

Sistem *Preventive Maintenance* Berbasis Web dengan Menggunakan Algoritma *Priority Scheduling* pada PT. Beta Pharmacon

Bayu Priyatna

Universitas Buana Perjuangan Karawang

bayu.priyatna@ubpkarawang.ac.id

Topan Trianto

Universitas Ma'soem

topant2018@gmail.com

Julifer P Manurung

Universitas Buana Perjuangan Karawang

si18.juliferp.manurung@mhs.ubpkarawang.ac.id

Nono Heryana

Universitas Buana Perjuangan Karawang

nonoheryana@staff.unsika.ac.id

Arip Solehudin

Universitas Buana Perjuangan Karawang

arip.solehudin@unsika.ac.id

ABSTRACT

PT. Beta Pharmacon in the actual preparation of Preventive Maintenance (PM) schedules and input of reports or preventive results data still uses Microsoft Excel, and preventive schedules that have been made are sometimes not following the actual because the machine cannot be maintained (the production process has not been completed) so that the schedule has been completed. Determined to change and to confirm whether preventive maintenance can be done every day, still by telephone. Based on the above problems, a web-based Preventive Maintenance System will be created using the priority scheduling algorithm. The web is built using the CodeIgniter framework and database using MySQL. With this system, it can facilitate scheduling of preventive maintenance, data collection of machines and tools in the company, search for data on preventive maintenance results, coordination of the implementation of preventive maintenance, so that it can provide the information needed effectively and efficiently.

Keywords: *Priority Scheduling, CodeIgniter, MySQL, Maintenance, Monitoring.*

ABSTRAK

PT. Beta Pharmacon secara aktual dalam penyusunan jadwal *Preventive Maintenance* (PM) dan input laporan atau data hasil *preventive* masih menggunakan *Microsoft Excel*, dan jadwal *preventive* yang sudah dibuatkan terkadang tidak sesuai dengan aktual dikarenakan mesin belum bisa dilakukan perawatan (proses produksi yang belum selesai), sehingga jadwal yang sudah ditentukan berubah dan untuk konfirmasi apakah bisa dilakukan *preventive maintenance* setiap harinya masih melalui telepon. Berdasarkan masalah diatas akan dibuatkan Sistem *Preventive Maintenance* berbasis web dengan menggunakan *algoritma priority scheduling*. Web dibangun menggunakan *framework CodeIgniter* dan database menggunakan *MySQL*. Dengan adanya sistem ini dapat mempermudah dalam penjadwalan *preventive maintenance*, pendataan mesin dan alat yang ada di perusahaan, pencarian data hasil

preventive maintenance, koordinasi tentang pelaksanaan *preventive maintenance*, sehingga dapat menyediakan informasi yang dibutuhkan secara efektif dan efisien.

Kata Kunci : *Priority Scheduling, CodeIgniter, MySQL, Maintenance, Monitoring.*

1. PENDAHULUAN

Teknologi informasi berkembang dengan sangat pesat, seperti perkembangan pada industri perangkat keras dan perangkat lunak. Dibandingkan dengan beberapa tahun yang lalu teknologi informasi yang ada jauh lebih baik. Perkembangan teknologi informasi pada saat ini menjadikan keberadaan komputer sangat diperlukan, agar pekerjaan manusia dapat dikerjakan dengan lebih mudah, karena komputer bisa bekerja dengan cepat, multi tasking, tepat dan akurat. Selain itu komputer dapat digunakan untuk menyimpan data dalam jumlah yang besar dengan aman, dan dapat di tampilkan kembali sesuai dengan kebutuhan.

PT. Beta Pharmacon merupakan salah satu member dari *Dexa Group*, perusahaan yang bergerak dalam bidang farmasi (obat-obatan) yang terletak di Kawasan Industri Suryacipta, Jl. Surya Madya Kav I-18C2, Karawang Jawa barat. Perusahaan yang mengedepankan ketahanan dan performa mesin, pemeliharaan dan perawatan mesin harus dilakukan secara rutin. Hal ini untuk menjamin dan memastikan mesin dapat berfungsi dengan baik saat proses produksi, serta dapat mendukung tercapainya hasil produksi (output). Jadwal perawatan mesin (*Preventive maintenance*) dapat mengurangi *down time* (waktu mesin berhenti karena kerusakan) sehingga permasalahan yang akan terjadi pada mesin sudah dapat diketahui terlebih dahulu.

Penjadwalan *Preventive Maintenance* pada *PT Beta Pharmacon* saat ini masih dilakukan dengan menghitung tanggal pelaksanaan *preventive maintenance* secara manual. Selain itu dokumentasi laporan pelaksanaan *preventive maintenance* belum ada, sehingga kurang termonitoring. Pencarian data *preventive maintenance* masih membutuhkan waktu lama karena terdiri dari beberapa *file excel* dan koordinasi pelaksanaan *Preventive maintenance* masih dilakukan melalui komunikasi telepon sehingga proses pelaksanaan *Preventive maintenance* bisa terhambat karena koordinasi yang kadang berlangsung dengan lama.

