interval*. Keluaran dari penelitian ini adalah alat bantu penjadwalan PM dalam bentuk perangkat lunak berupa *web application*. Perangkat lunak dikembangkan dengan menggunakan metodologi *Waterfall Model* dan prinsip *Object-Oriented Modelling*. Hasilnya, penjadwalan dalam perangkat lunak dapat menghasilkan penghematan total biaya perawatan dibandingkan kebijakan perusahaan.

*Kata Kunci: Tindak perawatan, teori keandalan, pengembangan perangkat lunak, Least-cost Preventive Maintenance Interval*

### Pendahuluan

PT X merupakan salah satu pelaku industri manufaktur dengan kompetensi inti di bidang desain, pengembangan, dan manufaktur produk aviasi. Perusahaan telah menerapkan kebijakan tindak perawatan pencegahan (*preventive maintenance* atau PM) pada level mesin. Tindakan PM dilakukan pada level mesin yang berarti tindak PM dilakukan pada seluruh subsistem pada mesin terlepas dari *failure history* masing-masing subsistem. Setiap subsistem dikenai tindakan PM dengan frekuensi tetap dalam suatu tahun. Dari data yang diperoleh, pada selang waktu PM, masih ditemukan kerusakan yang sebagian besar ditemukan pada level subsistem. Kondisi tersebut menandakan adanya variasi performansi pada level subsistem.

Nilai selang waktu pelaksanaan PM ditentukan oleh perusahaan secara intuitif karena belum adanya alat bantu analisis performansi. Di sisi lain, subsistem yang lebih sering rusak berpeluang membutuhkan tindak perawatan pencegahan yang lebih intensif. Adapun jika tindak perawatan pencegahan semakin sering dilakukan, maka akan mengurangi jam produktif serta menambah total biaya perawatan. *Output* dari penelitian ini adalah menghasilkan alat bantu penjadwalan kegiatan perawatan pencegahan berdasarkan hasil analisis performansi subsistem dengan turut mempertimbangkan total biaya perawatan. Interval ditentukan menggunakan metode *Least-cost Preventive Maintenance Interval* [1]. Alat bantu yang dihasilkan berupa perangkat lunak. Perangkat lunak dikembangkan dengan metodologi *Waterfall Model* [4].

### Metode Penelitian

Metodologi penelitian adalah sebagai berikut:

- 1) Tahap Analisis Sistem, *System analysis* merupakan tahap analisis kebutuhan sistem dalam pengembangan perangkat lunak. Pada tahap ini dilakukan analisis terhadap:
  - Kebijakan Tindak perawatan pencegahan
  - Sistem Manajemen Perawatan
  - Strukturisasi Kebutuhan Sistem
- 2) Tahap Perancangan Sistem dan Perangkat Lunak. Pada tahapan ini dilakukan hal-hal sebagai berikut:
  - Evaluasi Data Masukan

- Algoritma Penentuan Interval Waktu Penjadwalan
  - Identifikasi Use Case
  - Perancangan Perangkat Lunak
- 3) Pengujian Perangkat Lunak. Pada tahap ini dilakukan pengujian fungsi perangkat lunak [5].

