

# Implementasi Aplikasi Laporan Kinerja *Watchmaker* Dan *Polisher* di *Service Center* Jam Tangan X Dengan Metode RAD

Ikhwana Fakhri Nur Aziz<sup>1</sup>, Febrianto<sup>2</sup>, Muhammad Darwis<sup>3</sup>, Retno Hendrowati<sup>4</sup>,

<sup>1,2,3,4</sup>Program Studi Teknik Informatika, Universitas Paramadina, Indonesia

Email: <sup>1</sup>ikhwana.aziz@students.paramadina.ac.id, <sup>2</sup>febrianto@students.paramadina.ac.id,

<sup>3</sup>muhammad.darwis@paramadina.ac.id, <sup>4</sup>retno.hendrowati@paramadina.ac.id

**Abstrak** - Salah satu proses operasional rutin yang dilakukan oleh *service center* jam tangan X, perusahaan perbaikan jam tangan mewah, adalah mencatat laporan kinerja para teknisi (*watchmaker* dan *polisher*) sebagai dasar penilaian dan perhitungan komisi. Namun, saat ini proses tersebut masih dilakukan secara manual menggunakan *Microsoft excel* sehingga memakan waktu lama dan rentan kesalahan karena melibatkan banyak lembar kerja dan tabel untuk setiap nama teknisinya. Oleh karena itu penulis melakukan pengembangan aplikasi laporan kinerja *watchmaker* dan *polisher* (SiLaKi-WaPo) berbasis *web* di *service center* jam tangan X yang bertujuan untuk memudahkan pengelolaan laporan kinerja. Aplikasi ini dirancang dengan menggunakan metode pengembangan *Rapid Application Development* (RAD) dan memanfaatkan tools UML sehingga efisiensi waktu untuk pengembangan dapat tercapai. Beberapa tools pemrograman yang dipakai adalah JavaScript, HTML, CSS, MongoDB dan NodeJS serta GitHUB untuk melakukan *version control*. Aplikasi yang dihasilkan dalam penelitian ini diuji menggunakan metode pengujian *Black-Box* dengan pendekatan *Use case Testing* dan hasilnya diketahui bahwa seluruh fitur berjalan dengan baik. Dengan adanya aplikasi SiLaKi-WaPo berbasis *web*, pengguna dapat mengelola laporan kinerja *watchmaker* dan *polisher* dengan lebih mudah, cepat dan efisien.

**Kata Kunci:** Aplikasi berbasis *web*, Pengelola laporan kinerja, RAD, Metode *Black-Box*

**Abstract** - One of the routine operational processes conducted by the X wristwatch service center, a luxury wristwatch repair company, is recording the performance reports of technicians (*watchmakers* and *polishers*) as the basis for assessment and commission calculations. However, currently, this process is still carried out manually using *Microsoft Excel*, resulting in a time-consuming and error-prone procedure due to the involvement of multiple worksheets and tables for each technician. Therefore, the author developed a web-based performance report application for *watchmakers* and *polishers* (SiLaKi-WaPo) at the X wristwatch service center, aiming to streamline performance report management. This application is designed using the *Rapid Application Development* (RAD) methodology and utilizes UML tools to achieve development time efficiency. Several

*programming tools used include JavaScript, HTML, CSS, MongoDB, NodeJS, and GitHub for version control. The resulting application in this research was tested using the Black-Box testing method with the Use Case Testing approach, and the results revealed that all features functioned properly. With the web-based SiLaKi-WaPo application, users can manage watchmaker and polisher performance reports more easily, quickly, and efficiently.*

**Keywords:** Web-based application, Performance report management, RAD, Black-Box Method

## I. PENDAHULUAN

Memasuki era digital yang semakin berkembang pesat, penggunaan aplikasi berbasis *web* telah menjadi salah satu tren baru dalam meningkatkan efisiensi operasional dan pengolahan data. Aplikasi berbasis *web* memiliki banyak keunggulan dibandingkan dengan aplikasi desktop, yaitu dapat dioperasikan di berbagai platform yang memungkinkan penggunaannya pada berbagai sistem operasi dan perangkat yang berbeda. Keuntungan yang lain adalah semua pengguna memiliki akses ke versi aplikasi yang sama, serta menghindari masalah kompatibilitas perangkat lunak yang sering kali terjadi pada aplikasi desktop [1]. Dalam lingkup industri dan bisnis, aplikasi berbasis *web* memungkinkan perusahaan untuk merancang solusi yang sesuai agar dapat diakses dari berbagai perangkat dan sistem operasi, menciptakan efisiensi operasional dan meningkatkan produktivitas [2].

Dengan perkembangan teknologi yang terus berlanjut, perancangan dan pengembangan aplikasi berbasis *web* menjadi semakin penting untuk menjawab tantangan dan kebutuhan di dunia yang terus berubah ini [3]. Hal ini tidak terkecuali dalam industri layanan perbaikan jam tangan, khususnya di sebuah *service center* jam tangan X untuk menunjang kegiatan operasional kesehariannya.

