

ANALISIS & PENGEMBANGAN APLIKASI PAYROLL BERBASIS WEB PADA INSTITUSI PERGURUAN TINGGI SWASTA (STUDI KASUS DI UNIVERSITAS SWASTA XYZ)

Imron Rizki Maulana, Muhammad Fazly Qusyairy, Giri Purnama, Salwa Putri Kamilah

Teknik Informatika, Universitas Sains Indonesia

Jl. Akses Tol No.50, Gandasari, Kec. Cibitung, Kabupaten Bekasi, Jawa Barat 11650

Imron.rizki@sains.ac.id

ABSTRAK

Pengelolaan penggajian di Universitas Swasta XYZ saat ini masih dilakukan secara manual menggunakan aplikasi Microsoft Excel. Proses ini menghadapi kendala krusial berupa kurangnya transparansi, tingginya risiko human error, serta kompleksitas variabel penggajian dosen yang meliputi tunjangan fungsional, pendidikan, hingga kehadiran. Penelitian ini bertujuan mengembangkan sistem informasi payroll berbasis web untuk mengotomatisasi perhitungan gaji dan penyediaan slip gaji digital. Pengembangan sistem dilakukan menggunakan metode *System Development Life Cycle* (SDLC) model *Waterfall*, dengan bahasa pemrograman PHP, framework Laravel, dan database MySQL. Pengujian sistem dilakukan menggunakan metode *Black Box Testing*. Hasil penelitian berupa aplikasi payroll yang mampu mengintegrasikan data kepegawaian, memvalidasi perhitungan gaji secara otomatis, dan menyediakan akses mandiri bagi karyawan. Penerapan sistem ini terbukti meningkatkan efisiensi waktu pemrosesan gaji serta meminimalisir kesalahan perhitungan pada institusi.

Kata kunci : Gaji, Laravel, Payroll, SDLC, UML, Waterfall

1. PENDAHULUAN

Perkembangan teknologi informasi berlangsung dengan sangat cepat, ditandai oleh penggunaan sistem berbasis komputer yang mendukung aktivitas individu maupun organisasi, termasuk institusi pendidikan. Hal ini telah menjadi kebutuhan penting untuk memperoleh informasi. Informasi yang diperoleh kemudian dimanfaatkan sebagai alat pendukung dalam berbagai kegiatan, terutama dalam pengelolaan sumber daya manusia. Sumber daya manusia memegang peranan krusial bagi organisasi atau institusi dalam mencapai tujuan mereka. Oleh karena itu, perkembangan organisasi sangat bergantung pada kinerja setiap karyawan. Sebaliknya, karyawan juga membutuhkan penilaian terhadap potensi diri mereka untuk mengasah keterampilan serta mendapatkan penghasilan atau gaji demi kelangsungan hidup [1].

Sistem informasi memiliki peran krusial dalam merubah struktur maupun proses operasional sebuah organisasi atau institusi agar dapat mencapai tujuan yang diinginkan. Selain itu, sistem informasi penggajian yang efektif harus dilengkapi dengan aturan dan standar guna membantu pengelolaan keuangan organisasi secara lebih teratur [2]. Dengan penerapan sistem informasi, proses perhitungan gaji dapat dilakukan secara lebih objektif. Kesalahan dalam penggajian bisa berdampak serius bagi perusahaan maupun karyawan yang bekerja di dalamnya. Oleh karena itu, diperlukan perancangan sistem informasi penggajian untuk meminimalkan terjadinya kesalahan tersebut [3].

Penggajian karyawan merupakan aspek krusial dalam sebuah organisasi, baik itu perusahaan maupun instansi pemerintah, di mana upah atau gaji diberikan kepada pekerja sebagai imbalan atas jasa dan pekerjaan mereka selama periode tertentu. Setiap

perusahaan memiliki kebijakan pengelolaan *payroll* yang berbeda karena standar penentuan besaran gaji tiap pegawai tidak sama. Oleh sebab itu, diperlukan sistem manajemen *payroll* yang efektif. Namun, saat ini masih banyak organisasi atau perusahaan yang belum mampu mengelola penggajian dengan baik karena masih menggunakan metode konvensional dalam perhitungan gaji. Situasi ini menimbulkan berbagai masalah seperti kesulitan dalam pencatatan gaji, keterlambatan pembayaran kepada pegawai, serta ketidakakuratan laporan *payroll* [4].

