

PERANCANGAN APLIKASI POINT OF SALES BERBASIS WEB PADA TOKO NIAGA

Cindy Angelline ¹⁾ Tony ²⁾ Manatap Dolok Lauro ³⁾

^{1,2)} Program Studi Sistem Informasi, Fakultas Teknologi Informasi Universitas Tarumanagara
Jl. Letjen S. Parman No. 1, 11440, Indonesia

³⁾ Program Studi Teknik Informatika, Fakultas Teknologi Informasi Universitas Tarumanagara
Jl. Letjen S. Parman No. 1, 11440, Indonesia

email : cindy.825220019@stu.untar.ac.id¹⁾ tony@fti.untar.ac.id²⁾ manataps@fti.untar.ac.id³⁾

ABSTRAK

Usaha dagang memiliki berbagai jenis kegiatan, mulai dari grosir hingga eceran, yang membutuhkan sistem pengelolaan transaksi dan pencatatan stok yang efisien. Pada era digital saat ini, metode pencatatan manual dinilai kurang efektif karena berisiko menimbulkan kesalahan, kehilangan data, dan keterlambatan dalam pembuatan laporan keuangan. Penelitian ini bertujuan merancang aplikasi Point of Sales (POS) berbasis web untuk membantu usaha dagang dalam mengelola transaksi, data produk, serta laporan penjualan secara terintegrasi. Metode pengembangan sistem menggunakan Agile Scrum, yang memungkinkan pembangunan dilakukan secara fleksibel melalui tahapan analisis kebutuhan, perancangan sistem, implementasi, hingga pengujian. Perancangan sistem meliputi use case diagram, class diagram, sequence diagram, serta model basis data dengan notasi Crow's Foot. Implementasi aplikasi menggunakan bahasa pemrograman HTML dan PHP, basis data MySQL, serta framework Bootstrap untuk tampilan antarmuka.

Kata Kunci

Point of Sales, Aplikasi Web, Agile Scrum, Manajemen Transaksi, Toko Grosir

1. Pendahuluan

Usaha dagang memiliki berbagai jenis kegiatan, mulai dari grosir hingga eceran, yang memerlukan sistem pengelolaan transaksi, pencatatan penjualan, serta pengaturan stok barang yang efisien. Seiring meningkatnya aktivitas transaksi, metode pencatatan manual sering kali menimbulkan kendala berupa kesalahan pencatatan harga, duplikasi data, kehilangan informasi, serta keterlambatan dalam penyusunan laporan keuangan [1]. Hal ini menyebabkan proses operasional menjadi kurang optimal dan berpotensi menimbulkan kerugian.

Dalam era digital saat ini, teknologi informasi menjadi solusi utama untuk meningkatkan efisiensi operasional,

termasuk pada dunia bisnis dan usaha [2]. Aplikasi *Point of Sales* (POS) berbasis web menawarkan kemampuan pengelolaan transaksi penjualan, pencatatan data produk, serta pembuatan laporan secara otomatis, cepat, dan akurat. Penelitian terdahulu menunjukkan bahwa penggunaan sistem POS dapat mempercepat proses transaksi sekaligus mengurangi resiko kesalahan *input* data [2].

Penelitian ini bertujuan untuk merancang dan mengimplementasikan aplikasi POS berbasis web dengan metode *Agile Scrum*. Perancangan dilakukan melalui identifikasi kebutuhan fungsional dan non-fungsional, pemodelan sistem menggunakan diagram UML, serta perancangan basis data dengan notasi *Crow's Foot*. Implementasi dilakukan dengan bahasa pemrograman HTML, CSS, PHP, basis data MySQL, serta *framework Bootstrap* untuk tampilan antarmuka.

Kontribusi utama dari penelitian ini adalah menghasilkan aplikasi POS berbasis web yang mampu meningkatkan efisiensi operasional, meminimalisir kesalahan pencatatan, serta mendukung digitalisasi usaha dagang agar lebih adaptif terhadap perkembangan teknologi.

2. Dasar Teoritik dan Metodologi

2.1 Dasar Teori

2.1.1 Website

Website merupakan kumpulan halaman web dalam satu domain yang saling terhubung dan menyajikan berbagai informasi. Halaman pertama yang muncul ketika domain diakses disebut *homepage*, sedangkan keseluruhan isi domain disebut *website* [3]. Menurut Widia, et al. [3], *website* dapat dibedakan berdasarkan beberapa kategori. Dari jenisnya, *website* terbagi menjadi *website* dinamis yang kontennya selalu diperbarui seperti portal berita dan *website* statis yang kontennya jarang berubah seperti profil organisasi. Berdasarkan tujuan, terdapat personal web yang berisi informasi individu,

corporate web untuk perusahaan, portal web yang menyediakan berbagai layanan, forum web sebagai media diskusi, serta jenis lain seperti e-Government, e-Banking, dan e-Payment. Sementara itu, berdasarkan bahasa pemrograman, *website* dibagi menjadi *server side* yang bergantung pada server seperti PHP dan ASP, serta *server client side* yang cukup dijalankan melalui browser dengan bahasa HTML.

