

RANCANG BANGUN APLIKASI KASIR TABUNG GAS MENGGUNAKAN SCAN BARCODE PADA UD MAKMUR BOJONEGORO

Moh.Abdullah Mohtar, Awalludiyah Ambarwati

Sistem Informasi, Universitas Narotama Surabaya
Jalan Arief Rachman Hakim 51, Sukolilo, Surabaya
abdillah.m112@gmail.com

ABSTRAK

Di era digital yang berkembang pesat, penerapan sistem kasir yang efisien menjadi sangat penting bagi berbagai jenis bisnis, mulai dari toko retail, usaha tingkat kecil, hingga menengah. Penelitian ini berfokus pada pengembangan aplikasi kasir berbasis *website* untuk UD MAKMUR, sebuah bisnis yang menghadapi kendala dalam pelabelan dan pencatatan transaksi tabung gas yang masih menggunakan cara manual. Masalah utama yang dihadapi adalah label pada tabung gas yang sering pudar dan kerap sulit terbaca, serta pencatatan transaksi secara manual yang memakan waktu dan rentan terhadap kesalahan. Untuk mengatasi masalah ini, sistem kasir berbasis *barcode* dirancang menggunakan metode *Rapid Application Development* (RAD), yang memungkinkan fleksibilitas dalam menyesuaikan kebutuhan yang berubah selama proses pengembangan. Sistem ini bertujuan untuk meningkatkan efisiensi bisnis dengan mengotomatisasi proses transaksi dan meningkatkan akurasi pelacakan label. Hasil dari sistem aplikasi dalam penelitian ini telah memenuhi tahap validasi pengujian menggunakan metode *black box testing* melalui kolaborasi antara peneliti dan pihak UD MAKMUR. Hasil pengujian menunjukkan bahwa aplikasi telah memenuhi kebutuhan fungsional dan non-fungsional. Penelitian ini diharapkan memberikan kontribusi terhadap diskursus akademik tentang sistem kasir berbasis *website* dan dapat menjadi referensi untuk penelitian selanjutnya.

Kata kunci : Aplikasi kasir, *Rapid Application Development* (RAD), Sistem *barcode*.

1. PENDAHULUAN

Penerapan sistem kasir yang efisien menjadi sangat penting bagi berbagai jenis bisnis, mulai dari toko retail, usaha kecil, hingga menengah dalam rangka meningkatkan efisiensi bisnis dengan cara mengintegrasikannya kedalam sebuah sistem. Pada observasi awal yang dilakukan peneliti pada UD MAKMUR didapati bahwa pelabelan masih menggunakan cat, spidol dan sistem kerik secara manual sehingga dalam distribusinya ketika tabung terkena percikan air hujan maupun udara lembab maka tanda label tersebut kerap menjadi samar baik karena korosi maupun luntur.



Gambar 1. Labeling menggunakan cara manual

Serta didapati pula bahwa UD MAKMUR masih menggunakan metode pencatatan transaksi manual pada kertas, sehingga pada kondisi lapangan saat ini

kerap terjadi pencarian data transaksi pelanggan yang memakan waktu lebih, dikarenakan harus mencari dan menyaring informasi secara manual pada buku transaksi. Kondisi ini menunjukkan adanya potensi untuk meningkatkan proses transaksi bisnis dengan mengadopsi teknologi baru, seperti aplikasi kasir berbasis *scan barcode*.

Meskipun telah banyak aplikasi kasir yang tersedia di pasaran, masih ada beberapa tantangan yang perlu diatasi. Beberapa bisnis berkembang mungkin mengalami kesulitan dalam memilih aplikasi kasir yang sesuai dengan kebutuhan mereka, terutama dikarenakan banyaknya opsi yang tersedia dan penyesuaian kondisi infrastruktur perangkat teknologi dan kondisi bisnis yang telah berjalan juga bervariasi.

Penelitian ini bertujuan untuk meningkatkan efisiensi bisnis dengan mengotomatisasi proses transaksi dan meningkatkan akurasi pelacakan label di UD MAKMUR. Berdasarkan kebutuhan ini, peneliti tertarik untuk membangun aplikasi kasir yang efisien, serta merekomendasikan sistem *scan barcode* dalam proses transaksi. Dalam penelitian ini, metode *Rapid Application Development* (RAD) dipilih karena fleksibilitasnya dalam mengubah kebutuhan dan persyaratan selama siklus pengembangan. Pendekatan iteratif memungkinkan pemangku kebutuhan dan pengembang untuk saling berkomunikasi serta memastikan bahwa perangkat lunak yang dihasilkan memenuhi kebutuhan dan ekspektasi pemangku kebutuhan.

2. TINJAUAN PUSTAKA

4.1. Penelitian terdahulu

Untuk mendukung penelitian ini, peneliti menyertakan beberapa referensi penelitian terdahulu yang relevan. Referensi ini digunakan untuk pertimbangan dan mengembangkan ide serta sistem yang telah ada sebelumnya.