Sistem penggajian di Universitas Swasta XYZ saat ini masih menggunakan aplikasi mandiri berbasis *Excel*, di mana setiap penambahan data dilakukan secara manual. Hal ini menimbulkan beberapa kendala, seperti proses pencetakan slip gaji yang juga dilakukan secara manual dan harus didistribusikan kepada banyak karyawan, serta risiko kesalahan dalam perhitungan gaji. Selain itu, tantangan lain dalam proses *payroll* di Universitas Swasta XYZ adalah banyaknya parameter perhitungan yang harus diperhitungkan karena variasi jenis karyawan dosen. Parameter tersebut meliputi gaji pokok, tunjangan jabatan, tunjangan pendidikan S3, rekap kehadiran, BPJS ketenagakerjaan dan kesehatan baik pribadi maupun yayasan, pajak penghasilan (PPH), kenaikan transisi, serta tunjangan jabatan rangkap. Berdasarkan permasalahan yang dihadapi oleh Universitas Swasta XYZ serta temuan dari penelitian, dibutuhkan sebuah sistem penggajian baru yang terhubung dengan basis data untuk mempermudah proses perhitungan dan pengumpulan data oleh perusahaan, sehingga dapat mengurangi kesalahan yang terjadi pada sistem sebelumnya [5]. Tujuan dari penelitian adalah dapat meningkatkan transparansi dalam proses penggajian, sehingga setiap karyawan dapat dengan mudah

mengakses informasi terkait gaji mereka secara jelas dan akurat, serta meminimalkan potensi ketidakjelasan atau kesalahpahaman mengenai komponen gaji. Sebagai rencana penyelesaian masalah, pengembangan sistem ini akan dilakukan menggunakan metodologi *System Development Life Cycle* (SDLC) model *Waterfall*. Pendekatan ini dipilih karena karakteristiknya yang terstruktur, dimulai dari analisis kebutuhan mendalam, perancangan sistem menggunakan *Unified Modeling Language* (UML), hingga implementasi dan pengujian. Secara teknis, aplikasi akan dibangun menggunakan *Framework* Laravel dengan basis data MySQL untuk memastikan keamanan, kecepatan akses, dan kemudahan pengembangan lebih lanjut. Solusi ini diharapkan dapat menghasilkan manajemen penggajian yang akurat, efektif, dan transparan bagi seluruh civitas akademika Universitas Swasta XYZ.

2. TINJAUAN PUSTAKA

Penelitian ini dapat berhasil dilakukan berkat dasar dari penelitian sebelumnya yang menjadi acuan. Penelitian sebelumnya menyediakan kerangka kerja, data, dan wawasan yang sangat berharga untuk memahami konteks serta metode yang relevan:

Penelitian pertama dilakukan oleh [1] yang berfokus pada pengembangan aplikasi payroll berbasis web di lingkungan perguruan tinggi. Meskipun objek penelitian sama-sama mengenai payroll universitas, perbedaan mendasar terletak pada penggunaan teknologi, di mana penelitian tersebut menggunakan *Framework* CodeIgniter 3, sedangkan penelitian ini menerapkan *Framework* Laravel yang menawarkan arsitektur lebih modern dan skalabilitas yang lebih baik.

Penelitian kedua dilakukan oleh [2] yang merancang sistem informasi penggajian untuk Badan Usaha Milik Desa (BUMDes) Gading Sari guna mengurangi kesalahan perhitungan manual. Metode yang digunakan adalah *Waterfall*, mulai dari analisis hingga pemeliharaan. Fitur utamanya mencakup pencatatan kehadiran, data karyawan, dan perhitungan gaji otomatis yang menghasilkan slip gaji digital. Meskipun berhasil mendigitalkan proses manual, studi ini berfokus pada struktur organisasi desa yang variabel penggajiannya relatif lebih sederhana dibandingkan institusi pendidikan tinggi.

Penelitian ketiga dilakukan oleh [3] yang membahas sistem penggajian untuk personil proyek menggunakan metode *Waterfall*. Penelitian ini menekankan pada analisis kebutuhan untuk mengatasi ketidakefisienan pembuatan laporan gaji personil yang bersifat dinamis. Sistem ini dirancang untuk menerbitkan gaji secara otomatis dan meminimalisir kesalahan hitung. Namun, fokus penelitian ini adalah pada personil proyek yang masa kerjanya berbasis kontrak atau durasi proyek, sehingga strukturnya berbeda dengan penggajian dosen dan staf universitas yang memiliki komponen tunjangan tetap, fungsional, dan struktural yang lebih kompleks.

2.1. Manajemen Penggajian di Perguruan Tinggi

Sistem penggajian di perguruan tinggi memiliki karakteristik unik dibandingkan perusahaan dagang atau jasa umum. Komponen gaji tidak hanya berdasarkan kehadiran, tetapi juga melibatkan variabel akademis seperti jabatan fungsional (Asisten Ahli hingga Guru Besar), tunjangan pendidikan, serta tugas tambahan struktural. Kompleksitas ini menuntut sistem informasi yang mampu melakukan kalkulasi kondisional secara otomatis dan akurat untuk menghindari *human error* yang sering terjadi pada perhitungan manual menggunakan *spreadsheet* [6].

