

WATERFALL SEBAGAI MODEL PENGEMBANGAN SISTEM PERSEDIAAN APOTEK BERORIENTASI OBJEK

Muhamad Syarif

Program Studi Sistem Informasi, Universitas Bina Sarana Informatika PSDKU Kota Pontianak
Jl. Abdurrahman Saleh No. 18A, Kota Pontianak, Kalimantan Barat
muhammad.mdx@bsi.ac.id

Abstract - The rapid development and growth of information technology has a positive impact on companies engaged in industry, sales and services. One form of information system that is commonly developed is web-based. Inventories are considered important to be managed properly and must follow a policy pattern in accordance with consumer demand so that inventory can be optimal so that it does not result in losses due to expired products or products that are not available when consumers want to buy them, meaning that they can meet the demands of existing demands without carrying too much storage. big. Pharmacy is a business engaged in the sale of medicines and health products as well as health services. Inventory management processes at pharmacies are still many who manage their data using recordings on physical media (books) or using Ms. Excel only on every purchase and sale transaction. Based on the importance of inventory management and the problems that occur in the conventional recording process, inventory management at pharmacies should switch to computerized management or digitally stored by developing an inventory system. Computerized data management can certainly produce reports that are faster and structured and reduce the possibility of human error, therefore this study will develop a pharmacy inventory system that utilizes the Waterfall software development model, the system developed will be object-oriented.

Keywords - Pharmacy, Pharmacy Inventory, Waterfall, Object Orientation.

Abstrak - Pesatnya perkembangan dan pertumbuhan teknologi informasi berdampak positif bagi perusahaan yang bergerak dibidang industri, penjualan dan jasa. Salah satu bentuk sistem informasi yang umum dikembangkan adalah berbasis web. Persediaan dianggap penting untuk dikelola dengan baik dan harus mengikuti pola kebijakan sesuai dengan permintaan konsumen agar persediaan dapat optimal tidak mengakibatkan kerugian karena produk yang telah expired atau produk yang tidak tersedia ketika konsumen ingin membelinya, artinya dapat memenuhi tuntutan permintaan yang ada tanpa melakukan penyimpanan yang terlalu besar. Apotek adalah usaha yang bergerak dibidang penjualan obat-obatan dan produk kesehatan serta pelayanan kesehatan. Proses pengelolaan persediaan pada Apotek saat ini masih banyak yang mengelola datanya menggunakan pencatatan di media fisik (buku) atau menggunakan Ms. Excel saja pada setiap transaksi pembelian maupun penjualannya. Berdasarkan pentingnya pengelolaan persediaan dan permasalahan yang terjadi pada proses pencatatan konvensional tersebut, maka sudah seharusnya pengelolaan persediaan pada Apotek dapat beralih menjadi pengelolaan secara terkomputerisasi atau tersimpan secara digital dengan mengembangkan sistem persediaan. Pengelolaan data secara terkomputerisasi tentu dapat menghasilkan laporan yang lebih cepat dan terstruktur serta mengurangi kemungkinan terjadinya kesalahan manusia, oleh sebab itu penelitian ini akan mengembangkan sebuah sistem persediaan apotek yang memanfaatkan model pengembangan perangkat lunak Waterfall sistem yang dikembangkan akan berorientasi objek.

Kata Kunci - Apotek, Persediaan Apotek, Waterfall, Orientasi Objek.

I. PENDAHULUAN

Perkembangan sistem informasi sekarang sangat cepat dan pesat, tidak sedikit yang menggunakan sistem informasi untuk membantu kemudahan dalam bekerja. Salah satu bentuk sistem informasi yang mudah dikembangkan adalah berbasis web. Sistem informasi berbasis web tidak hanya digunakan untuk menampilkan informasi saja, namun dapat digunakan untuk berdialog dengan data sehingga memberikan informasi untuk mengambil sebuah keputusan manajerial[1].

