

PERANCANGAN APLIKASI *INDIVIDUAL PERFORMANCE PLANNING* DAN *PERFORMANCE APPRAISAL* MENGGUNAKAN METODE *WATERFALL* PADA PT. CENTURY BATTERIES INDONESIA

Najmil Hayah, Garno, Betha Nurina Sari
Informatika, Universitas Singaperbangsa Karawang
Karawang, Indonesia
najmilhayh6@gmail.com

ABSTRAK

Proses digitalisasi di PT Century Batteries Indonesia telah dimulai sejak tahun 2020, bertujuan untuk memudahkan akses data dari mana saja dan mengurangi penggunaan kertas, sehingga mengurangi risiko kehilangan dokumen akibat kesalahan manusia. Proses approval juga menjadi lebih terstruktur dengan adanya sistem aplikasi. Departemen HRD adalah salah satu yang memanfaatkan digitalisasi ini untuk proses pengisian laporan *Individual Performance Planning* (IPP) dan *Performance Appraisal* (PA). Penelitian ini bertujuan untuk merancang dan membangun aplikasi IPP dan PA berbasis *website* yang dapat mengatasi masalah visibilitas, transparansi, dan menyediakan mekanisme yang jelas. Metodologi penelitian menggunakan *System Development Life Cycle* (SDLC) dengan metode *waterfall*. Pada tahap *requirements definition*, dilakukan analisis masalah dan kebutuhan untuk *admin*, karyawan, dan atasan. Pada tahap *system and software design*, dibuat diagram UML, seperti *use case diagram*, *activity diagram*. Implementasi kode menggunakan CodeIgniter 4 dengan *template* Admin LTE. Pengujian dilakukan dengan menggunakan metode *black box testing* dan menghasilkan 10 dari 10 pengujian berjalan sesuai dengan *test case* yang diberikan, memastikan fungsionalitas aplikasi sesuai dengan kebutuhan pengguna. Uji coba *User Acceptance Testing* (UAT) menunjukkan aplikasi dapat membantu karyawan, atasan, dan admin dalam pelaporan dan monitoring IPP dan PA. Aplikasi ini kemudian diserahkan kepada perusahaan untuk proses *deploy* selanjutnya. Aplikasi ini diharapkan dapat membantu karyawan, atasan, dan kepala seksi people development HRD selaku *admin* dalam proses pelaporan dan *monitoring* laporan IPP dan PA.

Kata kunci : *penilaian kinerja karyawan, perencanaan kinerja, aplikasi, waterfall, black box testing*

1. PENDAHULUAN

Dalam lingkungan kerja yang dinamis, perencanaan kinerja individu atau *Individual Performance Planning* (IPP) sangat penting untuk mencapai tujuan organisasi dan pengembangan karyawan. Rencana kinerja ini mencakup aktivitas pekerjaan rutin dan proyek perbaikan, serta menggunakan prinsip SMART (*Specific, Measurable, Relevant, Time-Phased*) untuk memastikan tujuan yang jelas dan terstruktur. Penilaian kinerja atau *Performance Appraisal* (PA) dilakukan secara kolaboratif antara karyawan dan atasan, dengan kriteria yang sudah ditetapkan untuk menjaga struktur dan kejelasan dalam proses penilaian.

Metode *manual* yang digunakan saat ini, seperti menggunakan excel, menciptakan tantangan terkait visibilitas, transparansi, dan pelacakan perubahan. Hal ini mengakibatkan kesulitan dalam mengetahui siapa yang sudah mengisi IPP dan PA, serta memantau perubahan yang dilakukan oleh atasan. Oleh karena itu, perusahaan berencana untuk beralih ke portal berbasis *website* untuk Menyusun dan mengelola IPP dan PA yang akan memudahkan proses pengisian, persetujuan, dan pemantauan perubahan secara *real-time*.

Portal berbasis *website* ini akan mengintegrasikan proses persetujuan secara *digital*, mengeliminasi kebutuhan cetak dokumen, dan

mempercepat alur kerja. Setiap atasan akan memiliki akses terbatas sesuai hierarki organisasi, memastikan keamanan dan akurasi data. Portal ini juga akan menyediakan riwayat perubahan yang terperinci, memungkinkan pelacakan setiap perubahan yang dilakukan, dan memberikan opsi untuk mengunduh data laporan IPP dan PA oleh karyawan, *admin*, dan atasan.

Pengembangan dan pembangunan aplikasi ini menggunakan metode *waterfall*, yang membuat kualitas sistem informasi cenderung tinggi karena proses implementasinya yang dilakukan berdasarkan beberapa tahapan yang terstruktur [1]. Dengan menerapkan metode ini pada pembangunan dan pengembangan aplikasi, diharapkan proses penyusunan dan penilaian kinerja akan lebih terstruktur, mengurangi kesalahan manusia, dan meningkatkan efisiensi serta transparansi dalam manajemen kinerja individu di perusahaan.

2. TINJAUAN PUSTAKA

2.1. Kinerja Karyawan

Evaluasi kinerja merupakan proses yang dilakukan oleh perusahaan untuk mengukur efektivitas pekerja berdasarkan kriteria yang telah ditentukan, seperti sikap dan disiplin, yang memungkinkan perusahaan untuk memahami dan mengoptimalkan kinerja individu [2]. Dalam rangka

mencapai manajemen yang efektif, pentingnya peran karyawan yang kompeten dan pembinaan serta pelatihan menjadi krusial untuk meningkatkan keterampilan dan mempertahankan kesetiaan mereka terhadap perusahaan [3].