## HASIL DAN PERANCANGAN

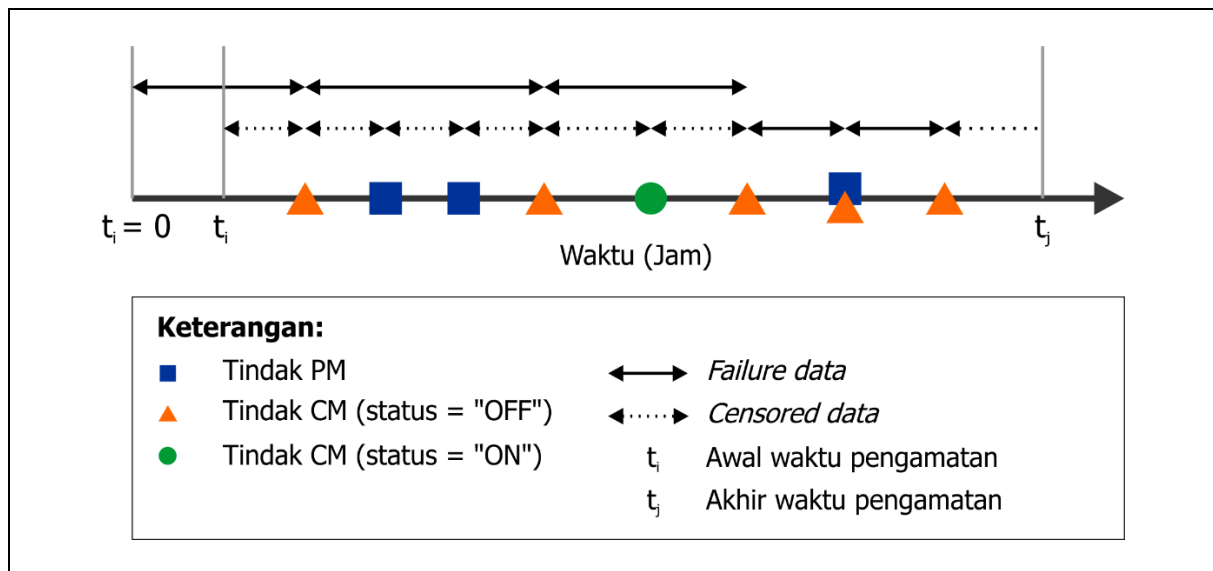
Data kerusakan diperoleh dari hasil proses ekspor yang dilakukan pada CMMS milik perusahaan. Hasil identifikasi data dapat dilihat pada **Tabel 1**. Informasi pada **Tabel 1** menunjukkan bahwa detail data tindak perawatan mencapai level subsistem. *Jobtype* menunjukkan jenis tindak perawatan yang dilakukan, yakni bernilai salah satu dari “PM”, “CM” (*corrective maintenance*), atau “PR” (*preventive replacement*). *Status* menunjukkan kondisi mesin atau subsistem saat tindak perawatan dilakukan. Atribut data *status* dapat bernilai “ON” atau “OFF”.

**Tabel 1. Hasil Identifikasi Data**

<b>Kolom</b>	<b>Keterangan</b>	<b>Type Data</b>
Codification	Kode mesin	<i>Nominal</i>
Description	Deskripsi mesin	<i>Nominal</i>
Manufacturer	<i>Vendor</i> mesin	<i>Nominal</i>
FacilityType	Jenis fasilitas	<i>Nominal</i>
SubSystem	Nama subsistem	<i>Nominal</i>
Status	Status mesin	<i>Nominal</i>
IssuedDate	Tanggal dan waktu dimulainya tindak perawatan	<i>Ratio</i>
DeliveryDate	Tanggal dan waktu selesainya tindak perawatan	<i>Ratio</i>
TotDowntime	Lama waktu <i>downtime</i>	<i>Ratio</i>
TotMTTR	Lama waktu perbaikan	<i>Ratio</i>
JobType	Jenis tindak perawatan	<i>Nominal</i>

Setiap baris data yang ada belum mengandung nilai *time between failures* (TBF), sehingga diperlukan pengumpulan data nilai TBF terlebih dahulu. Nilai TBF menunjukkan selang waktu sejak mesin atau subsistem beroperasi dalam kondisi normal hingga terjadinya kerusakan [2] [3]. Kerusakan (*failure*) didefinisikan sebagai peristiwa hilangnya (*termination*) kemampuan objek dalam melaksanakan fungsi sebagaimana objek tersebut dirancang dan dibangun.

Pada data historis tindak perawatan dalam CMMS, kejadian tersebut ditandai oleh adanya tindak CM dengan status “OFF”. Adapun, data historis yang ada tidak hanya mengandung jenis kejadian tersebut, sehingga berakibat kepada diperlukannya penapisan data masukan (*input data censoring*). Proses penapisan mengikuti diagram pada Gambar 1.

