

Pengembangan Aplikasi Perawatan Berdasarkan Life-Cycle Cost Berbasis Website

1st Farhan Ramadhan
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia

farhanramadhann@student.telkomuniversity.ac.id

2nd Rohmat Saedudin
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia

rdrohmat@telkomuniversity.ac.id

3rd Seno Adi Putra
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia

adiputra@telkomuniversity.ac.id

Abstrak —Mesin yang semakin canggih memiliki kecenderungan untuk menjadi kompleks. Kompleksitas ini mendatangkan masalah baru yaitu perawatan yang lebih intens. Untuk mengatasinya manusia membuat teori dan praktik perawatan yang berusaha menanggulangi kerusakan, perencanaan, atau pengadaan dana secara tepat sehingga aset perusahaan digunakan dengan lebih efektif. Salah satu teori yang dapat digunakan adalah *Life-Cycle Cost* (LCC). LCC adalah perhitungan yang bertujuan dalam memprediksi biaya jangka panjang dalam menggunakan mesin selama masa hidupnya. Biaya seperti bahan baku, pembelian mesin, upah tenaga kerja, listrik dan biaya untuk mendukung kemampuan operasional menjadi masukan untuk memperkirakan biaya jangka panjang agar perencanaan anggaran dapat ditentukan secara garis besar. Hal ini dapat mengurangi kemungkinan terjadinya pembengkakan dana ketika terjadi kerusakan atau jika kebutuhan untuk menambah mesin muncul. Sebuah aplikasi kalkulator berbasis website yang menghitung menggunakan metode LCC dapat digunakan untuk membantu proses perbaikan dan perencanaan keuangan. Data perhitungan yang telah dilakukan akan disimpan dengan tujuan memudahkan akses terhadap data tersebut. Aplikasi akan dikembangkan menggunakan metode *waterfall* dan memanfaatkan framework Laravel. Perencanaan sistem menggunakan model *Unified Modelling Language* (UML). Aplikasi akan di uji menggunakan metode *Black Box Testing*. Penelitian ini bertujuan untuk memberikan manfaat dalam meningkatkan efisiensi proses bisnis dari Serambi Indonesia.

Kata kunci— *Life-Cycle Cost*, perawatan, *waterfall*, *black box testing*

I. PENDAHULUAN

Dengan adanya revolusi industri 2.0 dan 3.0, produksi yang sebelumnya dilakukan sepenuhnya oleh manusia mulai dibantu oleh mesin sampai akhirnya proses dapat diotomatiskan sepenuhnya oleh mesin. Mesin dinilai lebih menguntungkan oleh perusahaan karena dapat mengurangi risiko terjadi kesalahan yang rentan dilakukan oleh manusia atau disebut juga dengan *human error*, melakukan tugas dengan cepat dan efisien, dan masih banyak manfaat lainnya.

Namun seiring dengan waktu, beberapa komponen atau keseluruhan mesin tersebut dapat mengalami kegagalan sistem dikarenakan beberapa komponen mengalami keausan sehingga krusial untuk memelihara mesin tersebut. Dampak kerusakan yang terjadi beragam dari jumlah, peran, dan

komponen yang rusak. Mesin dapat diganti secara keseluruhan jika dampaknya terlalu besar dan lebih mudah atau murah untuk mengganti mesin tersebut. Jika dampak kerusakan tidak terlalu besar, komponen yang telah rusak dapat diganti oleh komponen baru.

Perawatan harus dilakukan secara tepat supaya memaksimalkan potensi, efisiensi dan umur operasi mesin agar pengguna dapat meminimalisir biaya perawatan. Perhitungan biaya, dan daftar penggantian komponen umumnya dilakukan secara manual oleh manusia yang rentan melakukan kesalahan sehingga terjadi biaya lebih, penggantian komponen yang salah atau tidak tepat, dan kerusakan berlebih yang dialami oleh mesin.

Salah satu metode yang digunakan untuk menghitung biaya jangka panjang adalah LCC. Metode ini meliputi biaya keseluruhan dari awal pembelian, perawatan, operasional hingga akhir pembongkaran. Sehingga penentuan anggaran perbaikan dapat dilakukan dengan tepat. Hal ini dapat direalisasikan melalui perawatan dengan bantuan aplikasi untuk memudahkan pengguna dalam merencanakan penambahan atau penggantian mesin.

II. KAJIAN TEORI

A. Mean Time To Repair (MTTR)

MTTR melambangkan rata-rata waktu yang digunakan untuk memperbaiki mesin ketika terjadi kerusakan saat beroperasi normal. Waktu untuk melakukan reparasi meliputi proses seperti deteksi, diagnosa, akuisisi suku cadang, dan perbaikan mesin itu sendiri. Keseluruhan proses perbaikan disebut juga dengan *downtime*.

$$MTTR = \frac{TDH}{NF}$$

Dimana:

TDH = *Total Downtime Hours*

NF = *Number of Failures*

Untuk menentukan rata-rata waktu yang dibutuhkan untuk memperbaiki sebuah mesin, dapat dilakukan dengan menghitung total jam mesin tidak beroperasi karena kerusakan dibagi jumlah kerusakan yang terjadi. MTTR dan MTBF dinyatakan dalam bentuk jam dan digunakan untuk optimisasi perawatan.