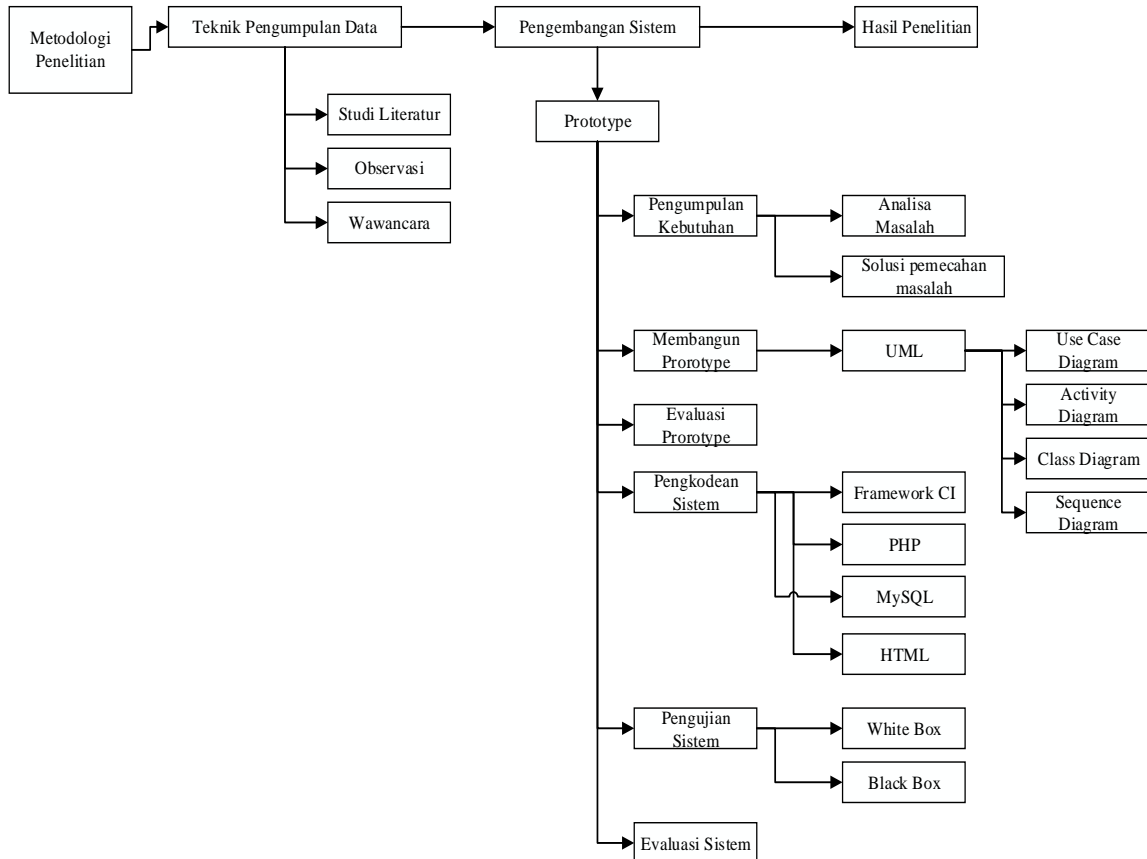
Berdasarkan latar belakang masalah tersebut dapat dispesifikan permasalahan agar dapat diuraikan dengan baik dengan harapan mendapatkan solusi untuk masalah tersebut dengan merancang sistem informasi *Preventive Maintenance* secara otomatis.

2. METODOLOGI

Dalam penyusunan penelitian ini, menggunakan susunan tahapan penelitian dengan model pengembangan sistem *Prototype*. Adapun diagram alur penelitian dapat dilihat pada gambar 1. Metode dalam pengumpulan data pada penelitian yaitu :

1. Studi Pustaka
Metode ini menggunakan jurnal-jurnal, laporan yang berkaitan dengan judul yang diangkat sebagai referensi. Informasi tersebut diambil dari berbagai sumber, baik dari dalam maupun dari luar perusahaan.
2. Observasi
Penulis langsung melakukan pengamatan terhadap kegiatan yang berhubungan dengan masalah yang diambil, agar memperoleh data dan keterangan yang lebih lengkap. Pengamatan dilakukan secara langsung ke *PT Beta Pharmacon*.
3. Wawancara
Penulis melakukan wawancara langsung dan Tanya jawab dengan orang-orang yang

terlibat seperti teknisi engineering, teknisi machinery, teknisi utility & Facility, beserta supervisor masing-masing bagian tersebut untuk mengetahui proses transaksi yang terjadi saat proses *Preventive maintenance* [17].



Gambar 1. Diagram Alur Penelitian

Tahapan dalam pengembangan sistem menggunakan *Sytem Development Life Cycle (SDLC)* yang berpatokan terhadap model dan proses yang digunakan oleh pengembang untuk membuat sistem perangkat lunak dan menguraikan proses, yaitu untuk mendapatkan solusi atas permasalahan yang ada [1]. Pengembangan sistem ini menggunakan metode Prototype. Prototype adalah suatu metode pengembangan sistem, yang terdiri atas model fisik kerja sistem dan berfungsi sebagai versi awal dari sistem. Dengan metode prototyping ini akan dihasilkan prototype sistem sebagai perantara pengembang dan pengguna agar dapat berinteraksi dalam proses kegiatan pengembangan sistem informasi [2]. Keuntungan dalam menggunakan metode *prototype* yaitu :

- 1) Menghemat waktu dalam pengembangan sistem
- 2) Penentuan kebutuhan lebih mudah diwujudkan
- 3) Klien berpartisipasi aktif dalam pengembangan sistem, sehingga hasil perangkat lunak mudah disesuaikan dengan kebutuhan dan keinginan pelanggan
- 4) Pengembang dapat lebih mudah dalam menentukan kebutuhan pelanggan [3].