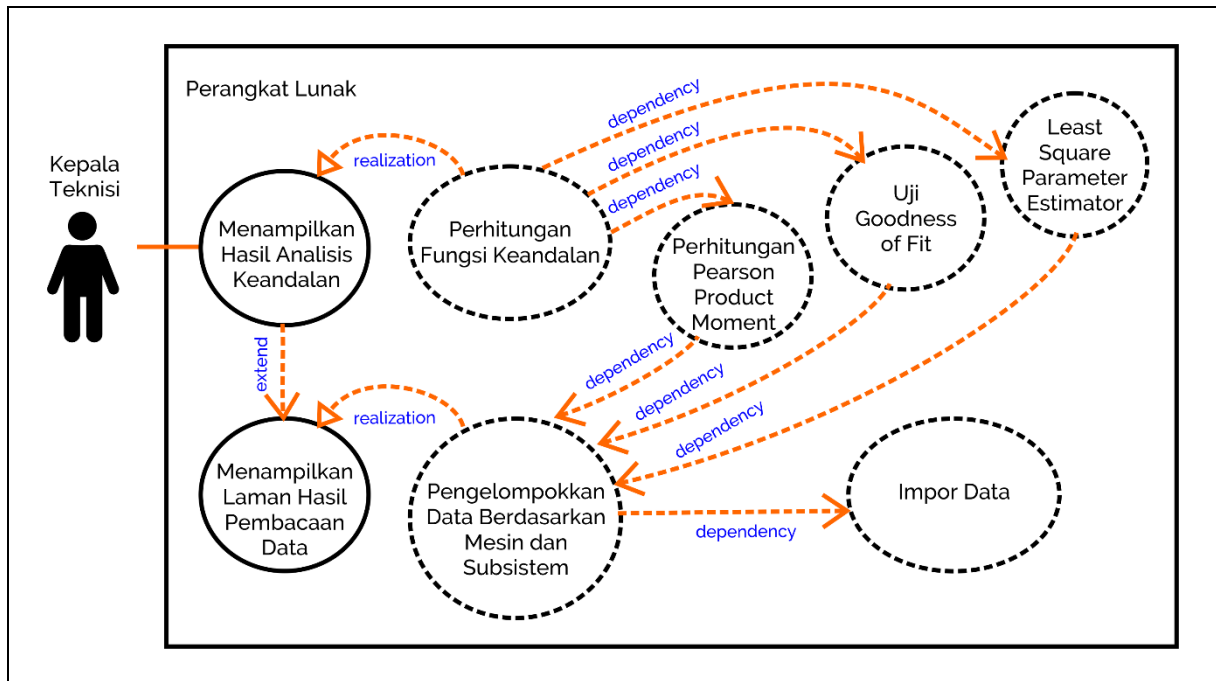


Gambar 1. Diagram Penapisan Data Masukan

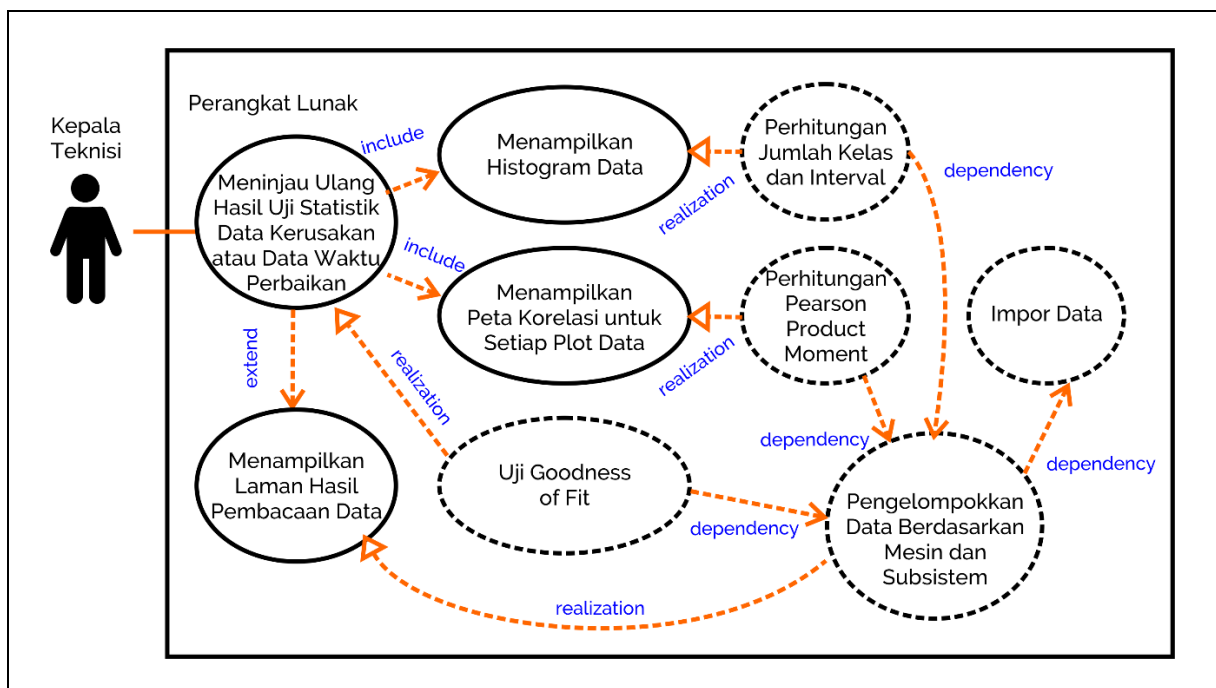
Algoritma penentuan interval waktu penjadwalan untuk masing-masing subsistem adalah sebagai berikut:

1. Kumpulkan data tanggal terjadinya kerusakan.
2. Jika jumlah data  $\geq 4$ , lanjut ke langkah (3). Lainnya, lanjut ke langkah (10).
3. Hitung nilai *time between failures* (TBF) untuk setiap kejadian kerusakan.
4. Identifikasi distribusi data dengan melakukan uji statistik *Goodness of Fit* (GoF). Pengujian dilakukan dengan metode pengujian Kolmogorov-Smirnov untuk distribusi normal, pengujian Bartlett untuk distribusi eksponensial, dan pengujian Mann untuk distribusi Weibull.
5. Eliminasi distribusi yang tidak lulus uji GoF. Jika tidak ada distribusi yang lulus uji statistik GoF, lanjut ke langkah (10). Lainnya, lanjut ke langkah (6).
6. Jika hanya terdapat satu distribusi yang lulus uji GoF, distribusi tersebut terpilih untuk menggambarkan distribusi data kerusakan, lalu lanjut ke langkah (8).
7. Jika terdapat lebih dari satu distribusi yang lulus uji GoF, hitung nilai *Pearson Product Moment* ( $r$ ) data terhadap setiap distribusi. Pemilihan jenis distribusi dilakukan berdasarkan nilai  $r$  tertinggi.
8. Estimasi parameter distribusi menggunakan metode *Least Square Parameter Estimator*.
9. Perhitungan Interval waktu penjadwalan.
10. Jika sampai tahap ini belum ditemukan nilai interval waktu penjadwalan, berarti (a) jumlah data tidak memadai atau (b) distribusi data tidak dapat digambarkan menggunakan distribusi normal, lognormal, eksponensial, atau Weibull. Maka, interval waktu penjadwalan mengikuti kebijakan *default* dari perusahaan, yaitu pada 2000 dan 4000 jam operasional.
11. Selesai.