Salah satu kebutuhan yang diperlukan dalam mengelola suatu usaha adalah persediaan. Persediaan dianggap penting untuk dikelola dengan baik dan

harus mengikuti pola kebijakan sesuai dengan permintaan konsumen agar persediaan dapat optimal tidak mengakibatkan kerugian karena produk yang telah expired atau produk yang tidak tersedia ketika konsumen ingin membelinya, artinya dapat memenuhi tuntutan permintaan yang ada tanpa melakukan penyimpanan yang terlalu besar. Kesalahan dalam pengelolaan persediaan dapat menyebabkan penumpukan pada penyimpanan yang berakibat pada tingginya biaya simpan dan kemungkinan terjadinya kerusakan produk semakin besar[2].

Apotek merupakan usaha yang bergerak dibidang penjualan obat-obatan dan produk kesehatan serta pelayanan kesehatan. Setiap kali adanya konsumen yang datang untuk membeli selalu menanyakan

ketersediaan obat yang diinginkan, dan proses pencarian ini tentunya memakan waktu serta terus berulang, sehingga proses ini harus diimbangkan dengan pengelolaan persediaan yang baik pula agar dapat memberikan informasi dengan cepat dan akurat. Proses pengelolaan persediaan pada apotek masih banyak yang memanfaatkan pada media fisik (buku) saja pada setiap transaksi pembelian obat ke PBF (Perusahaan Besar Farmasi) maupun penjualan obat ke konsumen.

Proses pencatatan tersebut sering kali mengakibatkan terjadinya beberapa permasalahan, permasalahan tersebut seperti lamanya proses pencarian ketersediaan obat dan harga obat. Permasalahan juga terjadi ketika proses pemeriksaan persediaan produk, produk yang rusak, dan expired perlu dicatat secara berkala sehingga mengurangi resiko pemberian obat expired ke konsumen. Dan dengan adanya catatan persediaan yang baik, maka dapat mendukung manajerial dalam membuat keputusan obat mana saja yang perlu disediakan dengan jumlah yang banyak ataupun jumlah yang sedikit.

Dalam proses pengembangan sistem informasi baik berbasis website maupun desktop, pengembang perlu memperhatikan metode yang tepat sesuai kebutuhan proses pengembangan, salah satunya adalah model Waterfall, dimana model ini memiliki 5 tahap yaitu Analisis kebutuhan perangkat lunak, Desain, Pembuatan Kode Program, Pengujian, dan Pendukung atau Pemeliharaan. Perlu untuk memperhatikan detail pada setiap tahapan karena perulangan pada tahap pengembangan akan berdampak pada kebutuhan waktu yang semakin lama dan biaya yang semakin besar. Salah satu tahap yang perlu dikerjakan dengan baik agar tidak terlalu berdampak ketika adanya kesalahan dalam analisa kebutuhan yaitu tahapan desain. Tahap desain dapat digambarkan menggunakan diagram, yaitu diagram basis data dan diagram pemodelan sistem.

Diagram UML adalah salah satu diagram yang sering digunakan untuk memberikan gambaran pemodelan berbasis objek, diagram ini dapat mempermudah bagian *programmer* dan *stakeholder* dalam memahami rancangan dari sistem yang akan dikembangkan nantinya, sehingga dapat mengurangi kemungkinan terjadinya kesalahan dalam proses pengembangan.

A. Sistem Informasi

Sistem informasi merupakan kumpulan sub-subsistem yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan, yaitu mengelola data menjadi informasi yang berguna bagi penerimanya[1].

Sistem informasi merupakan suatu kombinasi teratur dari orang-orang, *hardware*, *software*, jaringan komunikasi dan sumber daya data yang dapat melakukan pengelolaan data seperti menambahkan

data, mengubah data, dan menyebarkan informasi dalam sebuah organisasi[3].

B. Model Pengembangan Perangkat Lunak Waterfall

Model *Waterfall* merupakan salah satu metode pengembangan perangkat lunak yang tahapannya digambarkan seperti air terjun yang menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau berurutan dimulai dari analisa, desain, pengkodean, pengujian, dan pendukung[4].