Dalam proses perancangan suatu sistem dengan menggunakan model *prototype*, terdapat beberapa langkah-langkah yang dapat dilakukan yaitu [4]:

- 1) Pengumpulan Kebutuhan

Mengidentifikasi apa saja yang dibutuhkan oleh Pengguna. Menganalisa sistem yang

berjalan agar mampu menemukan model yang dibutuhkan oleh pemakai sistem.

- 2) **Membangun Prototype**
Menganalisa sistem yang terkait untuk mendapatkan gambaran *prototype* yang dibutuhkan. Lalu membuat perancangan yang dapat disajikan kepada pengguna. Dalam penelitian ini, pemodelan sistem dibangun dengan *Unified Modelling Language (UML)* yang merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan *Diagram* teks-teks pendukung [5]. UML sendiri juga memberikan standar penulisan sebuah *system blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem software [6]. UML mendefinisikan *Diagram-Diagram* sebagai *Use case Diagram, Class Diagram, Activity Diagram, Sequence Diagram* [7].
- 3) **Evaluasi Prototype**
Evaluasi ini dilakukan oleh Pengguna untuk menentukan *prototype* yang dibangun sudah sesuai dengan yang dibutuhkan. Jika sudah sesuai maka akan dilanjutkan ke tahap selanjutnya, dan jika belum sesuai maka harus mengulang ke tahap sebelumnya [8].
- 4) **Pengkodean Sistem**
Dalam proses ini model *prototyping* yang telah disetujui dan disepakati diterjemahkan atau diaplikasikan kedalam bentuk kode (*coding*) sesuai dengan bahasa pemrograman yang relevan. Pemrograman menggunakan Bahasa web programming Hypertext Preprocessor (PHP) [9], menggunakan toolkit framework CodeIgniter [10], menggunakan database sebagai tempat menyimpan data [11], dan MySQL sebagai pengakses database melalui jaringan [12].
- 5) **Pengujian Sistem**
Pengujian dilakukan dengan menggunakan metode blackbox dan whitebox. Pengujian whitebox difokuskan pada internal sistem yaitu source code program [13],[14]. Pengujian Blackbox dilakukan pengujian aplikasi sistem yang melibatkan user, yang bertujuan untuk mengetahui segi kekurangan terhadap sistem aplikasi yang telah dibangun [15],[16].

3. ANALISA DAN PERANCANGAN SISTEM

Berdasarkan hasil analisa maka dapat dirumuskan sebagai berikut:

- 1) Membangun sistem yang dapat mengatur jadwal *preventive maintenance*.
- 2) *User* dapat melihat dan mengubah jadwal *preventive maintenance* yang sudah ditentukan.
- 3) Database sebagai media untuk menyimpan data-data mesin, data jadwal dan data hasil pelaksanaan *preventive maintenance*.
- 4) Semua data transaksi *preventive maintenance* tersimpan dan dapat diakses dengan cepat saat dibutuhkan.

Sistem *preventive maintenance* yang dirancang menggunakan *Unified Modeling Language (UML)*, dan dengan menggunakan pemrograman Bahasa PHP dan *Framework Codeigniter*. Sedangkan perancangan sistem *database* menggunakan *class diagram* dan mengimplementasikan dengan MySQL.

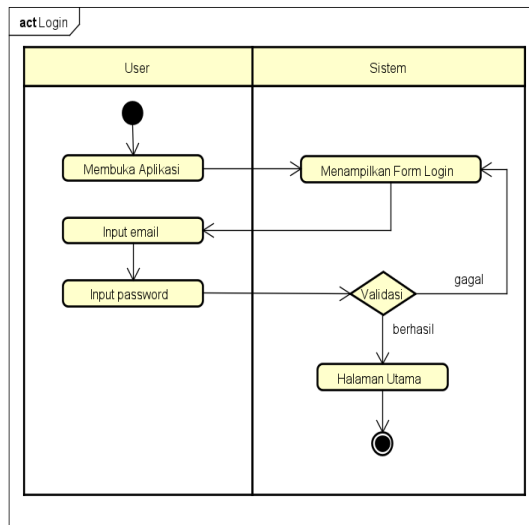
1) *Use Case Diagram*

Use Case Diagram menggambarkan interaksi antara pengguna sistem dengan sistem itu sendiri, yang digunakan untuk menggambarkan fungsionalitas dari sebuah sistem. Pada sistem usulan yang diajukan terdapat terdapat admin dan member sebagai aktornya. *Member* yang dimaksud adalah masing-masing *user per department* yang terlibat dalam kegiatan *preventive maintenance*. *Usecase Diagram Preventive Maintenance* dapat dilihat pada gambar 2.

