

Aplikasi untuk Pengelolaan Rantai Pasok Kue Kering (Studi Kasus: CV Anta Boga Kharisma, Kota Bandung)

1st Sabila
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

sabilabil@student.telkomuniversity.ac.id

2nd Asti Widayanti
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

astiwidayanti@tass.telkomuniversity.ac.id

3rd Anak Agung Gde Agung
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

aagdeagung@tass.telkomuniversity.ac.id

Abstrak - CV. Anta Boga Kharisma merupakan perusahaan yang bergerak di bidang distribusi dan pengelolaan kue kering. Sistem distribusi barang dagang pada CV Anta Boga Kharisma diawali dengan pembelian bahan baku kemudian perusahaan melakukan proses produksi. Kemudian dilanjut dengan proses permintaan pesanan dari distributor dan *reseller* kepada perusahaan. Setelah itu perusahaan akan melakukan pengiriman produk, jika pesanan dari distributor dan *reseller* tidak mencukupi stok produksi maka pengiriman akan dikirim sebagian. Lalu perusahaan akan menerbitkan *invoice*, tetapi *invoice* akan dikirim jika semua produk sudah diterima, selanjutnya distributor atau *reseller* melakukan transaksi pembayaran. Proses pencatatan dan pengelolaan pada setiap transaksi pada perusahaan ini masih dilakukan dengan manual menggunakan excel sehingga belum terintegrasi satu sama lain yang dapat menyebabkan kesalahan dalam melakukan pencatatan. Maka dari itu untuk mengatasi permasalahannya dibuatkan aplikasi pengelolaan rantai pasok kue kering. Aplikasi ini dibuat dengan menggunakan *Software Development Life Cycle* (SDLC) dengan model *prototype*, bahasa pemrograman PHP, perancangan menggunakan *Unified Modeling Language* (UML), serta framework CI 4. Dapat diperoleh kesimpulan bahwa aplikasi pengelolaan rantai pasok kue kering dapat membantu perusahaan untuk menghasilkan laporan keuangan yang baik berdasarkan pengelolaan transaksi yang fungsionalitasnya dapat melakukan transaksi permintaan, pengiriman barang, penerbitan *invoice*, pencatatan pembayaran, membuat jurnal umum, membuat laporan buku besar, dan membuat laporan laba rugi.

Kata Kunci: distribusi, permintaan, pasok, CI 4.

Abstract - CV. Anta Boga Kharisma is a company engaged in the distribution and management of pastries. The merchandise distribution system at CV Anta Boga Kharisma begins with the purchase of raw materials and then the company carries out the production process. Then proceed with the process of requesting orders from distributors and resellers to the company. After that, the company will deliver the product, if orders from distributors and resellers do not have sufficient production stock, then partial shipments will be sent. Then the company will issue an *invoice*, but the *invoice* will be sent if all products have been received, then the distributor or reseller makes a payment transaction. The

process of recording and managing every transaction at this company is still done manually using excel so that it is not yet integrated with each other which can cause errors in recording. Therefore, to overcome the problem, a pastry supply chain management application was created. This application was created using *Software Development Life Cycle* (SDLC) with a *prototype* model, PHP programming language, design using *Unified Modeling Language* (UML), and CI 4 framework. It can be concluded that the application of pastry supply chain management can help companies to produce reports good finance based on transaction management whose functionality can make request transactions, deliver goods, issue *invoices*, record payments, make general journals, make general ledger reports, and make profit and loss reports.

Keywords: distribution, demand, supply, CI 4.

I. PENDAHULUAN

Produksi adalah sebagian kegiatan yang menghasilkan barang (baik barang jadi maupun barang setengah jadi), bahan-bahan industri, bahan suku cadang (komponen). Distribusi adalah suatu kegiatan untuk pemindahan barang dari tahap produsen kepada konsumen didalam bagian *supply chain*.

CV Anta Boga Kharisma merupakan perusahaan manufaktur yang bergerak dalam bidang penjualan kue kering yang terletak di daerah Antapani Bandung. Perusahaan ini menargetkan produksi 10.000 toples setiap tahunnya, biasanya CV Anta Boga Kharisma akan mulai memproduksi kue minimal 6 bulan menjelang lebaran idul fitri.

CV Anta Boga Kharisma memproduksi 20 jenis kue kering, diantaranya *kaastangel*, nastar keju, nastar klasik, salju keju, putri gurun, sagu keju, *choco cizz*, *lemon cookies*, *florentine*, *bokepootjes mede cookies*, *choco mede*, *choco modelling*, *choco rice*, lidah kucing, lidah kucing *blueberry*, lidah kucing coklat, *green tea*, *butter peanut*, *mylo cookies*. Untuk varian kemasan terbagi menjadi dua diantaranya kemasan toples

Sistem distribusi barang dagang pada CV Anta Boga Kharisma diawali dengan pembelian bahan baku kemudian perusahaan melakukan proses produksi. Kemudian dilanjut dengan proses permintaan pesanan dari distributor dan *reseller* kepada perusahaan. Setelah itu perusahaan akan melakukan pengiriman produk, jika pesanan dari distributor dan *reseller* tidak mencukupi stok produksi maka pengiriman akan dikirim sebagian. Lalu perusahaan akan menerbitkan *invoice*, tetapi *invoice* akan dikirim jika semua produk sudah diterima, selanjutnya distributor atau *reseller* melakukan transaksi pembayaran.

