

SISTEM WEB POINT OF SALES (POS) BERBASIS UNIFIED MODELLING LANGUAGE (UML) DENGAN BLACKBOX TESTING UNTUK BARBERSHOP NAGATA

Wisnu Sukma Maulana¹, Julia Fajaryanti², Aldrick Subhan Zul Hanafi³

¹ Fakultas Teknologi Industri, wisnu_maulana@staff.gunadarma.ac.id, Universitas Gunadarma

² Fakultas Teknologi Industri, julia@staff.gunadarma.ac.id, Universitas Gunadarma

³ Fakultas Teknologi Industri, aldrick@student.gunadarma.ac.id Universitas Gunadarma

ABSTRACT

Nagata Barbershop has quite a lot of transactions carried out every day, and there are also plans to expand the business by adding branches. Problems arise during this time when recording transactions manually using paper, there is a mismatch with the data that should be used and it is difficult to manage transaction reports. Barbershop Nagata needs an application that can increase the accuracy of recording transactions, make it easier to manage financial reports and can also be used long term if you want to add branches with the same system. The aim of this research is the results of implementing the solutions used to overcome this problem. Choosing a web-based application is a supports for applications that can be accessed any time and can be implemented on different platforms. The Point of Sales (POS) application at the Nagata Barbershop has been tested using black box testing method.

Keyword: application, website, point of sales, blackbox testing

ABSTRAK

Barbershop Nagata memiliki banyak transaksi setiap hari, dan adanya rencana untuk menambah cabang. Permasalahan yang timbul selama ini ketika pencatatan transaksi manual dengan kertas mengalami ketidakcocokan dengan data seharusnya dan sulitnya mengelola laporan transaksi. Sehingga, Barbershop Nagata butuh aplikasi yang dapat meningkatkan akurasi pencatatan transaksi, kemudahan pengelolaan laporan keuangan dan juga dapat digunakan jangka panjang jika ingin melakukan penambahan cabang dengan sistem yang sama. Tujuan penelitian ini adalah hasil implementasi solusi yang digunakan untuk mengatasi masalah tersebut. Pemilihan aplikasi berbasis web merupakan factor yang mendukung aplikasi dapat diakses kapan saja dan dapat diimplementasikan pada platform yang berbeda. Aplikasi Point of Sales (POS) pada Barbershop Nagata telah diuji coba dengan metode blackbox testing.

Kata Kunci: aplikasi, website, point of sales, blackbox testing.

1. PENDAHULUAN

Barbershop Nagata yang berada di Jl. Raya Sukahati 101 Barbershop Salon Nagata RT 02 / RW03 Cibinong, Kabupaten Bogor sampai hari ini masih menggunakan pencatatan manual pencatatan transaksi dengan menulis di buku kas.

Barbershop Nagata memiliki cukup banyak transaksi yang dilakukan setiap hari, juga adanya rencana untuk melakukan ekspansi usah dengan menambah cabang. Permasalahan yang timbul selama ini ketika pencatatan transaksi manual dengan kertas mengalami ketidakcocokan dengan data seharusnya dan sulitnya mengelola laporan transaksi. Sehingga, Barbershop Nagata menginginkan suatu aplikasi yang dapat meningkatkan akurasi pencatatan transaksi, kemudahan pengeolaan laporan keuangan dan juga dapat digunakan jangka panjang jika ingin melakukan penambahan cabang dengan sistem yang sama.

Di rancang sebuah aplikasi berbasis sebagai alat pembukuan layanan jasa, diperlukan agar dapat menghasilkan perhitungan rekapan yang akurat dan akuntabel. Dengan memanfaatkan koneksi internet sebagai salah satu media perhitungan point of sales, perusahaan dapat menghitung lebih akurat dan akuntabel.

TELAAH PUSTAKA

Aplikasi Kasir dan Keuangan

Aplikasi adalah penggunaan dalam suatu komputer, intruksi atau pernyataan yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi *Output* dan Kasir merupakan bentuk pemberian layanan yang diberikan oleh produsen baik terhadap pengguna barang diproduksi maupun jasa yang ditawarkan. Menurut (Anwar Muthohari, 2016) Hal yang paling penting dalam suatu usaha adalah kualitas pelayanan yang diberikan konsumen akan merasa puas jika pelayanan yang diberikan sangat baik.

Unified Modelling Language (UML)

Unified Modeling Language (UML) adalah tujuan umum, perkembangan, bahasa pemodelan di bidang rekayasa perangkat lunak, yang dimaksudkan untuk menyediakan cara standar untuk memvisualisasikan desain sistem. UML awalnya termotivasi oleh keinginan untuk membakukan sistem notasi yang berbeda dan pendekatan untuk desain perangkat lunak yang dikembangkan oleh Grady Booch, Ivar Jacobson dan James Rumbaugh di *Rational Software* di 1994-1995, dengan pengembangan lebih lanjut yang dipimpin oleh mereka melalui tahun 1996.