2) *Activity Diagram*

a. *Activity Diagram Login Actor*

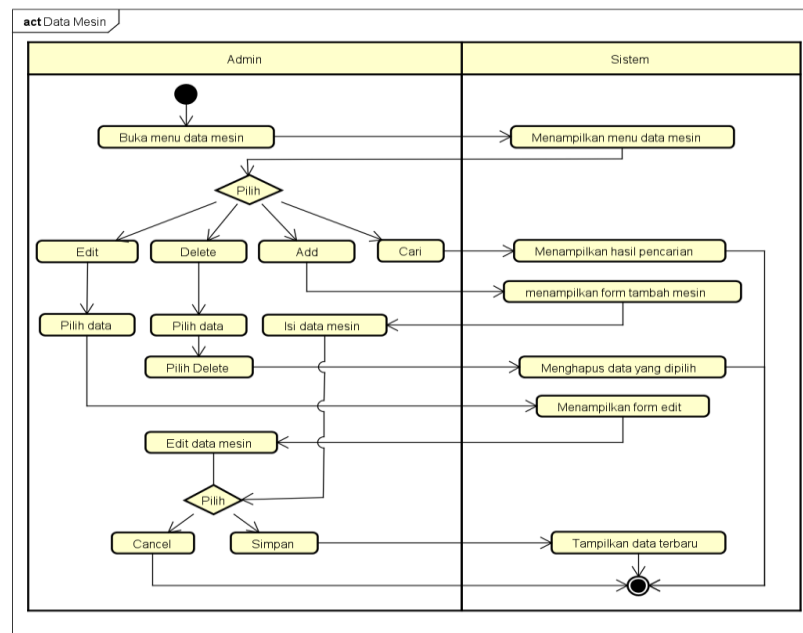
Aktivitas dimulai dari *Admin, member, supervisor (user)* membuka aplikasi dan aplikasi



Gambar 3. Activity Diagram Login Actor

b. Activity Diagram Data Mesin

Aktivitas dimulai saat admin memilih menu data mesin, kemudian sistem akan menampilkan halaman data mesin. Data mesin dapat dikelola oleh admin yaitu *Edit*, *Delete*, *add*, dan *cari*. Data yang akan di *Edit*, kemudian sistem akan menampilkan *form Edit* data mesin, lalu pilih simpan untuk melakukan penyimpanan data dan *cancel* untuk membatalkan *Edit* data. *Delete* data akan menghapus data yang dipilih. *Add* data untuk menambahkan data mesin baru, sistem akan menampilkan *form* tambah mesin, kemudian *admin* dapat mengisi data mesin dan pilih simpan untuk menyimpan data mesin baru dan *cancel* untuk membatalkan. Kemudian sistem akan otomatis meng-*update* data dan menampilkan data terbaru sesuai data terbaru. Activity Diagram data mesin dapat dilihat dalam gambar 4.



Gambar 4. Activity Diagram Data Mesin

3) Sequence Diagram

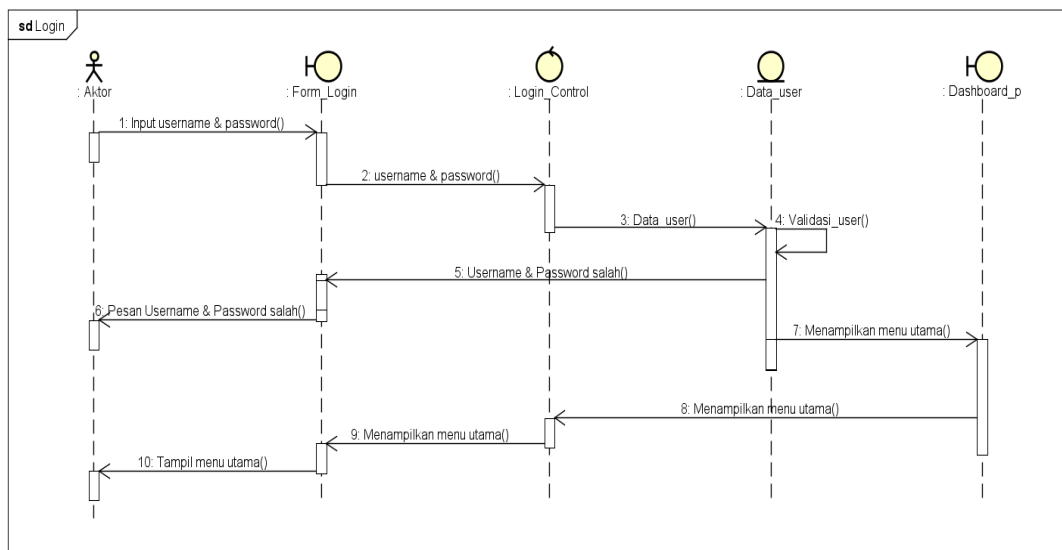
a. Sequence Diagram Login

Pada perancangan ini digambarkan hubungan antara actor dan sistem saat proses *login*. Actor menginput *username* dan *password* pada *form login*, kemudian *username* dan