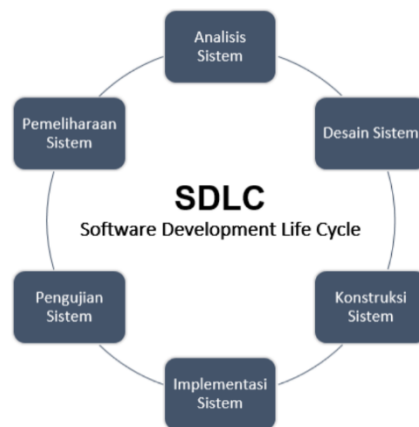
Data permintaan pesanan, proses produksi, pengiriman barang, dan transaksi pembayaran masih terpisah. untuk proses pencatatan dan pengelolaan pada setiap transaksi pada perusahaan ini masih dilakukan dengan manual menggunakan *excel* sehingga belum terintegrasi satu sama lain yang dapat menyebabkan kesalahan dalam melakukan pencatatan. Maka dari itu proyek akhir ini diharapkan bisa memberikan solusi bagi perusahaan dalam setiap melakukan proses rantai pasok kue dari mulai pencatatan permintaan pesanan, pencatatan penjualan, pencatatan sistem pengiriman barang, jurnal umum, buku besar, dan laporan laba/rugi.

II. METODE Pengerjaan

Aplikasi proyek akhir ini menggunakan pengembangan metode *software development life cycle* (SDLC) dengan model *prototyping*. SDLC adalah siklus perancangan atau pengembangan perangkat lunak yang bertujuan untuk menyelesaikan masalah secara efektif dan menghasilkan sistem yang berkualitas sesuai dengan keinginan pengguna atau sesuai tujuan perancangan sistem. SDLC dalam rekayasa perangkat lunak adalah proses membuat dan memodifikasi perangkat lunak dengan menggunakan model atau metode pengembangan perangkat lunak tersebut.

Konsep SDLC mendasari banyak jenis metodologi pengembangan perangkat lunak. Ada 3 jenis metode siklus hidup sistem yang paling banyak digunakan, yaitu: SDLC tradisional, SDLC *prototyping*, SDLC berorientasi pada objek.

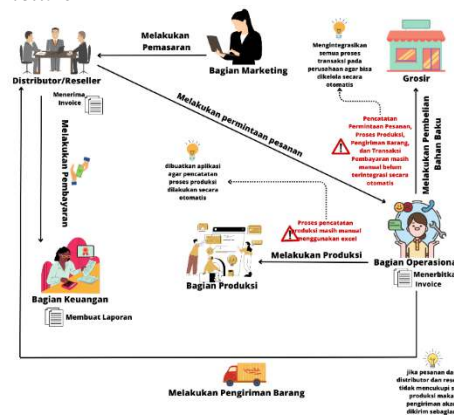
SDLC dalam pengembangan perangkat lunak memiliki fungsi antara lain sebagai sarana komunikasi antara tim pengembang dan pengguna/pelanggan. Dengan SDLC juga berfungsi untuk memberi peran dan tanggung jawab secara jelas antara pengembang, desainer, analis bisnis, dan manajer proyek. Fungsi lain dari SDLC adalah dapat memberikan gambaran yang jelas tentang *input* dan *output* dari satu tahap ke tahap berikutnya. SDLC memiliki tahapan-tahapan pengerjaan yang dilakukan oleh analis sistem dan *programmer* dalam membangun sebuah perangkat lunak. Ada 6 tahapan dalam SDLC antara lain: analisis sistem, desain sistem, konstruksi sistem, implementasi sistem, pengujian sistem, dan pemeliharaan sistem [1].



GAMBAR 1- 1
Tahapan Pengembangan Sistem SDLC

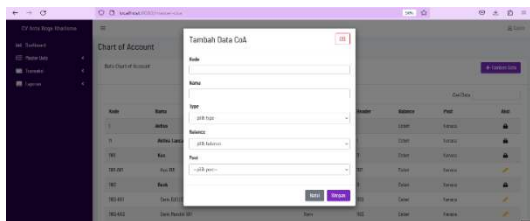
III. HASIL DAN PEMBAHASAN

A. Rich Picture



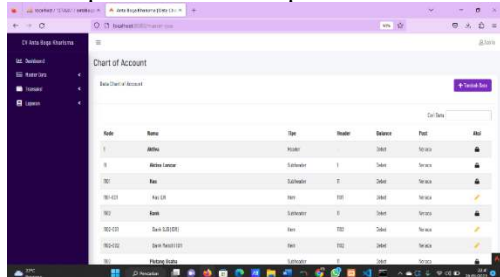
GAMBAR 3- 1
Rich Picture

Berdasarkan gambar, untuk memenuhi aktivitas produksi pada perusahaan CV Anta Boga Kharisma terdapat kegiatan pembelian bahan kepada grosir, proses rantai pasok kue dari perusahaan diawali dengan pembelian bahan baku kemudian