2.2. Website

Sebuah website terdiri dari berbagai halaman yang menyajikan informasi dalam bentuk teks, gambar, animasi, dan suara, yang bisa statis atau dinamis. Halaman-halaman ini saling terkait dan terstruktur, terhubung melalui tautan, dan diakses melalui jaringan [4].

2.3. Code Igniter

Pengembangan sistem informasi yang berbasis website telah mengalami kemajuan pesat, dengan pengenalan berbagai bahasa pemrograman yang mendukung penggunaan *framework*. *Framework* ini membantu dalam mempercepat proses pembuatan aplikasi, baik itu untuk *website* atau *desktop*, dengan mengorganisir kode secara lebih rapi dan meningkatkan efisiensi penggunaan waktu dan sumber daya oleh pengembang. Codeigniter, sebuah *framework* open-source yang diperkenalkan oleh Rick Ellis pada tahun 2006, adalah contoh populer yang menggunakan arsitektur MVC (*model*, *view*, *controller*) untuk memudahkan pembangunan *website* dinamis [5].

2.4. SQL Server

Sebuah Sistem Manajemen Basis Data Relasional (RDBMS) memanfaatkan SQL untuk mengelola dan memproses data dengan efisien. Dalam hal ini, *Data Definition Language* (DDL) digunakan untuk mendefinisikan dan memodifikasi struktur objek database melalui perintah seperti *create*, *rename*, *drop*, *alter*, dan *truncate*. *Data Control Language* (DCL) berperan dalam mengatur akses pengguna dengan perintah *grant* dan *revoke*. *Data Manipulation Language* (DML) mengurus manipulasi data secara langsung, dengan perintah seperti *merge*, *delete*, *insert*, dan *update*. SQL Server, dikembangkan oleh Microsoft, adalah salah satu contoh sistem manajemen database relasional yang mendukung pengambilan keputusan melalui fitur-fitur seperti *business intelligence* dan analisis data. Microsoft juga mengintegrasikan ekstensi khusus, Transact-SQL (T-SQL), untuk memperluas kemampuan pengolahan data SQL Server [6].

2.5. System Development Life Cycle (SDLC)

SDLC (System Development Life Cycle) adalah rangkaian langkah sistematis yang digunakan untuk mengembangkan atau memodifikasi sistem perangkat lunak dengan menggunakan metode dan model yang dianggap *best practice* berdasarkan pengalaman pengembangan sistem sebelumnya. Model-model dasar dalam SDLC meliputi *waterfall*, *prototype*, RAD (*Rapid Application Development*), *iterative*, dan

lainnya, yang masing-masing memberikan kerangka kerja untuk melaksanakan proses pengembangan sistem [7].

2.6. Waterfall

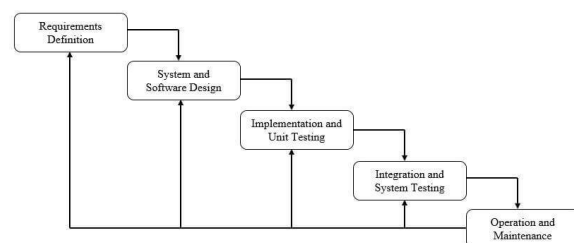
Metode *waterfall* adalah pendekatan yang sering digunakan dalam pengembangan perangkat lunak dan sistem, yang dikarakterisasi oleh prosesnya yang linier. Menurut Sommerville, metode ini berlangsung dari tahap awal perencanaan dan berlanjut secara bertahap hingga tahap pemeliharaan akhir. Dalam metode ini, setiap tahap berikutnya hanya dimulai setelah tahap sebelumnya telah sepenuhnya selesai [8].

2.7. Unified Modeling Language (UML)

Berbagai model pengembangan perangkat lunak tersedia, dengan pendekatan yang mencakup metode *procedural* hingga berorientasi objek. *Unified Modeling Language* (UML) adalah salah satu alat perancangan yang populer, difokuskan pada pendekatan berorientasi objek. UML digunakan untuk mendeskripsikan dan merancang sistem perangkat lunak, terutama dalam pengembangan yang menggunakan pemrograman berorientasi objek [9].

3. METODE PENELITIAN

Metode yang digunakan dalam pengembangan perangkat lunak dalam penelitian ini adalah metode *waterfall* yang merupakan bagian dari siklus hidup pengembangan perangkat lunak sistem atau System Development Life Cycle (SDLC) dan terdiri dari beberapa tahapan, yaitu analisis, desain, pembuatan kode, implementasi, pengujian dan pemeliharaan. Metode *waterfall* dipilih karena kebutuhan dan tujuannya sudah ditetapkan sejak awal, serta setiap fase dalam model ini memiliki spesifikasi dan hasil yang jelas. Hal ini memudahkan penilaian kemajuan dan manajemen proyek [9]. Tahapan yang terdapat di dalam metode *waterfall* ini terdiri dari *requirements definition*, *system and software design*, *implementation and unit testing*, *integration and system testing*, *operation and system maintenance* [8].