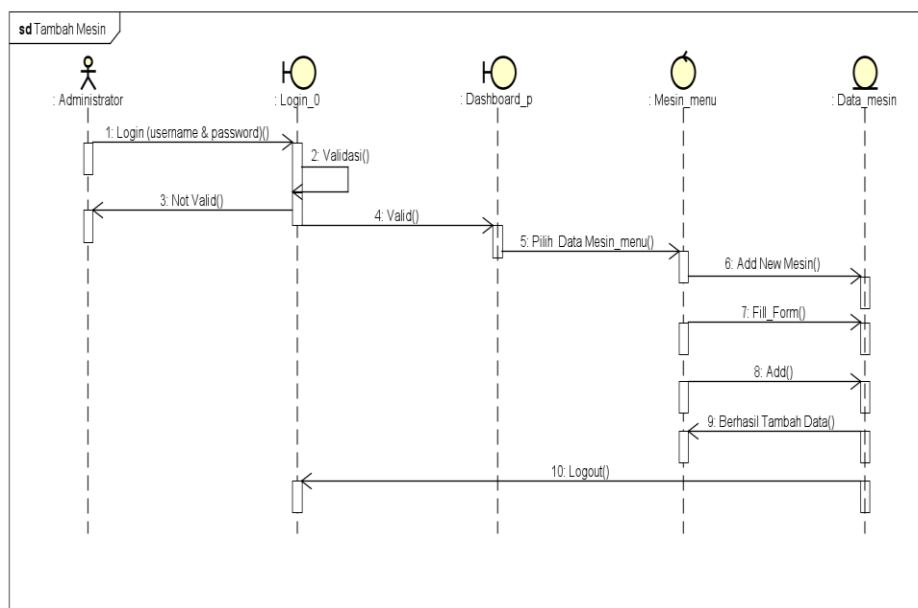
password akan di validasi, jika salah maka akan ada pesan dari sistem, dan jika benar maka sistem akan masuk dan menampilkan menu utama ke actor. *Sequence Diagram login* dapat dilihat dalam gambar 5.

b. *Sequence Diagram* Tambah Data Mesin

Pada perancangan ini digambarkan hubungan antara admin dan sistem saat proses tambah data mesin. Pada halaman utama setelah berhasil login, admin memilih Data Mesin dan memilih tombol Add, kemudian sistem akan menampilkan form penambahan data mesin untuk diisi oleh admin. Setelah semua data sudah terisi, pilih add. Kemudian sistem akan memberikan informasi bahwa data mesin sudah berhasil ditambah dan kembali ke tampilan data mesin terbaru. *Sequence Diagram* tambah mesin dapat dilihat dalam gambar 6.



Gambar 5. *Sequence Diagram Login*

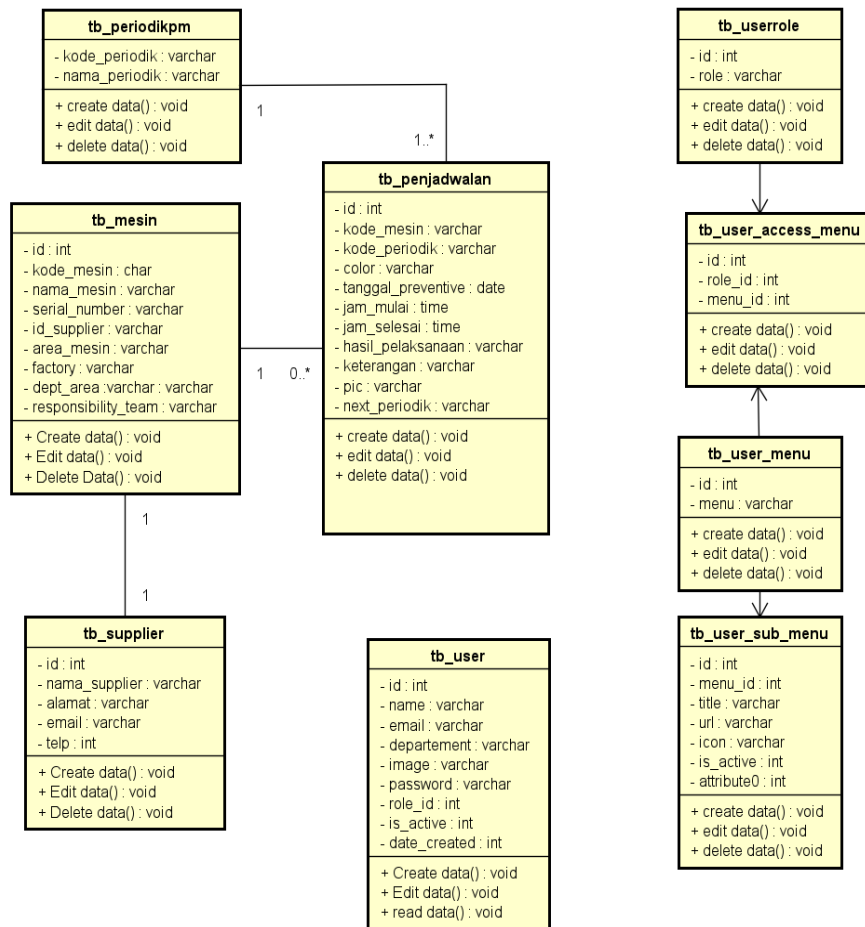


Gambar 6. *Sequence Diagram Tambah Data Mesin*

4) *Class Diagram*

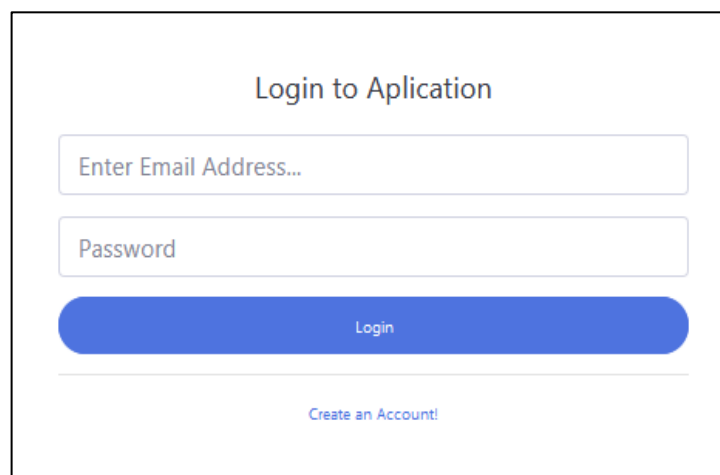
Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem [15]. Kelas memiliki apa yang disebut atribut dan

metode atau operasi. *Class diagram preventive maintenance* dapat dilihat pada gambar 7.