2.2. Kerangka Kerja Pengembangan Sistem

Untuk menangani kompleksitas data tersebut, penelitian ini memanfaatkan Laravel, sebuah kerangka kerja PHP yang menerapkan konsep *Model-View-Controller* (MVC). Penggunaan Laravel dipilih karena fitur keamanan bawaan (*built-in security*) dan *Eloquent ORM* yang memudahkan pengelolaan relasi database yang kompleks antar tabel karyawan, jabatan, dan komponen gaji [7].

Metodologi pengembangan yang diterapkan adalah SDLC Model *Waterfall*. Metode ini dipilih karena kebutuhan sistem penggajian di universitas bersifat pasti (*fixed requirements*) dan membutuhkan dokumentasi yang jelas di setiap tahapannya, mulai dari analisis kebutuhan (*Analysis*), perancangan sistem (*Design*), penulisan kode (*Code*), hingga pengujian (*Testing*) [8]. Pendekatan linier ini menjamin bahwa setiap modul perhitungan gaji telah divalidasi sebelum melangkah ke tahap implementasi, sehingga meminimalisir *bug* fatal pada saat sistem digunakan untuk transaksi keuangan [9].

3. METODE PENELITIAN

3.1. Pengumpulan Data

Teknik pengumpulan data kualitatif melibatkan tiga prinsip utama yang harus dijalankan oleh peneliti, yaitu observasi, wawancara, dan analisis dokumen [10]. Proses pengumpulan data dilakukan dengan wawancara dan observasi:

- Observasi: dilakukan dengan mengamati data berupa contoh template yang dipakai dalam proses perhitungan gaji serta komponen-komponen penggajian yang digunakan oleh bagian SDM di Universitas Swasta XYZ.
- Wawancara: dilakukan dengan pihak SDM Universitas Swasta XYZ untuk membahas secara rinci alur penggajian berdasarkan data hasil observasi. Tujuan dari wawancara ini adalah untuk memahami secara mendalam proses tersebut agar dapat dijadikan acuan dalam pengembangan aplikasi *payroll*.

3.2. Metode Perancangan

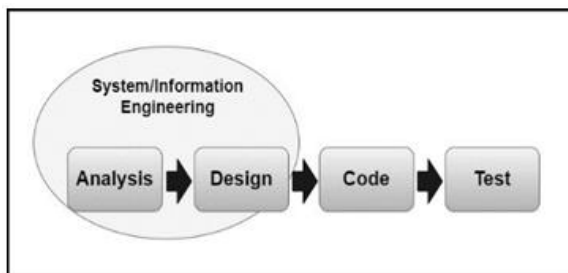
Metode perancangan sistem dilakukan dengan menggunakan UML (*Unified Modeling Language*). UML adalah metode desain yang dipakai untuk memvisualisasikan suatu masalah berdasarkan hasil

analisis yang telah dilakukan sebelumnya [11]. Tujuan penggunaan UML adalah untuk menyediakan pemodelan yang jelas dan mudah dipahami dalam proses pengembangan serta pertukaran model, sehingga dapat dimengerti oleh berbagai pihak secara luas. Dalam perancangan sistem UML terdapat beberapa jenis diagram yang umumnya sering digunakan, berikut beberapa diagram yang digunakan dalam proses perancangan UML sistem *payroll* Universitas Swasta XYZ:

- Use Case Diagram* menggambarkan setiap perilaku sistem yang dikembangkan dengan mendeskripsikan interaksi antara aktor dan sistem secara rinci.
- Activity Diagram* memperlihatkan alur kerja atau rangkaian aktivitas dalam sebuah sistem atau proses bisnis, termasuk menu-menu yang tersedia dalam perangkat lunak.
- Class Diagram* menunjukkan struktur sistem dengan menentukan kelas-kelas yang akan dibuat sebagai bagian dari sistem tersebut.

3.3. Model Waterfall

Sistem pengembangan perangkat lunak yang sistematis dikenal sebagai Software Development Life Cycle (SDLC). Ada banyak model SDLC yang dapat digunakan, masing masing dengan metode dan fitur unik [12]. Namun, metode waterfall adalah salah satu model pengembangan perangkat lunak yang bersifat linear dan sekuensial yang digunakan penulis dalam penelitian ini. Model ini terdiri dari beberapa tahap, yang masing-masing harus diselesaikan secara bertahap. Fokus metode Waterfall adalah:

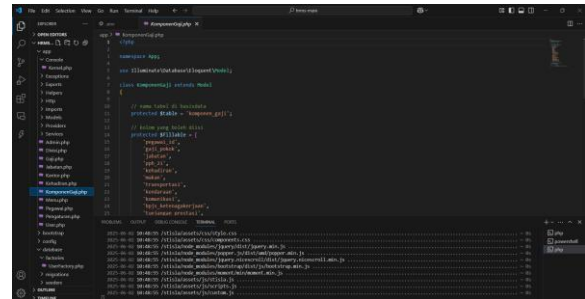


Gambar 1. Metode Waterfall

Adapun alur proses yang dilakukan pada gambar 1 adalah sebagai berikut:

- Analysis**
Tahap analisis meliputi identifikasi pengguna sistem, aktivitas yang akan mereka lakukan, serta waktu dan tempat penggunaan sistem tersebut.
- Design**
Proses desain menetapkan cara kerja sistem, mencakup perangkat lunak, perangkat keras, infrastruktur jaringan, formulir, dan laporan yang juga termasuk pengelolaan basis data.
- Code**
Tahap pengkodean adalah proses di mana desain sistem diubah menjadi bahasa yang dapat

dimengerti oleh komputer. Pada tahap ini, programmer bertanggung jawab untuk menuliskan instruksi sesuai dengan permintaan pengguna. Dengan kata lain, pengkodean adalah kegiatan menulis serangkaian kode yang mengikuti aturan penulisan (syntax) dari bahasa pemrograman yang digunakan.



Gambar 2. Pengkodean

Merujuk pada Gambar 2, spesifikasi teknis yang digunakan dalam pembangunan sistem ini meliputi beberapa komponen utama. Dari sisi *back-end*, sistem dikembangkan menggunakan bahasa pemrograman PHP versi 8.3 yang didukung oleh Composer 2.4.4 dan *framework* Laravel. Untuk antarmuka pengguna, digunakan *library* Bootstrap. Seluruh proses pengodean dilakukan melalui editor Visual Studio Code, dengan dukungan basis data MySQL dan server lokal XAMPP v3.3.0.

d) Testing

Tahap pengujian (*testing*) dilakukan sebagai langkah verifikasi akhir sebelum sistem digunakan. Proses ini bertujuan untuk memvalidasi bahwa seluruh fungsi dalam sistem telah beroperasi secara optimal dan selaras dengan spesifikasi kebutuhan yang telah didefinisikan sebelumnya.

4. HASIL DAN PEMBAHASAN

4.1. Analysis

Tahap analisis dilakukan dengan mengidentifikasi sistem yang perlu dikembangkan untuk memenuhi kebutuhan pengguna. Proses identifikasi ini dilakukan melalui observasi dan wawancara dengan pihak-pihak yang terlibat langsung dalam proses penghitungan gaji karyawan di Universitas Swasta XYZ.

4.2. Design

Tahap desain dimulai dengan proses perancangan sistem menggunakan UML (*Unified Modeling Language*) yang bertujuan untuk memvisualisasikan rancangan sistem serta alur proses yang akan dijalankan.

Penggambaran yang penulis sampaikan, terdapat pada diagram-diagram berikut:

4.3. Use Case Diagram

Tujuan utama pembuatan *Use Case Diagram* adalah untuk memperlihatkan pihak-pihak yang terlibat dalam pengoperasian sistem. Pihak yang

disebut sebagai aktor ini cakupannya luas, tidak hanya terbatas pada pengguna (manusia), tetapi juga mencakup sistem eksternal di luar aplikasi utama.

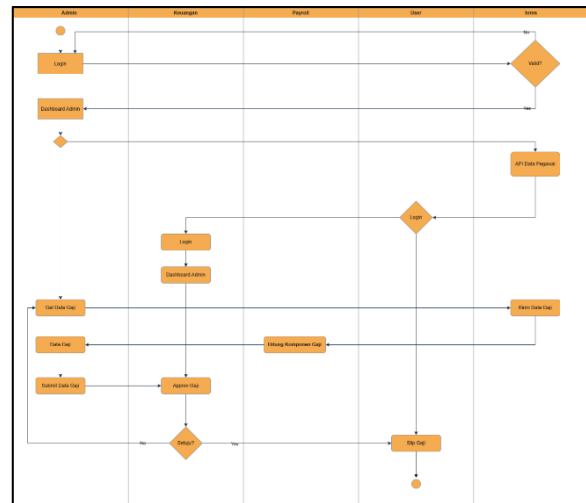


Gambar 3. Use Case Diagram dari Sistem Payroll.