Gambar 1. Metode *waterfall*

Pada gambar 1, dapat dilihat bahwa terdapat beberapa tahapan dalam metode *waterfall*. Berikut merupakan penjelasan lebih lanjut terkait metode *waterfall* yang digunakan dalam perancangan aplikasi IPP dan PA.

- Requirements definition.* Pada tahap ini, analisis terhadap permasalahan dilakukan melalui wawancara dengan pengguna. Setelah itu, dilakukan analisis kebutuhan untuk menentukan apa saja yang diperlukan dalam pengembangan aplikasi berdasarkan hasil analisis permasalahan. Solusi dari permasalahan akan ditemukan setelah analisis permasalahan dan kebutuhan dilakukan.
- System and software design.* Tahap ini melibatkan pembuatan desain perangkat lunak dengan *wireframe* dan desain sistem menggunakan diagram *use case activity*, *class diagram*, dan *sequence diagram*, dengan perangkat lunak *draw.io*.
- Implementation.* Pada tahap ini, kode ditulis menggunakan framework CodeIgniter 4, dan pengembangan database dilakukan dengan menggunakan SQL Server serta MSSQL sebagai DBMS.
- Integration and system testing.* Pada tahap ini, pengujian dilakukan menggunakan metode *black box testing* untuk memastikan bahwa sistem secara keseluruhan memenuhi persyaratan fungsional dan dapat berinteraksi dengan baik antar komponennya.
- Operation and maintenance.* Pada tahap ini, aplikasi yang telah selesai dikembangkan diperkenalkan dan diserahkan kepada pengguna. Pemeliharaan dilakukan untuk mengatasi masalah yang mungkin muncul serta menangani atau menambahkan fitur baru jika diperlukan.

4. HASIL DAN PEMBAHASAN

Pembahasan dari hasil perancangan aplikasi *individual performance planning* (IPP) dan *performance appraisal* (PA) adalah sebagai berikut.

4.1. Requirements Definition

Pada tahap ini, terdapat tiga aktivitas utama: analisis permasalahan, analisis kebutuhan, dan penyusunan solusi. Dalam analisis permasalahan, beberapa isu ditemukan berdasarkan wawancara, diskusi, dan survei dengan user kepala seksie people development HRD. Untuk admin, masalah utama adalah kurangnya kontrol akses, kesulitan monitoring IPP dan PA, kesalahan dalam penyimpanan dan approval data, serta tidak adanya sistem pengelolaan jadwal. Untuk karyawan, masalahnya mencakup kurangnya kontrol akses, manajemen pelaporan IPP dan PA yang kurang efisien, sulitnya melacak perubahan laporan, dan kebutuhan untuk mengunduh laporan. Sedangkan untuk atasan, masalahnya meliputi kurangnya kontrol akses, sulitnya monitoring IPP dan PA, serta pemberian *feedback* yang tidak sesuai alur.

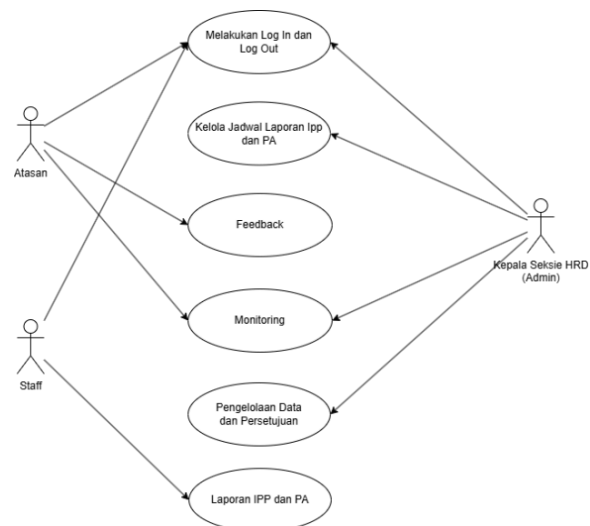
Berdasarkan analisis permasalahan tersebut, dilakukan analisis kebutuhan. Admin memerlukan aplikasi yang dapat membatasi akses pengguna, memonitor data, memberikan dukungan teknis, dan mengelola jadwal pelaporan. Karyawan

membutuhkan aplikasi yang dapat membatasi akses pengguna, mengisi dan mengelola IPP dan PA, mengunggah *file*, dan melihat riwayat penyuntingan. Atasan memerlukan aplikasi yang dapat membatasi akses pengguna, memonitor data, me-review dan menyetujui laporan, serta membatasi proses *review* dan *approval*.

Solusi permasalahan yang diambil adalah mengubah metode pengisian hingga *approval* dari yang sebelumnya menggunakan excel dan kertas menjadi digital melalui aplikasi berbasis *website*. Aplikasi ini bertujuan untuk meningkatkan kontrol akses, *monitoring*, dukungan teknis, pengelolaan jadwal, dan efisiensi manajemen pelaporan. Admin akan memiliki hak akses terbatas untuk memonitor laporan, memberikan dukungan teknis, dan mengelola periode pelaporan. Karyawan akan memiliki hak akses terbatas untuk melaporkan, mengedit, menghapus, mengunggah *file*, melihat riwayat, dan mencetak laporan. Atasan akan memiliki hak akses terbatas untuk *monitoring*, memberikan *feedback*, me-review, dan menyetujui laporan, serta membatasi alur *review* dan *approval*.