Salah satu kegiatan operasional *service center* jam tangan X yaitu mengelola data laporan kinerja teknisi (*watchmaker* dan *polisher*) setiap bulannya. Kegiatan tersebut saat ini masih dilakukan secara manual dengan menggunakan aplikasi *microsoft excel* sebagai aplikasi utama untuk mencatat dan mengelola data laporan kinerja. Data laporan kinerja yang dimaksud adalah rincian pekerjaan yang dilakukan oleh para teknisi yang nantinya akan menjadi laporan bulanan dalam hitungan point yang nantinya akan dikonversi menjadi komisi/bonus oleh

management. Metode yang digunakan saat ini dinilai sudah memanfaatkan teknologi informasi, namun masih memiliki sejumlah kekurangan dimana proses pengolahan data dapat memakan waktu yang cukup lama dan memiliki potensi terjadi kesalahan (*human error*) yang tinggi akibat pengguna harus berhadapan dengan banyak tabel dan lembar kerja [4].

Oleh karena itu, dalam penelitian ini penulis mengembangkan aplikasi sistem laporan kinerja *watchmaker* dan *polisher* (SiLaKi-WaPo) berbasis *web*. Aplikasi tersebut dirancang menggunakan metode *Rapid Application Development* (RAD), dimana metode ini dapat digunakan secara efektif untuk mendesain dan membuat aplikasi yang sangat spesifik sesuai kebutuhan dalam waktu pengembangan yang relatif lebih singkat, karena metode pengembangan aplikasi ini berfokus pada kecepatan penyelesaian pengembangan aplikasi dengan memanfaatkan komponen-komponen yang sudah tersedia dan memiliki fungsi serupa dengan kebutuhan dirancangnya aplikasi[5][6].

Aplikasi berbasis *web* modern memerlukan halaman yang dinamis, artinya pengguna dapat berinteraksi dengan halaman antarmuka aplikasi tanpa perlu melakukan refresh atau reload halaman [7]. Untuk memenuhi hal itu, pemilihan tools dan bahasa pemrograman yang sesuai sangat diperlukan. Pada perancangan aplikasi ini penulis menggunakan environment Node.JS [8] dengan mengombinasikan bahasa pemrograman Javascript, HTML dan CSS sebagai bahasa yang dapat berjalan dengan baik pada sisi klien atau server [9][10], serta menggunakan MongoDB sebagai sistem manajemen basis datanya. MongoDB dipilih karena fleksibilitasnya dalam penyimpanan data sehingga lebih mudah digunakan dibandingkan dengan manajemen basis data lainnya [11][12].

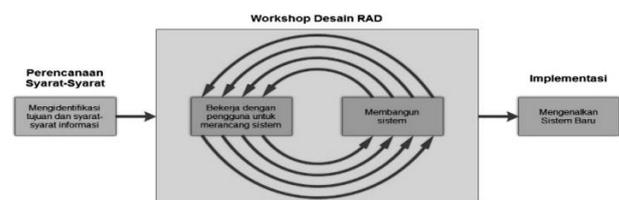
Setelah rancangan aplikasi berhasil dibuat, untuk memastikan kinerjanya dapat diandalkan maka perlu dilakukan pengujian terlebih dahulu sebelum aplikasi benar-benar siap untuk digunakan. Pengujian ini menggunakan metode *Black-Box Testing* dengan pendekatan *Use case Testing* terhadap fungsi fitur pada aplikasi. Setiap fitur pada rancangan aplikasi akan dicoba untuk mengetahui apakah terdapat bug pada fitur tertentu sehingga akan mengganggu keseluruhan sistem aplikasi [13].

Sebagai bahan referensi, beberapa penelitian sebelumnya yang memiliki kesamaan fokus adalah penelitian [14] dengan judul “Rancang Bangun Pengolahan Data Penilaian Kinerja Pegawai pada CV. Mitra Bersama Kabupaten Kayu Agung”. Pada penelitian tersebut aplikasi dikembangkan menggunakan PHP dan MySQL bertujuan untuk menggantikan proses penilaian secara manual. Selanjutnya penelitian [15] dengan judul “Perancangan Aplikasi Pengelolaan Dan Evaluasi Kinerja Pegawai Paud Ar Rosyidin Berbasis *Web*”, perancangan ini dimaksudkan untuk meningkatkan efisiensi dalam pengelolaan dan evaluasi kinerja pegawai paud ar-rosyidin. Kemudian penelitian [16] dengan judul “Aplikasi Manajemen Kinerja Pegawai Dengan Metode Sosiometri pada PT. Siba Prima Utama Feed Mill”. Pada penelitian tersebut aplikasi

dikembangkan untuk melakukan penilaian kinerja pegawai dengan mencatat *point*, peneliti juga menggunakan metode *Black-Box* untuk menguji aplikasi.