Pada Gambar 3, Sistem payroll yang dikembangkan di Universitas Swasta XYZ melibatkan empat aktor utama, yaitu *Admin*, *Finance*, *User*, dan *HRMS*. Dari keempat aktor tersebut, yang berinteraksi langsung dengan sistem adalah *Admin*, *Finance*, dan *User*. Sedangkan aplikasi *HRMS* merupakan sistem eksternal pihak ketiga yang menyimpan data karyawan tendik dan dosen. Data tersebut mencakup informasi identitas, status kepegawaian, jabatan, golongan pangkat/jabatan fungsional, pendidikan terakhir, pengalaman kerja, pelatihan atau kursus yang pernah diikuti serta profil lengkap karyawan lainnya.

4.4. Activity Diagram

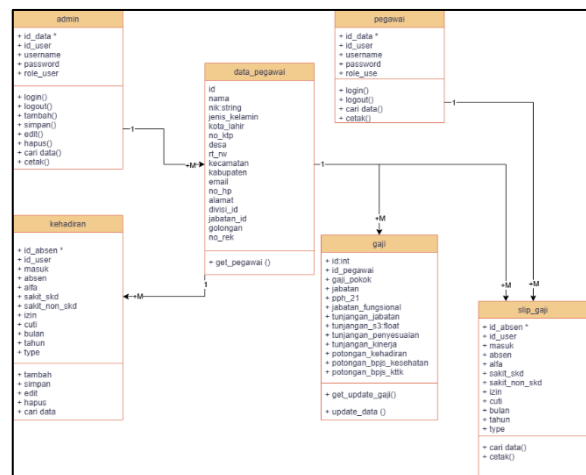
Activity Diagram menggambarkan alur kerja sistem payroll yang dimulai dari *Admin* melakukan *login*. Proses validasi login dilakukan melalui API pada *HRMS*, dan jika berhasil, *Admin* akan diarahkan ke halaman *dashboard*. Selanjutnya, *Admin* dapat memberikan akses *payroll* kepada setiap pegawai yang terdaftar di *HRMS*. Setelah itu, *Admin* mengambil data gaji dari *HRMS* dan mengirimkannya ke sistem *payroll* untuk diproses perhitungannya. Hasil perhitungan tersebut kemudian ditampilkan dalam formulir gaji karyawan di *dashboard Admin*. Data gaji ini selanjutnya dikirimkan oleh *Admin* ke bagian *Finance* untuk mendapatkan persetujuan. Jika permintaan ditolak, status data akan diperbarui menjadi ditolak dan *Admin* akan mengambil kembali data tersebut untuk proses selanjutnya. Namun jika disetujui, slip gaji akan dibuat dan diserahkan kepada *User*. Untuk penggambaran alurnya digambarkan pada Gambar 4 berikut:



Gambar 4. Activity Diagram dari Sistem Payroll.

4.5. Class Diagram

Class diagram sistem *Payroll* menggambarkan hubungan antar tabel dalam basis data. Hubungan antar tabel tersebut dapat dilihat pada Gambar 5, yang menunjukkan 6 (enam) kelas utama yang digunakan untuk menyimpan data-data yang diperlukan dalam proses penggajian.



Gambar 5. Class Diagram dari Sistem Payroll.

4.6. Code

Tahap berikutnya adalah proses *development*, di mana aplikasi mulai dikembangkan. Pembuatan aplikasi ini menggunakan *framework* *Laravel* dan *database* *MySQL*. *Laravel* adalah *framework* *PHP* modern yang menyediakan berbagai fitur lengkap untuk mempercepat pengembangan aplikasi *web* dengan struktur kode yang rapi dan mudah dipelihara [13]. Sedangkan *MySQL* merupakan salah satu *DBMS* populer yang menggunakan bahasa *SQL* untuk mengelola database dan bersifat *open source*, sehingga dapat digunakan secara gratis [13].

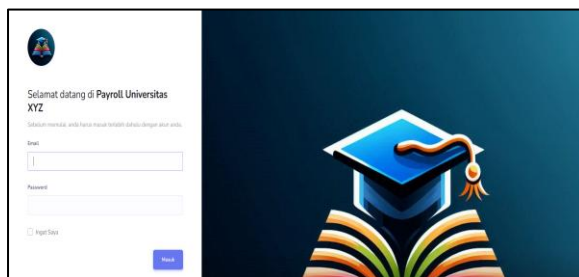
Pada proses pengembangan aplikasi yang dilakukan berdasarkan hasil dari perancangan yang dibuat. Secara *role* terdapat 3 jenis pengguna yaitu:

- Admin:** Bertanggung jawab untuk memasukkan data karyawan beserta informasi gaji, serta mengelola hak akses setiap pengguna. Tugas ini biasanya dilakukan oleh staf bagian SDM.
- Finance:** Berperan sebagai pihak yang memberikan persetujuan terhadap data gaji yang telah dimasukkan.
- User:** Merupakan karyawan yang memiliki akses untuk memeriksa dan mencetak slip gaji mereka sendiri.