4.2. System and Software Design

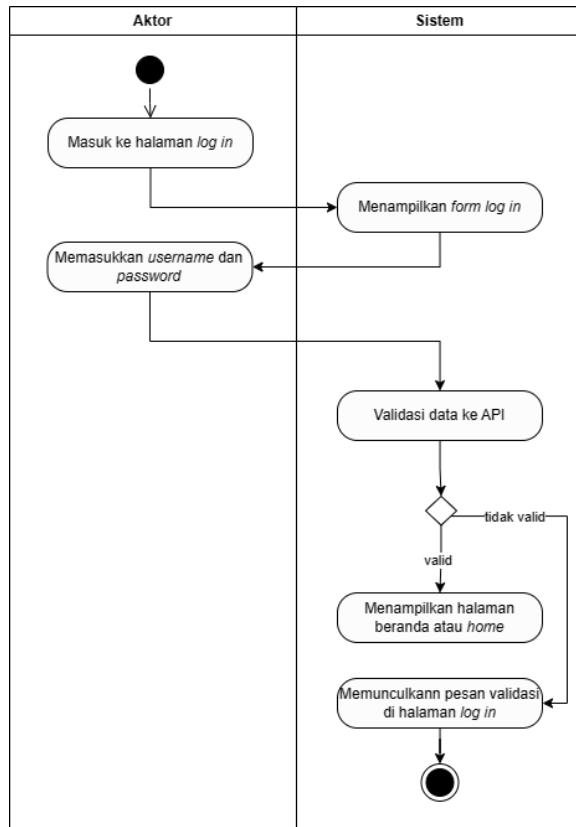
Pada tahap ini, dirancang kebutuhan UML yang dibutuhkan, seperti *use case diagram* dan *activity diagram*. Berikut merupakan *use case diagram* dan *activity diagram* yang dibuat berdasarkan skenario yang terjadi pada *use case diagram* dari perancangan aplikasi IPP dan PA.



Gambar 2. Use case diagram aplikasi IPP dan PA

Pada gambar 2, dapat dilihat bahwa terdapat sebuah *use case diagram* yang merupakan model yang digunakan untuk menggambarkan perilaku (*behavior*) dari sistem informasi yang akan dikembangkan [4]. Terdapat tiga aktor dalam *use case diagram* ini yang menggambarkan pengguna aplikasi, yaitu kepala seksie divisi HRD yaitu seksie *people development* sebagai *admin*, *staff* atau karyawan, dan atasan. *Admin* memiliki 4 *use case*,

yaitu melakukan *log in* dan *log out*, kelola jadwal laporan IPP dan PA, *monitoring*, serta pengelolaan data dan persetujuan. Atasan memiliki tiga *use case*, yaitu melakukan *log in* dan *log out*, *feedback*, dan *monitoring*. Staff memiliki dua *use case*, yaitu melakukan *log in* dan *log out* serta laporan IPP dan PA.



Gambar 3. Activity diagram melakukan log in

Pada gambar 3, dapat dilihat bahwa terdapat *activity diagram* yang dibuat berdasarkan skenario dari *use case* melakukan *log in* dan *log out*.

4.3. Implementation

```

57 public function loginToExternalApi() {
58   $api_url = 'https://portal3.incoe.asstra.co.id/production_control_v2/public/api/login';
59   $this->validation = \Config\Services::validation();
60   $this->loginModel = new LoginModel();
61
62   $data = $this->request->getPost();
63   $validate = $this->validation->run($data, 'masuk');
64   $errors = $this->validation->getErrors();
65
66   if ($errors) {
67     return $this->response->setJSON($errors)->setStatusCode(400);
68   } else {
69     $api_data = [
70       'username' => $data['username'],
71       'password' => $data['password']
72     ];
73
74     $options = [
75       'http' => [
76         'method' => 'POST',
77         'header' => 'Content-type: application/x-www-form-urlencoded',
78         'content' => http_build_query($api_data)
79       ]
80     ];
81
82     $context = stream_context_create($options);
83     $response = file_get_contents($api_url, false, $context);
84
85     $authData = json_decode($response, true);
  
```

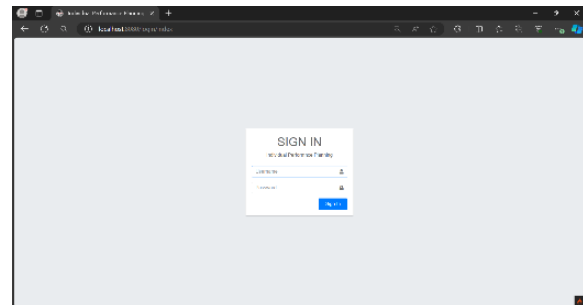
Gambar 4. Function untuk log in

Pada tahap implementasi, desain yang telah direncanakan sebelumnya dalam bentuk diagram diterjemahkan ke dalam kode atau sintaks

pemrograman. Proses implementasi kode dilakukan menggunakan Visual Studio Code sebagai editor kode, dengan bantuan framework CodeIgniter 4, serta memanfaatkan template AdminLTE yang didasarkan pada Bootstrap sebagai kerangka kerja CSS untuk membantu pembuatan antarmuka.