2.1.2 Grosir

Perdagangan grosir adalah sistem penjualan produk dalam jumlah besar kepada pembeli, seperti distributor atau toko, dengan harga yang lebih rendah dibandingkan harga eceran [4]. Usaha grosir termasuk dalam kategori *Business to Business* (B2B) karena transaksi dilakukan antar pelaku usaha, bukan langsung kepada konsumen akhir. Produk yang dibeli grosir biasanya dijual kembali oleh pengecer kepada konsumen akhir (*Business to Consumer* atau B2C) dengan harga lebih tinggi untuk memperoleh keuntungan [5]. Menurut Adyas, et al. [4][4], perdagangan grosir dapat dibedakan menjadi beberapa jenis, antara lain grosir tradisional yang menyalurkan produk dari produsen ke pembeli grosir, grosir e-commerce yang memanfaatkan internet sebagai platform penjualan, grosir tingkat tinggi yang melibatkan penjualan dalam jumlah besar pada industri sejenis, grosir modal di mana pembeli menahan barang hingga terjual, grosir *drop shipping* yang mengirimkan produk langsung dari produsen kepada konsumen akhir, serta grosir *online* yang memanfaatkan situs web untuk menjual produk dalam jumlah besar.

2.1.3 Point of Sales

Point of Sales (POS) adalah sistem yang digunakan untuk melakukan transaksi di toko dengan memanfaatkan perangkat seperti mesin, tablet, atau komputer [6]. POS berfungsi sebagai titik pembayaran ketika pelanggan membeli barang, sekaligus mendukung kelancaran proses transaksi agar lebih efisien dan menguntungkan pemilik bisnis. Sistem ini banyak diterapkan pada berbagai jenis usaha, seperti toko ritel, restoran, hingga minimarket, karena mampu mempercepat proses jual beli serta menjaga keamanan transaksi [7]. Proses transaksi dianggap selesai saat pelanggan melakukan pembayaran di kasir setelah sistem menghitung total belanja. Selain menghemat waktu, POS juga menyimpan setiap catatan penjualan secara digital sehingga mempermudah pemantauan laporan penjualan [7].

2.2 Metodologi

2.2.1 Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC) merupakan kerangka kerja yang mencakup seluruh tahapan pengembangan perangkat lunak, mulai dari

pengumpulan kebutuhan, analisis, implementasi, pengujian, hingga pemeliharaan [8]. Sepanjang proses SDLC, dilakukan berbagai pengukuran untuk mengevaluasi kualitas dan efisiensi pengembangan agar proyek dapat mencapai keberhasilan. Tahapan umum dalam SDLC meliputi perencanaan kebutuhan, analisis sistem, desain, implementasi, pengujian, penerapan, dan evaluasi.

SDLC sendiri memiliki beberapa metodologi yang dapat digunakan, antara lain *Waterfall*, *Spiral*, *Incremental*, *Rational Unified Process (RUP)*, *Rapid Application Development (RAD)*, *Agile*, dan *Rapid Prototyping* [9]. Metode *Waterfall* memiliki karakteristik tahapan yang berurutan, sehingga tahapan berikutnya tidak dapat dimulai sebelum tahapan sebelumnya selesai. Kelemahan metode ini terletak pada kesulitannya menghadapi perubahan kebutuhan; jika terjadi perubahan *requirement*, maka pengembangan harus dimulai dari awal sehingga dapat menghambat keseluruhan proses.

Sebaliknya, metode *Agile* menawarkan pendekatan yang lebih fleksibel dengan membagi siklus pengembangan menjadi sprint, yaitu iterasi pengembangan yang memungkinkan perubahan dilakukan bahkan setelah tahap perencanaan selesai. Hal ini membuat *Agile* lebih sesuai untuk pengembangan perangkat lunak dengan kebutuhan yang dinamis. Untuk mendukung prinsip *Agile*, salah satu kerangka kerja yang banyak digunakan adalah *Scrum*, yang berfokus pada sprint dengan durasi maksimal 30 hari. Setiap sprint mencakup beberapa aktivitas, seperti *Sprint Planning*, *Daily Scrum*, *Sprint Review*, dan *Sprint Retrospective* [9].