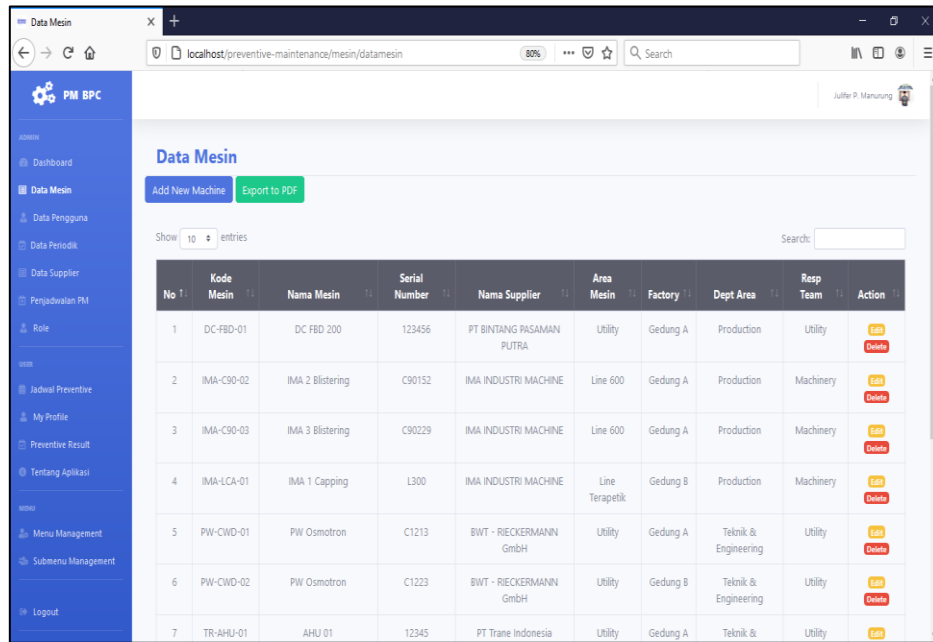
Gambar 7. *Class Diagram*

- 5) Pengkodean Sistem
 - a. Tampilan Login Admin dan Member



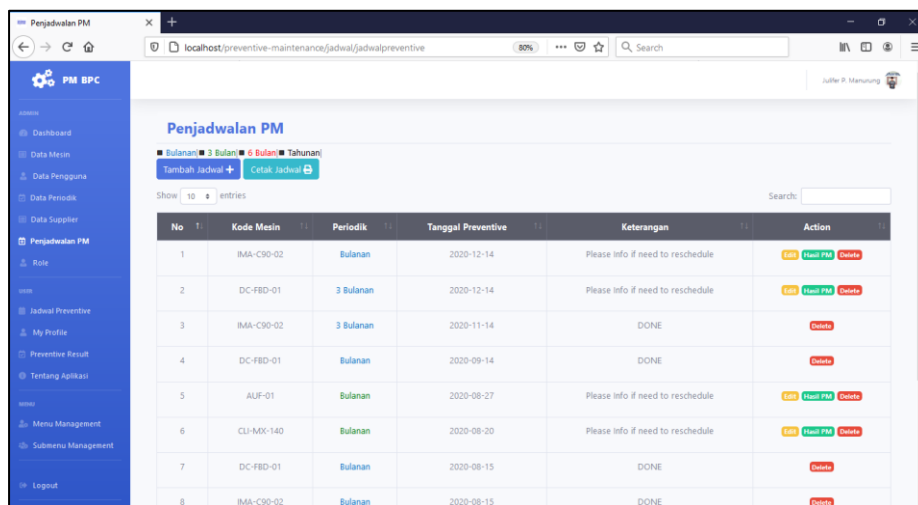
Gambar 11. *Tampilan Login Actor*

b. Tampilan Data Mesin



Gambar 12. Tampilan Data Mesin

c. Tampilan Jadwal Preventive Maintenance



Gambar 13. Tampilan Jadwal Preventive Maintenance

6) Pengujian Sistem

a. Black Box Testing

Pengujian *Black Box* dilakukan untuk menguji sistem terkait prosedur dan alur logika dalam kode program yang dibuat.

Tabel 1. Pengujian *Black Box* Login Actor

Pengujian dan Hasil Uji				
Yang diuji	Data Masukan	Yang diharapkan	Hasil Pengujian	Status
Login	Email dan Password	Username dan password benar, maka akan	Admin masuk ke halaman	Berhasil

		menampilkan halaman dashboard administrator	dashboard admin	
<i>Login</i>	<i>Email dan Password</i>	Username dan password salah, maka tidak bisa masuk ke menu utama dan kembali kehalaman login	Admin tidak masuk ke menu utama dan kembali kehalaman login	Berhasil