Penelitian [1] berfokus dalam mengembangkan aplikasi kasir berbasis web untuk toko sembako yang sebelumnya menggunakan sistem manual, menyebabkan antrian panjang dan ketidakakuratan data. Sistem baru menggunakan metode *Waterfall* yang terdiri dari tahap analisis kebutuhan, desain, implementasi, pengujian, dan pemeliharaan. Aplikasi ini membantu mempercepat proses transaksi, pencatatan stok, dan pembuatan laporan penjualan.

Penelitian [2] merancang aplikasi kasir berbasis web untuk toko Alomgada Kids yang menjual produk susu dan diapers. Sistem yang dikembangkan membantu dalam pencatatan transaksi, pengelolaan stok, dan pembuatan laporan penjualan, yang sebelumnya dilakukan secara manual. Aplikasi ini dirancang dengan menggunakan model pengembangan *Waterfall*, yang meliputi analisis kebutuhan, desain, implementasi, dan pengujian. Hasil penelitian menunjukkan bahwa sistem yang terkomputerisasi memberikan kemudahan proses transaksi serta pencarian data.

Penelitian [3] mengembangkan sistem informasi kasir berbasis web untuk Toko Raya Computer, yang sebelumnya menggunakan sistem manual yang memperlambat transaksi dan mempengaruhi kinerja karyawan. Implementasi sistem baru ini berhasil mempercepat proses transaksi, meningkatkan akurasi data, dan mempermudah pembuatan laporan yang mendukung pengambilan keputusan strategis. Hasilnya, toko mampu mengatasi keterbatasan sistem manual, meningkatkan efisiensi operasional, dan membuka peluang untuk pertumbuhan bisnis yang berkelanjutan di masa depan. Kesimpulannya, penerapan sistem ini tidak hanya mempercepat transaksi sehari-hari tetapi juga memberikan kemampuan pengelolaan data yang lebih akurat dan terintegrasi.

4.2. My Structure Query Language (MySQL)

MySQL adalah salah satu aplikasi DBMS (*Database Management System*) yang sudah sangat banyak digunakan oleh para pemrogram aplikasi web. MySQL merupakan sebuah sistem *database* relasional, sehingga dapat mengelompokkan informasi ke dalam tabel-tabel atau grup-grup informasi yang berkaitan. Setiap tabel memuat bidang-bidang yang terpisah, yang mempresentasikan setiap bit informasi. MySQL menggunakan indeks untuk mempercepat proses pencarian terhadap baris informasi tertentu[4].

4.3. Hypertext Preprocessor (PHP)

PHP adalah bahasa yang dirancang secara khusus untuk penggunaan pada web. PHP adalah *tool* untuk pembuatan halaman web dinamis. Pada awalnya PHP merupakan kependekan dari *Personal Home Page*. PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (*Form Interpreted*), yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data *form* dari web[4].

4.4. UML (Unified Modeling Language)

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan[5].

4.5. Rapid Application Development (RAD)

Rapid Application development (RAD) suatu proses memajukan ataupun meningkatkan perangkat lunak yang termasuk dalam teknik inkremental (bertingkat) serta menegaskan pada siklus pengembangan sedikit lebih cepat dan tepat untuk tujuan kualitas yang lebih baik. Dengan memanfaatkan *Rapid application development* (RAD) dapat meningkatkan prosedur kerja diawal fase dalam membangun aplikasi untuk menentukan kebutuhan *user*[6].

4.6. CodeIgniter

CodeIgniter merupakan sebuah aplikasi terbuka atau *open source* yang berfondasi dari *framework* PHP dengan model MVC atau juga biasa disebut *Model View Controller* yang digunakan untuk membangun suatu *website* dinamis dengan menggunakan kode PHP. CodeIgniter membantu memudahkan pengembang atau para *developer* untuk bisa membuat suatu aplikasi berbasis *website* dengan cepat dan tepat serta memberikan kemudahan dibandingkan dengan membuat menggunakan *koding* PHP murni atau membuat dari awal pemrograman[7].

4.7. Davidscotttufts PHP Barcode Library

Untuk menghasilkan *barcode* dalam aplikasi kasir ini, peneliti menggunakan library PHP Barcode dari David Scott Tufts yang tersedia di GitHub. *Library* ini memungkinkan pembuatan berbagai jenis *barcode* dengan mudah seperti dukungan terhadap Code 128, Code 39, Code 2of5, dan Codabar. *Library* PHP Barcode dari David Scott Tufts juga bersifat *Open source* dengan perizinan MIT yang diberikan antara lain adalah dalam penggunaan, menyalin,

memodifikasi, menggabungkan, dan mendistribusikan [8].

3. METODE PENELITIAN

Metode pengembangan sistem dalam penelitian ini menggunakan metode *Rapid Application Development* (RAD).



Gambar 2. Metode RAD (*Rapid Application Development*) [9]

Metode RAD dipilih karena pengembangan RAD berfokus pada setiap persyaratan pengembangan pada satu waktu. Keterlibatan pemangku kebutuhan saat mengembangkan produk dapat membantu meningkatkan kepuasan pemangku kebutuhan karena lebih banyak komunikasi yang terjadi saat mengembangkan produk dan pengguna dapat melihat kemajuan produk[10].

