

Rancangan Program Otomatisasi Otentikasi Pengguna Untuk Otorisasi Pada Website Dengan Python Dan Selenium Web Driver

Sulthan Muhammad Justice Islam Al Makassar¹, Thomas Budiman²,
Akmal Budi Yulianto³

Program Studi Teknik Informatika¹, Teknik Informatika², Sistem Informasi³
STMIK Jayakarta¹, STMIK Jayakarta², STMIK Jayakarta³

sultanalmakassari@gmail.com, thomas@stmik.jayakarta.ac.id,

Received: 27 Agustus 2022, **Revised:** 30 Agustus 2022, **Accepted:** 13 September 2022

Abstrak: Pada proses penjaminan mutu perangkat lunak terdapat permasalahan yang terjadi pada proses inspeksi ketika terjadi kesalahan pada server production, diantaranya adalah membutuhkan waktu untuk membuat data yang tiruan untuk menguji fitur yang mengalami masalah dan juga untuk mencari informasi terkait user, Penelitian ini bertujuan untuk mengembangkan program yang dapat membantu melakukan beberapa prosedur untuk membantu efisiensi dan efektifitas. Dibangun dengan menggunakan model pengembangan perangkat lunak bernama *Lean Software Development*, paradigma pemrograman yang digunakan berbasis *procedural programming* dan menganut konsep *imperative*. Hasil dari penelitian ini menunjukkan penggunaan program otomatisasi otentikasi pada website berbasis otorisasi dapat mempermudah dan meningkatkan proses penjaminan mutu perangkat lunak, serta pengujian kompatibilitas yang menunjukkan sistem dapat digunakan dengan baik pada beberapa peramban web.

Kata kunci: otomatisasi, otentikasi, dan otorisasi.

Abstract: *In the process of software quality assurance there are problems that occur in the inspection process when an error occurs on the production server, including the need for time to create artificial data to test features that experience problems and also to find information related to the user, this study aims to develop a program that can help perform some procedures to help efficiency and effectiveness. Built using a software development model called Lean Software Development, the programming paradigm used is based on procedural programming and adheres to the concept of imperative. the results of this study indicate the use of authentication automation programs on authorization-based websites can simplify and improve the software quality assurance process, as well as compatibility testing that shows the system can be used well on several web browsers, so as to help the development team or Software Quality Assurance team to more efficiently and effectively create data for code testing purposes and also access users who experience problems or errors on the production server.*

Keywords: automation, authentication, and authorization

I. PENDAHULUAN

Perkembangan teknologi informasi yang semakin massif dan kompleks membuat perkembangan perangkat lunak semakin pesat, hal ini didukung oleh perkembangan teknologi informatika yang ada, dengan demikian memudahkan para pengguna sistem informasi tersebut untuk lebih meningkatkan efektifitas dan efisiensi dari sistem.

Seiring perkembangan teknologi, penjaminan mutu perangkat lunak hadir karena meningkatnya kebutuhan perangkat lunak yang andal dan akurat untuk memenuhi kepuasan para pengguna perangkat lunak,



This work is licensed under a [Creative Commons Attribution 4.0 International License](http://creativecommons.org/licenses/by/4.0/).
<http://journal.stmikjayakarta.ac.id/index.php/JMIJayakarta>

maka dibutuhkan tim penjaminan mutu perangkat lunak pada sebuah infrastruktur dalam menguji fitur yang telah dibuat oleh para pengembang.

Namun karena pada tahap rilis sistem sering terjadi kesalahan pada sistem dan membutuhkan waktu yang tidak sedikit serta banyak nya persiapan yang dilakukan untuk memperbaikinya seperti mengakses akun user serta mengakses beberapa data terkait user, bahkan tim penjaminan mutu perangkat lunak harus beberapa kali membuat data untuk menguji fitur yang sedang di kembangkan, sehingga memengaruhi waktu dari proses penjaminan mutu perangkat lunak.

Oleh karena itu peneliti akan mengangkat tema otomatisasi untuk otentikasi login pada website yang memiliki banyak jumlah pengguna menggunakan teknologi Python dan Selenium webdriver. Yang nantinya akan menjadi sebuah program yang membantu tim pengembang dan tim penjaminan mutu perangkat lunak dalam efektifitas dan efesiansi untuk otorisasi pada perangkat lunak sehingga diharapkan metode ini dapat menggantikan metode manual yang tradisional.

