

Rancang Bangun Sistem Informasi *Quality Control* Berbasis *User Acceptance Testing* (UAT) Untuk *Project Digital* Pada PT ARG Solusi Teknologi

Dina Febrianti¹, Ahmaddul Hadi², Yeka Hendriyani³, Geovanne Farell⁴

^{1,2}Universitas Negeri Padang, Indonesia

Jl. Prof. Dr. Hamka Kampus UNP, Air Tawar Padang, Indonesia

*Corresponding author e-mail : dinafebriantii09@gmail.com

ABSTRAK

Penelitian ini bertujuan mengembangkan Sistem Informasi Manajemen Pengujian berbasis User Acceptance Testing (UAT) di PT. ARG Solusi Teknologi, sebuah perusahaan software house yang menghadapi kendala dalam memahami kebutuhan pengguna akhir, perubahan persyaratan proyek, dan pengujian manual yang kurang efisien. Sistem ini dirancang menggunakan framework Laravel, PHP, dan MySQL untuk mengoptimalkan proses pengujian akhir proyek secara lebih efisien dan akurat. Dengan melibatkan pengguna akhir dalam proses UAT, sistem ini bertujuan meningkatkan validasi produk sesuai standar kualitas yang diharapkan dan memastikan kesesuaian dengan kebutuhan pengguna. Implementasi sistem berbasis UAT ini diharapkan dapat meningkatkan kualitas proyek, efisiensi operasional, dan kepuasan pengguna terhadap hasil pengembangan perangkat lunak di PT. ARG Solusi Teknologi.

Kata kunci : Sistem Informasi, Manajemen Pengujian, *User Acceptance Testing*, PT. ARG Solusi Teknologi, *Laravel*, PHP.

ABSTRACT

This study aims to develop a Testing Management Information System based on User Acceptance Testing (UAT) at PT. ARG Solusi Teknologi, a software house company facing challenges in understanding end-user requirements, changing project specifications, and inefficient manual testing processes. The system is designed using the Laravel framework, PHP, and MySQL to optimize the final project testing process more efficiently and accurately. By involving end-users in the UAT process, the system aims to enhance product validation according to expected quality standards and ensure alignment with user needs. The implementation of this UAT-based system is expected to improve project quality, operational efficiency, and user satisfaction with the software development outcomes at PT. ARG Solusi Teknologi.

Keywords: *Information System, Testing Management, User Acceptance Testing, PT. ARG Solusi Teknologi, Laravel, PHP.*

I. PENDAHULUAN

Software house menyediakan jasa untuk membuat atau mengembangkan aplikasi. Dalam dunia yang sudah maju seperti sekarang ini, *software house* sangat penting bagi mereka yang ingin membuat aplikasi unik untuk mendukung bisnis atau perusahaan mereka, seperti mempromosikan barang dan jasa mereka, menjualnya, atau membuat sistem informasi bisnis mereka. Namun, mereka yang tidak bisa membuat atau ingin mendapatkan aplikasi yang mereka butuhkan dengan cepat sangat dibutuhkan oleh *software house*. *Software house* biasanya membuat *aplikasi desktop, web, atau mobile,*

menyediakan layanan multimedia, dan mendesain grafis. [1]

PT. ARG Solusi Teknologi atau yang sering disebut dengan Argenesia adalah salah satu *software house* yang menawarkan jasa pembuatan sistem informasi. Argenesia menyediakan jasa konsultasi sistem informasi hingga pembuatan aplikasi disesuaikan dengan kebutuhan pelanggan. Core utama yang dimiliki Argenesia ini adalah perancangan sistem informasi yang dimana meliputi kegiatan *requirement* dari *client*, analisa kebutuhan sistem, *design user interface* yang akan digunakan, coding untuk pembangunan sistem, testing, implementasi

sistem, serta penjelasan dan training sistem baru kepada *client*.

Salah satu tugas penting dalam pengembangan perangkat lunak adalah pengujian sistem/aplikasi yang telah dibuat agar sesuai dengan standar. Dimana sistem/aplikasi harus mudah digunakan, intuitif, dan memenuhi semua kebutuhan pengguna yang telah diidentifikasi.

User Acceptance Testing (UAT) adalah pengujian yang dilakukan oleh pengguna akhir, yaitu staf atau karyawan perusahaan yang berinteraksi dengan sistem. Pengujian ini bertujuan untuk memverifikasi apakah fungsi-fungsi dalam sistem telah berjalan sesuai dengan kebutuhan dan fungsinya.. [2]

Tujuan dari *user acceptance testing (UAT)* adalah untuk memvalidasi dan memastikan bahwa aplikasi perangkat lunak memenuhi kebutuhan dan persyaratan spesifik pengguna yang dituju sebelum dirilis secara resmi, memastikan fungsionalitas, kegunaan, dan keselarasan tingkat tinggi dengan skenario dunia nyata.[3]

Di era transformasi digital ini, kualitas perangkat lunak yang dihasilkan menjadi faktor krusial dalam menentukan keberhasilan suatu proyek. Oleh karena itu, fokus utama penelitian ini adalah pada pengembangan sistem *quality control*. Melalui *User Acceptance Testing (UAT)*, tim pengembang dapat melakukan validasi langsung terhadap kesesuaian produk yang dibuat dengan standar kualitas yang diharapkan serta memastikan kelayakan sesuai kebutuhan pengguna akhir.

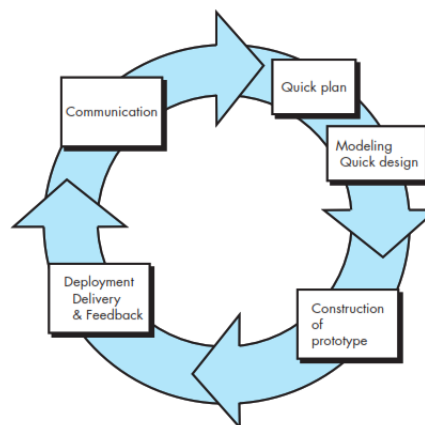
untuk mengatasi permasalahan tersebut di rancanglah sebuah aplikasi sistem informasi *quality control* berbasis *user acceptance testing (uat)* untuk project digital pada pt arg Solusi teknologi. Sistem informasi ini dikembangkan dengan menggunakan Bahasa pemrograman PHP dan dibuat dengan bantuan framework Laravel. Diharapkan dengan adanya Sistem informasi *quality control* berbasis *user acceptance testing (uat)* dapat membantu PT. ARG Solusi Teknologi meningkatkan dan memastikan bahwa produk yang dihasilkan sesuai dengan standar kualitas yang diharapkan.