B. Mean Time Between Failure (MTBF)

MTBF atau singkatan dari mean time between failure merupakan rata-rata selang waktu di antara dua kerusakan yang terjadi pada mesin yang beroperasi dengan normal. Normal berarti bahwa mesin berada dalam periode masa hidup yang dinilai efektif. Hal ini ditandai dengan rata-rata kegagalan yang konsisten.

$$MTBF = \frac{TUH - TDH}{NF}$$

Dimana:

TUH = Total Uptime Hours

TDH = Total Downtime Hours

NF = Number of Failures

Total Uptime Hours melambangkan akumulasi durasi dimana mesin bekerja dengan normal. Total Downtime Hours melambangkan akumulasi durasi dimana mesin sedang tidak bekerja dikarenakan mengalami kerusakan. Number of Failures melambangkan jumlah terjadinya kerusakan dalam periode tersebut.

C. Life Cycle Cost

Life Cycle Cost (LCC) terdefiniskan "LCC dari sebuah aset adalah jumlah dari semua kapital yang dikeluarkan untuk membantu aset tersebut semenjak desain dan manufaktur melalui operasi hingga akhir hidup dari aset tersebut. Jumlahnya terdiri dari biaya berulang atau tidak berulang dari siklus hidup aset secara keseluruhan termasuk harga komponen, biaya pemasangan komponen, biaya operasional, biaya peningkatan komponen, dan biaya lain yang tersisa pada akhir kepemilikan atau masa manfaatnya." [1]. LCC digunakan untuk menghitung, memprediksi, dan menganalisa jumlah biaya yang harus dikeluarkan untuk keberlangsungan aset.

Ada juga pendapat lain yang mengatakan bahwa LCC adalah suatu metode dalam mengevaluasi proyek dari seluruh biaya yang dibutuhkan dari awal tahap pengelolaan, pengoperasian, pemeliharaan, dan pembuangan dari sebuah proyek. Hasil dari LCC digunakan untuk bahan pertimbangan dalam proses pengambilan keputusan. [2]

$$Life\ Cycle\ Cost = BI + MS(OT + PT) + BP$$

Dimana:

BI = Biaya inisiasi

OT = Biaya operasional per tahun

PT = Biaya pemeliharaan per tahun

BP = Biaya pembongkaran

MS = Estimasi masa hidup

Biaya awal merupakan biaya yang dikeluarkan dalam merencanakan, pembelian atau pemasangan mesin. Biaya perawatan adalah biaya pemeliharaan agar mesin dapat digunakan secara efektif. Biaya perawatan meliputi perbaikan mesin contohnya seperti servis, penggantian bagian mesin, dan peningkatan kualitas mesin. Biaya operasional meliputi biaya untuk mengoperasikan mesin seperti bahan baku, gaji karyawan, dan listrik. Sedangkan biaya pembuangan adalah biaya untuk membongkar mesin ketika sudah melewati batas umur rencana pembangunan

atau mesin sudah tidak berfungsi secara optimal atau tidak bekerja sama sekali. Estimasi masa hidup adalah perkiraan periode dimana mesin dapat bekerja secara normal atau efektif. Jika usia mesin telah melewati estimasi masa hidup maka kecenderungan kerusakan meningkat dibandingkan di periode optimalnya.

Klasifikasi data diatas akan dijelaskan lebih detil pada rumus berikut.

$$OT = Cc + (G * TK)$$

Dimana:

Cc = Consumable Cost

G = Upah pegawai per tahun

TK = Jumlah tenaga kerja yang dikerahkan

Consumable cost merupakan akumulasi dari berbagai hal yang mendukung segi operasional mesin tersebut. Contohnya seperti listrik, bahan baku, bahan bakar, tinta, dll. Tenaga kerja merupakan pegawai yang dikerahkan untuk mengoperasikan mesin tersebut. Upah dikalikan jumlah pegawai yang dikerahkan pada lapangan.

$$PT = W * H \frac{MTTR}{MTBF}$$

Dimana:

W = Upah pegawai perawatan per tahun

H = Durasi penggunaan mesin per tahun

MTTR = Mean Time to Repair

MTBF = Mean Time Between Repair

Biaya perawatan pertahun dapat dihitung dengan mencari durasi penggunaan mesin per tahun. Durasi penggunaan mesin dikalikan upah pegawai perawatan. Rata-rata waktu perbaikan dibagi rata-rata waktu antara perbaikan digunakan untuk menentukan seberapa intens perawatan dilakukan.

$$BP = P + PB - S$$

Dimana:

P = Biaya pelepasan

PB = Biaya pengangkutan

S = Salvage value

Biaya pelepasan merupakan biaya untuk membongkar dan melepaskan mesin dari tempat instalasi. Setelah mesin dilepas maka akan diangkut untuk dibuang atau dijual. Salvage value adalah nilai jual mesin setelah dilepas.