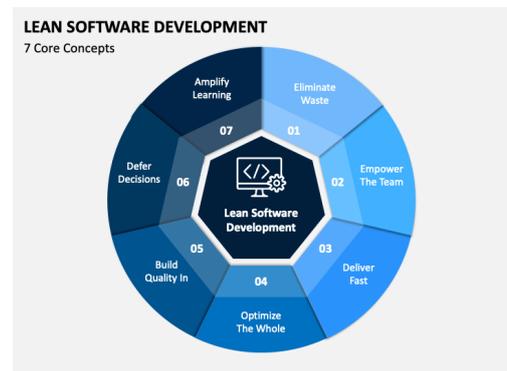
II. METODE DAN MATERI

2.1. Metode *Lean Software Development*

Lean adalah pendekatan sistematis untuk mengidentifikasi dan menghilangkan limbah melalui perbaikan secara terus-menerus, selalu meminta masukan kepada pelanggan dalam tujuan untuk mengejar kesempurnaan.

Lean Software Development adalah sebuah strategi yang dilakukan untuk mengurangi cacat dan menghemat waktu pengerjaan serta selalu memberikan nilai bisnis kedalam produk secara bertahap.[1]

Model ini merupakan metode pengembangan perangkat lunak sistematis, yang meliputi beberapa tahapan sebagai berikut :



Gambar 1: *Lean Software Development*

1. *Eliminate Waste*

Filosofi Lean memandang bahwa segala sesuatu yang tidak menambah nilai kepada pelanggan, merupakan limbah. Limbah yang dimaksud termasuk:

- Partially done work (Penundaan pekerjaan)
- Extra processes (Proses yang berlebihan)
- Extra features (Fitur yang berlebihan)
- Task switching (Beralih tugas)
- Waiting (Menunggu)
- Motion (Gerakan)
- Defects (Cacat)
- Management activities (Manajemen kegiatan)

Jika pekerjaan tertunda, bisa dikatakan pemborosan. Ini juga disebut pemborosan ketika seseorang mengkode, tetapi akhirnya ditinggalkan selama pengembangan. Proses penambahan, seperti penambahan dokumen dan fitur yang tidak sering digunakan pelanggan, juga merupakan pemborosan.

Orang yang berganti pekerjaan juga sia-sia. Menunggu aktivitas lain, tim, atau menunggu proses juga sia-sia. Tindakan yang diperlukan untuk menyelesaikan pekerjaan juga bisa dikatakan boros. Cacat atau kualitas yang buruk merupakan pemborosan. Manajer yang tidak menghasilkan nilai nyata juga merupakan pemborosan.

2. *Empower the Team*



This work is licensed under a [Creative Commons Attribution 4.0 International License](http://creativecommons.org/licenses/by/4.0/).
<http://journal.stmikjayakarta.ac.id/index.php/JMIJayakarta>

Eksekusi terbaik adalah mendapatkan detail dengan benar, dan tidak ada yang tahu detailnya lebih baik daripada orang yang benar-benar melakukan pekerjaan itu. Melibatkan pengembang dalam perincian keputusan teknis sangat penting untuk mencapai keunggulan.

Mereka yang berada di garis depan menggabungkan pengetahuan detail halus dengan kekuatan banyak pikiran. Ketika dilengkapi dengan keahlian yang diperlukan dan dipandu oleh para pemimpin, mereka akan membuat keputusan teknis dan proses yang lebih baik.

Karena keputusan dibuat terlambat dan dilaksanakan dengan cepat, tidak mungkin pemerintah pusat mengatur kegiatan pengembang.

Jadi praktik Lean menggunakan teknik Empower the Team untuk menjadwalkan pekerjaan dan menyertakan mekanisme sinyal lokal sehingga pekerja dapat saling memberi tahu apa yang perlu dilakukan. Dalam pengembangan perangkat lunak ramping, mekanisme tim Pemberdayaan adalah kesepakatan untuk secara teratur memberikan versi perangkat lunak yang berfungsi lebih lancar. Sinyal lokal terlihat melalui grafik, rapat harian, integrasi yang sering, dan pengujian komprehensif.