Dari definisi konsep RAD, dapat dilihat bahwa pengembangan aplikasi dengan menggunakan metode RAD dapat dilakukan dalam waktu yang relatif lebih cepat. Tahapan RAD terdiri dari 3 tahap yang terstruktur dan saling bergantung disetiap tahap yaitu:

a. Fase *Requirements Planning*

Dalam fase pertama yakni peneliti melakukan *brainstorming* dengan berdiskusi dengan narasumber untuk mengkaji berbagai kebutuhan maupun data yang akan dipakai ataupun diproses kedalam sistem yang akan dikembangkan. Lalu peneliti menganalisis data yang telah disampaikan oleh narasumber ke dalam beberapa bagian, yakni identifikasi permasalahan, kebutuhan fungsionalitas, kebutuhan non fungsionalitas dan proses bisnis usulan.

b. Fase *Workshop Design*

Hasil yang telah dicapai pada tahap awal lalu dibuat setiap modul perancangan, dan diperlihatkan hasilnya kepada pengguna untuk memperoleh umpan balik, dan proses ini berlanjut sampai seluruh modul yang diperlukan selesai dikembangkan oleh peneliti. Pada tahap ini peneliti melakukan perancangan modul menggunakan diagram UML dan *prototype user interface*.

c. Fase Implementasi

Pada tahap ini peneliti melakukan perancangan *user interface*, kode program serta pengujian pada aplikasi yang telah dirancang secara mandiri dan ketika fungsional yang dikembangkan dinyatakan berjalan sesuai sebagaimana mestinya, maka selanjutnya peneliti akan menghubungi pengguna untuk melakukan pengujian secara kolaboratif.

4.1. Proses Pengumpulan Data

Proses pengumpulan data dan informasi diperlukan agar peneliti mendapat data dan informasi yang relevan yang nantinya akan diolah ketahap penelitian lebih lanjut. Pengumpulan data yang dilakukan terbagi menjadi dua bagian, yaitu wawancara dan observasi, kedua proses pengumpulan data termasuk kedalam tahap perencanaan syarat-syarat (*Requirements Planning*) dalam metode RAD.

4.2. Identifikasi permasalahan

Setelah peneliti melakukan observasi dan wawancara terdapat beberapa permasalahan yang ditemukan. Berikut adalah beberapa permasalahan yang ditemukan:

- Pencatatan data transaksi masih menggunakan cara penulisan manual, seperti penulisan data waktu dan tanggal, nama pembeli, keterangan label, serta catatan bila ada kasus khusus. sehingga pada kondisi lapangan saat ini kerap terjadi pencarian data transaksi pelanggan yang memakan waktu yang lebih dikarenakan harus mencari dan menyaring informasi secara manual pada buku transaksi.
- Guna pembeda antara tabung gas atau pemberian label, pegawai memberi label ke tabung gas dengan spidol, cat dan sistem kerik secara manual sehingga dalam distribusinya ketika tabung terkena percikan air hujan maupun udara lembab maka tanda label tersebut kerap menjadi samar baik karena korosi maupun luntur
- Apabila pembeli jasa menggunakan tabung mereka sendiri maka opsi yang tersedia adalah labeling menggunakan spidol yang pada kondisi lapangnya kerap pudar karena tergesek secara tidak sengaja, atau mencatat ciri-ciri khusus pada tabung mereka yang nantinya akan dicatat pada buku riwayat transaksi, dikarenakan perbedaan regulasi pada tiap instansi bisnis cenderung berbeda yang kerap tidak mengijinkan penggunaan sistem kerik manual yang bersifat semi permanen.

4.3. Kebutuhan Fungsionalitas Sistem

Sistem yang dibuat harus dapat memenuhi kebutuhan fungsional sebagai berikut:

- Aplikasi kasir dapat memasukan data ke dalam basis data atau *database* agar tiap data dapat terintegrasi.
- Sistem dapat melakukan cetak label *barcode* sesuai dengan kode produk yang terdaftar.
- Sistem harus memiliki fitur untuk barang tabung gas yang dibeli maupun dikembalikan.
- Terdapat fitur riwayat transaksi yang terintegrasi dalam aplikasi.
- Mendukung fitur pembagian halaman atau navigasi dengan *pagination* untuk mengurangi beban pada *server* dan klien, sehingga meningkatkan kinerja aplikasi dan waktu respon.

- Menambahkan fitur pencarian berdasarkan rentang tanggal dan nama produk pada riwayat transaksi untuk memudahkan pencarian data.
- Menggunakan validasi tertentu pada fitur-fitur yang bersifat transaksional untuk meminimalisir kesalahan pengguna atau human error.
- Sistem aplikasi yang dirancang dapat mencetak nota transaksi.
- Sistem aplikasi memiliki tampilan responsif.

4.4. Kebutuhan Non Fungsionalitas Sistem

Selain kebutuhan fungsional, sistem yang dirancang juga harus memenuhi kebutuhan non fungsional diantaranya adalah sebagai berikut:

- Sistem dapat diakses oleh masing-masing *user* yang dibuat oleh *admin*.
- Sistem berjalan pada sistem operasi windows.