Tujuan utama dirancangnya aplikasi sistem laporan kinerja *watchmaker* dan *polisher* (SiLaKi-WaPo) di *service center* jam tangan X adalah untuk meningkatkan efisiensi dan kemudahan pengguna dalam melakukan tugasnya. Pengguna hanya perlu melakukan login pada halaman aplikasi, kemudian memasukan data pada form-input yang sudah disediakan sesuai instruksi, dan tidak perlu lagi berhadapan dengan tabel-tabel dan berpindah lembar kerja. Pada saat membuat laporan, perhitungan total perolehan poin untuk setiap nama teknisi akan muncul sekaligus dalam satu lembar laporan yang dihasilkan oleh aplikasi. Proses penarikan data untuk keperluan pembuatan laporan juga sangat mudah. Pengguna hanya perlu menuju pada menu search, untuk mencari data yang akan dijadikan laporan, kemudian pilih data berdasarkan kode teknisi, lalu tekan tombol cari. Setelah data muncul, data tersebut akan berbentuk tabel, tekan tombol “Cetak PDF” untuk membuat data laporan dalam format file pdf, dan data laporan siap digunakan sesuai kebutuhan [17].

## II. METODE PENELITIAN



Gambar 1. Pengembangan Perangkat Lunak Model RAD [5]

Untuk menyelesaikan penelitian ini penulis menggunakan model pengembangan aplikasi *Rapid Application Development* (RAD). RAD adalah suatu metode pengembangan perangkat lunak di mana proses pengembangannya dilakukan secara berulang (*incremental*) dan berfokus pada siklus pengembangan dalam periode waktu yang relatif singkat. Model RAD merupakan hasil adaptasi dari model waterfall dengan tujuan utama untuk menyelesaikan proses pengembangan dengan kecepatan tinggi [18]. Kecepatan pengembangan yang tinggi ini dicapai melalui pendekatan konstruksi berbasis komponen. Keberhasilan implementasi RAD sangat bergantung pada kemampuan pengembang aplikasi untuk menetapkan kebutuhan dan ruang lingkup pengembangan aplikasi secara rinci. Adapun detail langkah-langkah yang penulis lakukan dalam penyelesaian aplikasi SiLaKi-WaPo seperti pada Gambar 1.

Berdasarkan Gambar 1, tahapan-tahapan pengembangan aplikasi SiLaKi-WaPo dalam penelitian ini adalah:

1. Perencanaan. Tahapan pertama yang penulis lakukan adalah mengidentifikasi masalah dan mengumpulkan data-data pendukung sebagai alasan dan batasan dikembangkannya aplikasi. Pada pengembangan aplikasi SiLaKi-WaPo, permasalahannya adalah metode pengolahan data yang sebelumnya menggunakan *microsoft excel*

memerlukan waktu dan ketelitian, serta memiliki resiko *human error* yang tinggi sehingga diperlukan solusi yang lebih efektif. Untuk batasan perancangan sistemnya adalah mampu mempermudah penggunaan dan meminimalisir *human error* dalam pembuatan laporan.

2. Membangun sistem. Pada tahap ini penulis mulai membuat perancangan dengan memanfaatkan tools UML yang kemudian dilanjutkan membuat *prototype* aplikasi dengan pendekatan berbasis komponen, lalu menguji komponen tersebut apakah sesuai dengan kebutuhan atau tidak. Dalam hal ini, penulis melakukan pencarian *source code* rancangan aplikasi yang memiliki fungsi sejenis dengan kebutuhan yaitu create, read, update dan delete lalu membuat modifikasi tampilan serta fungsinya sesuai dengan kebutuhan perancangan.
3. Pengembangan rancangan. Tahap ini penulis bersama dengan *user* memodifikasi *prototype* yang ada sesuai dengan kebutuhan dan fungsi yang diperlukan. Penambahan fitur baru juga diperlukan agar fungsi yang lebih spesifik dapat dijalankan. Penulis menyesuaikan template dan menambahkan fitur “Cetak PDF” dalam pembuatan laporan. Pengumpulan feedback dari *user* salah satunya dengan melakukan pengujian *Black-Box* untuk menguji apakah fungsi dasar sistem dapat berjalan sesuai dengan yang diharapkan atau tidak.
4. Penerapan sistem. Pada tahapan ini adalah tahap terakhir yang penulis lakukan setelah rancangan sistem disetujui oleh *user*, dan melakukan perbaikan yang diperlukan agar sistem dapat berjalan dengan stabil dan dapat diandalkan untuk digunakan dalam kegiatan operasional.