Pada Perancangan sistem, metode yang akan digunakan sebagai panduan untuk tahapan dalam pembuatan website ini adalah metode *prototype*, yang memungkinkan pengembang untuk menguji dan memperbaiki fitur-fitur sistem secara bertahap sebelum penerapan penuh, metode ini memfasilitasi proses pengembangan yang lebih adaptif, memungkinkan penyesuaian berkelanjutan berdasarkan umpan balik dari pengguna dan pemangku kepentingan terkait. Dengan pendekatan ini, diharapkan sistem dapat lebih *responsive* terhadap kebutuhan pengguna akhir, mengoptimalkan fungsionalitas dan memperoleh umpan balik yang

memuaskan dari pengguna baik dalam kegiatan maupun dalam pengalaman menggunakan sistem yang dirancang.

II. METODE

Metode yang digunakan dalam perancangan website ini adalah metode *prototype*. Metode *prototype* merupakan salah satu pendekatan yang umum digunakan dalam pengembangan rekayasa perangkat lunak di dalam lingkungannya [4]. Penerapan metode *prototype* bertujuan untuk memperoleh gambaran yang jelas mengenai aplikasi yang akan dikembangkan melalui serangkaian tahapan.



Gambar 1. Metode *prototype*

Gambar 1 menunjukkan tahapan metode *prototype* yang terdiri dari beberapa langkah, yaitu *Communication* (komunikasi), *Quick Plan* (perencanaan awal), *Modelling Quick Design* (pemodelan desain cepat), *Construction of Prototype* (pembuatan prototipe), dan *Development Delivery and Feedback* (pengembangan, pengiriman, dan umpan balik). [5]. Berikut ini adalah tahapan dalam metode *prototype* yang dapat dijelaskan seperti berikut:

Communication (Komunikasi)

Proses ini mencakup keterlibatan yang mendalam antara pengembang dan pemangku kepentingan terkait. Hal ini bertujuan untuk mendapatkan pemahaman yang menyeluruh mengenai kebutuhan, tujuan, dan harapan mereka terhadap sistem yang direncanakan untuk dikembangkan. Dengan keterlibatan yang intensif, pengembang dapat memastikan bahwa setiap aspek yang penting bagi para pemangku kepentingan dipertimbangkan dan diakomodasi dalam desain dan pengembangan sistem.

Quick Plan (Perencanaan Awal)

Pada tahap ini, sebuah rencana kerangka kerja dan fitur utama yang akan diimplementasikan

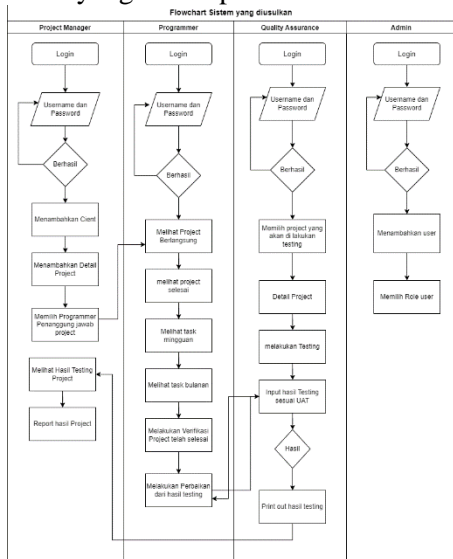
dirancang secara cepat berdasarkan hasil komunikasi sebelumnya.

a. Analisis Sistem Berjalan

Dalam analisis sistem yang sedang berjalan, tujuan adalah untuk memahami bagaimana sistem bekerja dan mengidentifikasi masalah yang dihadapi sistem tersebut. Hal ini bertujuan untuk menjadi dasar dalam menyusun usulan perancangan yang sesuai. [6]

b. Analisis Sistem yang diusulkan

Dalam menganalisis permasalahan yang terjadi di lapangan, sangat penting untuk mengidentifikasi akar masalah dan menemukan solusi yang tepat untuk mengatasinya. Dengan pendekatan ini, tindakan yang diambil dapat lebih sesuai dan efektif dalam menyelesaikan permasalahan yang dihadapi.



Gambar 2. Flowchart Sistem yang diusulkan

Modelling Quick Design (Pemodelan Desain)

Pada tahap ini melakukan perancangan sistem untuk membantu dalam mendefinisikan arsitektur sistem secara keseluruhan. Proses desain mengubah kebutuhan-kebutuhan menjadi bentuk karakteristik yang dimengerti perangkat lunak sebelum dimulai penulisan program.

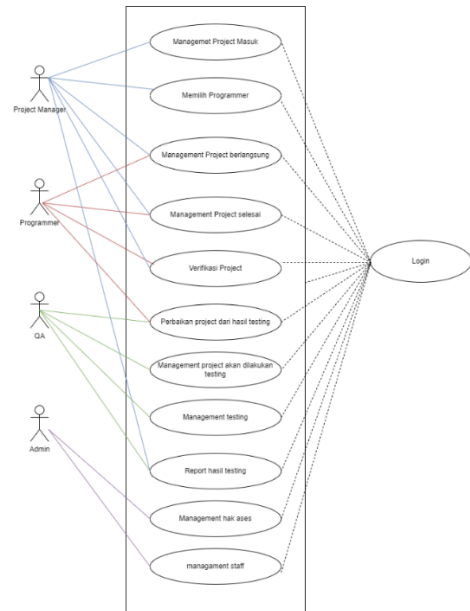
Perancangan Sistem

Perancangan sistem ini menggunakan pemodelan UML (Unified Modeling Language) yang memungkinkan untuk menggambarkan struktur, fungsi, dan proses dari sistem secara jelas dan terstruktur.

a. Use case diagram

Use case diagram merupakan pemodelan untuk menggambarkan kelakuan (behavior) sistem yang akan dibuat. Dengan menggunakan

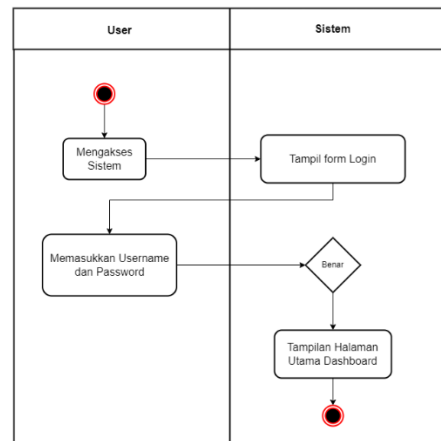
use case diagram, interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat dapat dideskripsikan. Singkatnya, use case diagram digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. [7]