4.5. Proses Bisnis Usulan

Proses bisnis usulan yang akan diimplementasikan kedalam sistem dibagi menjadi dua aktor, perincian dari kedua aktor adalah sebagai berikut:

- Admin* bertugas untuk mengelola berbagai aspek dari sistem, manajemen pengguna, manajemen produk hingga pengawasan transaksi.
- Kasir bertugas untuk memastikan kelancaran operasional dan kepuasan pelanggan, dengan otomatisasi banyak proses yang sebelumnya dilakukan manual, kasir dapat melayani lebih banyak pelanggan dalam waktu yang lebih singkat, meningkatkan efisiensi operasional toko.

Dalam perancangan aplikasi kasir yang akan dibangun, peneliti mengusulkan *website* sebagai *platform* dikarenakan *barcode scanner* pada saat ini mayoritas menggunakan konektivitas *usb* dan lebih mudah digunakan daripada menggunakan perangkat *mobile*.

Untuk *framework* yang akan digunakan adalah CodeIgniter dikarenakan dokumentasinya yang mudah ditemukan pada berbagai sumber di internet, pemilihan CodeIgniter sebagai *framework* yang dipilih juga berdasarkan bahasa pemrograman yang didukungnya yaitu PHP, dikarenakan PHP adalah pemrograman yang cukup dikenal oleh banyak pengembang dan memungkinkan pengembangan yang lebih fleksibel.

Bootstrap juga diusulkan peneliti dalam menyediakan HTML, CSS, Javascript. Bootstrap juga dilengkapi dengan fitur responsif yang mendukung untuk meningkatkan pengalaman penggunaan yang lebih baik.

Sedangkan basis data yang peneliti gunakan sebagai usulan adalah MySQL dikarenakan basis data ini bersifat *open source* atau dalam artian gratis serta memiliki fitur yang lengkap dalam mendukung proses perancangan aplikasi ini.

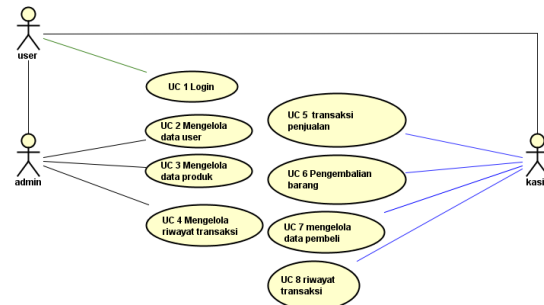
Terkait dengan bagaimana aplikasi kasir yang dirancang agar dapat berinteraksi dengan kode *barcode*

maka peneliti merekomendasikan penggunaan *barcode scanner*, serta untuk mencetak *barcode* berdasarkan kode unik dari sistem kasir nantinya, maka peneliti menggunakan *library* pendukung yang bernama *php-barcode* dari *davidscofftufts* yang berlisensi gratis dan *open source*[8].

4. HASIL DAN PEMBAHASAN

4.1. Use Case Diagram

Use case diagram digunakan selama proses analisis untuk menangkap *requirements sistem* dan untuk memahami bagaimana sistem seharusnya bekerja yang dapat dilihat pada Gambar 3. berikut:

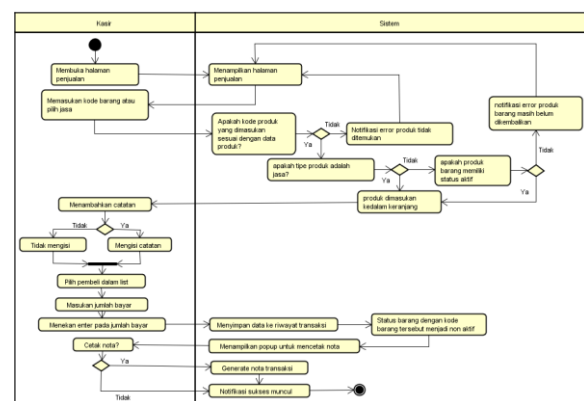


Gambar 3. Use case diagram

Pada Gambar 3. terdapat dua aktor yaitu kasir dan *admin* yang terdapat pada aplikasi yang akan dirancang, kedua aktor tersebut dapat dikategorikan sebagai *user* yang dapat melakukan *login*.

4.2. Activity Diagram

Activity diagram memodelkan alur kerja atau *workflow* sebuah proses bisnis dan urutan aktivitas dalam suatu proses yang dapat dilihat pada Gambar 4. sebagai berikut:

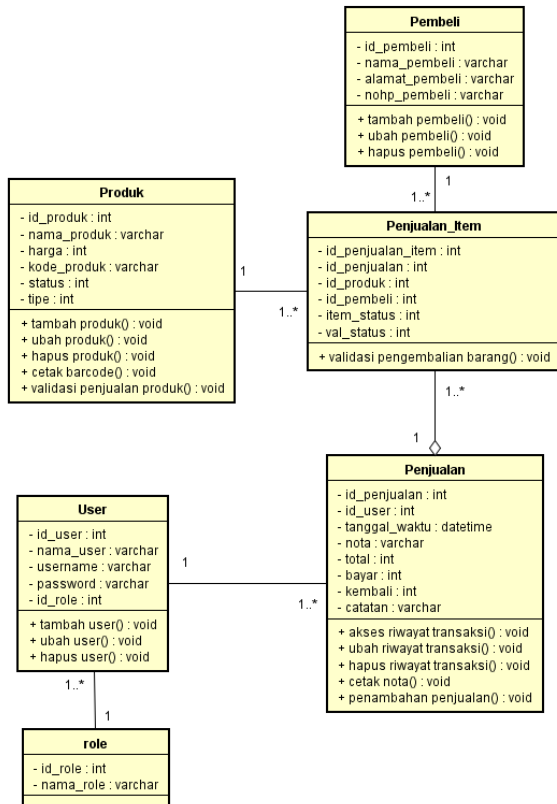


Gambar 4. Activity diagram penjualan

Gambar 4. adalah *Activity diagram* penjualan yang menjelaskan bagaimana sistem berinteraksi dengan proses pelayanan penjualan yang dilakukan oleh *user* dengan aktor kasir, mulai dari melakukan *scan barcode* hingga mencetak nota transaksi.

4.3. Class Diagram

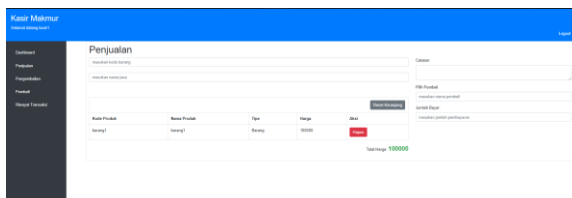
Class diagram memperlihatkan hubungan antar kelas dan penjelasan detail tiap kelas di dalam *model* desain (dalam *logical view*) dari aplikasi kasir yang akan dirancang yang dapat dilihat pada Gambar 5. berikut:



Gambar 5. Class diagram

4.4. Prototype User interface

Perancangan prototype *user interface* dalam tahap *workshop design* memungkinkan calon pengguna dan peneliti sebagai pengembang untuk melihat gambaran awal dari tampilan dan fungsi aplikasi. Perancangan prototype juga membantu *stakeholders* memahami bagaimana aplikasi akan bekerja secara visual dan interaktif serta dapat memberikan umpan balik langsung mengenai tata letak, navigasi, dan elemen interaksi yang dapat diubah atau disesuaikan sebelum pengembangan lebih lanjut dilakukan.



Gambar 6. Prototype user interface penjualan

Pada Gambar 6. terdapat form pengisian data penjualan seperti pengisian kode barang (untuk produk dengan tipe barang), nama jasa, catatan, nama pembeli, dan jumlah pembayaran. Selain itu calon

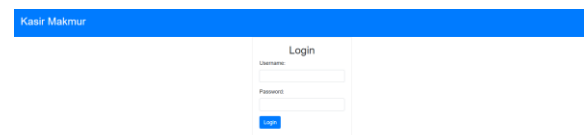
pengguna (sebagai kasir) juga dapat mencoba secara interaktif dari desain prototype yang telah dibuat.

4.5. User interface

Setelah desain prototype dari user interface telah disepakati maka peneliti akan mulai mengimplementasikan user interface tahap akhir yang termasuk dalam tahap *implementation* metode RAD.

4.6. User interface login

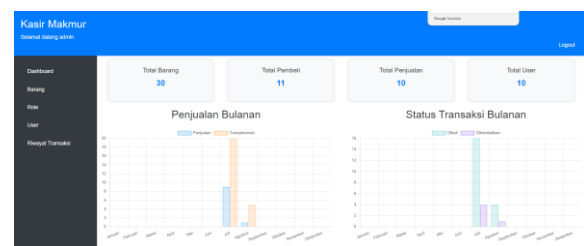
Menampilkan halaman *login* yang akan ditampilkan apabila *session user* masih belum aktif, atau belum melakukan *login* seperti pada Gambar 7.



Gambar 7. User interface login

4.7. User interface dashboard

Menampilkan halaman *dashboard* yang akan otomatis ditampilkan sistem ketika *user* berhasil melakukan *login*, halaman ini juga dapat diakses dengan menekan menu *dashboard* pada *sidebar* seperti pada Gambar 8.



Gambar 8. User interface dashboard

4.8. User interface tambah produk

Menampilkan halaman produk yang diakses oleh *user* sebagai aktor *admin*, dalam halaman ini, *admin* dapat melakukan penambahan data produk baik dengan tipe barang maupun jasa seperti pada Gambar 9.



Gambar 9. User interface tambah produk

4.9. User interface cetak barcode

Menampilkan halaman cetak *barcode*, sistem hanya melakukan cetak *barcode* pada produk yang ber-tipe barang serta dapat dilakukan cetak *barcode* dengan format pdf dengan menekan menu “cetak

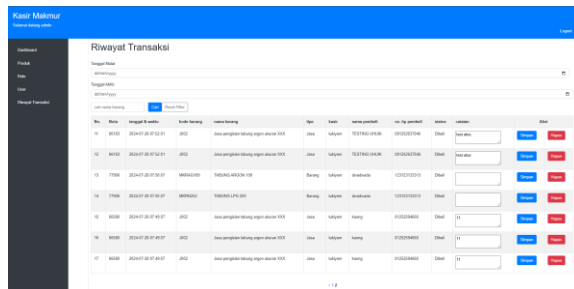
barcode” pada bagian pojok kanan atas seperti pada Gambar 10.