### III. HASIL DAN PEMBAHASAN

#### A. Perencanaan Kebutuhan

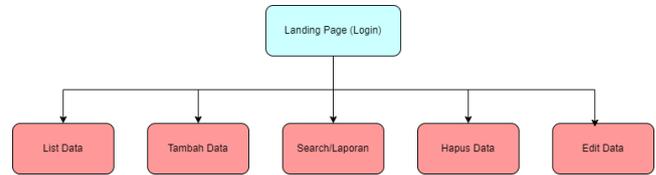
Dalam perancangan kebutuhan untuk merancang aplikasi sistem laporan kinerja *watchmaker* dan *polisher* (SiLaKi-WaPo) ini penulis memberikan rincian spesifikasi komputer yang digunakan dan perangkat lunak apa saja yang diperlukan. Adapun rincihannya sebagai berikut:

1. Spesifikasi Komputer:
  - Processor: Intel(R) Intel® Core™2 Duo E8400
  - Memory: 4 GB
  - VGA: Intel(R) HD Graphic 630
2. Perangkat Lunak yang digunakan:
  - Runtime Support: Node.Js
  - Kode Editor: VS Code atau Sublime Text
  - Browser: Google Chrome
  - CLI: CMD / PowerShell
  - Bahasa Pemrograman: JavaScript
  - Management Basis Data: MongoDB
  - Version Control: Git dan GitHub

Rincian kebutuhan yang penulis berikan diatas pada spesifikasi komputer bukanlah syarat atau spesifikasi minimum yang diperlukan untuk membuat aplikasi yang dirancang dapat berjalan dengan lancar, namun secara kebetulan perangkat yang tersedia memiliki spesifikasi

sedemikian rupa. Sedangkan untuk rincian perangkat lunak yang digunakan merupakan pilihan terbaik menurut versi penulis berdasarkan referensi yang didapat untuk membuat dan merancang sebuah aplikasi *web* menggunakan bahasa pemrograman JavaScript.

#### B. Perancangan Aplikasi

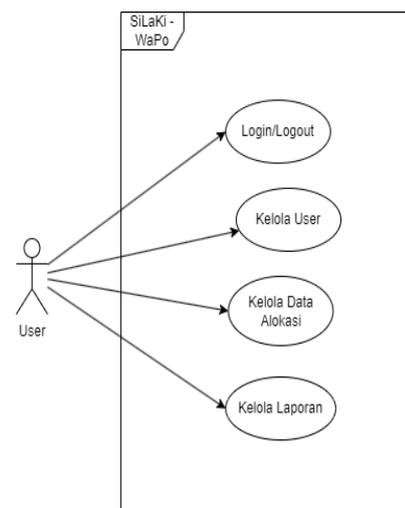


Gambar 2. SiteMap Aplikasi SiLaKi-WaPo

Aplikasi sistem laporan kinerja *watchmaker* dan *polisher* (SiLaKi-WaPo) di *service center* jam tangan X adalah aplikasi yang dirancang menggunakan bahasa pemrograman JavaScript dan dijalankan menggunakan teknologi antarmuka *web*. Seperti pada aplikasi berbasis *web* pada umumnya, aplikasi ini memiliki beberapa halaman situs yang saling terhubung sehingga menjadi sebuah aplikasi yang dapat digunakan.

Dari Gambar 2 diatas adalah peta situs dari aplikasi SiLaKi-WaPo, *user* bisa tau bahwa desain aplikasi cukup sederhana dan mudah untuk dioperasikan. Ketika *user* mengakses aplikasi, halaman pertama yang dijumpai adalah *Landing page*. *Landing page* ini berisi form untuk login kedalam aplikasi. Setelah login berhasil, *user* akan langsung diarahkan menuju halaman pengolahan data. Pada halaman tersebut, *user* memiliki akses langsung ke halaman halaman lain sekaligus tanpa perlu melakukan reload atau berpindah halaman. Maka dari itu, bentuk sitemap aplikasinya posisi halaman lain sejajar dengan halaman lainnya setelah *Landing page*.

#### 1. Use case Diagram.

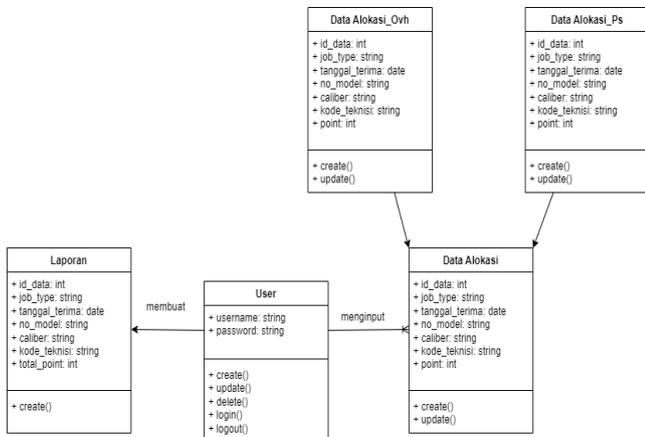


Gambar 3. Use Case Diagram Aplikasi SiLaKi-WaPo

UseCase Diagram aplikasi ini hanya memiliki satu aktor yaitu *user*. *User* adalah orang yang melakukan pengelolaan data laporan performa untuk *watchmaker* dan *polisher* di *service center* jam tangan X. *User* memiliki hak penuh untuk mengakses semua fungsi pada halaman aplikasi dengan

username dan password diberikan. User dapat mengelola user lain dengan cara menambah atau menghapusnya secara langsung melalui sistem basis data aplikasi.