Gambar 3. use case diagram

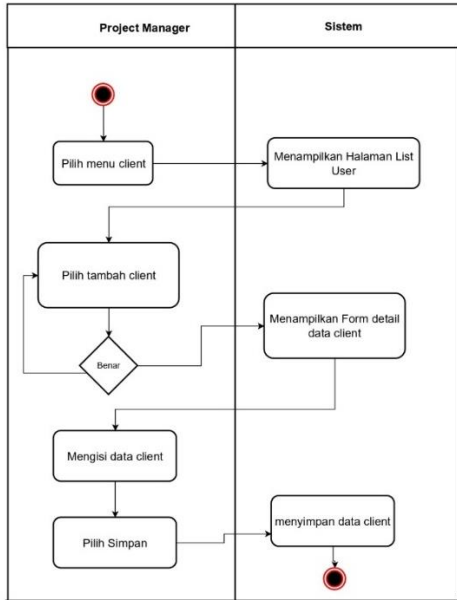
b. Activity diagram

Activity diagram menggambarkan alur kerja atau aktivitas dari suatu sistem atau proses bisnis. Penting untuk dicatat bahwa activity diagram menunjukkan aktivitas yang dilakukan oleh sistem, bukan tindakan yang dilakukan oleh aktor. Jadi, diagram ini berfokus pada aktivitas yang dapat dilakukan oleh sistem itu sendiri [8]. Aliran kerja dapat dilihat pada gambar dibawah ini:



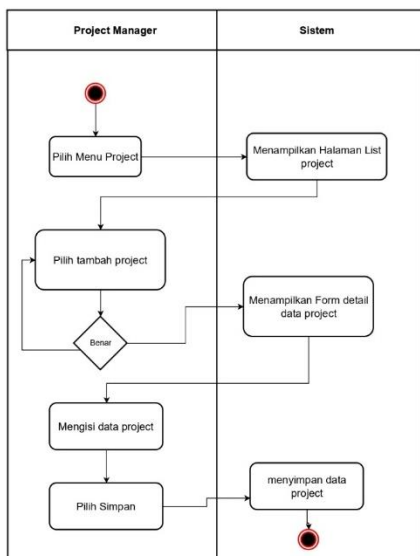
Gambar 4. Activity Diagram Login

Pada Gambar 4 acitivity login ini user mengisi username dan password. Username dan password yang telah dimasukkan akan dicek terlebih dahulu keabsahannya (validation). Jika data tersebut valid, maka sistem akan membuat session data sesuai dengan hak akses



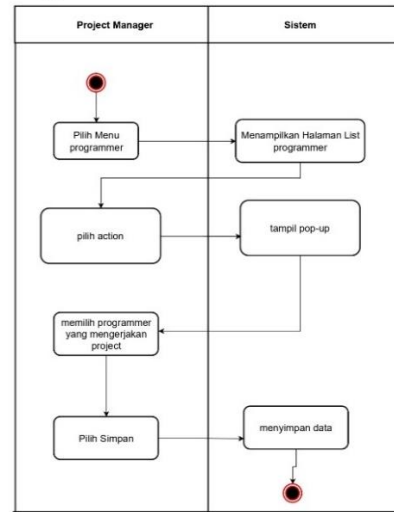
Gambar 5. Activity Diagram Menambahkan client

Pada gambar 5, pihak yang terlibat adalah project manager. Tujuan dari proses ini adalah untuk mencatat proyek yang akan dilaksanakan atau sedang berlangsung, dengan fokus pada kebutuhan klien dari perusahaan atau instansi tertentu. Proses ini dimulai dengan project manager.



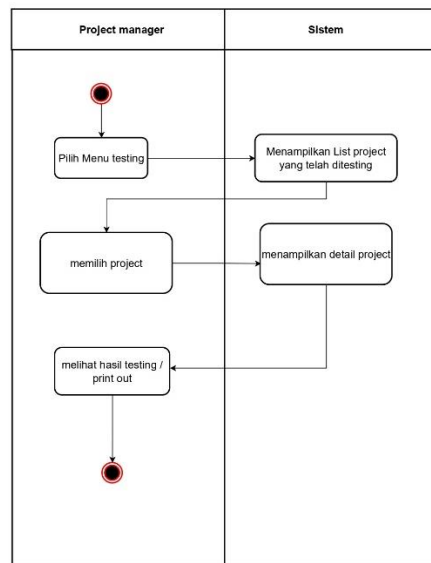
Gambar 6. Menambah Detail Project

Pada gambar 6, pihak yang terlibat adalah project manager. Tujuan dari proses ini adalah untuk menambahkan detail project yang dimulai dari menambahkan nama project, memilih client, tanggal dimulai, tanggal berakhir, deksripsi project.



Gambar 7. Activity Diagram Memilih Programmer

Pada gambar 7, Project Manager (PM) memiliki peran sentral dalam proses pemilihan programmer yang akan bertanggung jawab atas pelaksanaan suatu proyek. Setelah proses seleksi selesai, detail proyek akan disampaikan secara langsung kepada programmer yang terpilih.

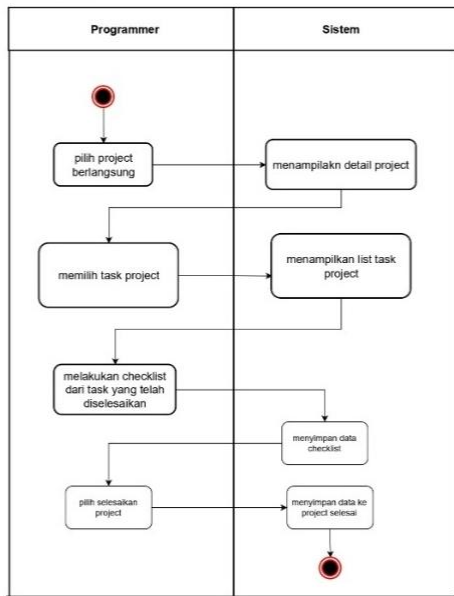


Gambar 8. Activity Diagram Report Hasil Testing

Pada gambar 8, proses pelaporan hasil pengujian dilakukan oleh Project Manager (PM),