Gambar 10. User interface cetak barcode

4.10. User interface riwayat transaksi admin

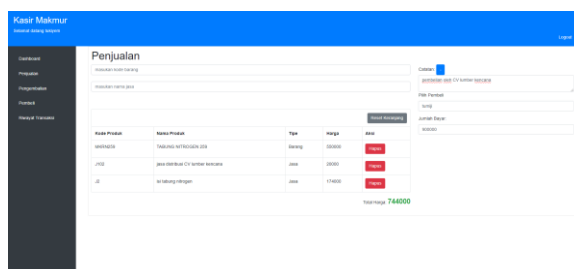
Menampilkan halaman riwayat transaksi *admin*, perbedaan dari riwayat transaksi *admin* dan kasir adalah riwayat transaksi *admin* dapat melakukan perubahan catatan dan menghapus data pada daftar riwayat transaksi, sedangkan riwayat transaksi kasir hanya dapat melihat dan mencari data yang dapat dilihat pada Gambar 11.



Gambar 11. User interface riwayat transaksi admin

4.11. User interface penjualan

Menampilkan halaman penjualan kasir, dimana kasir dapat melayani calon pembeli yang ingin melakukan transaksi, dan setelah produk yang memiliki tipe sebagai “barang” dipesan, maka status dari produk itu akan menjadi non-aktif dan tidak dapat dipesan lagi sebelum barang itu dikembalikan oleh pembeli seperti pada Gambar 12.

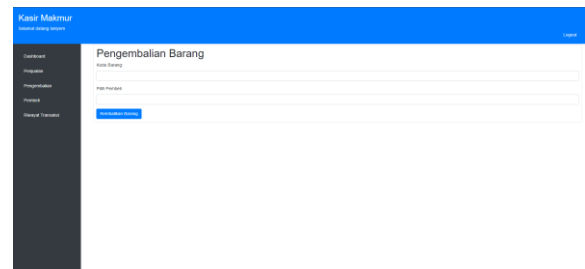


Gambar 12. User interface penjualan

4.12. User interface pengembalian barang

Menampilkan halaman pengembalian barang oleh kasir, dimana pembeli dapat melakukan pengembalian barang oleh user kasir, setelah barang dikembalikan oleh pembeli maka barang tersebut akan menjadi aktif kembali dan dapat dilakukan proses

penjualan sebagaimana mestinya lagi seperti pada Gambar 13.



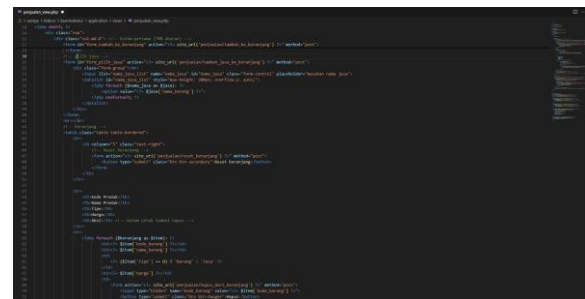
Gambar 13. User interface pengembalian barang

4.13. Kode program

Dalam tahap *implementation* RAD, kode program merujuk pada kumpulan instruksi yang ditulis dalam bahasa pemrograman untuk merealisasikan desain fungsional dari aplikasi yang dirancang. Berikut adalah beberapa potongan kode program yang telah dirancang

4.14. Kode program view penjualan

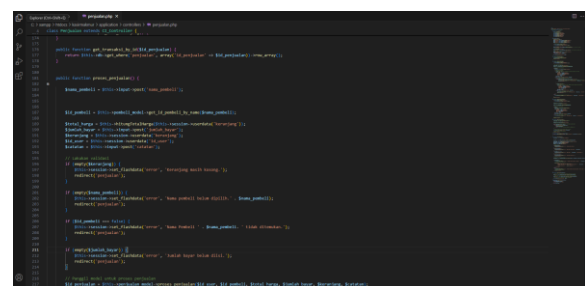
Menampilkan bagaimana kode dari *user interface* penjualan berinteraksi dengan *controller* penjualan dan menampilkan data ke pengguna.



Gambar 14. Kode program view penjualan

4.15. Kode program controller penjualan

Menampilkan bagaimana *controller* penjualan menangani permintaan pengguna, berinteraksi dengan *model*, dan memutuskan *view* mana yang akan ditampilkan. *Controller* berfungsi sebagai penghubung antara *model* dan *view*.

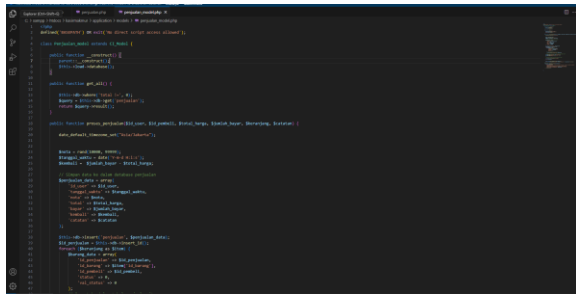


Gambar 15. Kode program controller penjualan

4.16. Kode program model penjualan

Menampilkan bagaimana *model* mengelola operasi basis data seperti mengambil, memasukkan,

memperbarui, dan menghapus data. Model berfungsi sebagai penghubung antara *controller* dan basis data.