GAMBAR 4- 3
Implementasi Proses Menambah Master Data COA

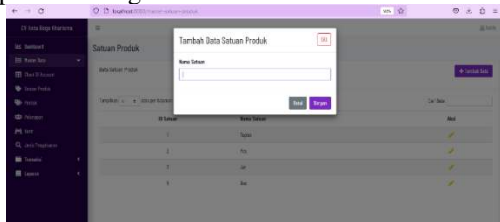
Berikut merupakan tampilan data coa yang telah berhasil ditambahkan. Halaman ini berfungsi untuk memberikan informasi kepada pengguna data akun yang tersedia. Berikut merupakan implementasi menampilkan master data coa:



GAMBAR 4- 4
Implementasi Proses Menampilkan Master Data Akun

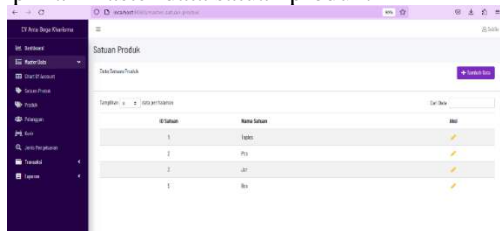
3. Implementasi Master Data Satuan Produk

Berikut ini merupakan implementasi master data satuan produk, berupa proses menambah data satuan produk, menampilkan data satuan produk, dan mengubah data satuan produk. Pada implementasi master data satuan produk terdapat tombol simpan yang berfungsi menyimpan data satuan produk sebagai berikut:



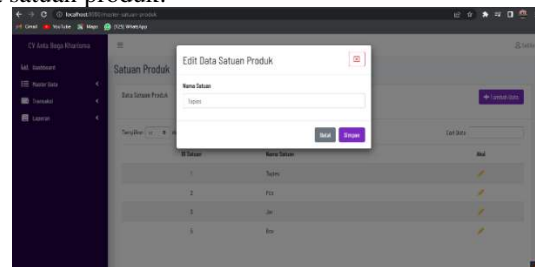
GAMBAR 4- 5
Implementasi Proses Menambah Master Data Satuan Produk

Berikut merupakan tampilan data satuan produk yang telah berhasil ditambahkan. Halaman ini berfungsi untuk memberikan informasi kepada pengguna data satuan produk yang tersedia. Berikut merupakan implementasi menampilkan master data satuan produk:



GAMBAR 4- 6
Implementasi Proses Menampilkan Master Data Satuan Produk

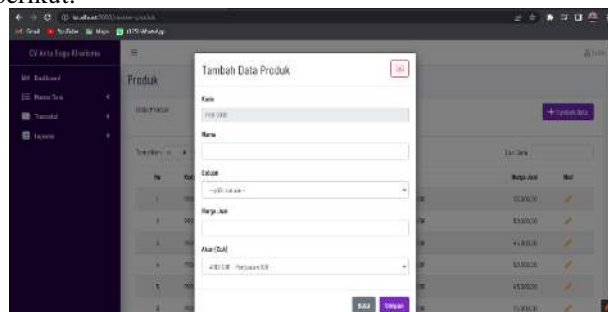
Pada halaman menampilkan master data satuan produk diatas terdapat *icon* warna kuning yang berfungsi untuk mengubah data pada master data satuan produk. Pengguna dapat mengubah data yang sebelumnya telah ditambahkan. Berikut adalah implementasi dari proses mengubah master data satuan produk:



GAMBAR 4- 7
Implementasi Proses Mengubah Master Data Satuan Produk

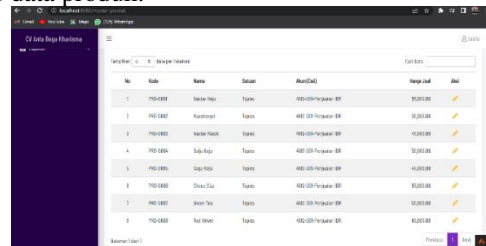
4. Implementasi master data produk

Berikut ini merupakan implementasi master data produk, berupa proses menambah data produk, menampilkan data produk, dan mengubah data produk. Pada implementasi master data produk pengguna mengisi data kode, nama, satuan, harga jual, akun (coa), setelah itu terdapat tombol simpan yang berfungsi menyimpan data produk sebagai berikut:



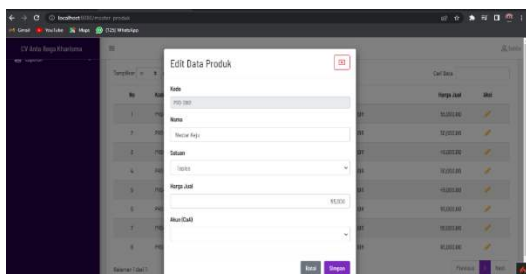
GAMBAR 4- 8
Implementasi Proses Menambah Data Produk

Berikut merupakan tampilan data produk yang telah berhasil ditambahkan. Halaman ini berfungsi untuk memberikan informasi kepada pengguna data produk yang tersedia. Berikut merupakan implementasi menampilkan master data produk:

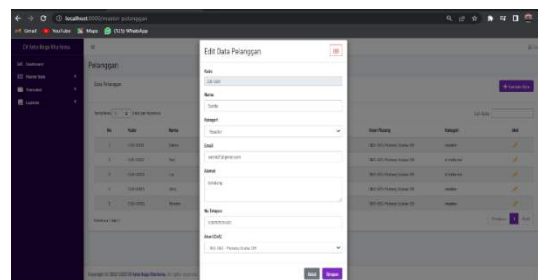


GAMBAR 4- 9 Implementasi Proses Menampilkan Master Data Produk

Pada halaman menampilkan master data produk diatas terdapat *icon* warna kuning yang berfungsi untuk mengubah data pada master data produk. Pengguna dapat mengubah data yang sebelumnya telah ditambahkan. Berikut adalah implementasi dari proses mengubah master data produk:



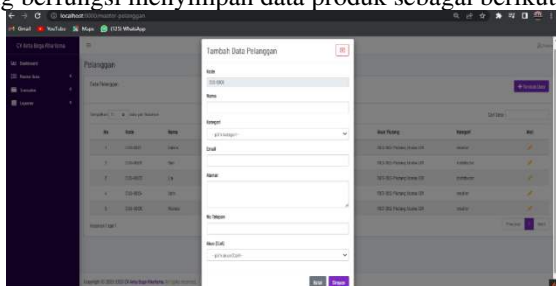
GAMBAR 4- 10
Implementasi Proses Mengubah Master Data Produk



GAMBAR 4- 13
Implementasi Proses Mengubah Master Data Pelanggan

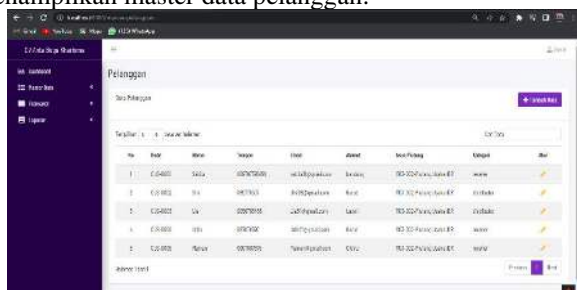
5. Implementasi master data pelanggan

Berikut ini merupakan implementasi master data pelanggan, berupa proses menambah data pelanggan, menampilkan data pelanggan, dan mengubah data pelanggan. Pada implementasi master data pelanggan pengguna mengisi data kode, nama, telepon, email, alamat, akun piutang, kategori, setelah itu terdapat tombol simpan yang berfungsi menyimpan data produk sebagai berikut:



GAMBAR 4- 11
Implementasi Proses Menambah Master Data Pelanggan

Berikut merupakan tampilan data pelanggan yang telah berhasil ditambahkan. Halaman ini berfungsi untuk memberikan informasi kepada pengguna data pelanggan yang tersedia. Berikut merupakan implementasi menampilkan master data pelanggan:

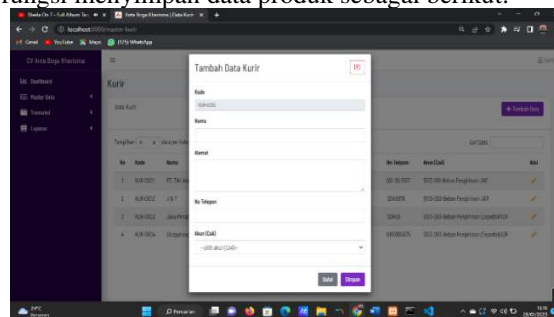


GAMBAR 4- 12
Implementasi Menampilkan Master Data Pelanggan

Pada halaman menampilkan master data pelanggan diatas terdapat *icon* warna kuning yang berfungsi untuk mengubah data pada master data pelanggan. Pengguna dapat mengubah data yang sebelumnya telah ditambahkan. Berikut adalah implementasi dari proses mengubah master data pelanggan:

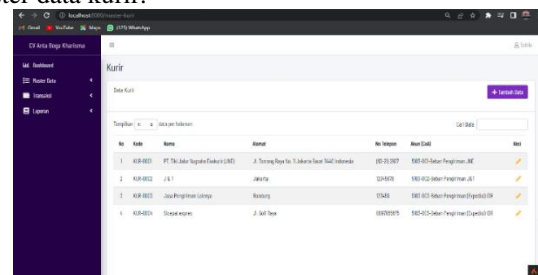
6. Implementasi master data kurir

Berikut ini merupakan implementasi master data kurir, berupa proses menambah data kurir, menampilkan data kurir, dan mengubah data kurir. Pada implementasi master data kurir pengguna mengisi data kode, nama, no telepon, alamat, akun coa, setelah itu terdapat tombol simpan yang berfungsi menyimpan data produk sebagai berikut:



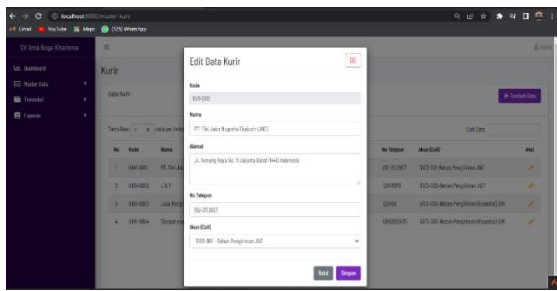
GAMBAR 4- 14
Implementasi Proses Menambah Master Data Kurir

Berikut merupakan tampilan data kurir yang telah berhasil ditambahkan. Halaman ini berfungsi untuk memberikan informasi kepada pengguna data kurir yang tersedia. Berikut merupakan implementasi menampilkan master data kurir:



GAMBAR 4- 15
Implementasi Proses Menampilkan Master Data Kurir

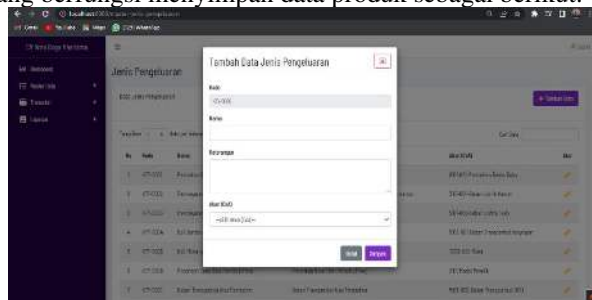
Pada halaman menampilkan master data kurir diatas terdapat *icon* warna kuning yang berfungsi untuk mengubah data pada master data kurir. Pengguna dapat mengubah data yang sebelumnya telah ditambahkan. Berikut adalah implementasi dari proses mengubah master data kurir:



GAMBAR 4- 16
Implementasi Proses Mengubah Master Data Kurir

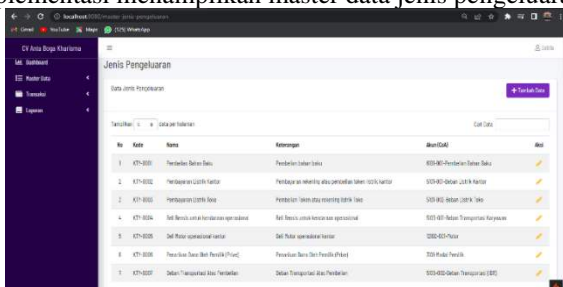
7. Implementasi master data jenis pengeluaran

Berikut ini merupakan implementasi master data jenis pengeluaran, berupa proses menambah data jenis pengeluaran, menampilkan data jenis pengeluaran, dan mengubah data jenis pengeluaran. Pada implementasi master data jenis pengeluaran pengguna mengisi data kode, nama, keterangan, akun coa, setelah itu terdapat tombol simpan yang berfungsi menyimpan data produk sebagai berikut:



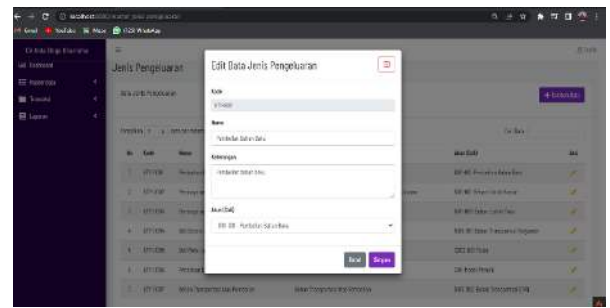
GAMBAR 4- 17
Implementasi Proses Menambah Data Jenis Pengeluaran

Berikut merupakan tampilan master data jenis pengeluaran yang telah berhasil ditambahkan. Halaman ini berfungsi untuk memberikan informasi kepada pengguna data jenis pengeluaran yang tersedia. Berikut merupakan implementasi menampilkan master data jenis pengeluaran:



GAMBAR 4- 18
Implementasi Proses Menampilkan Master Data Jenis Pengeluaran

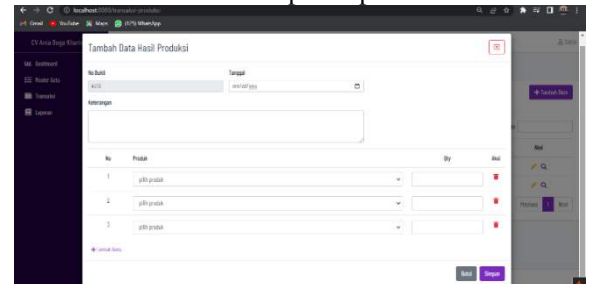
Pada halaman menampilkan master data jenis pengeluaran diatas terdapat *icon* warna kuning yang berfungsi untuk mengubah data pada master data jenis pengeluaran. Pengguna dapat mengubah data yang sebelumnya telah ditambahkan. Berikut adalah implementasi dari proses mengubah master data jenis pengeluaran:



GAMBAR 4- 19
Implementasi Proses Mengubah Master Data Jenis Pengeluaran

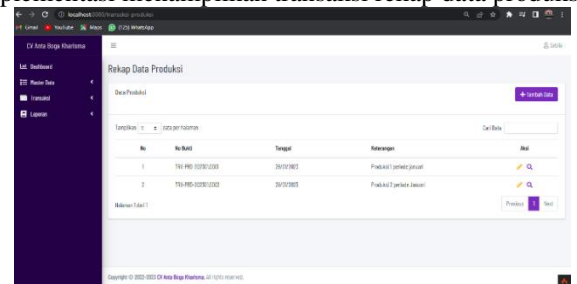
8. Implementasi transaksi

Berikut ini merupakan implementasi transaksi rekap data produksi. Pada halaman transaksi rekap data produksi *user* dapat menambah dan menampilkan data transaksi rekap data produksi. Pada halaman menambah data rekap produksi terdapat kolom yang harus diisi oleh *user* diantaranya adalah tanggal, dan keterangan. Sedangkan pada kolom inputan produk dan *quantity* diisi sesuai jumlah produk yang telah diproduksi. Tombol simpan berfungsi untuk menyimpan data rekap data produksi. Berikut implementasi dari proses menambahkan transaksi rekap data produksi:



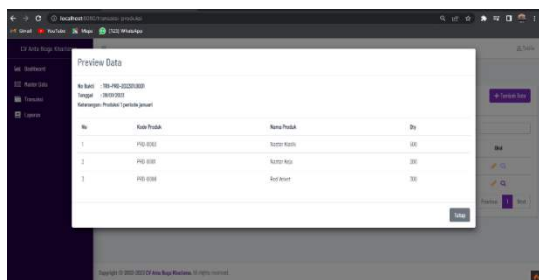
GAMBAR 4- 20
Implementasi Proses Menambahkan Transaksi Rekap Data Produksi

Berikut merupakan tampilan transaksi rekap data produksi yang telah berhasil ditambahkan. Halaman ini berfungsi untuk memberikan informasi kepada pengguna data rekap produksi yang telah diinput. Berikut merupakan implementasi menampilkan transaksi rekap data produksi:



GAMBAR 4- 21
Implementasi Proses Menampilkan Transaksi Rekap Data Produksi

Pada halaman diatas pada kolom aksi terdapat *icon* *search* yang berfungsi untuk *preview* data hasil produk yang telah di produksi. Berikut implementasi menampilkan *preview* transaksi rekap data produksi:

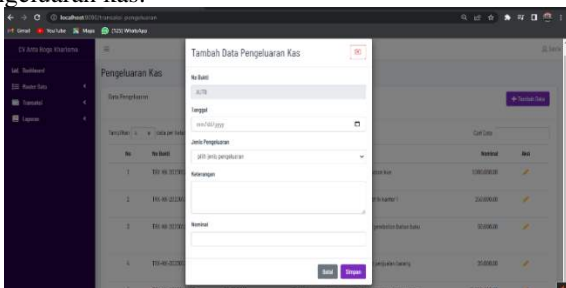


GAMBAR 4- 22

Implementasi Proses Menampilkan Preview Transaksi Rekap Data Produksi

9. Implementasi transaksi pengeluaran kas

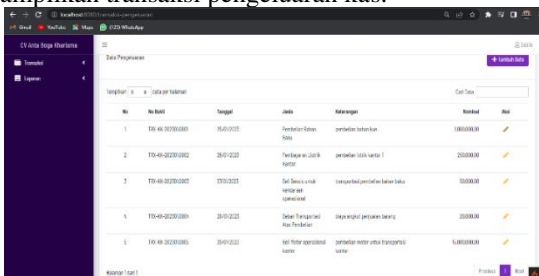
Berikut ini merupakan implementasi pengeluaran kas. Pada halaman transaksi pengeluaran kas *user* dapat menambah dan menampilkan data transaksi pengeluaran kas. Pada halaman menambah data pengeluaran kas terdapat kolom yang harus diisi oleh *user* diantaranya adalah tanggal, jenis pengeluaran, keterangan, dan nominal. Tombol simpan berfungsi untuk menyimpan data pengeluaran kas. Berikut implementasi dari proses menambahkan transaksi pengeluaran kas:



GAMBAR 4- 23

Implementasi Proses Menambah Transaksi Pengeluaran Kas

Berikut merupakan tampilan transaksi pengeluaran kas yang telah berhasil ditambahkan. Halaman ini berfungsi untuk memberikan informasi kepada pengguna data jenis pengeluaran yang tersedia. Berikut merupakan implementasi menampilkan transaksi pengeluaran kas:



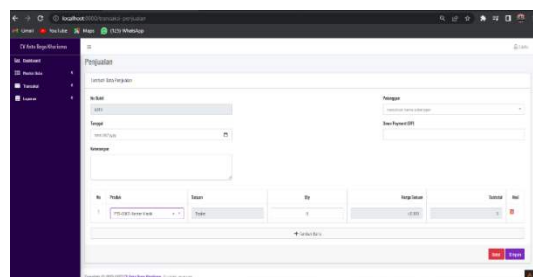
GAMBAR 4- 24

Implementasi Proses Menampilkan Transaksi Pengeluaran Kas

10. Implementasi transaksi penjualan

Berikut ini merupakan implementasi penjualan. Pada halaman transaksi penjualan *user* dapat menambah dan menampilkan data transaksi penjualan. Pada halaman menambah data penjualan terdapat kolom yang harus diisi oleh *user* diantaranya adalah tanggal, jenis, keterangan, pelanggan, dan *down payment*. Sedangkan pada kolom inputan produk dan *quantity* diisi sesuai produk yang dipesan. Tombol simpan berfungsi untuk menyimpan data

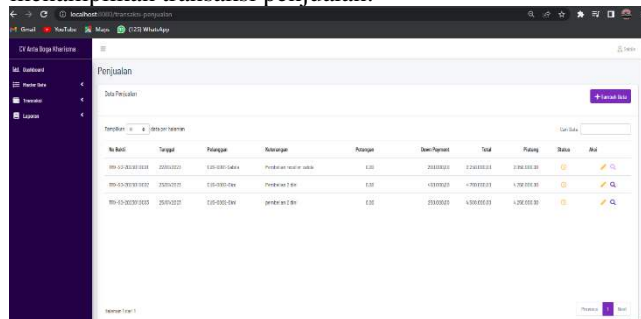
transaksi penjualan. Berikut implementasi dari proses menambahkan transaksi penjualan:



GAMBAR 4- 25

Implementasi Proses Menambah Transaksi Penjualan

Berikut merupakan tampilan transaksi penjualan yang telah berhasil ditambahkan. Halaman ini berfungsi untuk memberikan informasi kepada pengguna data penjualan yang telah diinput. Berikut merupakan implementasi menampilkan transaksi penjualan:

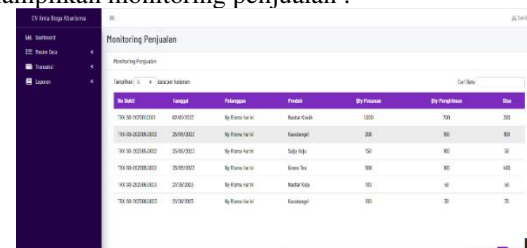


GAMBAR 4- 26

Implementasi Proses Menampilkan Transaksi Penjualan

11. Implementasi transaksi monitoring penjualan

Berikut ini merupakan implementasi monitoring penjualan. Pada halaman transaksi monitoring penjualan *user* dapat mencari data dan menampilkan data transaksi monitoring penjualan. Pada kolom cari data diisi nama produk. Transaksi monitoring penjualan ini merupakan tampilan dari produksi yang belum dikirim. Berikut adalah implementasi menampilkan monitoring penjualan :



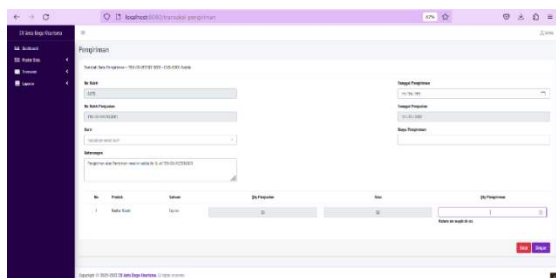
GAMBAR 4- 27

Implementasi Menampilkan Transaksi Monitoring Penjualan

12. Implementasi transaksi pengiriman

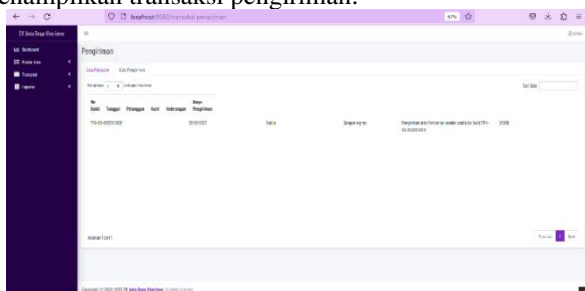
Berikut ini merupakan implementasi pengiriman. Pada halaman transaksi pengiriman *user* dapat menambah dan menampilkan data transaksi pengiriman. Sebelum masuk ke halaman menambah data pengiriman *user* diminta untuk memasukkan nama pelanggan yang sudah melakukan transaksi penjualan, setelah itu pada kolom aksi berfungsi untuk melanjutkan ke transaksi pengiriman. Pada halaman menambah data pengiriman terdapat kolom yang harus diisi

yaitu kurir, keterangan, tanggal pengiriman, tanggal penjualan, biaya pengiriman. Sedangkan pada kolom *quantity* diisi sesuai produk yang tersedia. Tombol simpan berfungsi untuk menyimpan data transaksi pengiriman. Berikut implementasi dari proses menambahkan transaksi pengiriman:



GAMBAR 4- 28
Implementasi Proses Menambahkan Transaksi Pengiriman

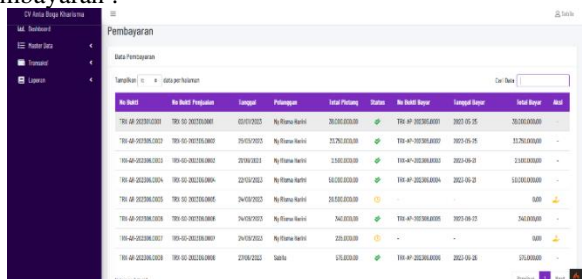
Berikut merupakan tampilan transaksi pengiriman yang telah berhasil ditambahkan. Halaman ini berfungsi untuk memberikan informasi kepada pengguna data pengiriman yang telah terkirim. Berikut merupakan implementasi menampilkan transaksi pengiriman:



GAMBAR 4- 29
Implementasi Proses Menampilkan Transaksi Pengiriman

13. Implementasi transaksi pembayaran

Berikut ini merupakan implementasi transaksi pembayaran. Pada halaman transaksi pembayaran *user* dapat mencari data, menampilkan data, dan pada tabel aksi *user* mengklik tombol pembayaran jika pembayaran sudah berhasil dilakukan. Berikut adalah implementasi transaksi pembayaran :

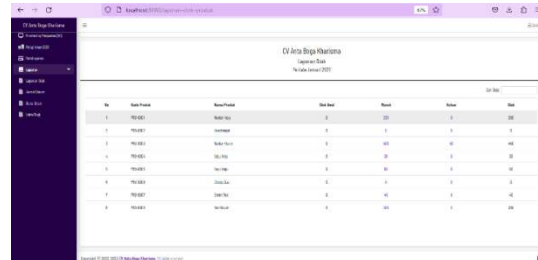


GAMBAR 4- 30
Implementasi Proses Menampilkan Transaksi Pembayaran

14. Implementasi laporan stok

Berikut merupakan tampilan isi data pada laporan stok. Pada halaman ini semua data jumlah produk yang telah diinputkan pada transaksi rekam data produksi berada pada