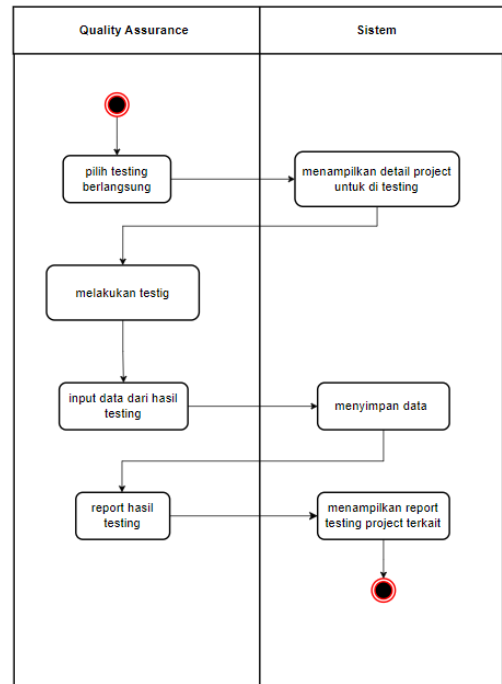
yang menggunakan opsi "Report Hasil Testing" yang telah disediakan dalam menu proyek.

menyebabkan sistem menampilkan detail lengkap dari proyek tersebut.



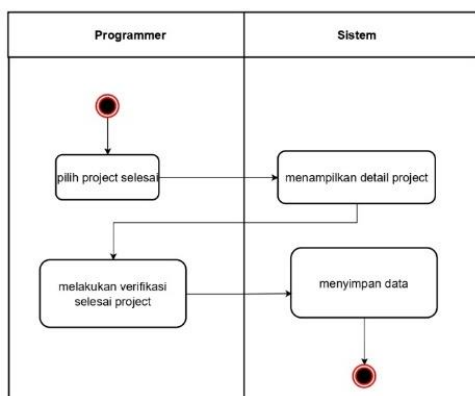
Gambar 9. Activity Diagram Project Berlangsung

Pada gambar 9, Proses ini mencerminkan tahapan yang terstruktur dan terkoordinasi dalam perjalanan pelaksanaan proyek, di mana programmer memperoleh akses terhadap informasi rinci terkait proyek yang memungkinkan mereka untuk memulai dan mengelola proyek dengan efisien.



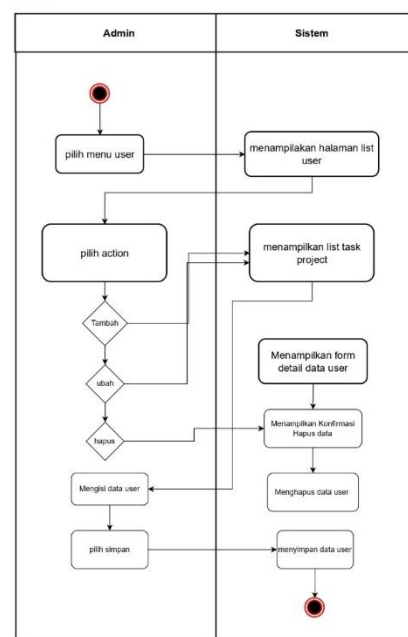
Gambar 11. Activity Diagram Testing Project

Dalam konteks diagram aktivitas ini, Quality Assurance (QA) berperan sebagai pihak yang terlibat secara langsung. Kegiatan yang esensial dalam lingkup ini adalah pengelolaan pengujian proyek. Tujuan utama dari proses ini adalah memfasilitasi pengguna untuk dapat dengan efisien mengelola pengujian dari proyek yang telah diselesaikan.



Gambar 10. Activity Diagram Verifikasi Project Selesai

Proses ini bertujuan untuk melakukan verifikasi terhadap kelengkapan proyek sebelum dilakukan pengujian, dengan tujuan utama melakukan pengecekan terakhir terhadap setiap tugas dan fitur. Langkah-langkah yang diambil melibatkan pemilihan opsi "Project Selesai" pada menu oleh programmer, yang kemudian

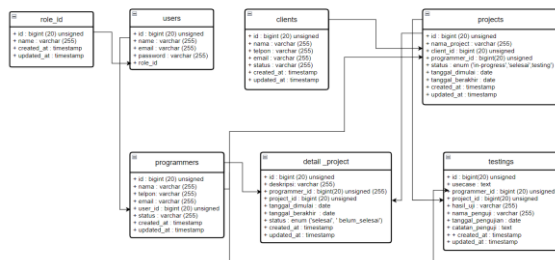


Gambar 12. Acitivity Diagram Kelola User

Pada activity diagram ini yang terlibat adalah admin. Aktivitas yang dilakukan dalam activity ini adalah mengelola user. Dalam kegiatan ini admin memilih menu user sehingga aplikasi menampilkan halaman daftar user. Kemudian admin pilih edit data user yang akan di edit datanya, untuk menghapus data user bisa memilih hapus dan untuk menambah user bisa memilih tambah.

c. *Class diagram*

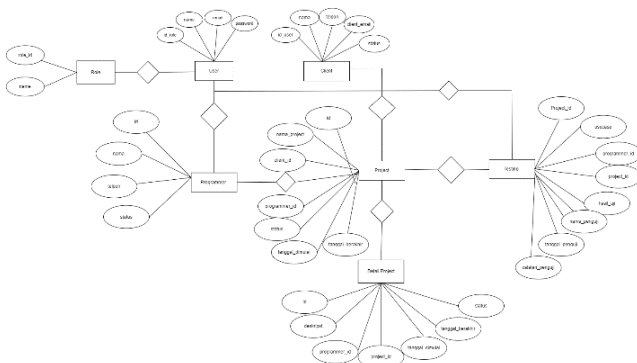
Class diagram adalah diagram yang digunakan untuk menggambarkan struktur kelas dalam sebuah sistem. Diagram ini membantu memodelkan struktur das'ar sistem atau aplikasi dengan menunjukkan entitas-entitas utama yang ada serta hubungan dan interaksi antar kelas [9].