Fungsi utama dari LCC adalah mengevaluasi operasional mesin ketika muncul permasalahan yang melibatkan akuisisi mesin. Perhitungan LCC dapat digunakan untuk mencari solusi-solusi alternatif yang membantu pembuatan keputusan. Hal-hal yang perlu diperhatikan juga usia rencana, biaya perbaikan, kualitas konstruksi, usia komponen, metode desain yang dianggap penting dan menjadi faktor-faktor yang mempengaruhi LCC. Manfaat dari analisis LCC adalah perencanaan biaya total cost atau total ownership cost (TOC) secara efektif sehingga cost term ownership yang paling pendek tercapai. LCC dapat

membantu bagian perawatan dalam memilih perlengkapan dan proses perbaikan berdasarkan pengeluaran total siklus masa hidup mesin. Perhitungan LCC dapat digunakan pada tahap perancangan, pengembangan, produksi, operasi, dan perawatan pada siklus hidup sesuai dengan estimasinya. [3]

Integrasi antara LCC dengan sektor produksi dapat diwujudkan dalam bentuk beberapa metode seperti model matematika, pembuatan aplikasi dengan tujuan optimisasi, dan pembuatan keputusan berdasarkan puluhan kriteria. Walaupun terdapat beberapa halangan seperti waktu, sumber daya, pengetahuan yang intensif pada bidang yang berbeda dapat dijadikan kesempatan dengan pengembangan metode yang lebih komprehensif. [4]

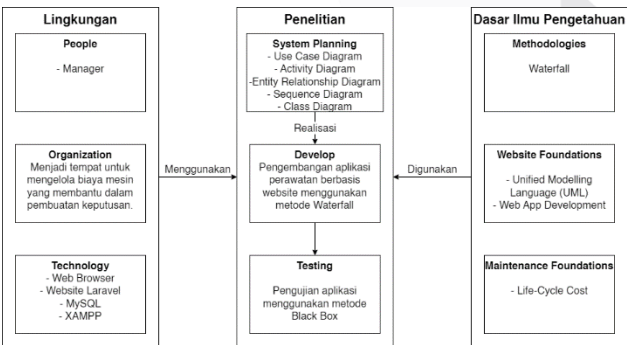
D. Black Box Testing

Black-Box Testing merupakan salah satu metode untuk menguji kelayakan aplikasi. Kasus uji ditentukan dari basis prasyarat atau spesifikasi program atau modul atau program yang tidak dipertimbangkan pada seleksi kasus uji [5]. Black-Box Testing dikenal juga dengan Functional Testing. Sesuai namanya metode ini menguji aplikasi dari segi fungsionalitas yaitu observasi hasil akhir aplikasi berdasarkan masukan tertentu sebelumnya tanpa mencoba menganalisa kode dari aplikasi tersebut. Sehingga tes ini tidak mengetahui latar belakang bagaimana sebuah aplikasi bekerja namun memfokuskan pada keluaran dan masukan yang sesuai dengan tujuan awal pembuatan aplikasi tersebut. Tester hanya mengetahui masukan yang diberikan kepada sistem, lalu menilai apakah keluaran sesuai dengan apa yang sistem seharusnya berikan.

III. METODE

A. Model Konseptual

Model konseptual digunakan untuk memvisualisasikan pengerjaan penelitian dalam sebuah kerangka yang lebih mudah dipahami. Kerangka ini memetakan ruang lingkup penelitian agar membantu menetapkan metode penyelesaian masalah dan merumuskan solusi untuk masalah tersebut.



GAMBAR 1
Model Konseptual

Hasil penelitian berupa aplikasi berbasis LCC dalam bentuk situs web yang dikelola oleh manager untuk menampilkan hasil perhitungan LCC yang dibutuhkan melalui aplikasi sehingga dapat membantu dalam memperkirakan biaya jangka panjang mesin. Data yang digunakan berasal dari organisasi sekaligus menjadi rujukan dalam perhitungan perawatan mesin. Pengembangan aplikasi menggunakan metode *Waterfall* dibantu dengan

pemetaan sistem menggunakan UML. Pengembangan aplikasi dibantu dengan teknologi kerangka *Laravel* sebagai situs web, *MySQL* dan *XAMPP* dalam pengelolaan dan penyimpanan data aplikasi. Sebelum aplikasi diluncurkan akan dilakukan pengujian menggunakan metode *Black Box Testing*.

IV. ANALISIS

A. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional dibutuhkan dalam pengembangan sistem dengan cara mendefinisikan proses kegiatan yang akan diimplementasikan pada aplikasi dan menjabarkan kebutuhan aplikasi agar dapat berjalan sesuai kebutuhan. [6]. Analisis kebutsuhan fungsional akan dijelaskan pada tabel berikut.