3. Deliver Fast

Di era perkembangan teknologi yang pesat ini, yang bertahan bukanlah yang terbesar, melainkan yang tercepat. Semakin cepat produk akhir dikirimkan tanpa cacat besar, semakin cepat umpan balik akan diterima dan dimasukkan ke dalam iterasi berikutnya. Kecepatan memastikan bahwa kebutuhan pelanggan terpenuhi hari ini, bukan kebutuhan mereka kemarin.

4. Optimize the Whole

Sistem yang kompleks membutuhkan keahlian yang mendalam di banyak domain yang berbeda. salah satu masalah terberat dalam pengembangan produk adalah bahwa para ahli dalam domain apa pun (misalnya, database atau interface) cenderung memaksimalkan kinerja bagian-bagian produk yang mewakili keahlian mereka sendiri, daripada berfokus pada kinerja sistem secara keseluruhan. Terlalu sering, ketika orang mendahulukan kepentingan sendiri, kepentingan umum diabaikan.

5. Build Quality In

Integritas konseptual berarti bahwa konsep sentral dari suatu sistem bekerja sama dengan keseluruhan yang mulus, terpadu, dan merupakan faktor penting dalam menciptakan integritas. perangkat lunak membutuhkan tingkat integritas ekstra dan harus mempertahankan kegunaannya dari waktu ke waktu. Perangkat lunak biasanya berkembang dengan anggun karena beradaptasi dengan masa depan. Perangkat lunak dengan integritas, dengan arsitektur yang koheren, dengan skor tinggi dalam kegunaan dan penerapan, pemeliharaan, kemampuan beradaptasi, dan ekstensibilitas. Penelitian menunjukkan bahwa integritas berasal dari kepemimpinan yang bijaksana, keahlian yang relevan, komunikasi yang efektif dan disiplin yang baik.

6. Defer decision

Sebagai pengembangan perangkat lunak, seringkali kita dikaitkan dengan beberapa ketidakpastian, hasil yang lebih baik harus dicapai dengan pendekatan berbasis pilihan. Semakin kompleks sebuah sistem, semakin besar kapasitas untuk perubahan yang harus dibangun ke dalamnya, sehingga memungkinkan pengadaan penundaan komitmen yang penting dan krusial, yang menyebabkan keterlambatan dalam menentukan segala sesuatunya.

7. Amplify learning

Pengembangan perangkat lunak adalah proses belajar yang berkesinambungan berdasarkan iterasi saat menulis kode. Software desain adalah proses pemecahan masalah yang melibatkan para pengembang, menulis kode serta apa yang telah mereka pelajari. nilai Software diukur dalam keandalan untuk digunakan dan tidak dalam kesesuaian dengan persyaratannya, untuk itu memahami apa yang pengguna bisnis benar-benar butuhkan dan apa yang pengembang pikirkan untuk kebutuhan pengguna.

2.2. UML (Unified Modelling Language)



This work is licensed under a [Creative Commons Attribution 4.0 International License](http://creativecommons.org/licenses/by/4.0/).
<http://journal.stmikjayakarta.ac.id/index.php/JMIJayakarta>

Unified Modelling Language (UML) Dalam perancangan sistem informasi, peneliti menggunakan perangkat lunak pemodelan berorientasi objek unified modelling language (UML). Peneliti menggunakan unified modelling language dengan tujuan agar kebutuhan sistem yang diinginkan dapat terpenuhi secara tepat sehingga permasalahan dari sistem yang ada dapat diselesaikan. Proses pemodelan UML dimulai dengan analisis kebutuhan sistem dan dilakukan pada use case diagram, kemudian berlanjut ke pembuatan class diagram, sequence diagram, dan activity diagram. Pemodelan perangkat lunak UML juga memberikan kemudahan dalam pengembangan perangkat lunak dan dapat mengetahui aliran atau aliran sistem yang diharapkan oleh perusahaan.[2] Contoh diagram dari UML salah satunya adalah *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*.