Gambar 13. *Class Diagram*

Perancangan ERD (Entity Relationship Diagram)

Dalam proses perancangan basis data, penting untuk membuat skema *Entity Relationship Diagram* (ERD) yang menggambarkan entitas (aktor) yang terlibat dalam basis data serta hubungan antar entitas tersebut. Berikut adalah rancangan ERD yang diusulkan untuk sistem ini:



Gambar 14. Rancangan ERD

Construction of Prototype (Pembentukan Prototype)

Setelah analisis desain selesai, dilakukan pembuatan prototipe melalui proses implementasi

rancangan tersebut dalam bentuk penulisan program. Sistem akan dibangun menggunakan *framework Laravel*, yang telah terbukti mampu mempercepat pengembangan aplikasi dan menyediakan struktur yang kokoh bagi aplikasi web yang kompleks.

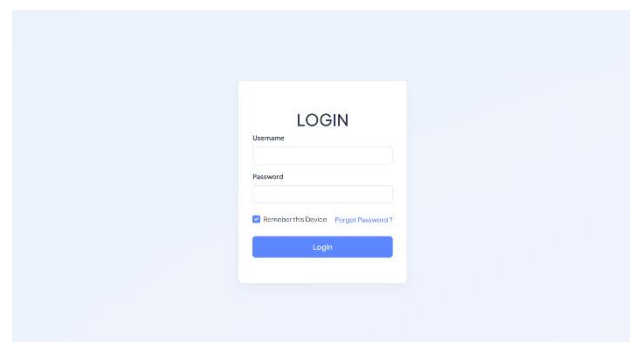
Development Delivery and Feedback

Tahap terakhir, yaitu *Development Delivery and Feedback*, melibatkan pengiriman versi prototipe kepada pemangku kepentingan untuk mendapatkan umpan balik awal. Sebelum pengiriman dilakukan, pengujian dilakukan untuk memastikan bahwa aplikasi berjalan sesuai dengan kebutuhan dan tujuan yang diharapkan. Umpan balik yang diterima kemudian digunakan untuk melakukan iterasi dan perbaikan terhadap prototipe sebelum diluncurkan secara penuh.

III. HASIL DAN PEMBAHASAN

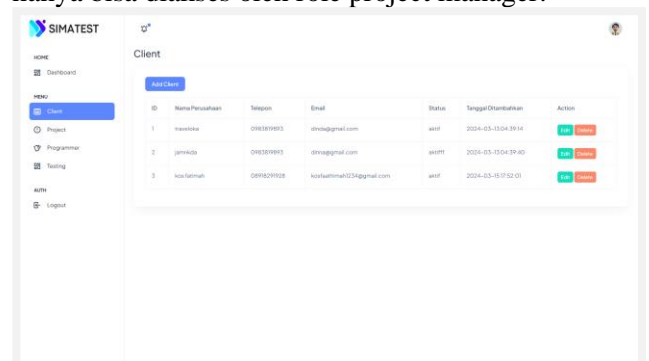
Hasil Rancangan Sistem

Hasil rancangan pada gambar 15 merupakan halaman login untuk semua user.



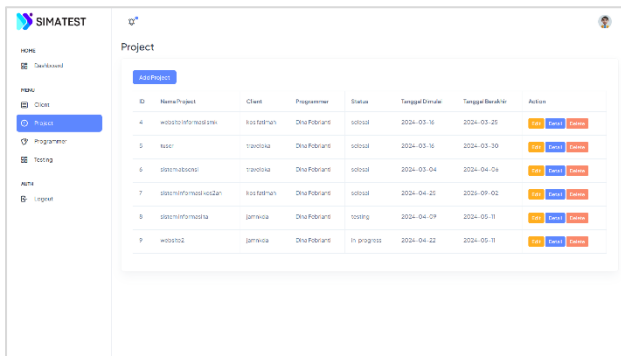
Gambar 15. halaman login

Pada gambar 16 merupakan halaman untuk manajemen client. Halaman manajemen client ini hanya bisa diakses oleh role project manager.



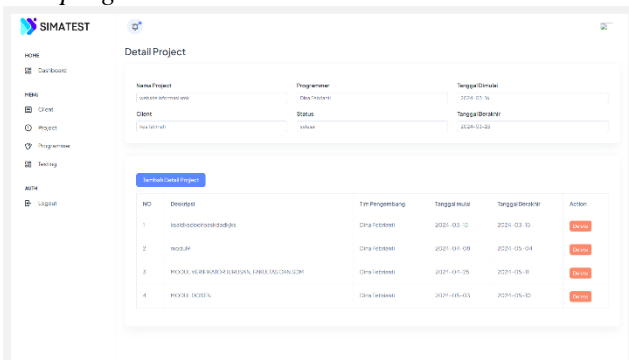
Gambar 16. halaman menu client

Pada gambar 17 *project manager* dapat menambahkan *project* yang akan dikerjakan dengan rentang waktu sesuai yang telah disepakati. Pada halaman ini *project manager* juga bisa mengedit/*update* data *project* pada *table project*.



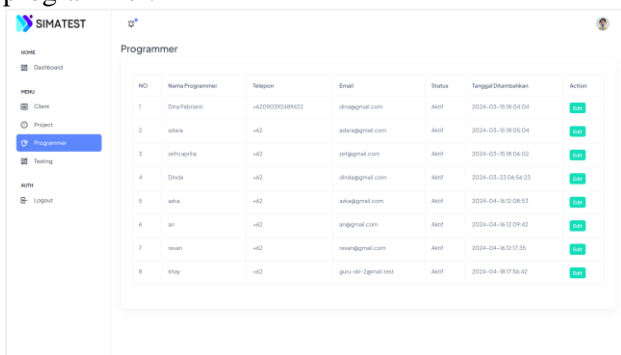
Gambar 17. Halaman Menu Project

Pada gambar 17 akan ditampilkan info dari *table project*, dimana info tersebut merupakan dari *project* yang sedang dikerjakan. Pada halaman ini untuk menambahkan detail *project* yang akan dikerjakan oleh *programmer*.



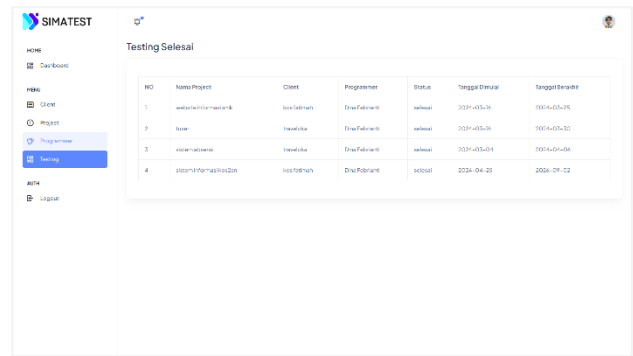
Gambar 18. Halaman Detail Project

Pada gambar 19 *project manager* dapat melakukan *update* data. Data ini otomatis bertambah jika *admin* menambahkan *user* dengan *role programmer*.