TABEL 1
Analisis Kebutuhan Fungsional

Tugas	Kebutuhan Fungsional	Aktor
Melakukan perhitungan LCC	Formulir untuk memasukkan nama mesin, biaya inisiasi, biaya operasional per tahun, biaya perawatan per tahun, biaya pembongkaran dan masa hidup mesin.	Manajer
	Verifikasi data yang diinput user, menolak data berupa angka negatif, unsur abjad, atau karakter spesial lainnya.	
	Menampilkan hasil perhitungan menurut masukan data manajer.	
Melihat riwayat perhitungan LCC	Halaman utama yang menampilkan riwayat perhitungan LCC manajer	
	Menyediakan filter dan pengurutan untuk menampilkan riwayat perhitungan berdasarkan nama mesin, tanggal, dll	
	Menyediakan tombol cetak untuk mencetak riwayat perhitungan LCC dalam bentuk PDF atau <i>Excel</i>	

Melakukan perhitungan LCC dan melihat riwayat perhitungan LCC merupakan fungsi utama dalam aplikasi ini. Manajer dapat melakukan perhitungan dengan data yang sesuai klasifikasi dan dapat melihat riwayat hasil perhitungan yang telah dilakukan. Manajer juga dapat melakukan ekspor data dalam bentuk *Excel*.

B. Analisis Kesenjangan

Analisis kesenjangan adalah proses perbandingan antara keadaan yang sudah ada sebelumnya dalam sebuah perusahaan dengan kondisi yang diinginkan atau target yang telah ditetapkan. Tujuan dari analisis kesenjangan adalah untuk mengidentifikasi perbedaan atau kesenjangan antara performa aktual dengan performa yang diharapkan. Analisis

kesenjangan digunakan untuk mengidentifikasi kekurangan pada proses bisnis. [7]

Efisiensi dan efektifitas ditekankan saat permodelan ulang proses bisnis dengan tujuan membantu pembuatan keputusan dan keuntungan bisnis. Analisis kesenjangan adalah alat dengan tujuan evaluasi dan perencanaan. Dengan cara membantu organisasi dalam memahami dimana mereka berada, dimana mereka ingin pergi, dan bagaimana cara terbaik untuk mencapai tujuan tersebut. [8]. Untuk melambangkan tingkat sebuah kondisi terpenuhi maka kebutuhan akan diklasifikasi sebagai N, P, dan F. N melambangkan tidak terpenuhi sama sekali, P berarti kebutuhan terpenuhi tetapi sebagian saja. Sedangkan F berarti kebutuhan terpenuhi secara menyeluruh.

TABEL 2
Analisis Kesenjangan

Kebutuhan	Kondisi Aktual	Fulfilment			Solusi
		N	P	F	
Menentukan biaya pengoperasian mesin jangka panjang.	Tidak ada prosedur akan penentuan biaya pengoperasian mesin jangka panjang.	✓			Mendefinisikan rumus untuk menentukan pengoperasian mesin jangka panjang dengan metode LCC.
Menyimpan data biaya pengoperasian mesin jangka panjang.	Data ditulis secara manual dengan media kertas.		✓		Membuat aplikasi yang dapat menyimpan sekaligus mengekspor data perhitungan berbasis LCC.
Kalkulator untuk perhitungan pengoperasian mesin jangka panjang	Tidak ada aplikasi atau sarana lain untuk memudahkan perhitungan pengoperasian jangka panjang.	✓			Membuat aplikasi kalkulator berbasis LCC.

V. PERANCANGAN

A. Diagram Kasus

Diagram kasus penggunaan adalah salah satu jenis diagram dalam UML yang digunakan untuk merepresentasikan interaksi antara pengguna dengan sistem. Diagram ini memberikan gambaran visual tentang bagaimana aktor menggunakan sistem untuk mencapai tujuan tertentu. Diagram ini membantu dalam memahami interaksi antara manajer dan sistem, sehingga memastikan

bahwa sistem yang dikembangkan memenuhi kebutuhan manajer dengan baik.

Dengan menggambarkan interaksi aktor dengan sistem maka pengembang dapat mengetahui apa saja yang manajer bisa lakukan atau tidak. Selain itu menggambarkan bagaimana aktor lain terlibat dalam pengoperasian sistem. Diagram ini juga membantu dalam tahap pengujian dengan membedakan aksi apa saja yang dapat dilakukan oleh masing-masing aktor. [9]

Dengan diagram ini, pengembang dapat semakin mengenal peran aktor dalam aplikasi dan menyesuaikan aplikasi jika seandainya perlu perubahan. Hal ini menjembatani komunikasi antara kebutuhan dengan pengembang aplikasi.