Menurut (Nuris, 2022) Pada tahun 1997 UML diadopsi sebagai standar oleh *Object Management Group*, dan telah dikelola oleh organisasi ini sejak. Pada tahun 2005 UML juga diterbitkan oleh *International Organization for Standardization* sebagai standar ISO disetujui. Sejak itu telah periodik direvisi untuk menutupi revisi terbaru dari UML. Beberapa *literature* menyebutkan bahwa UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misalnya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain :

1. *Use Case Diagram*

Use Case Diagram adalah salah satu jenis diagram dalam UML (*Unified Modeling Language*) yang digunakan untuk menggambarkan fungsionalitas sistem dari perspektif pengguna eksternal atau aktor. Diagram ini menunjukkan hubungan antara aktor dan *use case*, yang mewakili fungsionalitas yang diharapkan dari sistem. Aktor adalah entitas yang berinteraksi dengan sistem, seperti pengguna atau sistem eksternal lainnya. Komponen Utama dalam *Use Case Diagram*, adalah : Aktor (Entitas yang berinteraksi dengan sistem dari luar, bisa berupa pengguna manusia atau sistem lain), Use Case (Representasi dari fungsi atau layanan yang disediakan oleh sistem untuk actor), Sistem (Batasan yang menggambarkan sistem yang sedang dimodelkan), dan Asosiasi (Garis yang menghubungkan aktor dengan use case, menunjukkan interaksi antara mereka).

2. *Activity Diagram*

Activity Diagram adalah salah satu jenis diagram dalam *Unified Modeling Language (UML)* yang digunakan untuk memodelkan aliran kerja atau aktivitas dalam suatu sistem. Diagram ini menggambarkan langkah-langkah atau tindakan yang terjadi dalam sebuah proses, serta bagaimana aliran kontrol bergerak dari satu aktivitas ke aktivitas lainnya. *Activity Diagram* sangat berguna dalam memvisualisasikan alur logika bisnis, proses internal, dan alur kerja pengguna. *Activity Diagram* adalah representasi grafis dari aliran aktivitas dalam suatu proses atau sistem. Diagram ini menunjukkan urutan aktivitas, keputusan, perulangan, dan titik awal serta akhir dari proses. *Activity Diagram* membantu dalam memahami dan mendokumentasikan proses dinamis dan perilaku sistem secara keseluruhan.

3. *Class Diagram*

Class Diagram adalah representasi grafis dari kelas-kelas dalam sistem beserta hubungan di antaranya. Kelas adalah template atau *blueprint* untuk objek, yang merupakan instance dari kelas. Kelas memiliki atribut (data) dan metode (fungsi atau operasi) yang menggambarkan perilaku dari kelas tersebut. Hubungan antar kelas dapat berupa asosiasi, agregasi, komposisi, dan pewarisan (*inheritance*).

Framework Bootstrap

Bootstrap merupakan CSS yang populer dan digunakan secara luas untuk mengembangkan halaman web responsif dan *mobile-first*. Bootstrap menyediakan berbagai komponen siap pakai dan gaya standar yang dapat digunakan untuk mempercepat proses pengembangan web. Bootstrap adalah kerangka kerja *front-end* yang memudahkan dalam membuat tampilan web yang konsisten dan responsive juga bersifat *open-source*, awalnya dikembangkan oleh Mark Otto dan Jacob Thornton di Twitter. Bootstrap menyediakan kumpulan alat dan komponen berbasis HTML, CSS, dan JavaScript yang dirancang untuk mempermudah pembuatan desain web yang konsisten dan responsive [1]. *Framework* ini pertama kali

dirilis pada tahun 2011 dan sejak itu telah menjadi salah satu alat yang paling populer dalam pengembangan *web front-end*.

Black box Testing

Black box testing merupakan pengujian yang dilakukan untuk mengamati hasil input dan output dari perangkat lunak tanpa mengetahui struktur kode dari perangkat lunak. Pengujian ini dilakukan di akhir pembuatan perangkat lunak untuk mengetahui apakah perangkat lunak dapat berfungsi dengan baik. Pengujian perangkat lunak adalah sebuah elemen sebuah topik yang memiliki cakupan luas dan sering dikaitkan dengan verifikasi (*verification*) dan validasi (*validasi*) (V&V)” [2].

METODE PENELITIAN

Pengembangan proyek penelitian untuk migrasi sistem ke arsitektur *microservice* menggunakan pendekatan model SDLC. SDLC sendiri merupakan proses yang diikuti untuk proyek perangkat lunak, dalam organisasi perangkat lunak. Ini terdiri dari rencana terperinci yang menjelaskan bagaimana mengembangkan, memelihara, mengganti dan mengubah atau meningkatkan perangkat lunak tertentu. Siklus hidup mendefinisikan metodologi untuk meningkatkan kualitas perangkat lunak dan proses pengembangan secara keseluruhan [7].