a. Halaman log in

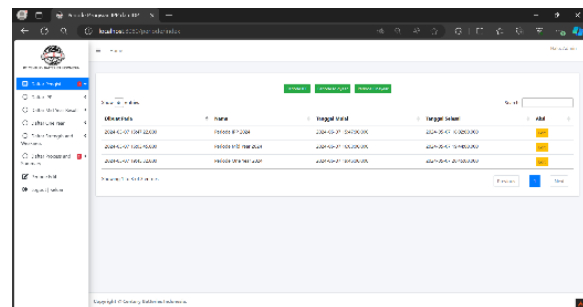
Halaman *log in* berfungsi untuk membantu aplikasi dalam melakukan *filtering* menu atau fitur untuk setiap pengguna yang akan mengakses aplikasi. Berisi dua buah *input*, yaitu *username* dan *password* yang akan dicek melalui API. Jika pengguna benar dalam memasukkan *username* dan *password*, aplikasi akan langsung mengarahkan pengguna ke halaman *home*. Tetapi jika ada yang salah, pengguna akan tetap berada di halaman *log in* dan aplikasi akan menampilkan pesan kesalahan.



Gambar 5. Halaman log in

b. Halaman edit periode

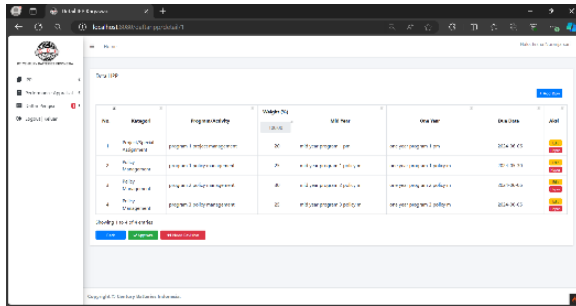
Halaman ini menampilkan jadwal dari periode IPP atau PA yang sedang berlangsung. Hanya dapat diakses oleh *admin* dan *admin* dapat melakukan penambahan periode baru, serta penyuntingan periode yang sudah ada.



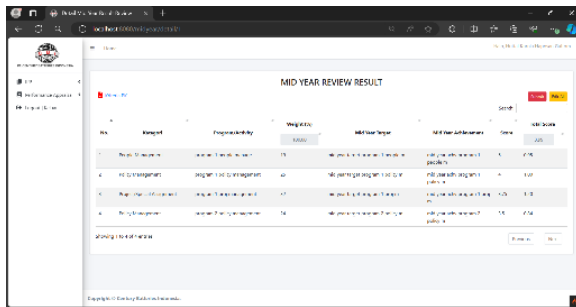
Gambar 6. Halaman edit periode

c. Halaman detail IPP

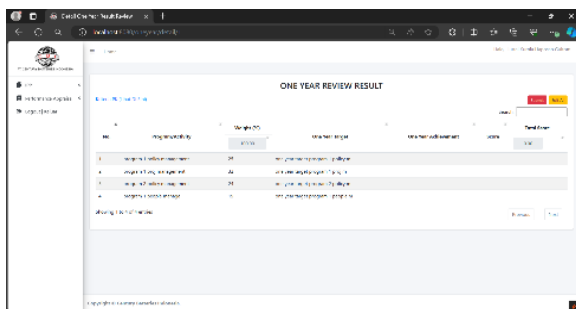
Karyawan maupun atasan dapat melakukan penyuntingan pada laporan IPP dan PA yang dibuat. Tombol atau fungsi *approve* dan *need revision* hanya akan tampil pada atasan. PA terdiri dari beberapa bagian lagi di dalamnya, seperti *mid year result review*, *one year result review*, *process and summary*, *strength and weakness*. Setiap bagian, memiliki kolom *input* masing-masing.



Gambar 7. Halaman detail IPP



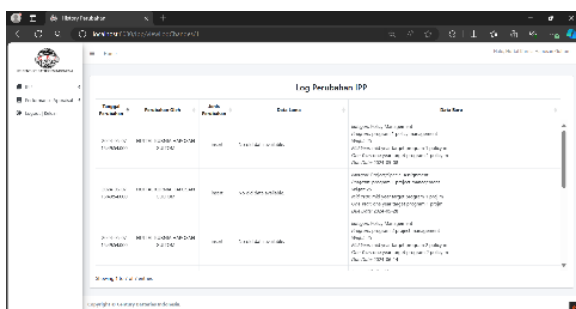
Gambar 8. Halaman detail mid year result review



Gambar 9. Halaman detail one year result review

d. Halaman log

Halaman *log* berisi keterangan berupa pengguna yang melakukan perubahan, jenis perubahan yang terjadi (*insert* dan *update*), serta perubahan data yang terjadi.



Gambar 10. Halaman log

4.5. Black Box Testing

Tahap pengujian ini menggunakan metode *black box testing* untuk menguji fungsionalitas program atau aplikasi. Pengujian menggunakan *black box testing* dipilih karena aplikasi dibutuhkan membutuhkan durasi waktu pengembangan yang singkat dan metode ini merupakan metode yang mengevaluasi fungsionalitas aplikasi berdasarkan spesifikasi eksternal dan kebutuhan pengguna, memastikan semua aspek dasar aplikasi sesuai dengan kebutuhan [11].

Pengujian dilakukan bersama dengan tiga pengguna aplikasi, yaitu kepala seksie departemen *people development* HRD dan salah dua anggota departemen *Management Information System* (MIS) untuk memastikan bahwa setiap fungsi yang ada berjalan sesuai dengan kebutuhan dengan memberikan skenario *input* dan *output* pada setiap fungsi dalam aplikasi. Berikut adalah hasil pengujian sistem menggunakan *black box testing*.