2.2.2 Agile dengan Scrum Menggunakan Goal

Agile merupakan pendekatan pengembangan perangkat lunak yang bersifat iteratif dan evolusioner karena berfokus pada kolaborasi tim serta penggunaan dokumen formal yang terbatas namun tepat guna. Tujuan utama metode ini adalah menghasilkan perangkat lunak yang berkualitas dengan biaya yang efektif dan waktu yang sesuai, meskipun kebutuhan klien dapat berubah sewaktu-waktu. *Agile* sangat relevan dalam konteks pengembangan perangkat lunak modern karena mampu menyesuaikan perubahan kebutuhan secara cepat. Salah satu prinsipnya adalah penyampaian hasil dalam jangka waktu singkat, biasanya antara dua hingga empat minggu.

Salah satu kerangka kerja yang banyak digunakan dalam implementasi *Agile* adalah *Scrum*. *Scrum* dapat dipahami sebagai penerapan prinsip *Agile* ke dalam proses yang lebih terstruktur dengan serangkaian aktivitas yang mendukung kolaborasi tim. Hubungan antara *Agile* dan *Scrum* dapat dianalogikan sebagai sifat dan tindakan, dimana *Agile* memberikan aturan adaptif sedangkan *Scrum* menyediakan praktik nyata untuk menjalankannya. Dalam penelitian ini, tahapan *Scrum* diterapkan pada proses perancangan aplikasi *Point of Sales* (POS) berbasis web.

Tahap pertama adalah *Sprint Planning*, yaitu analisis kebutuhan pengguna yang diselesaikan dalam periode tertentu sesuai kompleksitas. Pada perancangan aplikasi POS, proses ini dilakukan melalui observasi dan wawancara dengan pemilik toko sehingga menghasilkan daftar kebutuhan fitur. Selanjutnya kebutuhan tersebut diorganisasi dalam *product backlog* dan *sprint goal* menggunakan aplikasi manajemen proyek ClickUp, dengan status pekerjaan yang dibagi menjadi *to-do*, *in progress*, *ready to test*, dan *tested*.

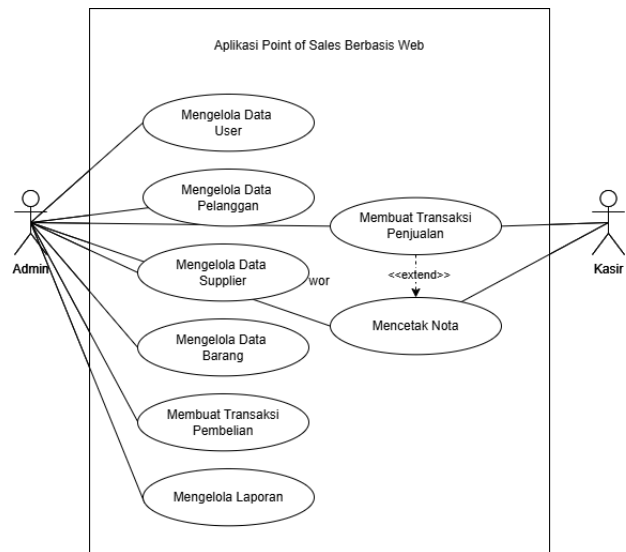
Tahap berikutnya adalah *Daily Scrum*, yakni pertemuan singkat selama 15 menit yang dilakukan setiap hari untuk membahas progres, hambatan, serta rencana kerja dalam 24 jam ke depan. Tahap ini berfungsi meningkatkan koordinasi tim serta mempercepat penyelesaian kendala yang muncul. Setelah itu, dilakukan *Sprint Review* di akhir setiap sprint, biasanya seminggu sekali, untuk mengevaluasi *backlog* item yang sudah selesai maupun yang tertunda, sekaligus menilai efektivitas kinerja tim.

Tahap terakhir adalah *Sprint Retrospective*, yaitu evaluasi yang dilakukan setelah *Sprint Review*. Fokusnya adalah menemukan kendala dan kekurangan yang terjadi pada sprint sebelumnya serta merumuskan strategi perbaikan pada sprint berikutnya. Pada perancangan aplikasi POS berbasis web ini, *retrospektif* dilakukan setiap akhir bulan dengan tujuan meningkatkan efektivitas kerja tim dan meminimalisir kendala yang sudah teridentifikasi.