Gambar 1 Tahapan-tahapan SDLC
Sumber: Satzinger, Jackson & Burd (2012)

1. Tahap Perencanaan
 - Pada tahap ini merancang *website* dilakukan dengan mengidentifikasi kebutuhan sistem, menentukan tujuan dan ruang lingkup proyek, serta melakukan studi kelayakan untuk memastikan bahwa proyek dapat dilaksanakan dengan sumber daya yang ada.
2. Tahap Analisis
 - Tahap kedua adalah analisis, dengan melakukan wawancara dengan Yulida, SE. untuk mendapatkan data harga setiap pelayanan pada barbershop dan informasi yang dibutuhkan untuk sistem, sehingga dapat dilakukan analisis kebutuhan pengguna dan sistem untuk memastikan bahwa semua kebutuhan bisnis dan teknis tercakup.
3. Tahap Perancangan
 - Tahap perancangan merupakan tahap dimana rancangan dari seluruh tahap sebelumnya diimplementasikan. Desain aplikasi menggunakan *UML*, *Class Diagram*, *Use-Case Diagram*, *Activity Diagram*.
4. Tahap Implementasi
 - Tahapan keempat adalah implementasi, yang menggunakan tools coding seperti PHP, XAMPP, dan alat-alat pendukung lainnya untuk mengembangkan sistem. Pada tahap ini, kode program ditulis dan diintegrasikan sesuai dengan desain yang telah dibuat.
5. Tahap Ujicoba
 - Tahap kelima adalah uji coba, dimana penelitian ini menggunakan metode *black box testing* untuk memeriksa fungsionalitas sistem tanpa melihat ke dalam struktur kode. Pengujian dilakukan untuk memastikan bahwa sistem berfungsi sesuai dengan spesifikasi dan kebutuhan pengguna. Tahap terakhir adalah pemeliharaan, yang melibatkan pemeliharaan sistem setelah implementasi, termasuk perbaikan *bug*,

pembaruan sistem, dan peningkatan fungsionalitas sesuai dengan kebutuhan pengguna yang berkembang.

HASIL DAN PEMBAHASAN

Hasil dari penelitian dengan metode SDLC dan perancangan menggunakan UML didokumentasi ke dalam beberapa tahapan sesuai dengan tahapan-tahapan yang mengikuti standar.

Tahap Perencanaan

Pada tahapan ini dilakukan dengan mengidentifikasi kebutuhan sistem, menentukan tujuan dan ruang lingkup proyek, serta melakukan studi kelayakan untuk memastikan bahwa proyek dapat dilaksanakan dengan sumber daya yang ada. Kebutuhan yang diperlukan, *Website* dapat menampilkan informasi sebuah layanan kasir yang dapat mempermudah untuk melakukan pencatatan transaksi menjadi sebuah digital dan juga sebuah halaman kelola untuk melakukan perubahan layanan, melihat transaksi, dan menambahkan sebuah *user*. Pada awal masuk *website* ini akan diarahkan terlebih dahulu kepada halaman *login* dengan halaman berisi kolom input *username*, *password* dan juga tombol untuk masuk ke *website*, setelah melakukan *login* halaman yang akan tampil adalah halaman kasir, disini *user* dapat mengisi nama pelanggan, memilih layanan, mengisi uang customer dan juga tombol kirim untuk mengarahkan ke halaman struk, pada halaman pertama terdapat juga *navigasi* untuk merubah sebuah tema warna dan juga beberapa bentuk dari sebuah *website*, navigasi selanjutnya adalah tombol kelola. Pada halaman kelola berisi tentang pendapatan mingguan, jumlah customer mingguan, mengubah menambahkan menghapus layanan, mengubah menambahkan menghapus *user*, dan juga melihat daftar transaksi yang sudah diisi melalui halaman kasir. Terdapat tombol juga untuk *export* hasil data transaksi ke Microsoft Excel.