Gambar 16. Kode program model penjualan

4.17. Validasi Pengujian Black box testing

Setelah peneliti mengimplementasikan hasil rancangan ke dalam aplikasi dan melakukan pengujian mandiri, maka peneliti melanjutkan pengujian dari aplikasi bersama dengan pihak UD MAKMUR menggunakan *black box testing* yang merupakan pengujian kualitas perangkat lunak dengan berfokus pada fungsionalitas perangkat lunak. Pengujian *black box* bertujuan untuk menemukan fungsi yang tidak benar, kesalahan antarmuka, kesalahan pada struktur data, kesalahan performansi, kesalahan inisialisasi dan terminasi[11]. skenario pengujian *black box testing* dapat dilihat pada Tabel 1.

Tabel 1. Pengujian *black-box testing*

No	Jenis user	Prasyarat skenario	Skenario	Hasil yang diharapkan	Hasil
1.	Admin	Halaman login ditampilkan	Login menggunakan data yang benar	Muncul notifikasi gagal login	Sukses
2.	Admin	Halaman login ditampilkan	Login menggunakan data yang salah	Sistem menampilkan halaman dashboard	Sukses
3.	Admin	Admin berhasil login	Menekan produk pada sidebar	Dapat mengakses daftar produk	Sukses
4.	Admin	Halaman produk ditampilkan	Menekan tambah produk dan mengisi data lalu tekan simpan	Produk berhasil ditambahkan pada daftar produk	Sukses
5.	Admin	Daftar produk memiliki data	Mengubah dan menghapus data produk	Dapat mengubah dan menghapus data pada daftar produk	Sukses
6.	Admin	Daftar produk memiliki data	Menekan cetak barcode	Halaman cetak barcode ditampilkan dan dapat mencetak dalam bentuk format pdf	Sukses
7.	Admin	Admin Berhasil login	Menekan role pada menu sidebar	Sistem menampilkan halaman role	Sukses
8.	Admin	Admin Berhasil login	Menekan user pada menu sidebar	Sistem menampilkan halaman user	Sukses
9.	Admin	Halaman user ditampilkan	Menekan tambah user dan mengisi data lalu tekan simpan	User berhasil ditambahkan pada daftar user	Sukses
10.	Admin	Daftar user memiliki data	Mengubah dan menghapus data user	Dapat mengubah dan menghapus data pada daftar user	Sukses
11.	Kasir	Admin sudah menambahkan user kasir	Login menggunakan data yang benar	Muncul notifikasi gagal login	Sukses
12.	Kasir	Admin sudah menambahkan user kasir	Login menggunakan data yang salah	Sistem menampilkan halaman dashboard	Sukses
13.	Kasir	Kasir berhasil login	Menekan pembeli pada menu sidebar	Sistem menampilkan halaman pembeli	Sukses
14.	Kasir	Halaman pembeli ditampilkan	Menekan tambah pembeli dan mengisi data lalu tekan simpan	Pembeli berhasil ditambahkan pada daftar pembeli	Sukses
15.	Kasir	Daftar pembeli memiliki data	Mengubah dan menghapus data pembeli	Dapat mengubah dan menghapus data pada daftar pembeli	Sukses
16.	Kasir	Kasir berhasil login	Menekan penjualan pada menu sidebar	Sistem menampilkan halaman penjualan	Sukses
17.	Kasir	Admin telah menambahkan produk barang dan mencetak barcode	Melakukan scan barcode pada kolom kode barang dan pilih atau cari jasa yang ditampilkan pada daftar jasa lalu tekan "enter"	Produk beserta detailnya berhasil ditambahkan pada keranjang penjualan	Sukses
18.	Kasir	Keranjang sudah berisi data produk	Tekan tombol "+" dan mengisi catatan	Kolom catatan dapat diisi	Sukses