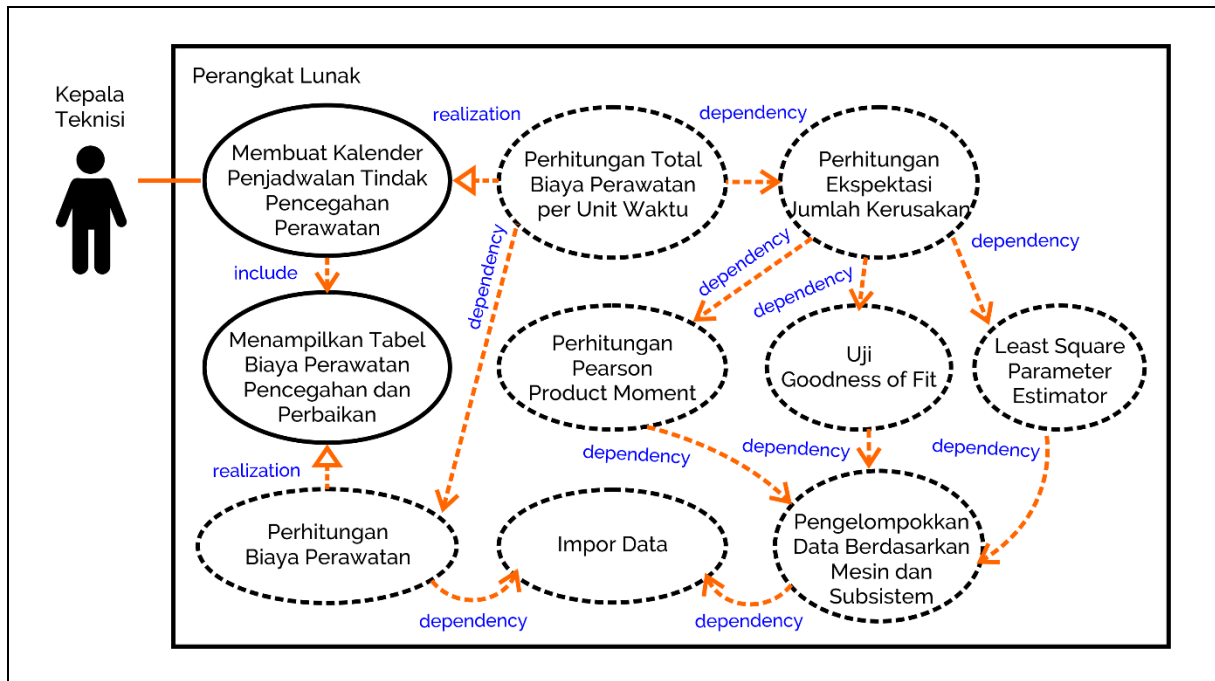
*Use cases* pada level *global* disesuaikan dengan tujuan penelitian untuk kemudian diurai menjadi *use case* yang lebih rendah. Tiga buah visualisasi *use case* pada level *global*, yaitu melakukan analisis keandalan, meninjau ulang hasil perhitungan, dan melakukan penjadwalan secara berturut-turut dapat dilihat pada Gambar 2, Gambar 3, dan Gambar 4.