2.3. Otomatisasi

Otomatisasi tidak hanya mengurangi pekerjaan manual, tetapi juga mengurangi biaya sekaligus meningkatkan efisiensi waktu yang dibutuhkan.[3] dari kutipan diatas dapat diartikan otomatisasi adalah proses cara kerja untuk pelaksanaan prosedur dan tata kerja dengan otomatis, tujuan dari otomatisasi adalah menyusun prosedur yang rapi, mudah, dan hasil dapat dibuat dengan cepat, akurat.

2.4. Otentikasi

Otentikasi adalah mekanisme yang dapat memverifikasi terhadap identitas asli pengguna.[4] dari kutipan tersebut otentikasi dapat diartikan sebagai cara untuk memastikan identitas seseorang, dengan beberapa prosedur tertentu sebagai syarat verifikasi identitas pengguna.

2.5. Otorisasi

Upaya untuk mengidentifikasi penyusup yang menyalahgunakan hak otorisasi istimewa sumber daya sistem tanpa izin dari cracker atau pengguna yang sah untuk memasuki sistem.[5]

Sehingga dapat diartikan bahwa otorisasi adalah hak yang sah untuk mengakses sebuah halaman ataupun sumberdaya yang ada di dalam sebuah sistem.

2.6. Website

Web adalah sebuah layanan dimana pengguna komputer dapat terhubung ke Internet dalam bentuk teks interaktif, gambar, suara, atau video, dengan keuntungan menghubungkan (linking) satu dokumen ke dokumen lain (hypertext), yang dapat diakses dengan browsing. akses perangkat.[6]

Dari pendapat tersebut dapat disimpulkan bahwa website merupakan layanan yang memiliki kumpulan fungsi dan konten yang tersusun sedemikian rupa menjadi halaman oleh para pengembang yang dapat di akses dengan alamat domain.

2.7. Python

Python merupakan bahasa pemrograman terkenal yang memiliki banyak keunggulan untuk mendukung pemrograman berorientasi objek dan dapat berjalan di berbagai platform sistem operasi seperti PC, Macintosh, UNIX, dll. [7]

2.8. Selenium Web Driver

Selenium Web Driver adalah alat kerangka otomatisasi yang mendukung banyak browser dan aplikasi Ajax dan mengambil pendekatan yang lebih modern dan stabil untuk mengotomatisasi operasi browser. [8]

Dari kutipan tersebut menjelaskan bahwa selenium web driver adalah framework yang sering digunakan oleh pengembang otomatisasi pada banyak browser, dimana selenium web driver sendiri mengimplementasikan metode yang cenderung modern dan stabil dalam otomatisasi.

III. PEMBAHASAN DAN HASIL

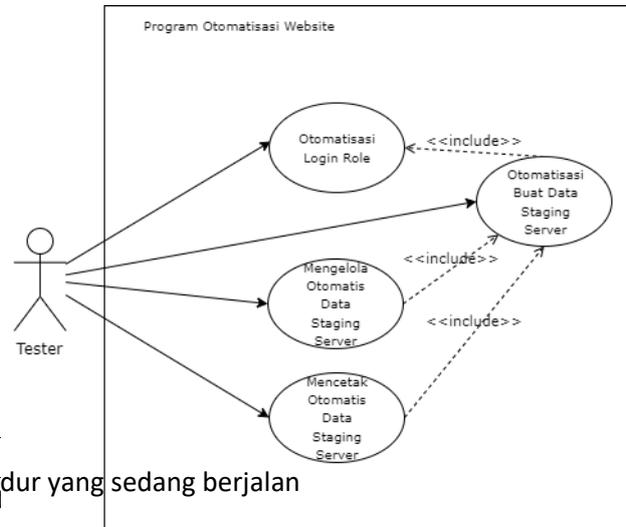
3.1. Analisis dan Perancangan



Tahap ini merupakan tahapan untuk memahami kebutuhan informasi sistem dengan mengumpulkan data informasi dan studi literatur dari sistem yang akan dikembangkan.