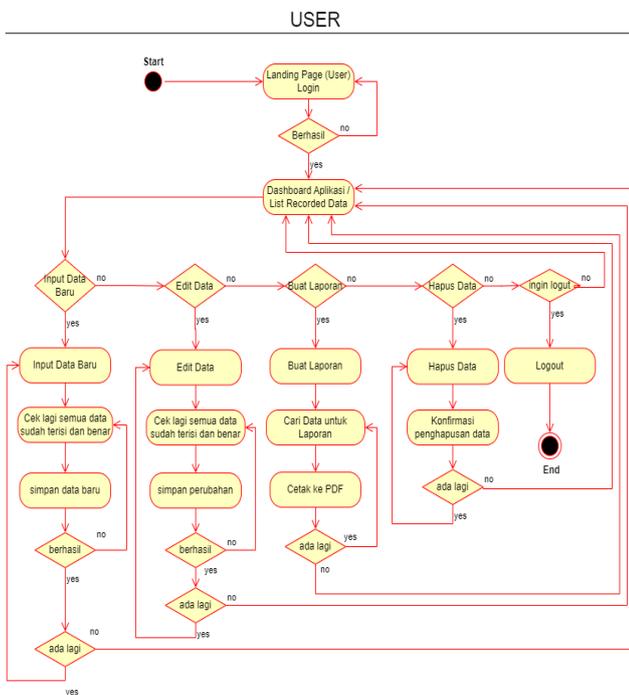
2. Class Diagram.



Gambar 4. Class Diagram Aplikasi SiLaKi-WaPo

Class Diagram diatas memberikan gambaran terhadap hubungan antar objek pada sistem aplikasi. Class User berisikan atribut username dan password untuk dapat melakukan operasi login-logout, dan melakukan manajemen data. Class Data\_Alokasi merupakan data yang nantinya akan diproses menggunakan aplikasi, data ini memiliki turunan berdasarkan jenis pekerjaan yang dilakukan oleh teknisi yaitu overhaul service dan quick service. Sedangkan Class Laporan adalah hasil yang dikeluarkan aplikasi setelah user melakukan generate report untuk masing-masing kode teknisi.

3. Activity Diagram.



Gambar 5. Activity Diagram Aplikasi SiLaKi-WaPo

Activity Diagram menggambarkan aktivitas apa saja yang dapat user lakukan dengan aplikasi SiLaKi-WaPo. Mulai dari verifikasi login pada Landing page, jika gagal user akan terus diminta untuk mencoba login hingga berhasil. Lalu ketika proses login berhasil, secara default aplikasi akan menampilkan halaman list recorded data yang berisikan hampanan data yang sudah berhasil ditambahkan ke aplikasi, jika ternyata belum ada data yang dimasukan maka halaman ini berisikan tabel yang kosong.

Selanjutnya, apakah user ingin melakukan sesuatu? dalam hal ini adalah melakukan input data baru, edit data, membuat laporan, menghapus data atau keluar dari aplikasi. Jika user ingin melakukan input data dan memastikan semua field sudah terisi dengan data yang benar, lalu klik simpan. Begitu pula dengan edit data, user harus memastikan semua data sudah benar sebelum melakukan penyimpanan atas perubahan yang dilakukan.

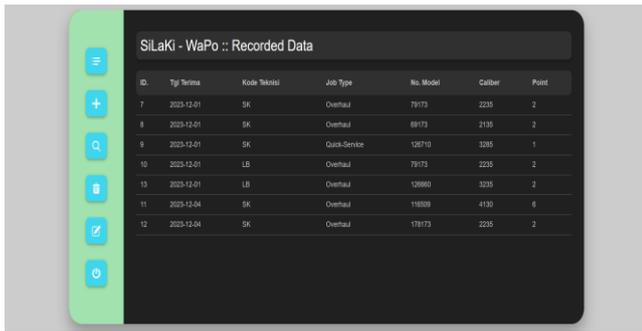
Pada proses membuat laporan user dapat mencari data yang akan diambil untuk dijadikan bahan laporan. Aplikasi menyediakan pilihan untuk memilih data berdasarkan kode teknisi (watchmaker atau polisher). Setelah menekan tombol cari, data hasil pencarian akan muncul, ketika data sudah muncul user dapat menekan tombol cetak pdf untuk membuat laporan dalam format file pdf. Jika dirasa semua sudah selesai, user bisa keluar aplikasi dengan menekan tombol logout.