Adapun mengenai aplikasi yang dikembangkan, berikut ini adalah uraian terkait fitur, fungsi, dan tujuan utama dari aplikasi tersebut.

4.7. Form Login

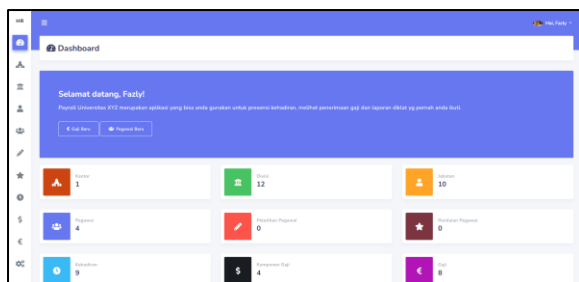
Halaman *login* seperti yang terlihat pada Gambar 6 digunakan oleh setiap pengguna yang ingin mengakses aplikasi *payroll* dengan memasukkan *email* dan *password* yang telah terdaftar di HRMS. Proses *login* pada aplikasi *payroll* dimulai setelah pengguna mengisi *email* dan *password*, kemudian aplikasi akan melakukan permintaan melalui API yang disediakan oleh HRMS untuk memeriksa apakah email tersebut terdaftar. Jika responsnya valid, pengguna akan diarahkan ke tampilan *Dashboard*.



Gambar 6. Form Login

4.8. Dashboard Admin dan Finance

Pada *dashboard admin* terdapat diagram untuk mengetahui pengeluaran gaji pada setiap bulannya. *Role* yang mempunyai akses pada menu ini adalah *admin* dan *finance*. Juga terdapat data yang memperlihatkan karyawan yang memiliki pinjaman. Berikut tampilan yang telah dibuat untuk *dashboard admin* dan *finance* pada Gambar 7:



Gambar 7. Tampilan Dashboard Admin dan finance

4.9. Dashboard User

Akses *dashboard user* dapat diakses untuk *role user* yang artinya merupakan semua karyawan.

Tampilan yang diperlihatkan adalah menu data gaji semua periode yang didapatkan. Berikut Tampilan *Form Data Gaji* yang telah di buat pada gambar 8 :

Gambar 8. Tampilan Data Gaji

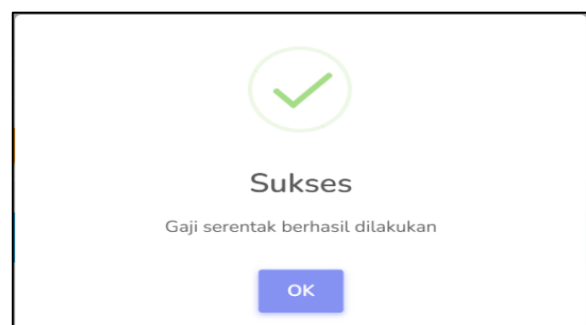
4.10. Form Penghitungan Gaji

Form ini digunakan untuk proses perhitungan gaji. Aplikasi *Payroll* secara otomatis akan menarik data setiap bulan dan pada tanggal tertentu dari HRMS, yang meliputi data karyawan, golongan, jabatan, serta komponen-komponen penggajian lainnya. Setelah data diperoleh, sistem akan melakukan perhitungan secara otomatis dan menampilkan hasilnya pada *form* penghitungan gaji yang kemudian dikirimkan ke bagian keuangan untuk mendapatkan persetujuan. Berikut Tampilan *Form Perhitungan Gaji* yang telah di buat pada gambar 9 :

Gambar 9. Tampilan Perhitungan Gaji

4.11. Form Persetujuan Gaji

Proses persetujuan gaji dilakukan oleh pengguna dengan peran di bagian keuangan. Mereka akan menyetujui gaji yang sudah dihitung dan dikirimkan oleh admin. Setelah disetujui, slip gaji akan secara otomatis dibuat. Berikut Tampilan *Form Persetujuan Gaji* yang telah di buat pada gambar 10 :



Gambar 10. Tampilan Persetujuan Gaji

4.12. Form Cetak Slip Gaji

Fitur cetak slip gaji tersedia untuk semua peran pengguna, memungkinkan mereka mencetak slip gaji sesuai dengan periode pembayaran yang diterima. Berikut Tampilan *Form Slip Gaji* yang telah di buat pada gambar 11 :