1. Analisis Sistem yang Sedang Berjalan

- a. Tim penjaminan mutu perangkat lunak mencari jenis user yang sama untuk mengakses masalah yang terjadi pada laporan user.
- b. Jika memiliki kasus yang di haruskan membuat data, tim penjaminan perangkat lunak melakukan pembuatan data secara manual.
- c. Tim penjaminan perangkat lunak harus membuka akun atau browser baru untuk mengakses lebih dari satu jenis user.



2. Evaluasi Prosedur Yang Sedang Berjalan

Berdasarkan hasil analisa terhadap sistem yang

No.	Permasalahan	
1.	Tim penjaminan mutu perangkat lunak mencari jenis user yang sama untuk mengakses masalah yang terjadi pada laporan user	berbagai jenis user.
2.	Jika memiliki kasus yang di haruskan membuat data, tim penjaminan perangkat lunak melakukan pembuatan data secara manual.	Membuat otomatisasi untuk mengirim data yang di butuhkan pada mandatory website.
3.	Tim penjaminan perangkat lunak harus membuka akun atau browser baru untuk mengakses lebih dari satu jenis user	Membuat otomatisasi untuk membuka sesi baru.

ada pada sistem manajemen dealer yang masih di jalankan secara manual. Maka dapat disimpulkan sebagai berikut :

3. Analisis Kebutuhan Pengguna

Berdasarkan analisis permasalahan dapat diperoleh bahwa permasalahan yang dihadapi sistem manajemen dealer adalah pengujian perangkat lunak yang manual yang kurang efektif serta efisien, dan memakan waktu cukup dalam proses pengujian.

Oleh karena itu dibutuhkan untuk membuat aplikasi yang dapat melakukan login dan pembuatan data secara otomatis dengan basis data yang terintegrasi.



Gambar : Use Case Diagram

3.2. Desain Sistem

1. Use Case Diagram

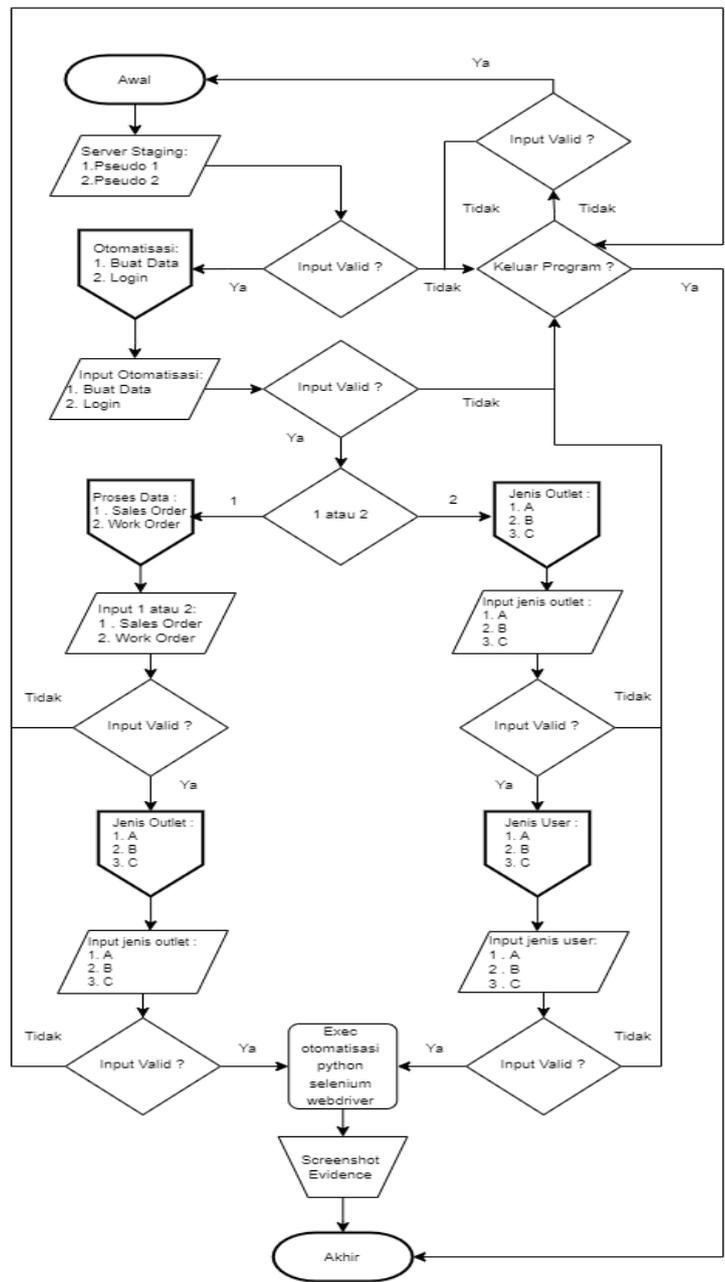
Proses sistem manajemen proyek yang sedang berjalan di Otomatisasi Otentikasi pengguna menggunakan *use case diagram* sebagai berikut :