C. Implementasi Aplikasi



Gambar 6. Landing page Aplikasi SiLaKi-WaPo

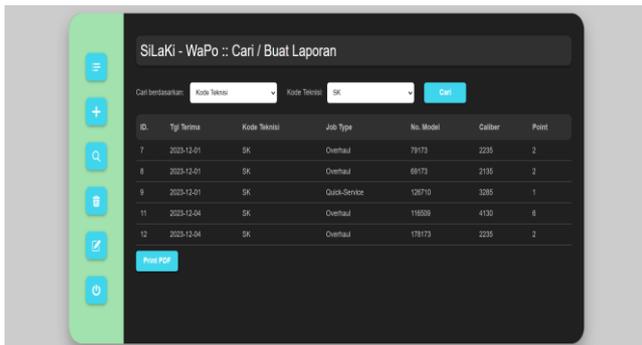
Saat aplikasi pertama kali diakses, user akan dihadapkan dengan form-login sebagai portal awal memasuki aplikasi. User harus melakukan input username dan Password sesuai dengan yang sudah diberikan. Gambar 6 adalah tampilan user interface login page dari aplikasi SiLaKi-WaPo. Desain UI dibuat seminimalis dan semudah mungkin untuk dioperasikan siapa saja yang menjadi user, sehingga tidak perlu dilakukan training atau pelatihan khusus dalam menggunakan aplikasi ini. Tampilan dari setiap halaman yang memiliki fungsi masing-masing akan diperlihatkan dimulai dari gambar berikut:



Gambar 7. Halaman recorded Data Aplikasi SiLaKi-WaPo

Setelah proses login berhasil, *user* secara default akan diarahkan ke halaman *recorded* data seperti yang dapat dilihat pada Gambar 7. Halaman ini berisikan data-data yang sudah diinput sebelumnya pada aplikasi. Detail data yang bisa dilihat adalah sesuai dengan data alokasi yang nantinya *user* juga akan memasukkan atau tambah data baru kedalam aplikasi. Isi data tersebut meliputi tanggal terima job, kode teknisi, jenis job, no. model, caliber, dan point yang diperoleh oleh teknisi. Untuk menambah data baru kedalam aplikasi. *User* harus mengisi semua field yang ada tanpa terkecuali. Jika ada salah satu *field* yang kosong atau tidak terisi dengan benar maka data tidak akan diproses dan tersimpan oleh aplikasi.

**D. Pengembangan Rancangan Aplikasi**

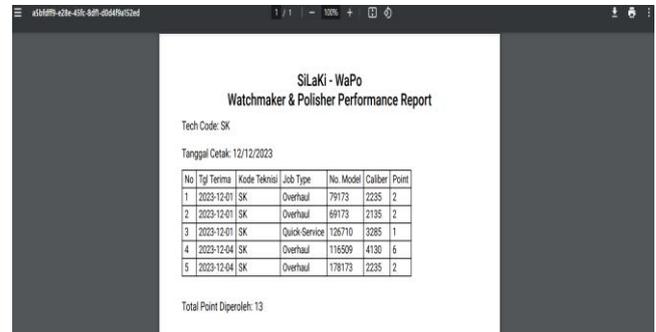


Gambar 8. Halaman Cari Data / Buat Laporan Aplikasi SiLaKi-WaPo

Pengembangan yang dilakukan adalah penyesuaian tampilan atau *user interface* aplikasi dari *source code* yang ada sehingga lebih enak dilihat dan mudah digunakan oleh *user*. Penyesuaian ini meliputi penyesuaian nama dari setiap tombol, atribut pada tabel serta penambahan fungsi untuk membuat laporan. Langkah yang harus dilakukan *user* untuk membuat laporan dengan aplikasi SiLaKi-WaPo ini pertama-tama harus masuk dulu ke halaman pencarian melalui tombol navigasi yang berada di samping sebelah kiri, seperti yang terlihat pada Gambar 8.

Pada halaman pencarian data untuk membuat laporan, *user* terlebih dahulu harus memasukan acuan pencarian data. Sistem aplikasi hanya menyediakan pilihan untuk memilih acuan pencarian berdasarkan kode teknisi. Setelah memilih acuan, pada *field* kode teknisi dapat diisi dengan pilihan kode teknisi yang tersedia. Lalu tekan tombol cari untuk memunculkan data.

Setelah data pencarian muncul, *user* dapat mencetak data tersebut kedalam file dengan format pdf. Adapun tampilan data dalam file pdf dapat dilihat pada Gambar 9. Dalam sekali klik tombol “cetak pdf” data laporan otomatis diproses dan dihitung jumlah akumulasi poin yang diperoleh, sehingga cara ini akan lebih mempermudah *user* dalam membuat laporan.