Tahap Analisis

1. Analisis Kebutuhan Fungsional

Website Point of Sales (POS) untuk Barbershop Salon Nagata dirancang untuk mempermudah kasir dalam melakukan pencatatan transaksi secara digital dan *owner* dalam melakukan pemantauan laporan keuangan. Pengguna yang mengunjungi website ini harus dapat mengaksesnya melalui perangkat seperti *desktop* dengan koneksi internet

2. Analisis Kebutuhan Non-Fungsional

Kebutuhan perangkat lunak yang digunakan dalam pembuatan website ini adalah sebagai berikut :

1. Sistem Operasi : Windows 10
2. Aplikasi : Microsoft Visual Studio Code, XAMPP, Browser Edge

Kebutuhan perangkat keras yang digunakan dalam pembuatan website ini adalah sebagai berikut :

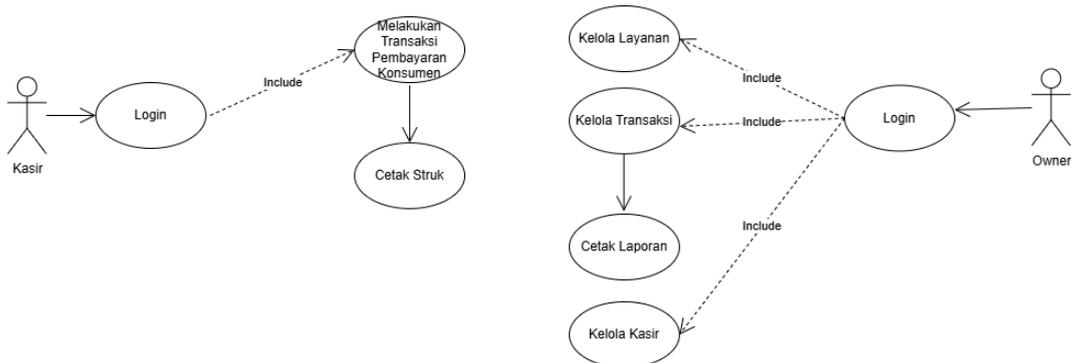
1. Processor : Intel I5 Generasi 8 atau terbaru
2. RAM : 8 GB atau lebih
3. Penyimpanan : 256 GB atau lebih
4. Sistem : 64 bit

Tahap Perancangan

Pada Tahapan ini, dirancang menggunakan UML dengan membuat *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*.

1. Use Case Diagram

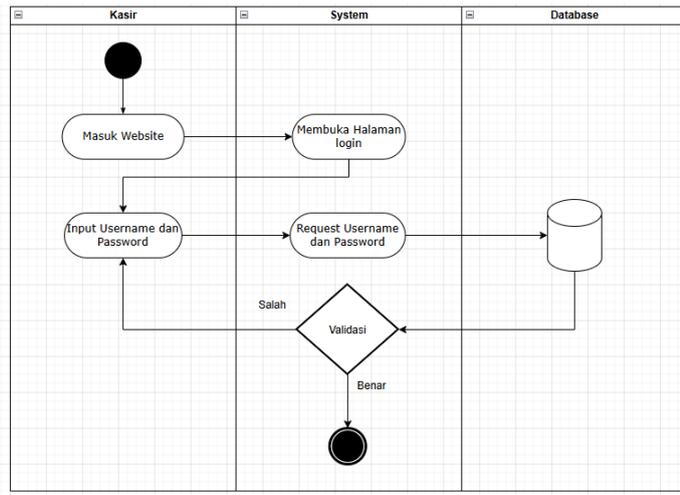
Use Case diagram adalah diagram yang menggambarkan kebutuhan sistem dari sudut pandang *user*, yang memperlihatkan hubungan-hubungan yang terjadi antara *actors* dengan *use case* dalam sistem. Pertama kasir akan membuka sebuah halaman *website* dan mengisi *form* sesuai pelanggan yang datang, yaitu berupa nama, jenis layanan, uang *customer*. Setelah kasir mengisikan maka seorang kasir dapat melihat detail layanan pada struk. Selanjutnya pemilik dapat mengakses halaman kasir maupun halaman kelola, dengan ini pemilik dapat melihat daftar transaksi dalam bentuk tabel yang telah diisi oleh sebuah kasir berupa nama pelanggan, nama layanan, jumlah pemasukan. Terlampir pada Gambar 2 berupa stuktur *use case* diagram