Berdasarkan pada gambar *use case diagram* diatas dapat disimpulkan bahwa terdapat 1 aktor dan 4 (empat) *use case* dalam proses otomatisasi otentikasi pada sistem manajemen dealer.

2. Flowchart

Proses sistem akademik yang sedang berjalan di dealer manajemen sistem menggunakan *Flowchart* sebagai berikut :

Berdasarkan pada gambar flowchart diatas dapat disimpulkan bahwa terdapat 1 (satu) Terminator start objek yang mengawali, terdapat 7 (tujuh) Input Output, terdapat 9 (sembilan) Decision yang menggambarkan suatu keputusan atau tindakan yang harus diambil pada kondisi tertentu, terdapat 5 (lima) connector off page yang menjadi konektor ke halaman lain dan terdapat 1 (satu) Terminator Stop untuk mengakhiri.





Gambar : Flowchart

3.3 Implementasi

Berikut adalah hasil skema implementasi dan perancangan yang telah dibuat :



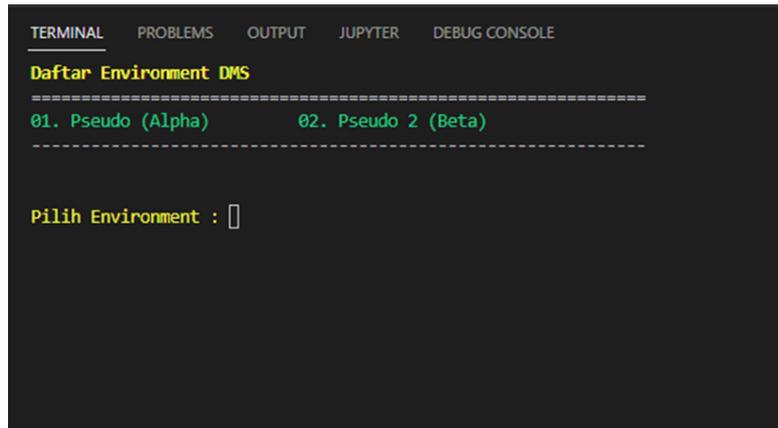
```

Windows PowerShell
├─ Pilih Server
│   └─ Pseudo 1 (Alpha)
│       └─ Pilih Jenis Otomatisasi
│           ├── 1. Buat Data
│           │   └─ Tipe Outlet DMS
│           │       └─ Jenis Flow Data
│           │           ├── 1. Sales Order
│           │           └─ 2. Work Order
│           └─ 2. Login
│               └─ Tipe Outlet DMS
│                   ├── 1. Office Head (HO)
│                   │   └─ Daftar Outlet Office Head (HO)
│                   │       └─ Daftar Role Office Head (HO)
│                   ├── 2. General Repair (GR)
│                   │   └─ Daftar Outlet General Repair (GR)
│                   │       └─ Daftar Role General Repair (GR)
│                   └─ 3. Body Paint (BP)
│                       └─ Daftar Outlet Jenis Body Paint (BP)
│                           └─ Daftar Role Body Paint (BP)
└─ Pseudo 2 (Beta)
    └─ Pilih Jenis Otomatisasi
        ├── 1. Buat Data
        │   └─ Tipe Outlet DMS
        │       └─ Jenis Flow Data
        │           ├── 1. Sales Order
        │           └─ 2. Work Order
        └─ 2. Login
            └─ Tipe Outlet DMS
                ├── 1. Office Head (HO)
                │   └─ Daftar Outlet Office Head (HO)
                │       └─ Daftar Role Office Head (HO)
                ├── 2. General Repair (GR)
                │   └─ Daftar Outlet General Repair (GR)
                │       └─ Daftar Role General Repair (GR)
                └─ 3. Body Paint (BP)
                    └─ Daftar Outlet Jenis Body Paint (BP)
                        └─ Daftar Role Body Paint (BP)
PS C:\Users\Cuzerblade\Downloads\Bot-DMS>
    
```