3. Hasil dan Pembahasan

3.1 Use Case Diagram

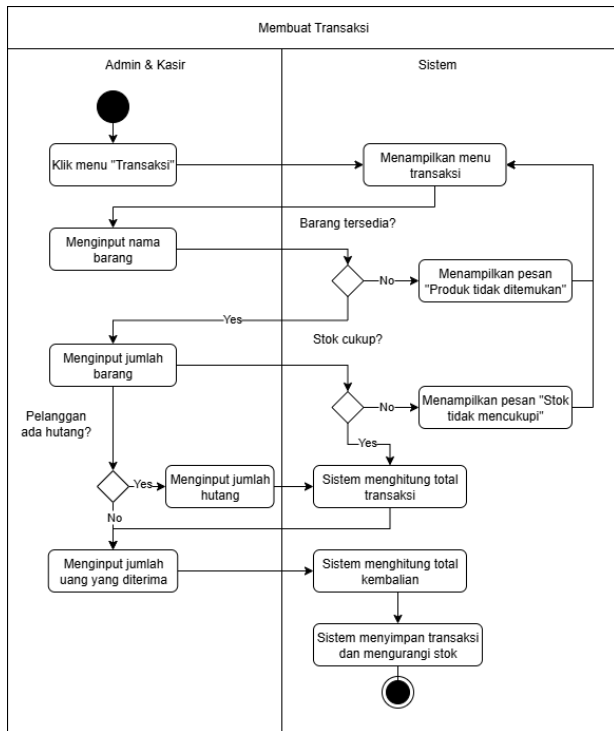
Use case merupakan sarana yang digunakan untuk menggambarkan persyaratan sebuah sistem serta bagaimana sistem tersebut seharusnya digunakan. Dalam penyusunannya, *use case* biasanya terdiri dari komponen aktor, *use case*, dan sistem [10]. *Use case diagram* sendiri adalah diagram yang menggambarkan hubungan antara aktor dan *use case* yang berfungsi sebagai alat bantu dalam menganalisis serta merancang sistem yang sedang dikembangkan [11]. Aktor dalam aplikasi POS berbasis web ada dua yaitu admin dan kasir, untuk *user* admin bisa mengakses semua fitur sedangkan *user* kasir hanya bisa membuat transaksi penjualan dan mencetak nota.



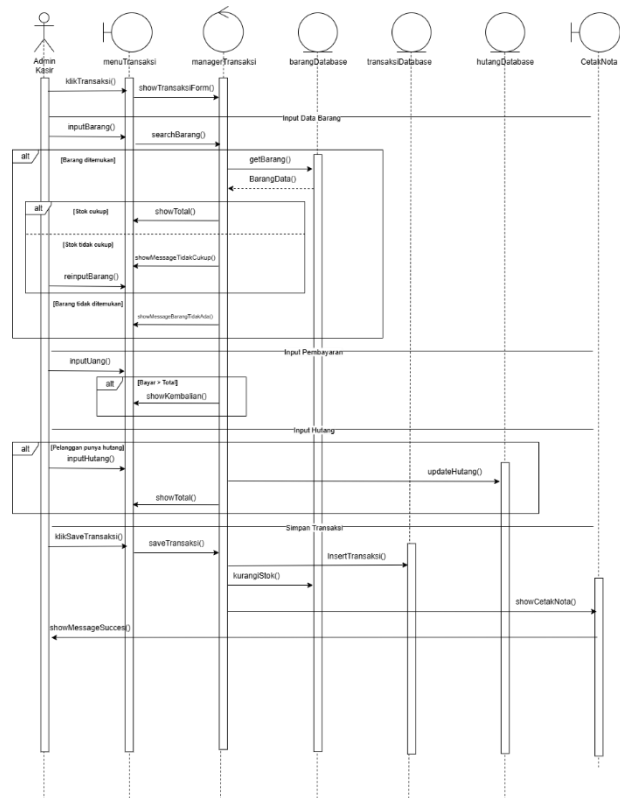
Gambar 1 Use Case Diagram

3.2 Activity Diagram

Activity diagram adalah diagram atau bentuk visual yang digunakan untuk menggambarkan aliran data maupun aliran kontrol dalam suatu sistem, serta menunjukkan aksi-aksi yang terstruktur dan dirancang untuk menggambarkan perilaku dinamis sistem [10]. Diagram ini berperan penting dalam memperlihatkan bagaimana suatu proses dimulai, dijalankan, hingga berakhir. Komponen utama dalam activity diagram terdiri dari beberapa elemen, yaitu Activity Node, yang merepresentasikan notasi dari proses yang bekerja dalam kontrol dan nilai data; Activity Edge, berupa garis yang menghubungkan aliran aksi secara langsung dari input menuju output; Initial State, digambarkan dengan lingkaran penuh yang menunjukkan awal proses; serta Decision, berbentuk ketupat dengan satu alur masuk dan dua atau lebih alur keluar yang mencerminkan kondisi berbeda [11]. Selain itu, terdapat pula *Fork* atau percabangan yang divisualisasikan sebagai garis hitam dengan satu alur masuk dan lebih dari satu alur keluar, *Join* atau penggabungan berupa garis hitam dengan beberapa alur masuk dan satu alur keluar yang mengharuskan semua aksi selesai sebelum dilanjutkan, serta Final State, berbentuk lingkaran penuh di dalam lingkaran kosong yang menandakan akhir dari suatu proses.