UNIVERSITAS SWASTA XYZ		STRUK PENERIMAAN GAJI	
Nama	Wahyuni Satrio	Id	12345678
Jabatan	Staf	Departemen	Manajemen
Masa Kerja	1 Tahun	Periode Gaji	Januari 2023
PERHITUNGAN			
A. Penghasilan Pokok		12,000,000	
B. Penghasilan Lain		1,000,000	
C. Penghasilan Bersih		13,000,000	
D. Potongan Pajak		1,655,520	
E. Potongan Lain		1,000,000	
F. Gaji Bersih		10,344,480	
G. Potongan Lain		1,000,000	
H. Gaji Bersih		9,344,480	
I. Potongan Lain		1,000,000	
J. Gaji Bersih		8,344,480	
K. Potongan Lain		1,000,000	
L. Gaji Bersih		7,344,480	
M. Potongan Lain		1,000,000	
N. Gaji Bersih		6,344,480	
O. Potongan Lain		1,000,000	
P. Gaji Bersih		5,344,480	
Q. Potongan Lain		1,000,000	
R. Gaji Bersih		4,344,480	
S. Potongan Lain		1,000,000	
T. Gaji Bersih		3,344,480	
U. Potongan Lain		1,000,000	
V. Gaji Bersih		2,344,480	
W. Potongan Lain		1,000,000	
X. Gaji Bersih		1,344,480	
Y. Potongan Lain		1,000,000	
Z. Gaji Bersih		344,480	

Gambar 11. Tampilan Slip Gaji

Tabel 1. Hasil Pengujian Aplikasi Payroll Dengan Black Box

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	Masuk ke sistem menggunakan akun yang sudah terdaftar di HRMS dengan memasukkan email dan password yang valid.	User berhasil masuk ke halaman Dashboard sesuai dengan peran masing-masing saat login.	Sesuai
2	Masuk menggunakan akun yang belum terdaftar di HRMS.	Menampilkan <i>Pop Up X</i> (Jika Salah)	Sesuai
3	<i>Admin</i> memeriksa data gaji yang telah diambil dari HRMS dan mengirimkannya ke bagian <i>Finance</i> untuk proses persetujuan.	Data gaji dikirimkan ke daftar persetujuan gaji di bagian <i>Finance</i> .	Sesuai
4	Bagian <i>Finance</i> memberikan persetujuan gaji melalui <i>form approval</i> .	Sistem memperbarui status gaji dan membuat slip gaji secara otomatis.	Sesuai
5	<i>User</i> mencetak slip gaji.	Sistem menghasilkan slip gaji dalam format PDF.	Sesuai
6	<i>Admin</i> mengakses <i>dashboard</i> yang khusus disediakan untuk <i>admin</i> .	Sistem menampilkan <i>dashboard admin</i> sesuai <i>module admin</i>	Sesuai
7	Bagian <i>Finance</i> mengakses <i>dashboard</i> khusus untuk <i>Finance</i> .	Sistem menampilkan data pengeluaran gaji per bulannya dan bisa di sesuaikan dengan fitur filer bulan.	Sesuai
8	<i>User</i> mengakses <i>dashboard</i> yang diperuntukkan khusus bagi <i>User</i> .	Sistem menampilkan <i>form</i> data gaji <i>user</i> dan bisa di cetak untuk slip gaji	Sesuai

Berdasarkan hasil pengujian pada Tabel 1, secara keseluruhan *form* yang dibuat telah diuji menggunakan skenario yang telah ditentukan dan menghasilkan *output* sesuai dengan harapan. Hal ini menunjukkan bahwa aplikasi tersebut berfungsi dengan baik, sehingga dapat digunakan oleh bagian SDM Universitas Swasta XYZ dalam mengelola penggajian karyawannya.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian dan pembahasan, dapat disimpulkan bahwa perancangan aplikasi payroll pada Universitas Swasta XYZ berhasil dilakukan dengan memvisualisasikan kebutuhan sistem melalui desain UML (*Use Case, Activity, dan Class Diagram*). Implementasi sistem menggunakan metode SDLC model *Waterfall* dengan bahasa pemrograman PHP dan framework Laravel telah menghasilkan aplikasi yang mampu menangani kompleksitas komponen gaji di lingkungan universitas. Selain itu, pengujian menggunakan metode *Black Box* menunjukkan bahwa seluruh fungsi aplikasi, mulai dari manajemen data karyawan, perhitungan gaji otomatis, hingga pencetakan slip gaji, telah berjalan sesuai skenario

3.4. Testing

Tahap akhir adalah pengujian sistem pada aplikasi. Uji coba ini dilakukan untuk memastikan bahwa aplikasi sudah memenuhi kebutuhan yang diharapkan. Pengujian dilakukan menggunakan metode *black box*, yaitu metode yang fokus pada pemeriksaan *output* berdasarkan data uji tanpa memperhatikan struktur internal, serta menguji fungsi-fungsi aplikasi [14]. Berikut hasil dari proses pengujian aplikasi dengan metode *black box*:

yang diharapkan dan bebas dari kesalahan fungsional. Sistem ini siap diimplementasikan untuk meningkatkan akurasi dan efisiensi manajemen penggajian.