Gambar 2. Use Case #1: Melakukan Analisis Keandalan Subsistem

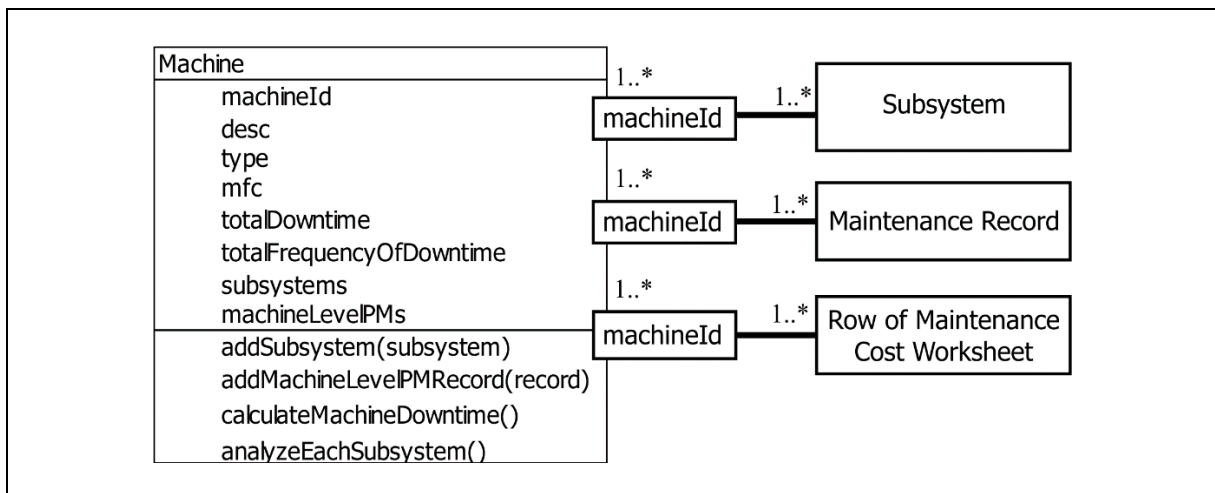


Gambar 3. Use Case #2: Meninjau Ulang Hasil Uji Statistik Data Waktu Kerusakan dan Data Waktu Perbaikan



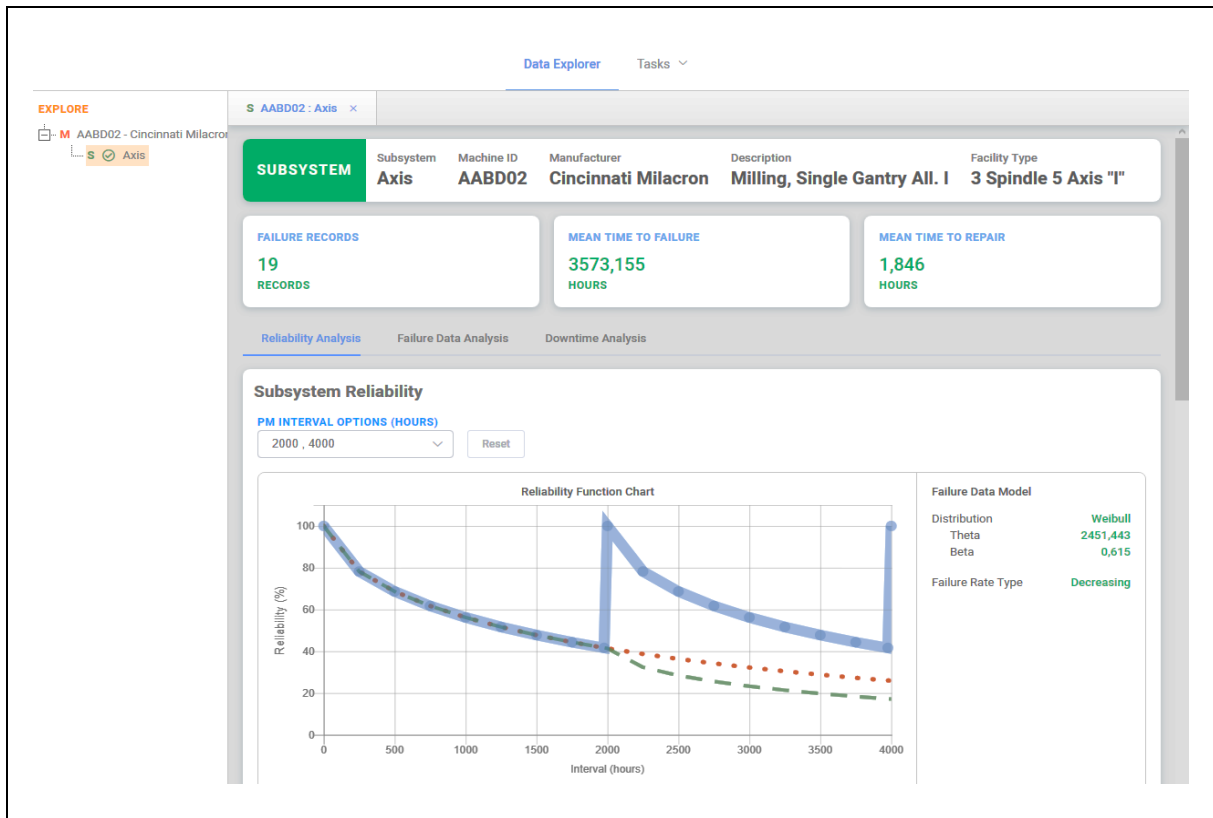
Gambar 4. Use Case #3: Membuat Kalender Penjadwalan Tindak Perawatan Pencegahan