Gambar 2 Activity Diagram Transaksi



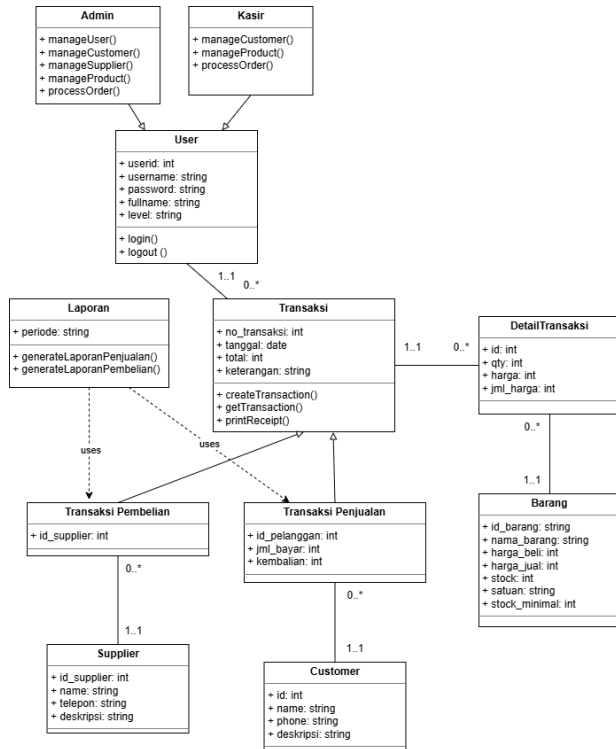
Gambar 3 Sequence Diagram Transaksi

3.3 Sequence Diagram

Sequence diagram merupakan diagram yang digunakan untuk menggambarkan proses yang dilakukan oleh pengguna serta kolaborasi antar objek yang saling berinteraksi dalam sistem [10]. Diagram ini menitikberatkan pada urutan pesan yang dipertukarkan di antara elemen-elemen sistem sehingga dapat memperlihatkan alur komunikasi secara runtut. Beberapa komponen utama yang terdapat dalam *sequence diagram* antara lain *Activations*, yang menunjukkan eksekusi fungsi dari suatu objek; *Actor*, yang merepresentasikan peran pengguna atau entitas eksternal yang melakukan serangkaian aksi; *Collaboration Boundary*, yaitu elemen yang menggambarkan lingkungan percobaan dan berfungsi memantau objek; serta *Parallel Vertical Lines*, berupa garis vertikal yang menandakan jalannya suatu proses atau *state* [11]. Selain itu, terdapat pula *Processes*, yang menggambarkan tindakan atau aksi yang dilakukan aktor dalam suatu waktu; *Window*, yang memperlihatkan halaman yang ditampilkan dalam suatu proses; serta *Loop*, yang merepresentasikan model logika yang memungkinkan terjadinya pengulangan beberapa kali dalam suatu interaksi. *Sequence diagram* ini menampilkan hubungan alur aktor dengan menu transaksi, manajer transaksi, *database* barang, *database* transaksi, *database* hutang, dan percetakan nota.

3.4 Class Diagram

Class diagram merupakan representasi visual dari struktur suatu sistem yang menggambarkan alur dan keterkaitan dalam *database* [10]. Diagram ini bersifat statis karena berfokus pada pemodelan struktur sistem, bukan pada perilaku dinamis atau interaksi secara langsung. Dengan demikian, *class diagram* lebih menekankan pada definisi hubungan dan atribut antar kelas dibandingkan alur proses. Dalam penggunaannya, *class diagram* memiliki beberapa jenis relasi yang umum digunakan, di antaranya adalah *Association*, yaitu hubungan yang menunjukkan adanya interaksi antar kelas dan biasanya digambarkan dengan garis berpanah terbuka yang menandakan aliran pesan satu arah; *Generalization*, yaitu hubungan yang bersifat hierarkis dari kelas yang lebih khusus menuju kelas yang lebih umum; serta *Constraint*, yaitu hubungan yang berfungsi membatasi atau memberikan aturan tertentu dalam sistem sehingga dapat mendefinisikan aspek non fungsional [11].