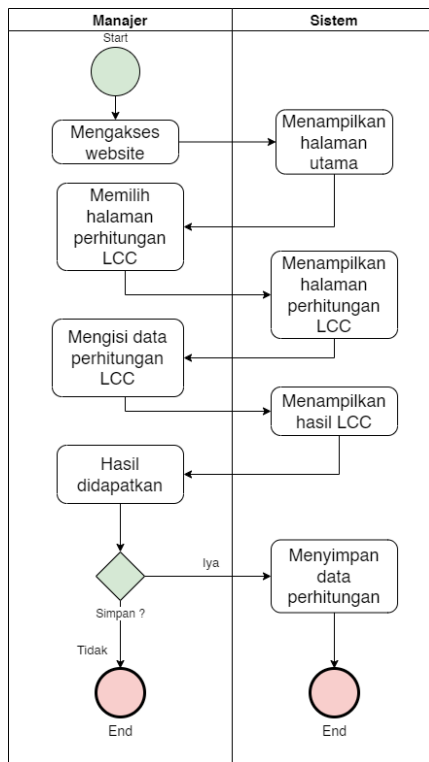


GAMBAR 2
Diagram Kasus

Diagram ini mengidentifikasi aktor dan perilaku sistem dalam proses analisis perancangan sistem. Berikut adalah fungsionalitas sistem pada aplikasi I-Maintenance. Aktor yang terlibat adalah manajer dan *admin*. Manajer dapat melakukan registrasi akun, masuk, keluar, melakukan perhitungan dan melihat riwayatnya. Sementara *admin* dapat mengelola manajer dengan menghapus akun manajer.

B. Diagram Aktivitas

Diagram Aktivitas adalah diagram yang menggambarkan urutan aktivitas yang terjadi dalam suatu proses sistem. Pembuatan diagram berdasarkan skenario kasus penggunaan yang telah dibuat sebelumnya. Diagram ini berisi aktivitas dan tindakan dalam suatu urutan yang merefleksikan hubungan aktivitas dan tindakan tersebut.



GAMBAR 3
Aktivitas Diagram Perhitungan

Pengguna yang sudah memasukan akun ke dalam aplikasi akan ditampilkan halaman utama dimana halaman perhitungan LCC dapat diakses. Pengguna memasukan data yang telah diklasifikasikan sesuai perhitungan LCC pada aplikasi ini dan memasukkannya pada halaman *input* data mesin. Data akan diterima dan diolah oleh aplikasi lalu menampilkannya kepada pengguna secara langsung. Hasil yang ditampilkan kepada pengguna mengikuti rumus sesuai dengan metode LCC. Pengguna dapat memilih tombol simpan untuk menyimpan data perhitungan. Data akan disimpan ke database agar dapat diakses pada halaman riwayat.

C. Diagram Relasi Entitas

Diagram relasi entitas adalah sebuah diagram yang menggambarkan hubungan antara entitas dalam sebuah basis data. ERD memberikan representasi visual mengenai struktur data yang ada dalam sistem. Hubungan tersebut menjelaskan kardinalitas dan opsionalitas antar entitas.

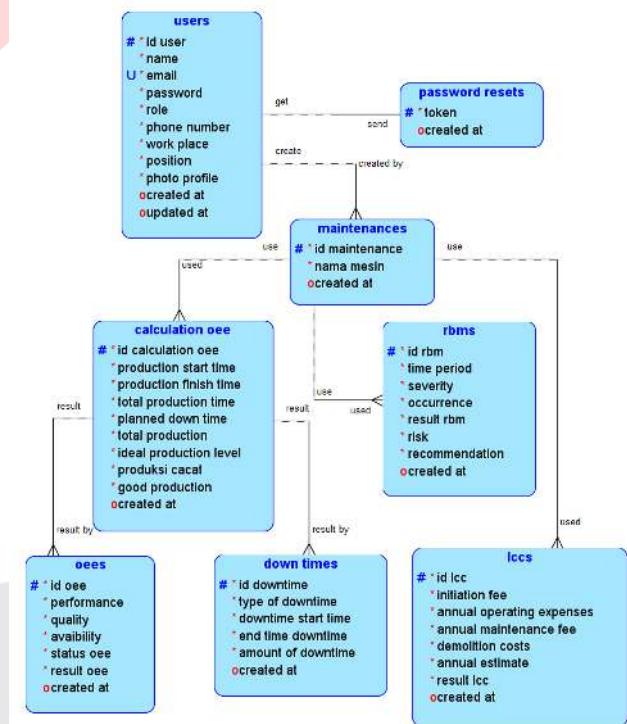
Terdapat empat entitas yang meliputi perhitungan LCC pada aplikasi yaitu *users*, *password resets*, *lccs*, dan *maintenances*.

Entitas *users* melambangkan pengguna yaitu manajer dan admin dan memiliki atribut utama yaitu *id user* sebagai *primary key*, beserta atribut lainnya yaitu *name*, *email*, *password*, *role*, *phone number*, *position*, *workplace* dan *photo profile*. Atribut *position* melambangkan kedudukan manajer dalam tempat dia bekerja sedangkan atribut *workplace* melambangkan tempat manajer bekerja. Manajer dengan admin dibedakan pada tabel *users* melalui atribut *role*.