Gambar 9. Hasil Laporan Cetak PDF Aplikasi SiLaKi-WaPo

**E. Pengujian Aplikasi**

Aplikasi sistem laporan kinerja *watchmaker* dan *polisher* (SiLaKi-WaPo) ini penulis melakukan pengujian terhadap fungsi dasar yang harus berjalan di sistem aplikasi menggunakan metode pengujian *Black-Box*. Berikut ini adalah hasil dari pengujian aplikasi menggunakan metode *Black-Box testing*:

Tabel 1. Hasil Pengujian *Black-Box* Aplikasi SiLaKi-WaPo

No	Use case Activity Testing dan App Response	Result
1	Login dengan <i>Username</i> dan <i>Password</i> salah, Aplikasi tidak bisa login.	Pass
2	Login dengan <i>Username</i> benar dan <i>Password</i> salah, Aplikasi tidak bisa login.	Pass
3	Login dengan <i>Username</i> salah dan <i>Password</i> benar, Aplikasi tidak bisa login.	Pass
4	Login dengan <i>Username</i> dan <i>Password</i> benar, Aplikasi bisa login.	Pass
5	Melakukan Tambah Data dengan <i>field</i> tidak terisi lengkap, aplikasi tidak bisa menyimpan.	Pass
6	Melakukan Tambah Data dengan <i>field</i> terisi lengkap, aplikasi bisa menyimpan.	Pass
7	Melakukan Edit Data dengan <i>field</i> tidak terisi lengkap, tanpa memasukan <i>Password</i> ,	Pass

	aplikasi tidak bisa menyimpan.	
8	Melakukan Edit Data dengan <i>field</i> tidak terisi lengkap, dengan memasukkan <i>Password</i> , aplikasi tidak bisa menyimpan.	<i>Pass</i>
9	Melakukan Edit Data dengan <i>field</i> terisi lengkap, tanpa memasukkan <i>Password</i> , aplikasi tidak bisa menyimpan.	<i>Pass</i>
10	Melakukan Edit Data dengan <i>field</i> terisi lengkap, dengan memasukkan <i>Password</i> , aplikasi bisa menyimpan.	<i>Pass</i>
11	Melakukan Hapus Data dengan memasukkan ID data yang salah dan tanpa memasukkan <i>Password</i> , aplikasi tidak bisa menghapus.	<i>Pass</i>
12	Melakukan Hapus Data dengan memasukkan ID data yang salah dan memasukkan <i>Password</i> , aplikasi tidak bisa menghapus	<i>Pass</i>
13	Melakukan Hapus Data dengan memasukkan ID data yang benar dan tanpa memasukkan <i>Password</i> , aplikasi tidak bisa menghapus.	<i>Pass</i>
14	Melakukan Hapus Data dengan memasukkan ID data yang benar dan memasukkan <i>Password</i> , aplikasi bisa menghapus	<i>Pass</i>
15	Membuat laporan dari pencarian data tanpa memilih acuan kode teknis, aplikasi tidak dapat melakukan pencarian data.	<i>Pass</i>
16	Membuat laporan dari pencarian data dengan memilih acuan kode teknis, aplikasi dapat melakukan pencarian data.	<i>Pass</i>
17	Melakukan Cetak PDF dari data yang berhasil dicari dengan aplikasi.	<i>Pass</i>

Berdasarkan Tabel 1, hasil pengujian aplikasi menggunakan metode *Black-Box* bahwa aplikasi laporan

kinerja *watchmaker* dan *polisher* (SiLaKi-WaPo) dapat menjalankan fungsi dasar sesuai yang diharapkan.

#### IV. KESIMPULAN DAN SARAN

##### A. Kesimpulan

Setelah menyelesaikan penelitian ini, kesimpulan dari penulis adalah:

1. Pengembangan aplikasi laporan kinerja *watchmaker* dan *polisher* (SiLaKi-WaPo) untuk membantu *service center* jam tangan X dalam menemukan solusi meningkatkan kemudahan dan efisiensi dalam mengelola data kinerja *watchmaker* dan *polisher*-nya. Aplikasi ini dikembangkan menggunakan metode RAD dan memanfaatkan tools UML dalam perancangannya. Beberapa tools lain yang digunakan adalah NodeJS untuk application environment dan MongoDB sebagai basis data aplikasi.
2. *User Interface* aplikasi didesain dengan prinsip kemudahan dan kesederhanaan sehingga memudahkan semua pengguna dalam berinteraksi menggunakan aplikasi ini. Sistem berbasis *website* menyediakan kompatibilitas yang tinggi dengan berbagai perangkat, sehingga penggunaan aplikasi ini tidak terbatas pada komputer desktop saja, namun bisa diakses melalui *web browser* dari perangkat manapun asalkan memiliki koneksi internet.
3. Aplikasi sistem laporan kinerja *watchmaker* dan *polisher* (SiLaKi-WaPo) dikembangkan menggunakan metode RAD dan sudah diuji dengan metode *Black-Box*. Dari hasil pengujian didapatkan bahwa aplikasi dapat berjalan sesuai dengan yang diharapkan.

##### B. Saran

Aplikasi SiLaKi-WaPo masih jauh dari kata sempurna, mengingat perkembangan teknologi informasi sangatlah cepat sehingga aplikasi masih sangat mungkin untuk dikembangkan lagi. Saran dari penulis terhadap siapapun yang nantinya akan mengembangkan aplikasi ini adalah selalu mengedepankan kepraktisan dalam setiap fitur yang akan dibuat. Jangan sampai tujuan awal diciptakannya aplikasi ini untuk mempermudah pekerjaan malah berbalik menjadi penghambat pekerjaan.