Tabel 1. Hasil Pengujian

No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil
1.	Melakukan <i>log in</i> menggunakan <i>username</i> dan <i>password</i> yang terdaftar	Menuju halaman <i>home</i>	Sesuai harapan
2.	Melakukan <i>input</i> periode	IPP dan PA dapat diisi sesuai dengan periode yang berjalan	Sesuai harapan
3.	<i>Submit</i> IPP	Data laporan IPP muncul di atas	Sesuai harapan
4.	<i>Submit</i> PA	Data laporan PA muncul di atas	Sesuai harapan
5.	<i>Approval</i> IPP	IPP sudah tidak dapat diisi dan dapat dilakukan <i>print out</i>	Sesuai harapan
6.	<i>Approval</i> PA	PA sudah tidak dapat diisi dan dapat dilakukan <i>print out</i>	Sesuai harapan
7.	<i>Upload file</i> IPP dan PA sebelum tahun 2024	<i>File</i> IPP dan PA muncul di halaman <i>index</i>	Sesuai harapan
8.	Menyimpan dan menghapus isi dari IPP dan PA	Muncul <i>history</i> berupa keterangan penyimpanan dan penghapusan yang terjadi	Sesuai harapan
9.	<i>Unsubmit</i> laporan IPP dan PA	IPP dan PA dapat di- <i>edit</i> kembali oleh karyawan	Sesuai harapan
10.	Tombol <i>need revision</i> diklik	IPP dan PA dikembalikan lagi kepada karyawan dan dapat dilakukan proses <i>edit</i>	Sesuai harapan

4.4. Integration and system testing

Pengujian perangkat lunak adalah metode yang digunakan untuk memastikan perangkat lunak berfungsi dengan efektif dan benar. Tanpa pengujian, sulit untuk mengetahui apakah suatu program memenuhi standar dan persyaratan yang dibutuhkan oleh pengguna [10].

Dari tabel 1 dapat dilihat bahwa 10 dari 10 skenario pengujian menggunakan *black box testing* yang diberikan *input* tertentu sesuai dengan skenario pengujian yang dibuat sesuai dengan hasil yang diharapkan.

4.6. User Acceptance Testing (UAT)

User Acceptance Testing (UAT) dilakukan untuk mengevaluasi aplikasi IPP dan PA dengan melibatkan sejumlah 20 responden dari berbagai tingkatan organisasi. Responden terdiri dari 3 Kepala Divisi, 5 Kepala Departemen, 10 Kepala Seksi, dan 2 *staff*. Evaluasi dilakukan menggunakan skala penilaian yang mencakup kriteria Sangat Setuju (SS), Setuju (S), Kurang Setuju (KS), dan Tidak Setuju (TS), yang masing-masing memiliki bobot penting dalam menentukan penerimaan dan kepuasan pengguna terhadap aplikasi sesuai dengan tabel 2. Tujuan utama dari UAT ini adalah untuk menilai sejauh mana aplikasi ini memenuhi harapan pengguna dalam hal kesesuaian fungsional, kemudahan penggunaan, serta keinginan untuk mengadopsi aplikasi ini secara rutin dalam lingkungan kerja.

Tabel 2. Bobot jawaban

Bobot	Jawaban
4	Sangat Setuju (SS)
3	Setuju (S)
2	Kurang Setuju (KS)
1	Tidak Setuju (TS)

Tabel 3. Pertanyaan Evaluasi

Kode Pertanyaan	Pertanyaan Evaluasi
Kesesuaian Fungsional	
Q1	1. Apakah fitur laporan kinerja dapat diakses dan berfungsi dengan baik?
Q2	2. Apakah sistem pengisian perencanaan kinerja individu sesuai dengan harapan?
Q3	3. Apakah proses <i>approval</i> laporan berjalan tanpa kesalahan?
Kemudahan Penggunaan	
Q1	1. Apakah letak dan navigasi aplikasi mudah dipahami?
Q2	2. Apakah instruksi penggunaan aplikasi jelas dan membantu?
Q3	3. Apakah anda merasa nyaman menggunakan fitur-fitur aplikasi ini??
Keinginan untuk Menggunakan	
Q1	1. Apakah aplikasi ini membantu dalam pekerjaan anda?
Q2	2. Apakah anda akan merekomendasikan aplikasi ini kepada rekan kerja?
Q3	3. Apakah anda berencana menggunakan aplikasi ini?

Tabel 3 menunjukkan pertanyaan-pertanyaan yang diajukan kepada responden. Dari pertanyaan evaluasi yang diberikan, didapatkan hasil seperti pada tabel 4.

Tabel 4. Hasil Evaluasi

Pertanyaan	Jawaban			
	SS	S	KS	TS
Kesesuaian Fungsional				
Q1	10	8	2	0
Q2	8	12	0	0
Q3	14	6	0	0
Kemudahan Penggunaan				
Q1	12	8	2	0
Q2	10	10	0	0
Q3	14	6	0	0
Keinginan untuk Menggunakan				
Q1	16	4	0	0
Q2	12	8	2	0
Q3	14	6	0	0

Dari tabel 4, yaitu hasil evaluasi yang melibatkan 20 responden, dihitung persentase nilai untuk setiap pertanyaannya agar memberikan gambaran yang lebih jelas dan terukur mengenai bagaimana aplikasi atau sistem dinilai oleh pengguna, serta memberikan arahan untuk perbaikan dan pengembangan berkelanjutan. Perhitungan persentase tersebut dapat dilihat di bawah ini.