Entitas *password resets* digunakan ketika manajer ingin merubah *password* atau lupa *password* ketika ingin menggunakan akunnya. Memiliki atribut *token* sebagai *primary key*. Dan atribut *created at* sebagai sebuah *timestamp*.

Entitas *lcc* melambangkan perhitungan LCC yang dilakukan berdasarkan masukan manajer. Entitas ini memiliki atribut utama yaitu *id lcc* sebagai *primary key*. *Initiation fee*, *annual operating expenses*, *annual maintenance free*, *demolition cost*, *annual estimate*, dan *result_lcc*. Atribut *result_lcc* melambangkan hasil perhitungan LCC yang telah dilakukan, sedangkan atribut lainnya kecuali atribut utama merupakan masukan yang digunakan untuk menghitung LCC.

Entitas *maintenances* memiliki atribut yaitu *id maintenance* sebagai *primary key*, *nama mesin*. Entitas ini melambangkan mesin agar dapat membedakan mesin dari masing-masing manajer sesuai metode perhitungan yang dilakukan.



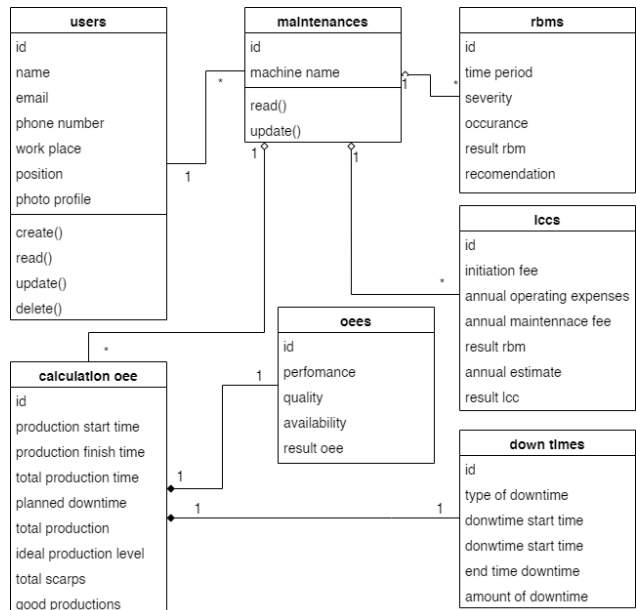
GAMBAR 4
Diagram Relasi Entitas

D. Diagram Kelas

Diagram kelas adalah salah satu jenis diagram yang digunakan dalam pemodelan perangkat lunak untuk menggambarkan struktur dan hubungan antara kelas-kelas dalam aplikasi. Diagram kelas memberikan gambaran visual tentang entitas-entitas (kelas), atribut-atribut yang dimiliki oleh kelas, serta hubungan dan interaksi antara kelas-kelas tersebut.

Terdapat tujuh kelas secara keseluruhan, yaitu *users*, *maintenances*, *rbms*, *lccs*, *calculation oee*, *oees*, dan *down times*. Untuk proses LCC menggunakan tiga kelas yaitu *users*, *maintenances*, dan *lccs*. Manajer dapat menambahkan dan menghapus

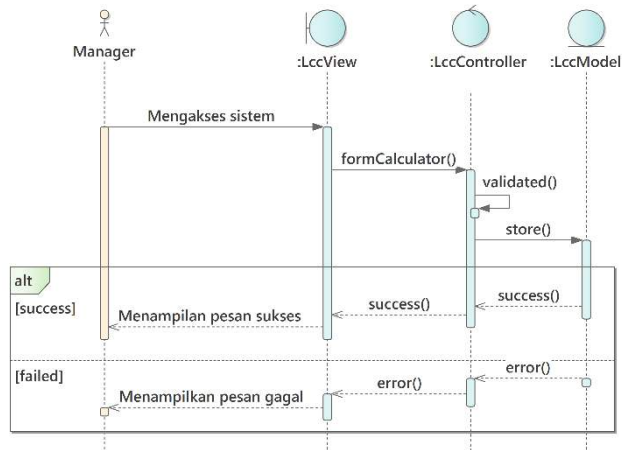
mesin atau perhitungan dengan metode `create()` dan `delete()`.



GAMBAR 5
Diagram Kelas

E. Diagram Urutan

Diagram urutan atau *sequence diagram* adalah jenis diagram yang digunakan dalam pengembangan sistem untuk menggambarkan interaksi antara objek-objek dalam sistem secara urutan kronologis. Diagram ini memberikan gambaran visual tentang bagaimana objek-objek berinteraksi satu sama lain dan mengirim pesan-pesan dalam urutan waktu tertentu.