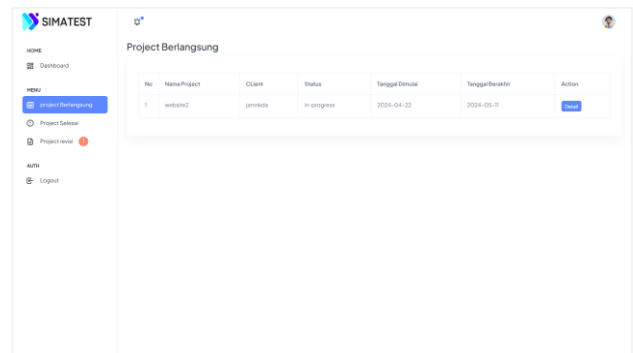
Gambar 19. Halaman Programmer

Pada gambar 20 merupakan gambar untuk melihat testing yang telah selesai. Dimana jika *testing* dan status sudah selesai maka *project* telah selesai dan bisa digunakan oleh *client*.



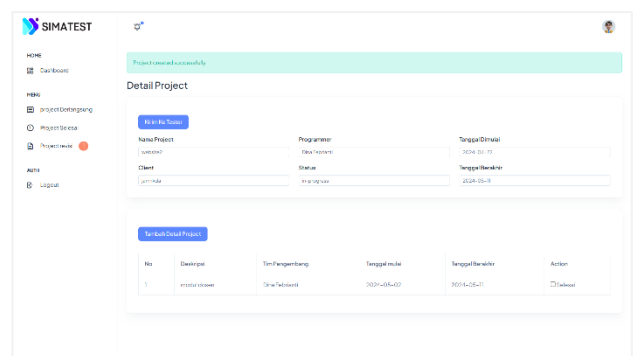
Gambar 20. Halaman Testing Selesai

Pada gambar 21 programmer dapat melihat *project* yang akan dikerjakan. di dalam halaman ini ada *button* detail yang berfungsi untuk meredirect ke halaman detail *project* berlangsung.



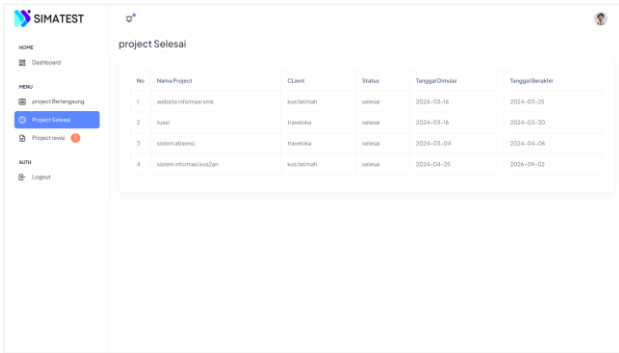
Gambar 21. Halaman Project Berlangsung

Pada gambar 22 merupakan halaman detail *project* berlangsung. Halaman ini hanya bisa diakses oleh *user programmer*. Pada halaman ini berfungsi melihat detail *project* atau *task* yang akan dikerjakan selama waktu yang ditentukan.



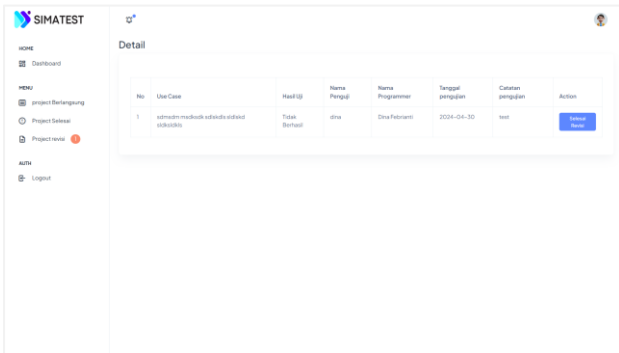
Gambar 22. Halaman Detail Project

Pada gambar 23 merupakan halaman menu *project* selesai. Halaman ini hanya bisa diakses oleh *programmer*. Disini user dapat melihat *project* yang telah selesai.



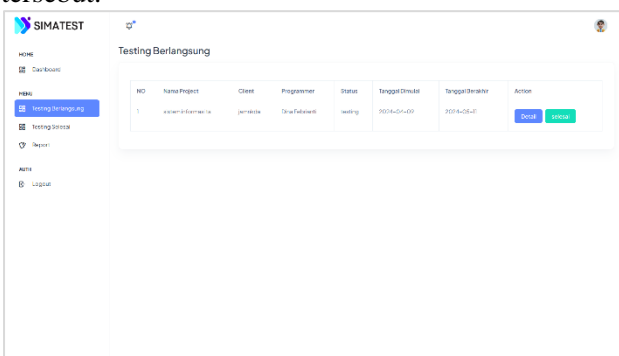
Gambar 23. Halaman Project Selesai

Pada gambar 24 ini akan tampil data *table project* dengan status testing. Pada halaman ini ada *button* detail, jika ingin melihat detail *project* yang direvisi silahkan klik detail. Maka akan di *redirect* ke halaman detail project *revisi*.



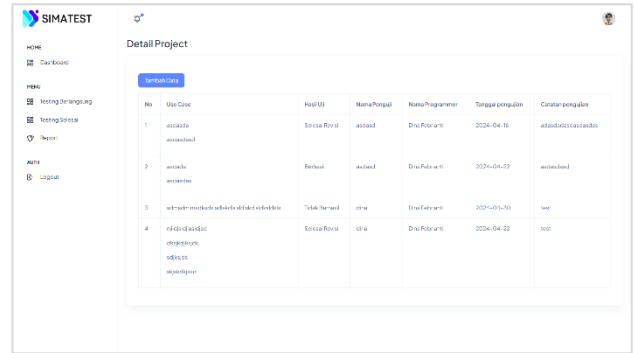
Gambar 24. Halaman Detail Project Revisi

Pada gambar 25 menampilkan *list project* yang sedang berlangsung *testing* sistem/ aplikasi yang telah dikirim oleh *user programmer* terlebih dahulu. Untuk menambahkan detail hasil pengujian dari *project* tersebut.