Tabel 2. Pengujian Black Box Menu Data Mesin

Pengujian dan Hasil Uji				
Yang diuji	Data Masukan	Yang diharapkan	Hasil Pengujian	Status
Button Add	Data-data mesin	Jika semua data mesin sudah terisi, maka data dapat disimpan, dan jika ada data yang tidak terisi, maka data tidak dapat disimpan dan tampil pemberitahuan pengisian data	Dapat melakukan penyimpanan data. Dan saat ada data yang tidak terisi, tidak dapat menyimpan data	Berhasil
Button <i>Edit</i>	Data mesin yang akan di <i>Edit</i>	Data mesin ter- <i>Edit</i> sesuai kebutuhan	Data mesin berhasil di <i>Edit</i>	Berhasil
Button <i>Delete</i>	Data mesin yang akan dihapus	Data mesin terhapus	Data mesin berhasil dihapus	Berhasil
Button Print				

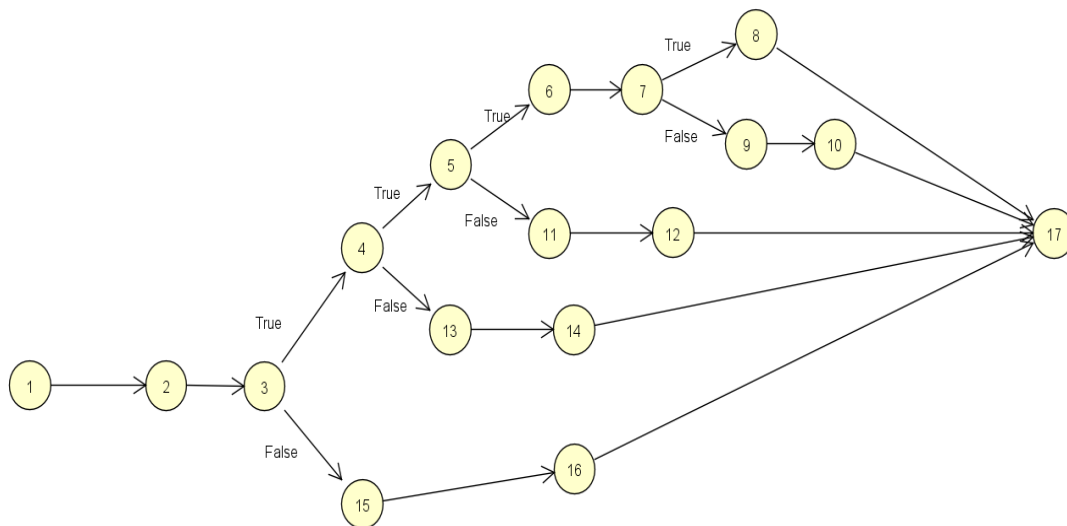
b. White Box Testing

Tahapan ini dilakukan pengujian *white box* terhadap *source code* program awal, berikut merupakan *source code*

Tabel 3. Pengujian White Box Source code login

No	Source Code
1	<code>private function _login() {</code>
2	<code>\$user = \$this->db->get_where('tb_user', ['email' => \$email])->row_array();</code>
3	<code>if (\$user) {</code>
4	<code>if (\$user['is_active'] == 1) {</code>
5	<code>if (password_verify(\$password, \$user['password'])) {</code>
6	<code>\$data = [</code>

	'email' => \$user['email'], 'role_id' => \$user['role_id']]; \$this->session->set_userdata(\$data);
7	if (\$user['role_id'] == 1) {
8	redirect('admin');
9	} else {
10	redirect('user');
11	} else {
12	\$this->session->set_flashdata('message', '<div class="alert alert-danger" role="alert"> Wrong password!</div>'); redirect('auth');
13	} else {
14	\$this->session->set_flashdata('message', '<div class="alert alert-danger" role="alert"> This Email has not been activated! Please contact Your Administrator.... </div>'); redirect('auth');
15	} else {
16	\$this->session->set_flashdata('message', '<div class="alert alert-danger" role="alert"> Email is not registered!</div>'); redirect('auth');
17	}



Gambar 14. Pengujian Whitebox Login Actor

Dengan data diatas, diketahui bahwa Jumlah panah / edge (E) = 20, Jumlah lingkaran / node (N) = 17, Predicate (P) = 4 kondisi, sehingga Cyclomatic Complexity untuk Flowgraph G adalah sebagai berikut :

$$V(G) = E - N + 2$$

$$V(G) = 20 - 17 + 2$$

$$V(G) = 5$$

$$V(G) = P + 1$$

$$V(G) = 4 + 1$$

$$V(G) = 5$$

Pada proses *login* terdapat 5 skenario yaitu :

- a. 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 17
- b. 1 – 2 – 3 – 4 – 5 – 6 – 7 – 9 – 10 – 17
- c. 1 – 2 – 3 – 4 – 5 – 11 – 12 – 17
- d. 1 – 2 – 3 – 4 – 13 – 14 – 17
- e. 1 – 2 – 3 – 15 – 16 – 17

4. PENUTUP

Dengan dirancangnya sebuah sistem informasi *Preventive maintenance* yang diuraikan pada bab-bab sebelumnya, maka dapat ditarik kesimpulan sebagai berikut:

- 1) Perancangan sistem *Preventive maintenance* dengan menggunakan model pengembangan *prototype* ini, dapat menjawab kebutuhan karyawan terkait permasalahan penjadwalan *Preventive maintenance* pada PT. Beta Farmacon.
- 2) Hasil dan jadwal *preventive* dapat menjadi lebih informatif yang dapat digunakan untuk kebutuhan penjadwalan *maintenance* dan bisa diakses oleh karyawan yang membutuhkan.
- 3) Jadwal yang sudah diatur dapat di *re-schedule* kembali oleh karyawan apabila jadwal tidak memungkinkan untuk dilakukan, sehingga proses *maintenance* dapat menyesuaikan terhadap waktu jalannya dan berhentinya mesin.