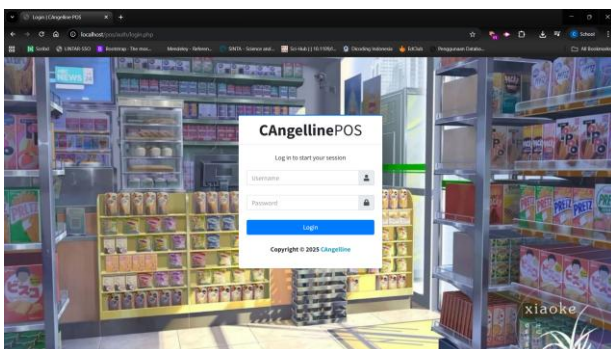


Gambar 4 Class Diagram

3.5 Tampilan Halaman Aplikasi

3.5.1 Halaman Login

Halaman ini merupakan pertama kali *user* akan diarahkan sebelum masuk kedalam aplikasi. Untuk *user* kasir *username* dan *password* diperoleh dari admin, untuk *user* admin *username* dan *password* ditentukan pemilik aplikasi dan admin lain.

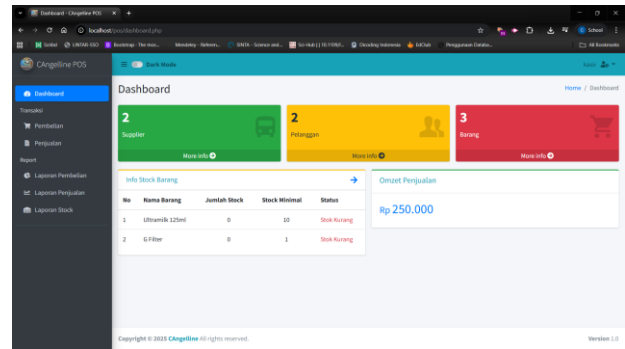


Gambar 5 Tampilan Halaman Login

3.5.2 Halaman Dashboard

Halaman *dashboard* merupakan halaman pertama pada aplikasi yang menampilkan jumlah *supplier*, jumlah

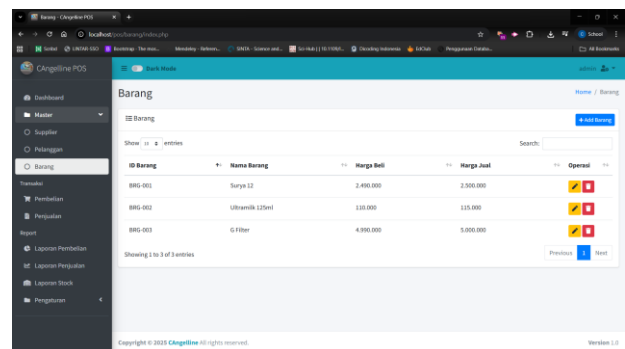
pelanggan, dan jumlah barang. Ditampilkan juga daftar stok barang yang kosong serta omzet penjualan.



Gambar 6 Tampilan Halaman Dashboard

3.5.3 Halaman Master Barang

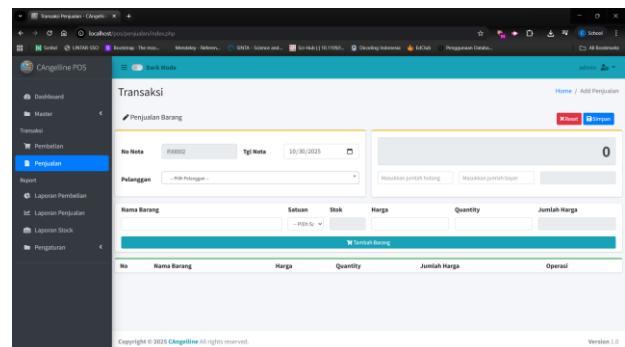
Pada halaman ini menampilkan nama barang, harga beli, harga jual, serta operasi untuk mengubah dan menghapus barang. Untuk operasi CRUD pada barang hanya dapat dioperasikan *user* admin.



Gambar 7 Tampilan Halaman Master Barang

3.5.4 Halaman Transaksi Penjualan

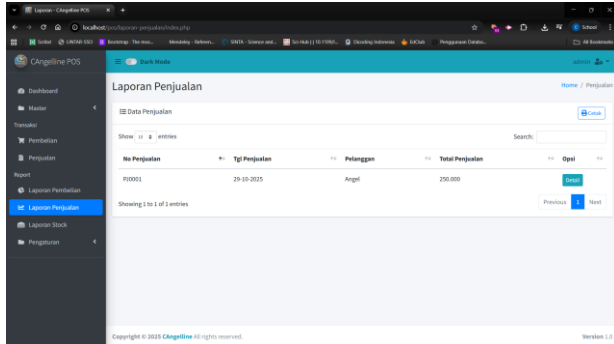
Pada transaksi penjualan *user* dapat mengisi nama pelanggan, nama barang, satuan barang, serta jumlah barang yang dibeli. Untuk transaksi penjualan terdapat juga kolom hutang dan juga bayar.