Perusahaan telah memiliki satu buah *server* yang selama ini digunakan untuk kepentingan CMMS. Adapun setiap personil Departemen *Maintenance* memiliki satu buah komputer. Maka, topologi yang digunakan dalam pengembangan perangkat lunak adalah *client-server*. Adapun perancangan arsitektur perangkat lunak dilakukan berdasarkan prinsip *Object-Oriented Modelling*. Objek dirancang sesuai dengan kebutuhan dalam *use case*. Contoh model objek hasil perancangan dapat dilihat pada Gambar 5.

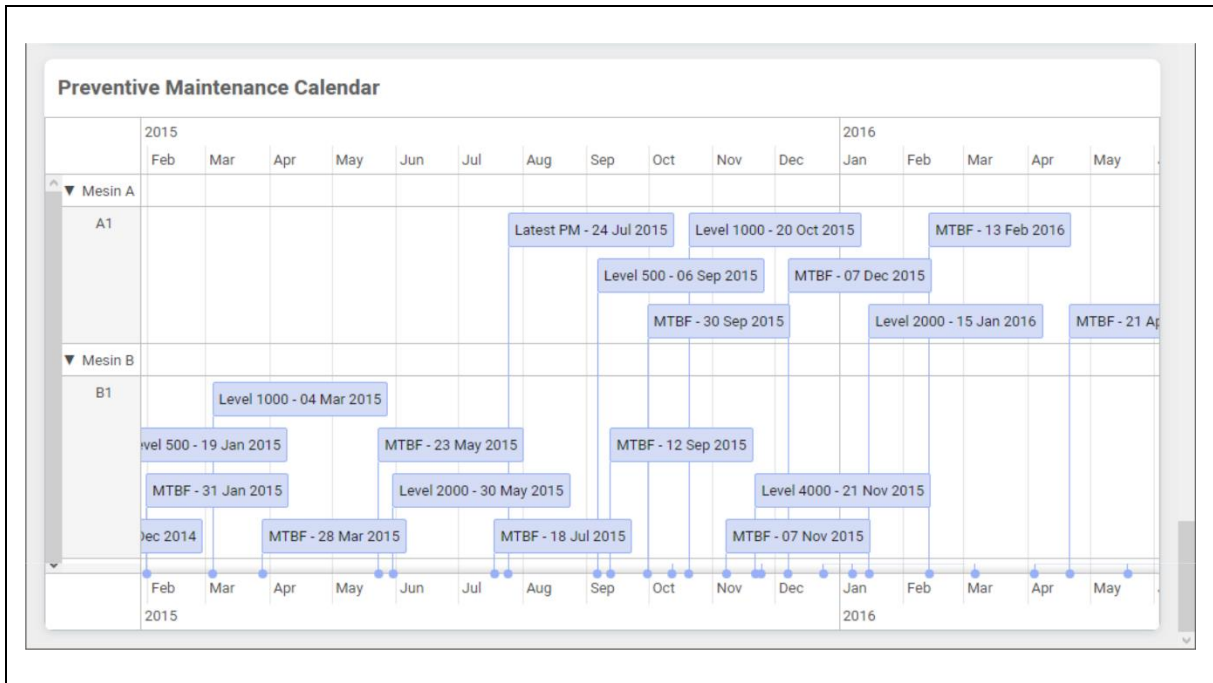


Gambar 5. Model Objek *Machine*

Perangkat lunak terbagi menjadi dua bagian, yakni *Data Explorer* dan *Tasks*. Laman *Data Explorer* berisikan daftar mesin dan subsistem berikut hasil analisis keandalan, analisis data waktu kerusakan, dan analisis *downtime*. Cuplikan tampilan *Data Explorer* dapat dilihat pada Gambar 6. Laman *Tasks* berisikan aksi yang dapat dilakukan setelah dilakukan analisis keandalan terhadap masing-masing subsistem. Pada saat ini, jenis *task* yang dapat dilakukan adalah penjadwalan tindak perawatan pencegahan. Cuplikan rancangan tampilan kalender hasil penjadwalan tindak perawatan dapat dilihat pada Gambar 7.