Gambar : skema menu program otomatisasi otentikasi untuk otorisasi

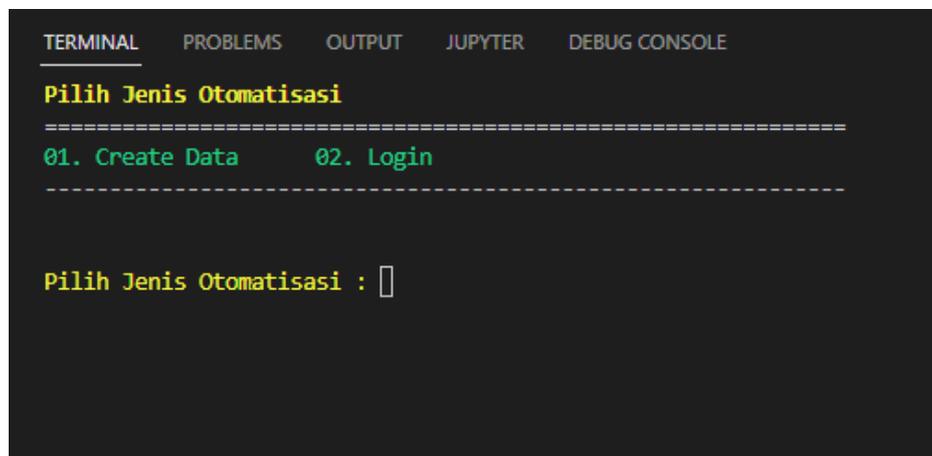
1. Halaman Pilih Server





Gambar : Daftar Environment Server

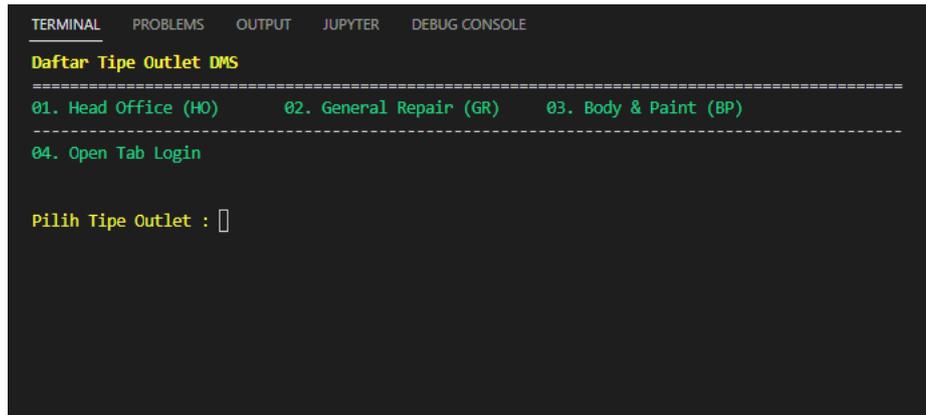
2. Halaman Pilih Jenis Otomatisasi



Gambar : Daftar Jenis Otomatisasi

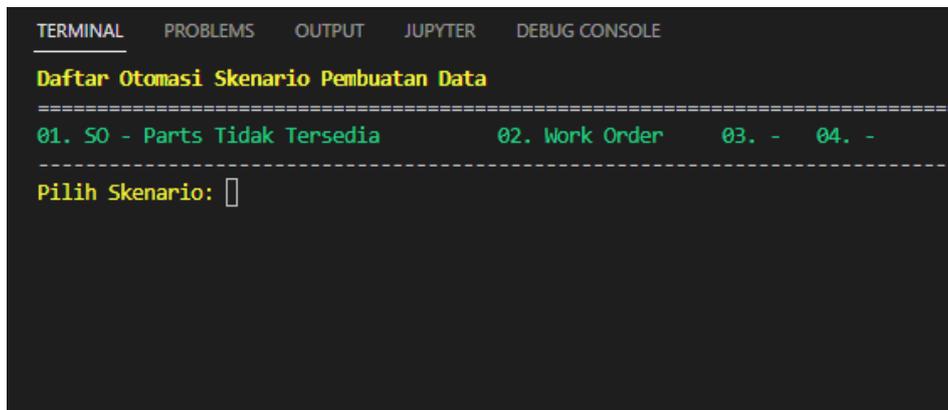
3. Halaman Pilih Jenis Outlet





Gambar : Daftar Jenis Otomatisasi Login Outlet

4. Halaman Otomastisasi Flow Data



Gambar : Daftar Jenis Otomatisasi Flow Data

IV. KESIMPULAN

Berdasarkan hasil penelitian yang peneliti lakukan pada program otomatisasi otektikasi untuk otorisasi, untuk website terutama pada website sistem manajemen dealer dapat membantu tim penjaminan mutu perangkat lunak untuk dapat lebih efisien dan lebih efektif dalam menangani kasus kerusakan dan kesalahan pada saat pengembangan sistem.

REFERENSI

- [1] Unadira and D. L. Badira, "Apa yang dimaksud dengan Lean Software Development?," *dictio.id*, 2018. <https://www.dictio.id/t/apa-yang-dimaksud-dengan-lean-software-development/15011/2> (accessed Jul. 05, 2022).
- [2] I. Wahyudi, S. Bahri, and P. Handayani, "Aplikasi Pembelajaran Pengenalan Budaya Indonesia," vol. V, no. 1, pp. 135–138, 2019, doi: 10.31294/jtk.v4i2.
- [3] I. K. Wairooy, "Software Testing," *socs.binus.ac.id*, 2020. <https://socs.binus.ac.id/2020/06/30/software-testing/> (accessed May 09, 2022).
- [4] M. N. A. Syahrin, "Pentingnya Menambahkan Otorisasi dan Otentikasi Dalam Pengembangan Sistem