Gambar 8 Tampilan Halaman Transaksi Penjualan

3.5.5 Halaman Laporan Penjualan

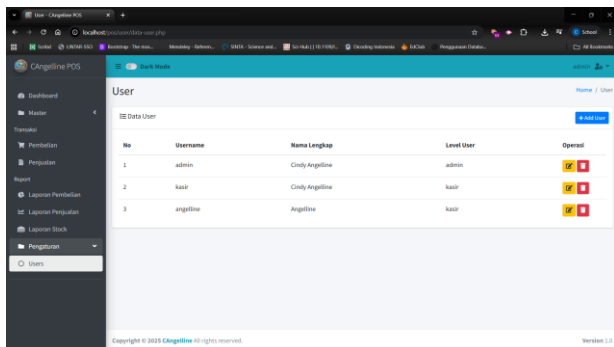
Laporan halaman berisi detail penjualan transaksi penjualan yang telah berhasil dan untuk menyimpan laporan dapat di tekan tombol *print* lalu memilih periode waktu yang diinginkan.



Gambar 9 Tampilan Halaman Laporan Penjualan

3.5.6 Halaman User

Halaman *user* berisi data-data *user* yang tersimpan dalam *database* dan hanya bisa dikelola oleh *user* admin. Tampilan *sidebar user* kasir tidak memiliki fitur pengaturan *user*.



Gambar 10 Tampilan Halaman User

3.6 Pengujian

Pengujian *User Acceptance Testing* (UAT) dilakukan untuk memastikan bahwa aplikasi POS berbasis web yang telah dikembangkan dapat berjalan sesuai dengan kebutuhan pengguna di Toko Niaga Jaya. Pengujian ini melibatkan *user* kasir dan admin.

Tabel 1 Hasil UAT Aplikasi POS Berbasis Web

No	Fitur Uji	Sangat Tidak setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1	Login (Kasir)	0%	0%	0%	0%	100%
2	Login (Admin)	0%	0%	0%	0%	100%
3	Mengelola Data User	0%	0%	0%	0%	100%
4	Mengelola Data Pelanggan	0%	0%	0%	0%	100%

5	Mengelola Data Supplier	0%	0%	0%	0%	100%
6	Mengelola Data Barang	0%	0%	0%	100%	0%
7	Membuat Transaksi Pembelian	0%	0%	0%	0%	100%
8	Membuat Transaksi Penjualan (Admin)	0%	0%	0%	100%	0%
9	Membuat Transaksi Penjualan (Kasir)	0%	0%	0%	100%	0%
10	Mencetak Nota (Admin)	0%	0%	0%	0%	100%
11	Mencetak Nota (Kasir)	0%	0%	0%	0%	100%
12	Mengelola Laporan	0%	0%	0%	0%	100%
13	Logout	0%	0%	0%	0%	100%

Berdasarkan hasil UAT dapat disimpulkan seluruh fitur utama berjalan sesuai dengan rancangan dan dapat diterima oleh *user*. Didapatkan juga umpan balik positif terkait kemudahan penggunaan dan tampilan antarmuka aplikasi. Namun, ditemukan saran untuk memperbesar kolom serta tulisan seperti jumlah bayar, dan total.

4. Kesimpulan

Berdasarkan rancangan dan metode yang digunakan dalam perancangan aplikasi POS maka didapatkan kesimpulan sebagai berikut:

1. Perancangan aplikasi POS berbasis web dirancang menggunakan metode *agile scrum* dengan tahapan analisis kebutuhan, perancangan sistem, implementasi, hingga pengujian.
2. Aplikasi POS berbasis web dibuat menggunakan bahasa pemrograman PHP, HTML, CSS, basis data MySQL, serta framework Bootstrap untuk antarmuka.
3. Aplikasi POS berfungsi untuk membuat dan mencetak transaksi, mencatat stok barang, hingga membuat laporan.
4. Hasil pengujian aplikasi POS menggunakan User Acceptance Testing (UAT) mendapatkan skor tanggapan “Setuju” dan “Sangat Setuju”, hasil ini menunjukkan fitur yang dirancang telah memenuhi kebutuhan user secara umum.
5. Aplikasi POS pada Toko Niaga Jaya telah diperbaiki berdasarkan masukan dari user sehingga saat ini aplikasi cukup menjawab permasalahan membuat nota secara manual, mencatat stok barang, hingga membuat laporan berdasarkan periode.