Gambar 6. Tampilan *Data Explorer*



Gambar 7. Tampilan Kalender Penjadwalan Tindak Perawatan Pencegahan

Verifikasi hasil perhitungan dilakukan dengan membandingkan hasil perhitungan dalam *Excel* dan perangkat lunak. Hasil perhitungan dalam *Excel* dapat dilihat pada Tabel 2. Hasil perhitungan dalam perangkat lunak dapat dilihat pada Gambar 8. Hasil perhitungan dalam perangkat lunak bersifat akurat sesuai dengan hasil perhitungan dalam *Excel*, sehingga pengujian disimpulkan berhasil.

Tabel 2. Hasil Perhitungan dalam Excel

Kode Subsystem	Set Data	Biaya Perawatan per Jam (Rupiah)			
		500, 1000, 2000, 4000	1000, 2000, 4000	2000, 4000	4000
A1	Normal (Increasing Failure Rate)	2492,5	3195	3140	3842,5
B1	Lognormal (Increasing- Decreasing Failure Rate)	3250	3952,5	4655	4600
C1	Eksponensial (Constant Failure Rate)	220	165	110	812,5
D1	Weibull (Decreasing Failure Rate)	4007,5	3195	3140	2327,5
E1	Weibull (Increasing Failure Rate)	2492,5	3195	3140	3085

Maintenance Cost per PM Interval Options						
Headers : <input type="text" value="Select"/>						
No.	Machine ID	Subsystem	Interval Options			
			[500, 1000, 2000, 4000]	[1000, 2000, 4000]	[2000, 4000]	[4000]
1	Mesin A	A1	2492,500	3195	3140	3842,500
2	Mesin B	B1	3250	3952,500	4655	4600
3	Mesin C	C1	220	165	110	812,500
4	Mesin D	D1	4007,500	3195	3140	2327,500
5	Mesin E	E1	2492,500	3195	3140	3085

Total Data Entries : 5 Data per Page : 5

Gambar 8. Hasil Perhitungan dalam Perangkat Lunak

## Kesimpulan

Kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut:

1. Kebijakan tindak perawatan pencegahan *block maintenance policy* yang semula dilakukan pada level mesin dengan interval 2000 dan 4000 jam operasional diurai menjadi level subsystem dan opsi interval waktu pelaksanaan tindak perawatan pencegahan untuk setiap subsystem dapat memiliki nilai yang bervariasi.
2. Variabel keputusan dalam *Least-Cost Preventive Maintenance Interval* adalah nilai interval waktu pelaksanaan, sehingga ditentukan fungsi pembatas interval waktu pelaksanaan tindak perawatan pencegahan yang disesuaikan dengan *Preventive Maintenance Instruction* yang dimiliki perusahaan, yakni 500, 1000, 2000, dan 4000 jam operasional.

**Daftar Pustaka**

- [1] Ebeling, Charles E. 1997, *Introduction to Reliability and Maintainability Engineering*. McGraw-Hill, New York.
- [2] Krasich, Milena. 2009, *How to Estimate and Use MTBF/MTBF?*. New Jersey, Institute of Electrical and Electronics Engineers.
- [3] Relex Software Corporation, 2009, *MTBF Versus MTBF: Theoretical Definitions and Alternative Uses for MTBF*. Relex, Helsinki.
- [4] Royce, Winston W. Royce., 1970, 'Managing the Development of Large Software Systems', *Proceedings IEEE Wescon Agustus 1970*, hal 1-9.
- [5] Valacich, Joseph S., George, Joey F., 2016, *Modern Systems Analysis and Design*. Pearson, New York.