Adapun saran-saran yaitu :

1. Aplikasi yang dibangun masih sederhana sehingga dapat melakukan pengembangan lebih lanjut seperti penambahan *form ceklist Preventive maintenance* semua alat dan mesin, sehingga *ceklist* dapat langsung di isi oleh teknisi melalui sistem.
2. Karyawan harus selalu melakukan *update* setiap hari untuk jadwal *preventive maintenance* pada sistem yang sudah dirancang, agar jadwal yang sudah ditentukan dapat terealisasi dengan baik.

DAFTAR PUSTAKA

- Abdul, R., Raden K. Hardinto, Annurrahman, dan D. Susanti. 2018. *Sistem Informasi Pembayaran Tambahan Penghasilan Berdasarkan Beban Kerja Pada Dinas Pendidikan Dan Kebudayaan Kabupaten Tanah Bumbu*. vol. 4, no. 1, pp. 1–11.
- Destiningrum, M. dan Q. J. Adrian. 2017. *Sistem Informasi Penjadwalan Dokter Berbassis Web Dengan Menggunakan Framework Codeigniter (Studi Kasus: Rumah Sakit Yukum Medical Centre)*. *J. Teknoinfo*, vol. 11, no. 2, pp. 30–37.
- Dwi, A., M. Priyatmoko, dan A. E. Burhandenny. 2019. *Mulawarman Berbasis Web Menggunakan Framework Codeigniter*. *J. Ind. Galuh*, vol. 1 No 2, pp. 69–79.
- Hananto, A. L. dan B. Priyatna. 2017. *Rancang Bangun Aplikasi Informasi Harga Produk*. *TechnoXplore J. Ilmu Komput. Teknol. Inf.*, vol. 2, no. 1, pp. 10–20.
- Hidayat, T. dan Sukisno. 2018. *Rancang Bangun Sistem Penerimaan Mahasiswa Baru Online dengan Model SDLC Metode Prototipe di Universitas Islam Syekh-Yusuf*. Ed. XVIII, vol. Volume 2, pp. 161–177.
- Hilabi, S. S., 2017. *Rancang Bangun Situs Responsif Di Universitas Buana Perjuangan*

Karawang Dengan Menggunakan Metode Perpaduan Grid System Dan Css Media Query. Techno Xplore J. Ilmu Komput. dan Teknol. Inf., vol. 2, no. 1, pp. 1–10.

Kurniawati, F. W., C. Rudianto, A. D. Manuputty, dan A. F. Wijaya, 2020. *Design of Internship Information Systems Using Prototyping Method in PT. Gramedia Asri Media. J. Inf. Syst. Informatics*, vol. 2, no. 1, pp. 89–104.

Nurmiati, S., Arkanda, dan N. U. Aryo. 2017. *Sistem Informasi Penjadwalan Fasilitas Berbasis Web Studi Kasus Pada Institut Sains dan Teknologi Nasional. Ejournal Kaji. Tek. Elektro*, vol. 2, no. 1, pp. 38–46.

Pradipta, A. A., Y. A. Prasetyo, dan N. Ambarsari. 2015. *Pengembangan Web E-Commerce Bojana Sari Menggunakan Metode Prototype. e-Proceeding Eng.*, vol. 2, no. 1, pp. 1042–1056.

Purnomo, D., 2017. *Model Prototyping Pada Pengembangan Sistem Informasi*. vol. 2, no. 2, pp. 54–61.

Romadhon, S. S. dan Desmulyati. 2019. “Vol . 3 No . 1 ISSN : 2597-3673 (Online) ISSN : 2579-5201 (Printed) ISSN : 2597-3673 (Online) ISSN : 2579-5201 (Printed),” vol. 3, no. 1, pp. 21–28.

Siregar, A. M., Faisal, S., Cahyana, Y., dan Priyatna, B. 2020. *Perbandingan Algoritme Klasifikasi Untuk Prediksi Cuaca. Jurnal Accounting Information System (AIMS)*, 3(1), 15-24.

Solehudin, A., Priyatna, B., & Heryana, N. *Analysis Effect of Zfone Security on Video Call Service in Wireless Local Area Network*.

Sofyan. A., P. Puspitorini, dan M. A. Yulianto, 2016. *Aplikasi Media Informasi Sekolah Berbasis SMS Gateway Dengan Metode SDLC (System Development Life Cycle)*, *J. Sisfotek Glob.*, vol. 6, no. 2, pp. 1–7.

Suendri. 2018. *Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem Informasi Remunerasi Dosen Dengan Database Oracle (Studi Kasus: UIN Sumatera Utara Medan)*. *J. Ilmu Komput. dan Inform.*, vol. 3, no. 1, pp. 1–9.

Utomo, D. W., D. Kurniawan, dan Y. P. Astuti. 2018. *Teknik Pengujian Perangkat Lunak Dalam Evaluasi Sistem Layanan Mandiri Pemantauan Haji Pada Kementerian Agama Provinsi Jawa Tengah. Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 9, no. 2, pp. 731–746.

Wira, D., T. Putra, dan R. Andriani. 2019. *Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD*. vol. 7, no. 1.