REFERENSI

- [1] H. A. Fernando Wigen, “PERANCANGAN SISTEM INFORMASI MANAJEMEN KEUANGAN

- BERBASIS WEB STUDI KASUS DI PT.SURYA WIGEN,” *J. Teknol. Digit. dan Sist. Inf.*, vol. 1, 2024, [Online]. Available: <https://ojsiibn1.indobarunasional.ac.id/index.php/JUT-EKDISI>
- [2] Z. Rashifah and E. S. Budi, “Rancangan Sistem Informasi Pada Kasir Berbasis Web,” *J. Sist. Komput. dan Inform.*, vol. 3, no. 4, p. 529, 2022, doi: 10.30865/json.v3i4.4241.
- [3] D. M. Widia and S. R. Asriningtias, *Cara Cepat dan Praktis Membangun Web Dinamis dengan PHP dan MySQL*. Universitas Brawijaya Press, 2021. [Online]. Available: <https://books.google.co.id/books?id=GnpYEAAQBAJ>
- [4] S. E. M. M. Mujito, S. E. M. M. C. Dr. Hari Muharam, M. B. A. Drs. Dasmanasyah Adyas., S. P. Anneu Fitriyanti, and I. Fathurrahman, *MANAJEMEN PEMASARAN: Sebuah Pengantar Untuk Pemula*. EDU PUBLISHER, 2023. [Online]. Available: <https://books.google.co.id/books?id=rGPBEEAAQBAJ>
- [5] F. Syahla Nur and Suparjimin, “Analisis Manajemen Risiko pada UMKM Grosir dan Eceran Toko KANNA.ID,” *PENG J. Ekon. Dan Manaj.*, vol. 2, no. 1, pp. 1681–1693, 2025, doi: <https://doi.org/10.62710/3aasj227>.
- [6] S. K. M. K. D. E. M. S. R. S. K. M. K. F. I. S. T. M. K. Sigit Susanto Putro, *BUKU AJAR PENGANTAR TEKNOLOGI INFORMASI DAN KOMPUTER*. PENERBIT KBM INDONESIA, 2024. [Online]. Available: <https://books.google.co.id/books?id=B3k2EQAAQBAJ>
- [7] Stefani Yohana Paula Bere, Ni Made Estiyanti, and Nengah Widya Utami, “Analisis Dan Perancangan Sistem Point Of Sales (POS) Pada Toko Harco Bali,” *Smart Techno (Smart Technol. Informatics Technopreneurship)*, vol. 5, no. 1, pp. 49–58, 2023, doi: 10.59356/smart-techno.v5i1.79.
- [8] C. Science, A. Kruglov, and G. Succi, *Developing Sustainable and Energy-Efficient Software Systems*. 2023.
- [9] R. Silhavy, *Intelligent Algorithms in Software Engineering: Proceedings of the 9th Computer Science On-line Conference 2020, Volume 1*. in Advances in Intelligent Systems and Computing. Springer International Publishing, 2020. [Online]. Available: <https://books.google.co.id/books?id=tEz2DwAAQBAJ>
- [10] R. Destriana, S. M. Husain, N. Handayani, and A. T. P. Siswanto, *Diagram UML Dalam Membuat Aplikasi Android Firebase “Studi Kasus Aplikasi Bank Sampah.”* Deepublish, 2022. [Online]. Available: <https://books.google.co.id/books?id=6bMEQAAQBAJ>
- [11] T. Arianti, A. Fa’izi, S. Adam, and M. Wulandari, “Perancangan Sistem Informasi Perpustakaan Menggunakan Diagram Uml (Unified Modelling Language),” *J. Ilm. Komput. Tera[an dan Inf.*, vol. 1, no. 1, pp. 19–25, 2022, [Online]. Available: <https://journal.polita.ac.id/index.php/politati/article/view/110/88>
- [12] R. Ferdiana and U. G. M. Press, *ENGINEERING DESIGN PADA SISTEM INFORMASI*. Gadjah Mada University Press, 2021. [Online]. Available: <https://books.google.co.id/books?id=E-5IEAAQBAJ>
- Cindy Angelline**, saat ini sebagai mahasiswa program studi Sistem Informasi Tarumanagara angkatan 2022.
- Tony**, memperoleh gelar S.Kom. pada tahun 2005 dari Universitas Tarumanagara, M.Kom. pada tahun 2010 dari Universitas Indonesia, dan Ph.D. pada tahun 2021 dari Curtin University. Saat ini sebagai staf pengajar di Fakultas Teknologi Informasi, Universitas Tarumanagara.
- Manatap Dolok Lauro Sitorus**, memperoleh gelar S.Kom. dari Universitas Tarumanagara, Jakarta pada tahun 2006. Pada tahun 2010 memperoleh MMSI dari Universitas Bina Nusantara, Jakarta. Saat ini sebagai Staf Pengajar Program Studi Teknik Informatika Universitas Tarumanagara.