GAMBAR 6
Diagram Urutan Perhitungan

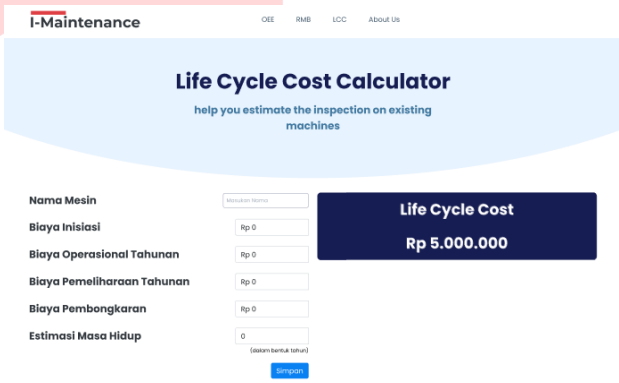
Manajer mengakses aplikasi dengan memasuki halaman perhitungan LCC. Halaman ini ditampilkan melalui komponen *view* yaitu `LccView`. Manajer menginputkan data perhitungan dan ditampung dalam fungsi `formCalculator()`. Data tersebut akan dialihkan kepada *controller* `LccController`. Pada komponen *controller*, perlu dilakukan validasi untuk mengecek kondisi akun sudah masuk atau belum karena data dapat tersimpan

hanya jika manajer memasuki sistem menggunakan sebuah akun.

Jika akun sudah masuk kedalam aplikasi, fungsi `store()` membawa data dan menyimpan data perhitungan melalui model `LccModel`. Data akan tersimpan jika semua data memenuhi kebutuhan aplikasi yaitu tidak ada kolom *input* yang kosong dan bentuk angka tidak boleh angka negatif. Jika data berhasil disimpan maka konfirmasi berhasil akan disampaikan kepada manajer dengan fungsi `success()`. Jika terjadi kegagalan menyimpan data, maka diberikan konfirmasi gagal melalui fungsi `error()`.

F. Desain

Desain aplikasi dibutuhkan sebagai tampilan depan pada proses pengembangan aplikasi yang berbentuk mock up. *Mock up* ini merupakan visualisasi tampilan yang menjadi acuan pada pengembangan bagian *front-end*. Sebaiknya hasil akhir aplikasi memiliki kesamaan yang tinggi atau sama persis dengan *mock up* agar aplikasi sesuai dengan tujuan awal.



GAMBAR 7
Desain Perhitungan

VI. PENGUJIAN

A. Black Box Testing

Black box testing adalah salah satu jenis pengujian perangkat lunak yang dilakukan dengan fokus pada *input* dan *output* yang dihasilkan oleh perangkat lunak tanpa mengetahui rincian implementasinya. Pendekatan ini mencerminkan cara pandang "kotak hitam" (*black box*) terhadap perangkat lunak, di mana penguji hanya melihat perangkat lunak sebagai sebuah entitas eksternal yang mengambil *input* dan menghasilkan *output* tanpa mengetahui cara kerjanya di dalamnya.

TABEL 3
Black Box Testing

Uji Kasus	Masukan	Ekspektasi	Hasil
Memasukan nama mesin, biaya inisiasi, biaya operasional tahunan, biaya pemeliharaan tahunan, biaya pembongkaran, dan estimasi masa hidup dengan tepat.	Nama mesin: Goss Community Biaya inisiasi: 95,000,000 Biaya operasional: 20,000,000 Biaya pemeliharaan: 8,500,000 Biaya pembongkaran: 47,500,000	Data tersimpan karena lengkap dan hasil LCC adalah 570.000.000 dan ditampilkan pesan "Sukses, berhasil"	Ditampilkan pesan "Sukses, berhasil menyimpan data" dengan hasil LCC 570.000.000. Data tersimpan pada database.

	Masa hidup: 15	menyimpan data".	
--	----------------	------------------	--

B. Pengujian Terhadap Mesin

Data yang dibutuhkan untuk melakukan pengujian terhadap mesin sudah didapatkan yaitu data biaya untuk penambahan mesin baru dan data biaya untuk merawat dan mengoperasikan mesin yang ada. Hasil berserta data biaya untuk penambahan mesin dan mesin yang sudah ada akan dijelaskan pada tabel berikut.

TABEL 4
Perhitungan LCC Mesin Baru

Data	Biaya
Biaya Inisiasi	Rp. 95,000,000
Biaya Operasional Tahunan	Rp. 20,000,000
Biaya Pemeliharaan Tahunan	Rp. 8,500,000
Biaya Pembongkaran	Rp. 47,500,000
Masa Hidup	15 Tahun
Nilai LCC	Rp. 570,000,000

TABEL 5
Perhitungan LCC Mesin Saat Ini

Data	Biaya
Biaya Inisiasi	Rp. 95,000,000
Biaya Operasional Tahunan	Rp. 20,000,000
Biaya Pemeliharaan Tahunan	Rp. 8,500,000
Biaya Pembongkaran	Rp. 47,500,000
Masa Hidup	15 Tahun
Nilai LCC	Rp. 570,000,000

Setelah mengetahui nilai LCC dari mesin baru dan mesin yang sudah ada, nilai LCC tersebut dapat digunakan untuk membandingkan keputusan untuk menambah mesin baru atau merawat mesin yang sudah ada. Maka dapat diputuskan bahwa lebih menguntungkan untuk menggunakan mesin yang sudah ada dengan nilai LCC yang lebih rendah yaitu Rp. 431,500,000 dibandingkan mesin baru yaitu Rp. 570,000,000.