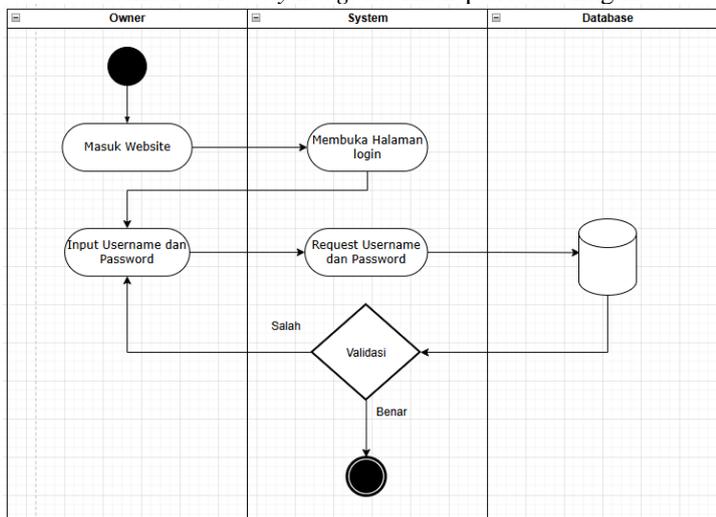
Gambar 2 Use Case Diagram Aplikasi

2. Activity Diagram

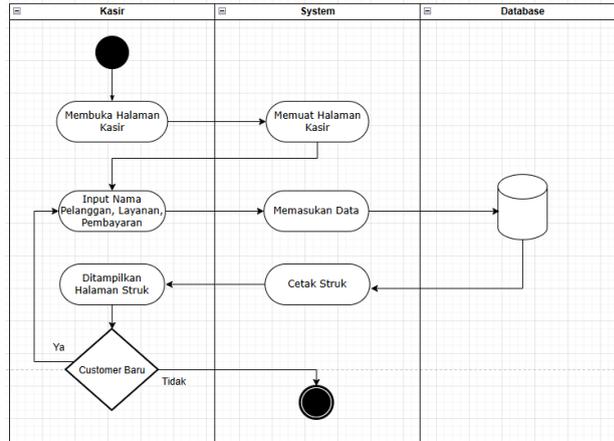
Selanjutnya pada *activity* diagram menjelaskan proses seorang kasir dan pemilik pada ini, pertama kasir dan *owner* mengakses dan masuk ke halaman *login* jika akun kasir belum dibuat, kasir wajib mendaftar terlebih dahulu melalui pemilik untuk diberikan akses masuk. Setelah kasir mendapatkan sebuah akses masuk maka akan diarahkan ke halaman utama kasir.



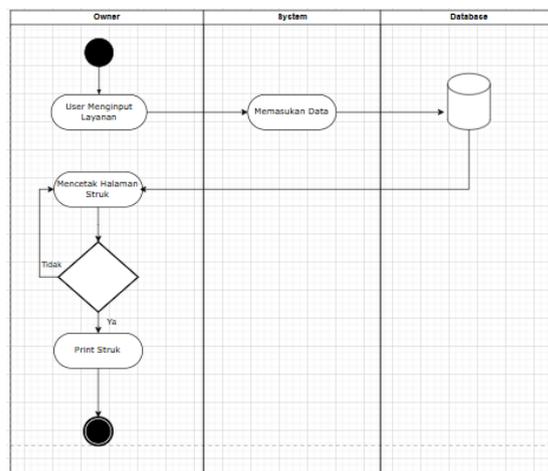
Gambar 3 Activity Diagram Kasir pada saat Log in



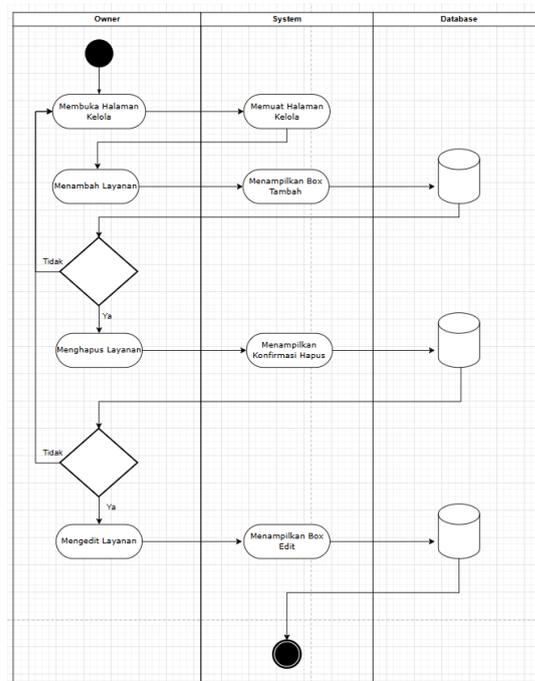
Gambar 4 Activity Diagram Owner pada saat Log in



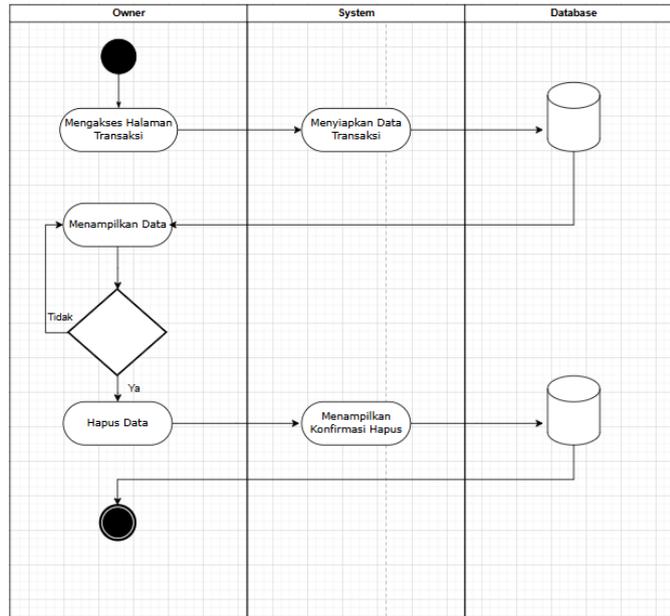
Gambar 5 Activity Diagram Owner Kasir Melakukan Transaksi Pembayaran Konsumen