DAFTAR PUSTAKA

- [1] R. Rinaldiansyah Nugraha and G. Purnama, "Web-Based Payroll Application Development in Higher Education Institutions (Case Study at XYZ University)," *Jtsi*, vol. 4, no. 2, pp. 335–345, 2023.
- [2] A. Yevance Mailoa, D. Trisnawarman, and Wasino, "Payroll System Design using Waterfall Method in Gadingsari Village Owned Enterprises," *International Journal of Application on Sciences, Technology and Engineering*, vol. 1, no. 2, pp. 403–410, 2023, doi: 10.24912/ijaste.v1.i2.403-410.
- [3] N. T. Santoso, H. Sifaulloh, A. Prasetyo, and M. A. Yaqin, "Analisis dan Perancangan Software Penggajian Personil Proyek Menggunakan Metode Waterfall," *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, vol.

- 5, no. 3, pp. 244–253, 2023, doi: 10.28926/ilkomnika.v5i3.336.
- [4] N. Hidayati, “Using the Waterfall Model in Developing an Employee Payroll Information System,” *International Journal of Information System & Technology Akreditasi*, vol. 7, no. 4, pp. 236–249, 2023.
- [5] S. N. Alisya, B. Harahap, and M. R. Wayahdi, “Design of a Website-Based Battuta University Employee Payroll System,” vol. 1, no. 4, pp. 16–21, 2024.
- [6] R. Setiawan, A. Ikhwana, and A. M. Muhidin, “Rancang Bangun Sistem Informasi Penggajian Di Perguruan Tinggi Swasta,” *Jurnal Algoritma*, vol. 14, no. 2, pp. 373–383, Feb. 2015, doi: 10.33364/algoritma/v.14-2.373.
- [7] F. Sinlae, E. Irwanda, Z. Maulana, and V. Eka Syahputra, “Penggunaan Framework Laravel dalam Membangun Aplikasi Website Berbasis PHP,” *Jurnal Siber Multi Disiplin*, vol. 2, no. 2, pp. 119–132, Jul. 2024, doi: 10.38035/jsmd.v2i2.186.
- [8] N. T. Santoso, H. Sifaulloh, A. Prasetyo, and M. A. Yaqin, “Analisis dan Perancangan Software Penggajian Personil Proyek Menggunakan Metode Waterfall,” *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, vol. 5, no. 3, pp. 244–253, Dec. 2023, doi: 10.28926/ilkomnika.v5i3.336.
- [9] D. R. Hidayat and D. Hamdani, “PERANCANGAN SISTEM INFORMASI PENGGAJIAN KARYAWAN BERBASIS WEBSITE DI PERUSAHAAN X MENGGUNAKAN METODE PROTOTYPE,” *Rabit : Jurnal Teknologi dan Sistem Informasi Univrab*, vol. 9, no. 2, pp. 271–284, Jul. 2024, doi: 10.36341/rabit.v9i2.4837.
- [10] F. Malahati, A. U. B. P. Jannati, Q. Qathrunnada, and S. Shaleh, “Kualitatif: Memahami Karakteristik Penelitian Sebagai Metodologi,” *Jurnal Pendidikan Dasar*, vol. 11, no. 2, pp. 341–348, 2023, doi: 10.46368/jpd.v11i2.902.
- [11] I. G. Anggraeni and A. Suryadi, “Penerapan Sistem Informasi Rukun Tetangga (Rt) Dengan Model Unified Modelling Language,” vol. 3, no. 2, pp. 146–151, 2022.
- [12] S. Anggrian, B. Y. Geni, J. Tj, D. Barat, J. Barat, and D. K. I. Jakarta, “KEPEGAWAIAN MENGGUNAKAN METODE WATERFALL (STUDI KASUS : PT . DOLA USAHA INDONESIA),” *JATI (Jurnal Mahasiswa Teknik Informatika) Vol. 8 No. 1, Februari 2024*, vol. 8, no. 1, pp. 1029–1035, 2024.
- [13] M. Rhifky Wayahdi, F. Ruziq, S. Hafiz, and N. Ginting, “Pelatihan Menjadi Backend Developer Dengan Framework Laravel Pada Siswa Dan Siswi SMK Swasta Free Methodist Medan Training To Become A Backend Developer With The Laravel Framework For Vocational School Students Free Methodist Medan,” *Pengabmas Nusantara*, vol. 6, no. 1, pp. 20–29, 2024.
- [14] D. Novianti and D. Anjani, “Pengujian Aplikasi E-Farmer Dalam Perhitungan Keuntungan Dengan Metode Blackbox Testing Boundary Value Analysis,” *JUNIF - Jurnal Nasional Informatika*, vol. 1, no. 2, pp. 76–81, 2020.