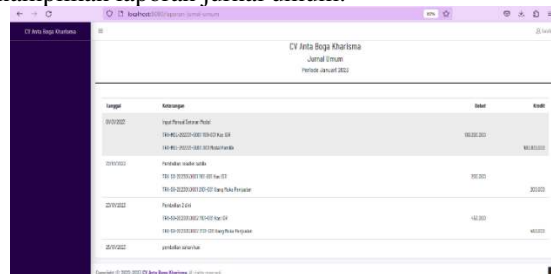
halaman ini. Laporan ini berfungsi untuk memberikan informasi kepada pengguna terkait data laporan stok yang tercatat. Berikut adalah implementasi menampilkan laporan stok:



GAMBAR 4- 31
Implementasi Proses Menampilkan Laporan Stok

15. Implementasi laporan jurnal umum

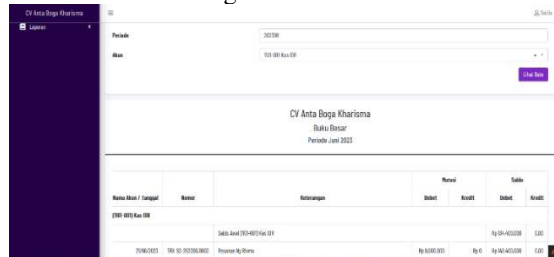
Berikut merupakan tampilan isi data pada laporan jurnal umum. Pada halaman ini semua data telah diinputkan pada transaksi pengeluaran kas, penjualan, piutang dan pengiriman berada pada halaman ini. Laporan ini berfungsi untuk memberikan informasi kepada pengguna terkait data jurnal umum yang tercatat. Berikut adalah implementasi menampilkan laporan jurnal umum:



GAMBAR 4- 32
Implementasi Proses Menampilkan Laporan Jurnal Umum

16. Implementasi laporan buku besar

Berikut merupakan tampilan untuk menampilkan isi data pada laporan buku besar. Pada halaman ini semua data yang berhasil di jurnal berada di halaman ini. Halaman ini berfungsi untuk memberikan informasi kepada pengguna data buku besar yang tercatat. Sebelum masuk ke tampilan buku besar *user* diminta untuk memilih akun yang akan ditampilkan dalam kolom akun. Implementasi menampilkan laporan buku besar sebagai berikut:

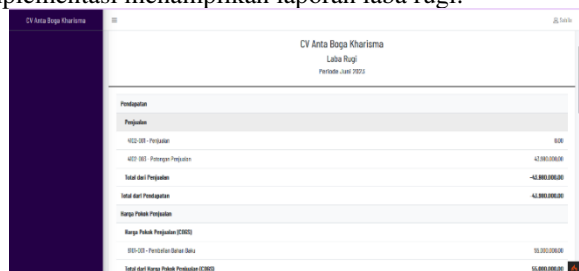


GAMBAR 4- 33
Implementasi Proses Menampilkan Buku Besar

17. Implementasi laporan laba rugi

Berikut merupakan tampilan isi data pada laporan laba rugi. Pada halaman ini semua data telah diinputkan. Laporan

ini berfungsi untuk memberikan informasi kepada pengguna terkait data laporan laba rugi yang tercatat. Berikut adalah implementasi menampilkan laporan laba rugi:



CV Anta Boga Kharisma	
Laba Rugi	
Periode: Jan 2023 - Jun 2023	
Penghasilan	
Penghasilan	
42,000,000.00	
42,000,000.00	
Total dari Penghasilan	42,000,000.00
Pengeluaran	
42,000,000.00	
Total dari Pengeluaran	42,000,000.00
Rugi Bersih	
0.00	
Total dari Rugi Bersih	0.00

GAMBAR 4- 34

Implementasi Proses Menampilkan Laporan Laba Rugi

V. KESIMPULAN

A. Kesimpulan

Dari pembahasan yang telah dijabarkan pada bab-bab sebelumnya maka dapat ditarik sebuah kesimpulan untuk perangkat lunak pada proyek akhir yang berjudul aplikasi berbasis web untuk pengelolaan rantai pasok kue kering ini sebagai berikut:

1. Aplikasi pada proyek akhir ini dapat melakukan transaksi rekap data produksi, pengeluaran kas, penjualan, pengiriman,
2. Aplikasi pada proyek akhir ini mampu melakukan jurnal umum, buku besar, dan laporan laba rugi secara otomatis,

B. Saran

Adapun saran yang dapat diberikan penulis untuk aplikasi ini adalah:

1. Perlunya menambahkan fitur untuk pencatatan persediaan kartu stok
2. Perlunya menambahkan fitur untuk pencatatan harga pokok persediaan

VI. REFERENSI

- [1] N. Ahmad, "Software Development Life Cycle," in *Analisa & Perancangan Sistem Informasi Berorientasi Objek*, Bandung, Widina Media Utama, 2022, p. 42.