#### DAFTAR PUSTAKA

- [1] K. Anita, E. R. Susanto, and A. D. Wahyudi, "APLIKASI LOWONGAN PEKERJAAN BERBASIS WEB PADA SMK CAHAYA KARTIKA," 2020. [Online]. Available: <http://jim.teknokrat.ac.id/index.php/sisteminformasi>
- [2] S. Khan, E. Rydow, S. Etemaditajbakhsh, K. Adamek, and W. Armour, "Web Performance Evaluation of High Volume Streaming Data Visualization," *IEEE Access*, vol. 11, no. January, pp. 15623–15636, 2023, doi: 10.1109/ACCESS.2023.3245043.
- [3] W. Wang, J. Guo, Z. Li, and R. Zhao, "Behavior model construction for client side of modern web applications," *Tsinghua Science and Technology*,

- vol. 26, no. 1, pp. 112–134, 2021, doi: 10.26599/TST.2019.9010043.
- [4] Z. Ding, H. Qiu, R. Yang, C. Jiang, and M. C. Zhou, “Interactive-Control-Model for Human-Computer Interactive System Based on Petri Nets,” *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1800–1813, 2019, doi: 10.1109/TASE.2019.2895507.
- [5] Suyanto and Andri, “IMPLEMENTASI RAPID APPLICATION DEVELOPMENT DALAM,” 2020.
- [6] B. Purnomosidi, *Pengenalan Lingkungan Pengembangan Aplikasi Berbasis Perangkat Bergerak*. Jakarta, 2022.
- [7] Sumarlin Tantik and Andriana Myra, “SISTEM ADMINISTRASI PENDAFTARAN PESERTA DIDIK BARU DENGAN CASHBASIS DENGAN MENGGUNAKANWEB DINAMIS,” *Jurnal Sains dan Manajemen*, vol. 10, no. 2, 2022.
- [8] OpenJSFoundation, “About Node JS.” [Online]. Available: <https://nodejs.org/en/about/>
- [9] A. Permatasari and S. Suhendi, “Rancang Bangun Sistem Informasi Pengelolaan Talent Film berbasis Aplikasi Web,” *Jurnal Informatika Terpadu*, vol. 6, no. 1, pp. 29–37, 2020, doi: 10.54914/jit.v6i1.255.
- [10] S. Mariko, “APLIKASI WEBSITE BERBASIS HTML DAN JAVASCRIPT UNTUK MENYELESAIKAN FUNGSI INTEGRAL PADA MATA KULIAH KALKULUS,” *Jurnal Inovasi Teknologi Pendidikan*, vol. 6, no. 1, pp. 80–91, 2019, doi: 10.21831/jitp.v6.1.22280.
- [11] BAMAUMA, “Jenis-Jenis Database yang Sering Digunakan,” Biro Administrasi Mutu Akademik dan Informasi. [Online]. Available: <https://bamai.uma.ac.id/2021/12/30/jenis-jenis-database-yang-sering-digunakan/>
- [12] Nasution, “Implementasi Mongo Db, Express Js, React Js Dan Node Js (Mern) Pada Pengembangan Aplikasi Formulir, Kuis, Dan Survei Online,” *Informatics Engineering*, pp. 1–160, 2021.
- [13] N. Made, D. Febriyanti, A. A. Kompiang, O. Sudana, and N. Piarsa, “Implementasi Black Box Testing pada Sistem Informasi Manajemen Dosen,” 2021.
- [14] Ubaidillah, “RANCANG BANGUN PENGOLAHAN DATA PENILAIAN KINERJA PEGAWAI PADA CV. MITRA BERSAMA KABUPATEN KAYU AGUNG,” *Jurnal Ilmu Komputer*, vol. 1, no. 1, pp. 1–7, 2020.
- [15] L. Norhan and N. Sukmawardani, “Perancangan Aplikasi Pengelolaan Dan Evaluasi Kinerja Pegawai Paud Ar Rosyidin Berbasis Web,” *Jurnal Manajemen Sistem Informasi*, vol. 01, no. 01, pp. 27–32, 2022.
- [16] D. Pramaishella and M. Maryam, “Aplikasi Manajemen Kinerja Pegawai Dengan Metode Sosiometri pada PT. Siba Prima Utama Feed Mill,” *Jurnal Manajemen Informatika (JAMIKA)*, vol. 13, no. 2, pp. 94–107, 2023, doi: 10.34010/jamika.v13i2.9557.
- [17] S. Alazmi and D. C. De Leon, “A Systematic Literature Review on the Characteristics and Effectiveness of Web Application Vulnerability Scanners,” *IEEE Access*, vol. 10, pp. 33200–33219, 2022, doi: 10.1109/ACCESS.2022.3161522.
- [18] D. Hariyanto, R. Sastra, F. E. Putri, S. Informasi, K. Kota Bogor, and T. Komputer, “Implementasi Metode Rapid Application Development Pada Sistem Informasi Perpustakaan,” 2021.