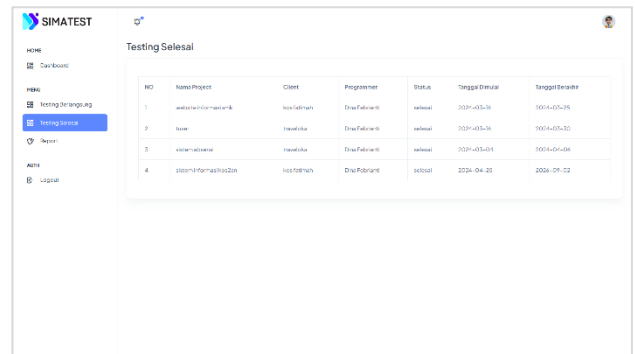
Gambar 25. Testing Berlangsung

Pada gambar 26 merupakan halaman detail testing berlangsung. Pada halaman ini hanya bisa diakses oleh *user quality assurance*. Pada halaman ini akan menampilkan hasil pengujian yang telah dilakukan. Jika ingin menambahkan data silahkan klik *tambah data*.



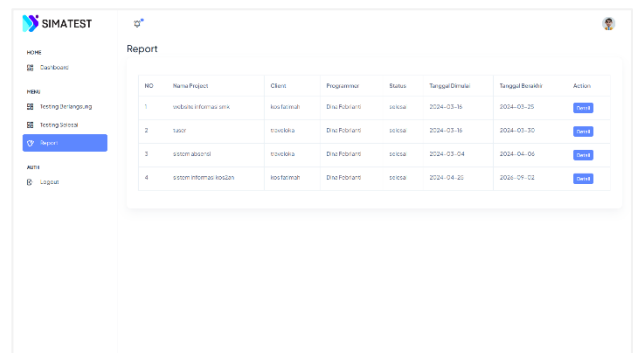
Gambar 26. Halaman Detail Testing

Pada gambar 33 merupakan menu halaman testing selesai. Pada halaman ini hanya diakses user *quality assurance*. Halaman ini menampilkan list testing telah selesai. Jika status telah selesai maka *project* tersebut sudah siap dipakai oleh *client*. status telah selesai maka *project* tersebut sudah siap dipakai oleh *client*.



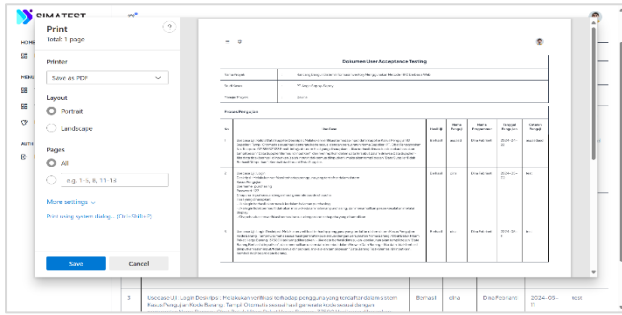
Gambar 27. Halaman Testing

Pada gambar 28 merupakan halaman menu *report*. Pada halaman ini hanya bisa diakses oleh *user quality assurance*. Halaman ini berfungsi untuk melihat *report* dari hasil pengujian yang telah dilakukan. Jika ingin melihat detail dari hasil pengujian silahkan klik *button* detail.



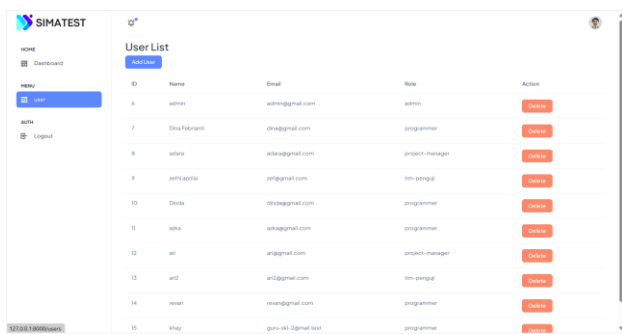
Gambar 28. Halaman Report

Pada gambar 35 merupakan detail *report* dari hasil pengujian. Halaman ini hanya diakses oleh *user quality assurance*. Jika ingin menjadikan pdf silahkan *ctrl+p* maka akan tampil



Gambar 29. Hasil Report

Pada gambar 30 merupakan tampilan menu *user*. Pada halaman ini hanya bisa diakses oleh *admin*. *Admin* bisa menambahkan *user* dan mendelete *user*.



Gambar 30. Halaman Report

Pengujian Sistem

Pengujian sistem dilakukan untuk mengevaluasi keberhasilan dan kualitas sistem yang telah dibangun sesuai dengan harapan yang diinginkan. Metode yang digunakan dalam pengujian adalah white-box testing, di mana struktur internal kode program dieksplorasi untuk memastikan bahwa setiap jalur dan kondisi yang mungkin terjadi telah diuji secara menyeluruh.

Rumus CC didasarkan pada jumlah *node* dan *edge* dalam *flowgraph* program. *Cyclomatic Complexity* (CC) merupakan sebuah metrik yang digunakan untuk mengukur kompleksitas logika dalam sebuah program [10] Secara matematis, rumus CC didefinisikan sebagai berikut:

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

Keterangan:

- V(G) : *cyclomatic complexity*.
- E : jumlah *edge* pada *flowgraph*.
- N : jumlah *node* pada *flowgraph*.

Rumus ini memberikan nilai minimum CC adalah 1 (satu) . Semakin tinggi nilai CC, semakin kompleks struktur aliran kontrol dari program tersebut. Oleh karena itu, CC sering digunakan sebagai indikator potensial untuk menemukan bagian-bagian kode yang mungkin memerlukan perhatian lebih dalam pengujian dan pemeliharaan.

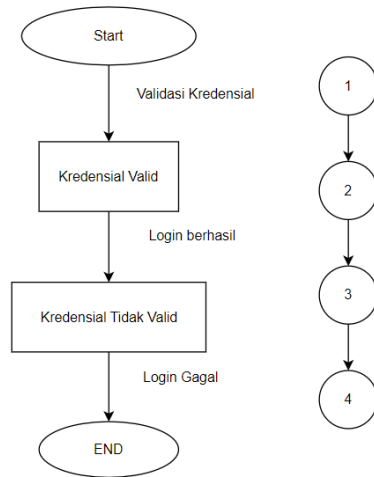
Berikut adalah penjelasan mengenai pengujian yang dilakukan dari salah satu *function* yang ada di dalam sistem.