VII. KESIMPULAN

Setelah melakukan analisa terhadap proses bisnis eksisting, ditemukan beberapa kekurangan yang dapat dikoreksi pada proses bisnis targeting. Kesenjangan yang berada di antara proses bisnis eksisting dengan targeting disertai dengan pengembangan aplikasi perawatan berdasarkan LCC menggunakan metode Waterfall dan didefinisikan dengan bentuk UML. Fitur utama aplikasi tersebut adalah kalkulator LCC dan ekspor data perhitungan yang telah dilakukan menjadi bentuk Excel. Untuk melakukan perhitungan LCC, manajer harus mengumpulkan data relevan dan mengklasifikasikan data sesuai dengan kebutuhan aplikasi. Data tersebut meliputi biaya inisial, biaya operasional, biaya pemeliharaan, biaya pembongkaran, dan estimasi masa hidup. Dikarenakan data harus diklasifikasikan sesuai standar aplikasi maka

perhitungan LCC spesifik atau kasus tertentu tidak dapat dilakukan.

Sebagai acuan dalam perhitungan LCC digunakan mesin cetak Goss Community pada Serambi Indonesia. Perbandingan dilakukan dengan menghitung nilai LCC penambahan mesin baru untuk menggantikan mesin yang sudah ada. Dikarenakan nilai LCC penambahan mesin yaitu Rp. 570,000,000 lebih besar daripada nilai LCC mesin yang sudah ada yaitu Rp. 431,500,000 sehingga lebih menguntungkan untuk menggunakan dan merawat mesin yang sudah ada.

Penelitian ini memiliki banyak kekurangan yang dapat diperbaiki atau bisa dikembangkan kembali. Dikarenakan batasan berupa data yang diterima harus sesuai dengan standar dari aplikasi ini, menurunkan kompatibilitas studi kasus. Maka alangkah baiknya untuk melakukan pengerjaan selanjutnya meliputi aspek terutama menyediakan pilihan untuk menerima puluhan data sesuai dengan klasifikasinya dan penambahan fitur menghitung dengan bunga.

REFERENSI

- [1] J. M. T. Farinha, *Asset maintenance engineering methodologies*. CRC Press, 2018.
- [2] S. Fuller and S. Petersen, "Life-cycle costing manual for the federal energy management program, NIST Handbook 135," 1996.
- [3] D. M. Frangopol and M. Liu, "Maintenance and management of civil infrastructure based on condition, safety, optimization, and life-cycle cost," *Structures and Infrastructure Systems*, pp. 96–108, 2019.
- [4] W. T. França, M. V. Barros, R. Salvador, A. C. de Francisco, M. T. Moreira, and C. M. Piekarski, "Integrating life cycle assessment and life cycle cost: a review of environmental-economic studies," *Int J Life Cycle Assess*, vol. 26, no. 2, pp. 244–274, 2021, doi: 10.1007/s11367-020-01857-y.
- [5] B. B. Agarwal, S. P. Tayal, and M. Gupta, *Software engineering and testing*. Jones & Bartlett Learning, 2010.
- [6] K. Chari and M. Agrawal, "Impact of incorrect and new requirements on waterfall software project outcomes," *Empir Softw Eng*, vol. 23, pp. 165–185, 2018.
- [7] W. Guo, Z. Yang, Q. Feng, C. Dai, J. Yang, and X. Lei, "A new method for band gap analysis of periodic structures using virtual spring model and energy functional variational principle," *Mech Syst Signal Process*, vol. 168, p. 108634, 2022.
- [8] B. Sapkota, S. Palaian, S. Shrestha, A. Ozaki, M. I. Mohamed Ibrahim, and M. Jakovljevic, "Gap analysis in manufacturing, innovation and marketing of medical devices in the Asia-Pacific region," *Expert Rev Pharmacoecon Outcomes Res*, vol. 22, no. 7, pp. 1043–1050, 2022.
- [9] R. Fauzan, D. Siahaan, S. Rochimah, and E. Triandini, "Use case diagram similarity measurement: A new approach," in *2019 12th International Conference on Information &*

- Communication Technology and System (ICTS)*, IEEE, 2019, pp. 3–7.
- [10] F. Ciccozzi, I. Malavolta, and B. Selic, “Execution of UML models: a systematic review of research and practice,” *Softw Syst Model*, vol. 18, pp. 2313–2360, 2019.