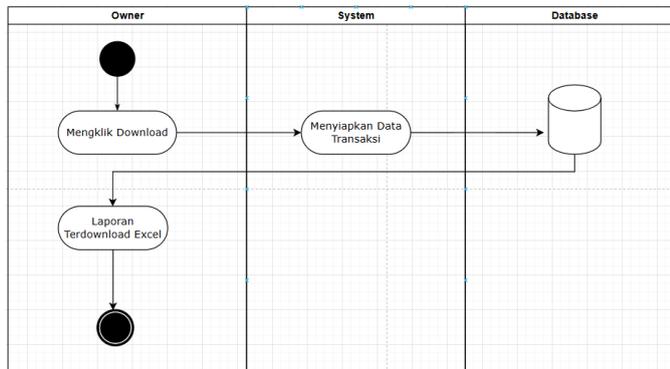
Gambar 6 Activity Diagram Kasir saat Cetak Struk



Gambar 7 Activity Diagram saat Kelola Layanan



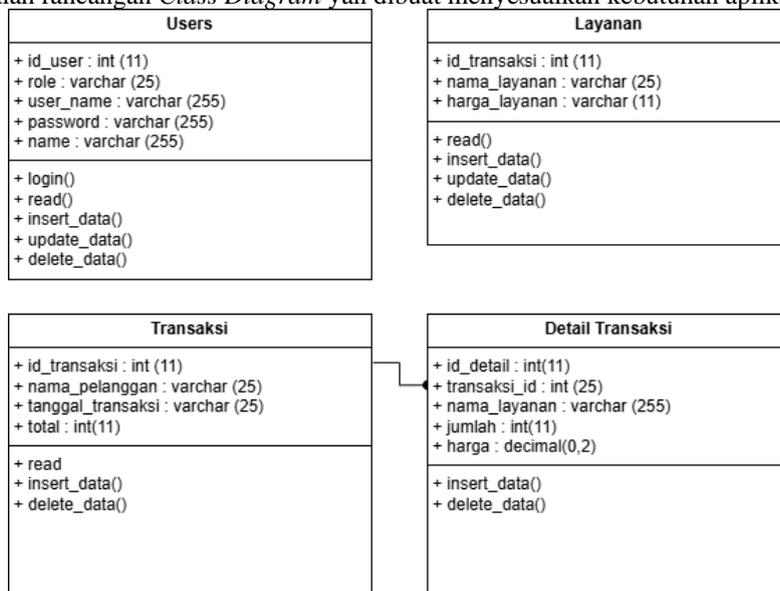
Gambar 8 Activity Diagram saat Kelola Transaksi



Gambar 9 Activity Diagram saat Cetak Laporan

3. Class Diagram

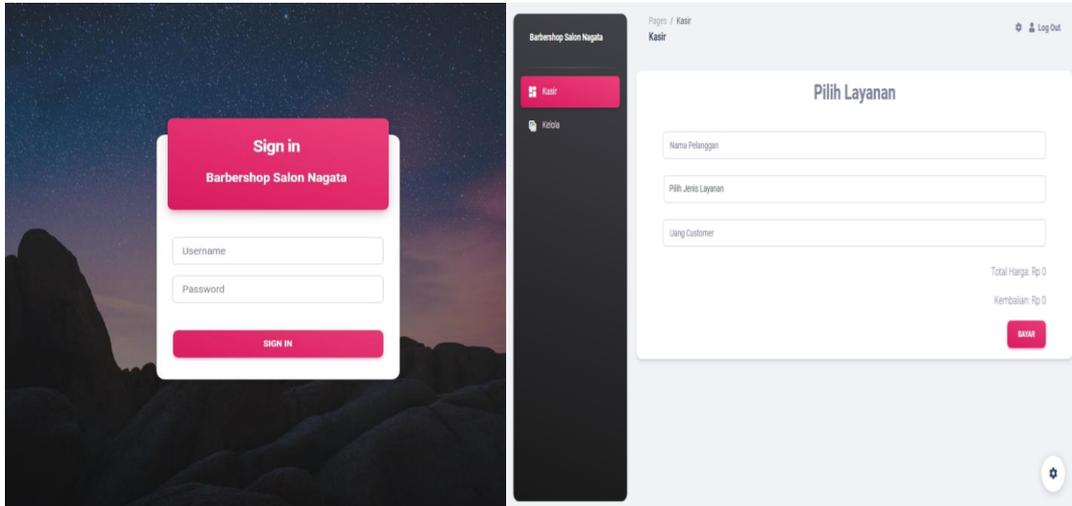
Berikut adalah rancangan *Class Diagram* yan dibuat menyesuaikan kebutuhan aplikasi.