This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).
<http://journal.stmikjayakarta.ac.id/index.php/JMIJayakarta>

Aplikasi,” *Kompasiana*, 2022.

https://www.kompasiana.com/nuralfisyahrin/61db941c1b796c25b414e532/pentingnya-menambahkan-otorisasi-dan-otentikasi-dalam-perancangan-sistem-aplikasi?page=2&page_images=1 (accessed Jun. 10, 2022).

- [5] D. Haryono, Y. Zulianda, Wirta, and Lusiana, “Sistem Pendeteksian Serangan Jaringan Local Area Network (Lan) Menggunakan Algoritma Naive Bayes,” *JOISIE (Journal ...)*, vol. 5, no. 1, pp. 1–8, 2021.
- [6] M. Sitinjak Daniel Dido Jantce TJ and J. Suwita, “Analisa Dan Perancangan Sistem Informasi Administrasi Kursus Bahasa Inggris Pada Intensive English Course Di Ciledug Tangerang,” *Ipsikom*, vol. 8, no. 1, pp. 1–19, 2020.
- [7] V. S. Ginting, K. Kusriani, and E. Taufiq, “Implementasi Algoritma C4.5 untuk Memprediksi Keterlambatan Pembayaran Sumbangan Pembangunan Pendidikan Sekolah Menggunakan Python,” *Inspir. J. Teknol. Inf. dan Komun.*, vol. 10, no. 1, pp. 36–44, 2020, doi: 10.35585/inspir.v10i1.2535.
- [8] R. Patil and R. Temkar, “Intelligent Testing Tool : Selenium Web Driver,” *Int. Res. J. Eng. Technol. R., Temkar, R. (2017). Intell. Test. Tool Selenium Web Driver. Int. Res. J. Eng. Technol. 4(6), 1920–1923.* <https://irjet.net/arch>, vol. 4, no. 6, pp. 1920–1923, 2017, [Online]. Available: <https://irjet.net/archives/V4/i6/IRJET-V4I6599.pdf>