No	Jenis user	Prasyarat skenario	Skenario	Hasil yang diharapkan	Hasil
19.	Kasir	Kasir telah menambahkan data pembeli	Pilih atau cari nama pembeli	Pembeli berhasil dipilih	Sukses
20.	Kasir	Data keranjang, nota dan pembeli terisi	Masukan jumlah bayar lalu tekan “enter”	Muncul notifikasi untuk memilih pencetakan nota	Sukses
21.	Kasir	Muncul popup untuk memilih pencetakan nota	Tekan “ok”	Nota dapat dicetak dengan format pdf	Sukses
22.	Kasir	Kasir berhasil login	Menekan pengembalian pada menu sidebar	Sistem menampilkan halaman pengembalian barang	Sukses
23.	Kasir	Penjualan sudah dilakukan dan berstatus belum dilakukan pengembalian barang	Melakukan scan barcode pada kolom kode barang dan pilih atau cari pembeli yang bukan pembelinya	Muncul notifikasi gagal karena data tidak valid	Sukses
24.	Kasir	Penjualan sudah dilakukan dan berstatus belum dilakukan pengembalian barang	Melakukan scan barcode pada kolom kode barang dan pilih atau cari pembeli sesuai dengan penjualan yang belum dilakukan pengembalian barang	Muncul notifikasi sukses melakukan pengembalian barang	Sukses
25.	Admin atau Kasir	Kasir pernah melakukan proses transaksi penjualan	Menekan riwayat transaksi pada menu sidebar	Sistem menampilkan halaman riwayat transaksi dan data transaksi sebelumnya tercatat didalamnya	Sukses
26.	Admin atau Kasir	Kasir pernah melakukan proses transaksi penjualan	Masukan kombinasi tanggal mulai, tanggal akhir, atau nama barang, lalu tekan cari	Data yang ditampilkan akan relevan sesuai dengan kriteria pencarian	Sukses
27.	Admin	Kasir pernah melakukan proses transaksi penjualan	Masukan perubahan pada kolom catatan pada data daftar riwayat transaksi, lalu tekan simpan	Catatan transaksi berhasil diubah	Sukses
28.	Admin	Kasir pernah melakukan proses transaksi penjualan	Menekan hapus pada daftar riwayat transaksi	Data pada riwayat transaksi berhasil dihapus	Sukses

5. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang dilakukan di UD MAKMUR, dapat disimpulkan bahwa aplikasi kasir yang dirancang telah memenuhi tahap validasi pengujian menggunakan metode *black box testing* melalui kolaborasi antara peneliti dan pihak UD MAKMUR. Hasil pengujian menunjukkan bahwa aplikasi telah memenuhi kebutuhan fungsional dan non-fungsional dengan fitur-fitur tambahan seperti pagination, *filter* pencarian tanggal dan waktu telah efektif dalam membantu manajemen dan pencarian data yang didapatkan peneliti melalui umpan balik yang diberikan ketika melakukan tahap pengujian. Fitur tambahan seperti pencetakan *barcode* dan dukungan *barcode scanner* juga berhasil mempersingkat proses bisnis.

DAFTAR PUSTAKA

- [1] M. Z. Akbar, M. A. Nur, M. F. Sabana, and T. Tanjung, “Perancangan Aplikasi Kasir Berbasis Website Pada Toko Sembako Menggunakan Metode Waterfall,” *OKTAL: Jurnal Ilmu Komputer dan Sains*, vol. 1, no. 8, pp. 1274–1281, 2022.
- [2] S. Bakhri, F. Hanif, and A. Haidir, “Rancang Bangun Aplikasi Kasir Penjualan Susu Berbasis Web Pada Alomgada Kids Jakarta,” *IJCIT (Indonesian Journal on Computer and Information Technology)*, vol. 5, no. 1, pp. 47–54, 2020.
- [3] C. R. Pakusadewa and U. Chotijah, “Perancangan Sistem Informasi Aplikasi Kasir Unit Pelayanan Jasa Toko Raya Computer Berbasis WEB,” *Jurnal Nasional Komputasi dan Teknologi Informasi*, vol. 6, no. 5, pp. 621–627, 2023.
- [4] A. Lutfi, “Sistem Informasi Akademik Madrasah Aliyah Salafiyah Syafi’iyah Menggunakan Php Dan Mysql,” *Jurnal AiTech*, vol. 3, no. 2, 2017.
- [5] P. Sulistyorini, “Pemodelan Visual dengan Menggunakan UML dan Rational Rose,” *Jurnal Teknologi Informasi DINAMIK*, vol. 14, no. 1, pp. 23–29, 2009.
- [6] G. Jinda, Y. D. Da Khwuta, and E. E. Sala, “Rancang Bangun Sistem Informasi Penyewaan Kos Putri Di Kelurahan Onekore,” *Jurnal Jupiter*, vol. 15, no. 1, pp. 135–142, 2023.
- [7] F. Rahman and S. Ratna, “Perancangan E-Learning Berbasis Web Menggunakan Framework CodeIgniter,” *Jurnal Ilmiah Technologia*, vol. 9, no. 2, pp. 95–100, 2018.

- [8] Davidscotttufts, "Source code for the article 'How To Create Barcodes in PHP' found at <http://davidscotttufts.com/2009/03/31/how-to-create-barcodes-in-php/> - Issues · davidscotttufts/php-barcode." Accessed: Aug. 27, 2024. [Online]. Available: <https://github.com/davidscotttufts/php-barcode>
- [9] S. Kosasi and I. D. A. E. Yuliani, "Penerapan Rapid Application Development Pada Sistem Penjualan Sepeda Online," *Jurnal SIMETRIS*, vol. 6, no. 1, pp. 27–36, 2015.
- [10] N. M. N. Daud, N. A. A. A. Bakar, and H. M. Rusli, *Implementing Rapid Application Development (RAD) Methodology in Developing Practical Training Application System*. IEEE, 2010.
- [11] Wijaya. Yahya Dwi and M. W. Astuti, "Pengujian Blackbox Sistem Informasi Penilaian Kinerja Karyawan Pt Inka (Persero) Berbasis Equivalence Partitions.," *Jurnal Digital Teknologi Informasi*, vol. 4, no. 1, pp. 22–26, 2021.