Gambar 11. Penilaian UAT

$$\% \text{ Skor Aktual} = \frac{\text{Skor Aktual}}{\text{Skor Ideal}} \times 100\%$$

Dengan keterangan, skor aktual merupakan hasil penjumlahan nilai kategori dikalikan dengan frekuensi. Dan skor ideal merupakan penjumlahan nilai yang diasumsikan jika responden memiliki skor tertinggi dari semua jawaban kuesioner. Pada kesesuaian fungsional, nilai persentasenya akan dihitung sesuai dengan rumus pada gambar 11, seperti berikut.

$$\begin{aligned} \text{Rata-rata nilai Q1} &= ((10 \times 4) + (8 \times 3) + (2 \times 2) + (0 \times 1)) / 20 = 3.4 \\ \text{Persentase Q1} &= (3.4 / 4) \times 100\% = 85\% \\ \text{Rata-rata nilai Q2} &= ((8 \times 4) + (12 \times 3) + (0 \times 2) + (0 \times 1)) / 20 = 3.4 \\ \text{Persentase Q2} &= (3.4 / 4) \times 100\% = 85\% \\ \text{Rata-rata nilai Q3} &= ((14 \times 4) + (6 \times 3) + (0 \times 2) + (0 \times 1)) / 20 = 3.7 \\ \text{Persentase Q3} &= (3.7 / 4) \times 100\% = 92.5\% \end{aligned}$$

Pada kemudahan penggunaan, nilai persentasenya akan dihitung sesuai dengan rumus pada gambar 11, seperti berikut.

$$\begin{aligned} \text{Rata-rata nilai Q1} &= ((12 \times 4) + (8 \times 3) + (2 \times 2) + (0 \times 1)) / 20 = 3.8 \\ \text{Persentase Q1} &= (3.8 / 4) \times 100\% = 95\% \\ \text{Rata-rata nilai Q2} &= ((10 \times 4) + (10 \times 3) + (0 \times 2) + (0 \times 1)) / 20 = 3.5 \\ \text{Persentase Q2} &= (3.5 / 4) \times 100\% = 87.5\% \\ \text{Rata-rata nilai Q3} &= ((14 \times 4) + (6 \times 3) + (0 \times 2) + (0 \times 1)) / 20 = 3.7 \\ \text{Persentase Q3} &= (3.7 / 4) \times 100\% = 92.5\% \end{aligned}$$

Pada keinginan penggunaan, nilai persentasenya akan dihitung sesuai dengan rumus pada gambar 11, seperti berikut.

$$\begin{aligned} \text{Rata-rata nilai Q1} &= ((16 \times 4) + (4 \times 3) + (0 \times 2) + (0 \times 1)) / 20 = 3.8 \\ \text{Persentase Q1} &= (3.8 / 4) \times 100\% = 95\% \\ \text{Rata-rata nilai Q2} &= ((12 \times 4) + (8 \times 3) + (2 \times 2) + (0 \times 1)) / 20 = 3.8 \\ \text{Persentase Q2} &= (3.8 / 4) \times 100\% = 95\% \\ \text{Rata-rata nilai Q3} &= ((14 \times 4) + (6 \times 3) + (0 \times 2) + (0 \times 1)) / 20 = 3.7 \\ \text{Persentase Q3} &= (3.7 / 4) \times 100\% = 92.5\% \end{aligned}$$

Dari hasil perhitungan di atas, dapat disimpulkan bahwa pada kesesuaian fungsional, pengguna memberikan penilaian yang sangat tinggi dengan persentase rata-rata sekitar 89.2%. Hal ini menunjukkan bahwa fitur-fitur aplikasi terkait laporan kinerja, pengisian perencanaan kinerja individu, dan proses *approval* laporan dianggap berfungsi dengan baik oleh pengguna. Mengenai kemudahan penggunaan, pengguna juga merasa aplikasi ini mudah dipahami dengan persentase rata-rata sekitar 91.5%, menunjukkan bahwa tata letak, navigasi, dan instruksi penggunaan aplikasi dinilai positif. Selain itu, dalam kategori keinginan untuk menggunakan, mayoritas pengguna menyatakan keinginan untuk menggunakan aplikasi ini dengan persentase rata-rata sekitar 94.2%, mencerminkan tingginya antusiasme untuk memanfaatkan aplikasi ini.

4.7. Operation and maintenance

Pada tahap ini, aplikasi yang sudah berjalan secara lokal, diserahkan kepada pihak IT perusahaan untuk selanjutnya dilakukan proses *deploy* agar dapat diakses oleh seluruh karyawan perusahaan. Lalu, saat ditemukan *error* atau *bug* pada aplikasi, dilakukan pemeliharaan dan perbaikan terhadap *error* atau *bug* tersebut.

5. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan, aplikasi *Individual Performance Planning* (IPP) dan *Performance Appraisal* (PA) menunjukkan bahwa aplikasi ini berhasil mengatasi berbagai permasalahan yang dihadapi oleh *admin*, karyawan, dan atasan dalam pengelolaan kinerja. Melalui analisis kebutuhan dan solusi permasalahan, aplikasi ini menyediakan fitur pembatasan akses, *monitoring data*, dukungan teknis, dan pengelolaan jadwal pelaporan yang efektif. Desain sistem dan perangkat lunak yang melibatkan *use case diagram* dan *activity diagram* memastikan interaksi pengguna dengan sistem berjalan lancar. Implementasi menggunakan Visual Studio Code, CodeIgniter 4, dan *template Admin LTE* berhasil mengembangkan fitur utama seperti halaman *log in*, *edit periode*, detail IPP dan PA, serta *log perubahan*.

Pengujian *black box testing* menunjukkan bahwa semua fungsi aplikasi beroperasi sesuai dengan harapan, memastikan kehandalan sistem. Dari hasil *User Acceptance Testing*, dapat disimpulkan bahwa pengguna memberikan penilaian tinggi terhadap kesesuaian fungsional (89.2%), kemudahan penggunaan (91.5%), dan keinginan untuk menggunakan aplikasi (94.2%). Dengan demikian, aplikasi ini tidak hanya menggantikan metode *manual* dengan digitalisasi, tetapi juga meningkatkan efisiensi dan efektivitas dalam manajemen kinerja, memberikan nilai tambah yang signifikan bagi organisasi. Untuk pengembangan lebih lanjut, disarankan perbaikan antarmuka agar lebih menarik

dan peningkatan keamanan aplikasi untuk melindungi data-data yang bersifat rahasia.

DAFTAR PUSTAKA

- [1] A. A. Wahid, "Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi," *J. Ilmu-ilmu Inform. dan Manaj. STMIK*, no. November, pp. 1–5, 2020.
- [2] D. Darmansah, S. Fernandez, I. K. Putri, M. Y. Fathoni, and S. Wijayanto, "Sistem Pendukung Keputusan Untuk Seleksi Jabatan Dengan Menggunakan Metode Topsis (Studi Kasus: Kantor Camat Lais)," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 3, pp. 2222–2233, 2022, doi: 10.35957/jatisi.v9i3.2534.
- [3] A. J. M. Novita, A. W. Rizqi, and H. Hidayat, "Aplikasi Penilaian Kinerja Karyawan Berbasis Website Menggunakan Metode Topsis Di Pt Petrokimia Gresik," *JUSTI (Jurnal Sist. dan Tek. Ind.)*, vol. 2, no. 3, p. 307, 2022, doi: 10.30587/justicb.v2i3.3913.
- [4] M. Tabrani, Suhardi, and H. Priyandaru, "Sistem Informasi Manajemen Berbasis Website Pada Unl Studio Dengan Menggunakan Framework Codeigniter," *J. Ilm. M-Progress*, vol. 11, no. 1, pp. 13–21, 2021, doi: 10.35968/m-pu.v11i1.598.
- [5] G. B. Sulistyono and P. Widodo, "Pemanfaatan Framework Codeigniter Untuk Pembuatan Sistem Informasi Perpustakaan Berbasis Web." *Indonesian Journal on Networking and Security*, Jakarta, pp. 2354–6654, 2021. [Online]. Available: https://repository.bsi.ac.id/index.php/unduh/item/337398/Jurnal_PIW_Pemanfaatan-Framework-Codeigniter-Untuk-Pembuatan-Sistem-Informasi-Perpustakaan-Berbasis-Web.pdf
- [6] D. Setiyadi and Herlawati, "Structured Query Language (SQL) untuk Purchase Order (PO) menggunakan SQL Server," *Bina Insa. ICT J.*, vol. 6, no. 1, pp. 75–88, 2019.
- [7] Y. Wahyudin and D. N. Rahayu, "Analisis Metode Pengembangan Sistem Informasi Berbasis Website: A Literatur Review," *J. Interkom J. Publ. Ilm. Bid. Teknol. Inf. dan Komun.*, vol. 15, no. 3, pp. 26–40, 2020, doi: 10.35969/interkom.v15i3.74.
- [8] W. Ningsih and H. Nurfauziah, "Perbandingan Model Waterfall Dan Metode Prototype Untuk Pengembangan Aplikasi Pada Sistem Informasi," *J. Ilm. METADATA*, vol. 5, no. 1, pp. 83–95, 2023, doi: 10.47652/metadata.v5i1.311.
- [9] K. Nistrina and L. Sahidah, "Unified Modelling Language (Uml) Untuk Perancangan Sistem Informasi Penerimaan Siswa Baru Di Smk Marga Insan Kamil," *J. Sist. Inf.*, vol. 04, no. 01, pp. 12–23, 2022.
- [10] Uminingsih, M. Nur Ichsanudin, M. Yusuf, and

- S. Suraya, "Pengujian Fungsional Perangkat Lunak Sistem Informasi Perpustakaan Dengan Metode Black Box Testing Bagi Pemula," *STORAGE J. Ilm. Tek. dan Ilmu Komput.*, vol. 1, no. 2, pp. 1–8, 2022, doi: 10.55123/storage.v1i2.270.
- [11] B. B. Sasongko, F. Malik, F. Ardiansyah, A. F. Rahmawati, F. D. Adhinata, and D. P. Rakhmadani, "Pengujian Blackbox Menggunakan Teknik Equivalence Partitions pada Aplikasi Petgram Mobile," *J. ICTEE*, vol. 2, no. 1, p. 10, 2021, doi: 10.33365/jictee.v2i1.1012.