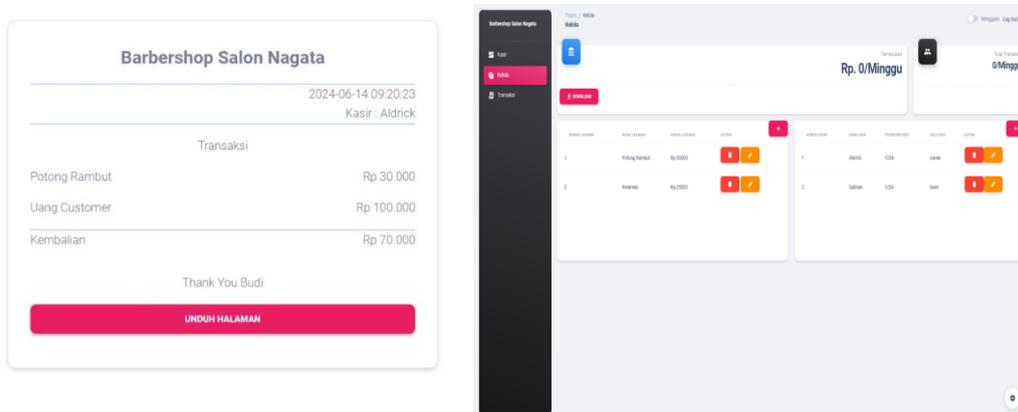
Gambar 10 Class Diagram

Tahap Implementasi

Pada tahapan implementasi, pembuatan website menggunakan *framework* bootstrap menggunakan *tools editor* Visual Studio Code.



Gambar 11 Tampilan Halaman Log in dan Halaman Dashboard



Gambar 12 Tampilan Halaman Kelola dan Transaksi

Tahap Ujicoba

Pada tahapan ujicoba, digunakan metode *black box testing*. Metode ini memastikan Website yang telah dibuat apakah sesuai dengan keinginan pemakai dan dapat digunakan oleh pemakai dan dioperasikan oleh operator sistem. Beberapa tingkatan pengujian sebelum mengimplementasikan sistem aplikasi yaitu pengujian bagian, dilakukan ke setiap program untuk memperhitungkan bahwa setiap program memenuhi spesifikasinya.

Tabel 1 Hasil Pengujian Aplikasi menggunakan *Black box Testing*

| No | Halaman | Skenario Pengujian | Hasil Yang Diharapkan | Hasil Actual |
|----|---------------|---|---|----------------|
| 1 | Halaman Login | Memasukan username dan password benar | Berhasil masuk kedalam halaman kasir | Berhasil Login |
| 2 | Halaman Login | Memasukan username salah dan password benar | Menampilkan Incorrect User name or password | Gagal Login |

| | | | | |
|----|-----------------|--|--|--|
| 3 | Halaman Login | Memasukan username benar dan password salah | Menampilkan Incorrect User name or password | Gagal Login |
| 4 | Halaman Login | Memasukan username yang salah dan password salah | Menampilkan Incorrect User name or password | Gagal Login |
| 5 | Halaman Login | Tidak mengisi username mengisi password | Menampilkan Username is required | Gagal Login |
| 6 | Halaman Login | Mengisi username tidak mengisi password | Menampilkan Password is required | Gagal Login |
| 7 | Halaman Login | Tidak mengisi sama sekali username dan password | Menampilkan Username Password is required | Gagal Login |
| 8 | Halaman Beranda | Username yang terdaftar sebagai role kasir | Mengeluarkan Output Halaman Kasir untuk menginput nama customer, nama layanan, uang customer dan juga perhitungan uang customer | Halaman tampil dengan baik button pada tempatnya, dan dapat diisi dengan baik |
| 9 | Halaman Beranda | Username yang terdaftar sebagai role owner | Mengeluarkan Output Halaman Kasir untuk menginput nama customer, nama layanan, uang customer dan juga perhitungan uang customer | Halaman tampil dengan baik button pada tempatnya, dan dapat diisi dengan baik |
| 10 | Halaman Struk | Username yang terdaftar sebagai role kasir | Mengeluarkan Output hasil dari input yang diisikan dihalaman kasir dengan baik | Halaman struk tampil dengan baik, file struk dapat di unduh dengan baik menjadi file xls |
| 11 | Halaman Struk | Username yang terdaftar sebagai role owner | Mengeluarkan Output hasil dari input yang diisikan dihalaman kasir dengan baik | Halaman struk tampil dengan baik, file struk dapat di unduh dengan baik menjadi file xls |
| 12 | Halaman Kelola | Username yang terdaftar sebagai role kasir | Halaman tidak dapat diakses oleh kasir | Kasir tidak dapat mengakses halaman kelola |
| 13 | Halaman Kelola | Username yang terdaftar sebagai role owner | Halaman dapat diakses oleh pemilik menghasilkan output perhitungan uang mingguan, jumlah customer mingguan, layanan, users, data transaksi | Owner dapat mengakses halaman dan dapat menambahkan mengubah menghapus dengan baik |