Function Login

```

1 public function login(Request $request)
2 {
3     $credentials = $request->validate([
4         'email' => 'required|email',
5         'password' => 'required',
6     ]);
7
8     if (Auth::attempt($credentials)) {
9         // Berhasil login, redirect ke halaman yang diinginkan
10        return redirect('/');
11    } else {
12        // Gagal login, kembali ke halaman login
13        return redirect('/login')->with('error', 'Invalid credentials');
14    }
15 }
    
```

Gambar 31. Code Function Login



Gambar 32. Flowgraph Function Login

Menghitung cyclomatic complexity:

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

$$V(G) = 3 - 4 + 2 = 1$$

Ket :

- E = 3 (jumlah Panah)
- N = 4 (Jumlah Node)

Berdasarkan perhitungan cyclomatic complexity untuk function destroy didapatkan 1 independent path dari potongan code pada gambar 31 seperti yang terlihat pada tabel

Path	Jalur	Skenario	Hasil
1	1-2-3-4	1. Start: Permintaan untuk login. 2. Validasi Kredensial: Kredensial yang diberikan divalidasi. 3. Kredensial Valid: Kredensial yang diberikan valid. 4. Login Berhasil: Pengguna berhasil login. 5. End: Proses selesai.	Berhasil

Berdasarkan hasil uji fungsi login yang telah dilakukan dengan menggunakan *cyclomatic complexity* (CC), terdapat satu jalur eksekusi yang dapat diidentifikasi. Berikut adalah pembahasan mengenai hasil uji fungsi login:

1. Jalur Eksekusi

Dari hasil perhitungan CC, terdapat satu jalur eksekusi yang melibatkan empat node atau langkah dalam fungsi login. Jalur tersebut adalah:

- a. Mulai: Permintaan untuk login.
- b. Validasi Kredensial: Kredensial yang diberikan divalidasi.
- c. Kredensial Valid: Kredensial yang diberikan valid.
- d. Login Berhasil: Pengguna berhasil login.
- e. Selesai: Proses selesai

2. Skenario

jalur eksekusi tersebut menggambarkan skenario di mana seorang pengguna mencoba untuk login ke sistem. Proses dimulai dengan pengguna mengirimkan permintaan login, kemudian sistem akan memvalidasi kredensial yang diberikan. Jika kredensial valid, pengguna akan berhasil login ke sistem.

3. Hasil Uji

Dari hasil uji tersebut, dapat disimpulkan bahwa fungsi login berhasil menjalankan proses autentikasi dengan baik, serta mengelola kasus ketika kredensial yang diberikan oleh pengguna valid dan menghasilkan login yang berhasil.

Dengan demikian, hasil uji fungsi login menunjukkan bahwa fungsi tersebut berfungsi sesuai dengan yang diharapkan dan berhasil mengelola proses autentikasi pengguna dengan baik.

IV. KESIMPULAN

Berdasarkan hasil perancangan sistem informasi quality control berbasis user acceptance testing dapat disimpulkan sebagai berikut

1. Sistem yang dihasilkan adalah sistem manajemen pengujian untuk memudahkan pengujian akhir proyek.
2. Sistem yang dihasilkan adalah sistem manajemen pengujian untuk memudahkan pengujian akhir proyek.
3. Metode user acceptance testing berhasil diterapkan pada website PT. ARG Solusi Teknologi, memastikan pengujian menyeluruh dengan melibatkan pengguna akhir, sehingga meningkatkan kualitas, keandalan, dan kepuasan pengguna, serta mengurangi risiko kesalahan pasca peluncuran.

DAFTAR PUSTAKA

- [1] Dwi Lestari, F., Irawan, B. and Reditiyamurti, Y. (2015) 'Analysis and Design Web-Based Applications for Software House Information System', 2(2), p. 3442.
- [2] Rahman, A. and Sumartayasa, K. (2023) 'Analisa Kualitas Website BPJS Kesehatan Dengan Metode Web Qual 4 . 0 Dan User Acceptance Testing Di Wilayah Kabupaten Ogan Komering Ulu', 19(2), pp. 237–243.
- [3] Setter, M. (2023). What Is User Acceptance Testing (UAT): Meaning, Definition. Retrieved from <https://usersnap.com/blog/user-acceptance-testing-right/>.
- [4] Siswidiyanto, S., Wijayanti, D., & Haryadi, E. (2020). Sistem Informasi Penyewaan Rumah Kontrakan Berbasis Web Dengan Menggunakan Metode Prototype. *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi dan Komunikasi*, 15(1), 16-23.
- [5] Aditya, R. and Pranatawijaya, V.H. (2021) 'Rancang Bangun Aplikasi Monitoring Kegiatan Menggunakan Metode Prototype', 1(June), pp. 47–57.
- [6] Arafah, T., & Mulyono, H. (2020). Analisis dan perancangan sistem informasi pelayanan pernikahan pada kantor urusan agama (KUA) Kecamatan Jambi Timur. *Jurnal Manajemen Sistem Informasi*, 5(3), 313–327. <http://ejournal.stikomdb.ac.id/index.php/manajemensisteminformasi/article/view/903>
- [7] Hadi, A., Budayawan, K. and Syukhri, S. (2017) 'Perancangan Sistem Informasi Dokumentasi Borang Akreditasi Program Studi', *Jurnal Teknologi Informasi dan Pendidikan*, 10(3), pp. 53–63. Available at: <https://doi.org/10.24036/tip.v10i3.19>
- [8] Hadi, A., Budayawan, K. and Syukhri, S. (2017) 'Perancangan Sistem Informasi Dokumentasi Borang Akreditasi Program Studi', *Jurnal Teknologi Informasi dan Pendidikan*, 10(3), pp. 53–63. Available at: <https://doi.org/10.24036/tip.v10i3.19>
- [9] Jaya, E. A., Febriyani, I., & Putra, R. (2022). Perancangan Sistem Informasi Kenaikan Pangkat Dan Golongan Guru Smkn 1 Sawahlunto Berbasis Web. *Jurnal Sains Dan Teknologi: Jurnal Keilmuan Dan Aplikasi Teknologi Industri*, 22(2), 379. <https://doi.org/10.36275/stsp.v22i2.546>
- [10] Andriyanto, A. et al. (2023) 'OKTAL : Jurnal Ilmu Komputer dan Science Otomatisasi Pengujian Aplikasi POS (Point Of Sale) Menggunakan Metode White Box', *OKTAL : Jurnal Ilmu Komputer dan Science*, 2(6), pp. 1769–1779