| | | | | |
|----|-------------------|--|---|--|
| 14 | Halaman Transaksi | Username yang terdaftar sebagai role kasir | Halaman tidak dapat diakses oleh kasir | Kasir tidak dapat mengakses halaman transaksi |
| 15 | Halaman Transaksi | Username yang terdaftar sebagai role owner | Halaman dapat diakses oleh pemilik, <i>history</i> kasir dapat dilihat dengan jelas dari data transaksi yang telah dilakukan oleh kasir | Owner dapat mengakses halaman transaksi, penempatan button tepat pada rancangan, owner juga dapat melihat dan menghapus transaksi. |

KESIMPULAN DAN SARAN

Berdasarkan hasil rancangan pada pembuatan Website *Point Of Sales* Barbershop Nagata Menggunakan PHP Dan MYSQL, sudah berhasil dibuat dan diakses pada alamat : <https://posnagata.fym.my.id/>. Aplikasi juga telah melalui tahapan ujicoba setelah dilakukan *hosting* pada *browser desktop* dan juga perangkat gawai. Dengan adanya website, Barbershop Salon Nagata telah mengoptimalkan dalam melayani transaksi konsumen dan juga pencatatan transaksi secara keseluruhan karena informasi mengenai transaksi telah tersimpan dan tercatat. Juga pemilik usaha dapat memantau pemasukan Barbershop melihat laporan keuangan secara *realtime*.

Sebagai saran dari penelitian ini, peneliti berharap adanya pengembangan website yang memudahkan user dalam berbagi informasi dan memperoleh informasi, serta owner dapat mengetahui tentang keuntungan bisnis yang diperoleh dan melakukan penambahan pada *database* pelanggan untuk melakukan pencatatan nomor telepon, *email* dan SMS sebagai bentuk promosi ataupun ucapan dihari spesial pelanggan.

DAFTAR PUSTAKA

- [1] Abdulloh, R. (2022). 7 Materi Pemrograman Web untuk Pemula 4: Bootstrap & MariaDB. Penerbit PT Elex Media Komutindo. Jakarta.
- [2] Anwar Muthohari, Bunyamin, & Sri Rahayu. (2016). *PENGEMBANGAN APLIKASI KASIR PADA SISTEM INFORMASI RUMAH MAKAN PADANG ARIUNG*. Sekolah Tinggi Teknologi Garut.
- [3] Ariani Sukanto, R., & Shalahuddin, M. 2016. Rekayasa Perangkat Lunak. Bandung. Informatika.
- [4] Dora, Sujit Kumar, and Dubey, Pushkar. 2013. "Software Development Life Cycle (Sdlc) Analytical Comparison And Survey On Traditional And Agile Methodology". *National Monthly Refereed Journal Of Research In Science & Technology*. Vol No.2, Issue No.8 ISSN 2277-1174. Abhinav Journal Publisher. Odisha.
- [5] Gani, L., & Achmad, A. (2019). Website dan HTML. Website Dan HTML, 58.
- [6] Hartono, T. (2021). *Hosting: Pengertian, Fungsi, dan Jenis Layanan yang Wajib Diketahui*. Jakarta Nuris Avrilia, & Muryan Awaludin. (2022). *SISTEM INFORMASI KEARSIPAN BERBASIS WEB PADA FAKULTAS EKONOMI DAN BISNIS UNIKA ATMA JAYA JAKARTA*.
- [7] Nuris Avrilia, & Muryan Awaludin. (2022). *SISTEM INFORMASI KEARSIPAN BERBASIS WEB PADA FAKULTAS EKONOMI DAN BISNIS UNIKA ATMA JAYA JAKARTA*.
- [8] Purwanto. 7 Februari 2017. *Metodologi System Development Life Cycle (SDLC)*. <https://medium.com/@purwanto.dev/metodologi-system-development-life-cycle-sdlc-2f0349df1364>.
- [9] Rohman, F. (2022). *Internet Adalah Jaringan Komputer, Ini Pengertian Dan Sejarahnya*. Jakarta
- [10] Yeni Susilowati. (2019). Modul E-Commerce-Teaching Factory For Students.pdf (H. Purnomo (ed.)).