Metode *Waterfall* ini sebenarnya adalah *linear sequential model*, yang sering disebut dengan *classic life cycle*. Disebut dengan *Waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan[5].

C. Framework CodeIgniter

Framework adalah komponen pemrograman yang siap digunakan ulang kapan saja sehingga *programmer* tidak harus membuat script yang sama untuk tugas yang sama[6]. *Framework* adalah kumpulan intruksi-intruksi yang dikumpulkan dalam *class* dan *function-function* dengan fungsi masing-masing untuk memudahkan *developer* dalam memanggilnya tanpa harus menuliskan *syntax* program yang sama berulang-ulang serta dapat menghemat waktu[7].

CodeIgniter adalah sebuah aplikasi *open source* yang berupa kerangka kerja atau *framework* untuk membangun website menggunakan bahasa pemrograman PHP[8]. CodeIgniter adalah sebuah *framework* PHP yang bersifat *open source* dan menggunakan metode MVC (*Model, View, Controller*) untuk memudahkan *developer* atau *programmer* dalam membangun sebuah aplikasi berbasis web tanpa harus membuatnya dari awal[7].

D. Diagram Unified Modeling Language (UML)

UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Sukanto dan Shalahuddin dalam[9].

UML singkatan dari *Unified Modelling Language* sebagai bahasa, berarti UML memiliki sintaks dan semantik. Saat kita membuat suatu model menggunakan konsep *Unified Modelling Language* ada berbagai aturan yang harus diikuti. Bagaimana elemen-elemen pada model-model yang dibuat berhubungan satu dengan yang lainnya harus sesuai dengan standar[10].

E. Object Oriented Programming (OOP)

Menurut Hermawan, Pemrograman berorientasi objek atau *object oriented programming* (OOP) merupakan suatu pendekatan pemrograman yang menggunakan *object* dan *class*. OOP memberikan kemudahan dalam pembuatan sebuah program, keuntungan yang didapat seperti *reliable*, *reusable*, *maintainable* dan *extendable*[11].

II. METODE PENELITIAN

Metode penelitian yang digunakan yaitu metode deskriptif yang berfungsi mengumpulkan data secara sistematis. Metode penelitian deskriptif merupakan metode penelitian yang berusaha menggambarkan dan menginterpretasi objek sesuai dengan apa adanya. Metode penelitian deskriptif juga biasa disebut dengan metode non-eksperimen, karena pada penelitian ini tidak melakukan kontrol manipulasi variabel penelitian[12].

Metode penelitian deskriptif ini didukung dengan metode pengembangan perangkat lunak model *Waterfall*, berikut tahapan dari penelitian yang dilakukan.

1. Analisis kebutuhan perangkat lunak
 Dalam tahapan ini, penulis melakukan analisis kebutuhan yang diperlukan dalam pembuatan sistem informasi persediaan pada Apotek berbasis web, dimulai dari analisis kebutuhan fungsional (kebutuhan fungsi sistem yang didapat oleh setiap pengguna), analisis kebutuhan non-fungsional (kebutuhan pendukung sistem, seperti *hardware* yang digunakan dan *software* pendukung) dengan menggunakan teknik pengumpulan data.
2. Desain
 Pada tahapan desain ini menggunakan diagram *Unified Modelling Language* (UML) yang terdiri dari *usecase diagram*, *activity diagram*, *class sdiagram*, *sequence diagram*, *component diagram* dan *deployment diagram*.
3. Pembuatan kode program
 Bahasa pemrograman yang digunakan adalah *hypertext markup language* (HTML), *hypertext preprocessor* (PHP), *Bootstrap*, *cascading style sheets* (CSS), *javascript* dan *jQuery*, *CodeIgniter* sebagai *framework* PHP.
4. Pengujian
 Pengujian dilakukan untuk mengetahui hasil dari program yang telah dibuat. Teknik pengujian yang dilakukan adalah dengan menggunakan metode *blackbox testing* atau pengujian yang berfokus pada fungsionalitas sistem.
5. Pendukung (*support*) atau pemeliharaan (*maintenance*)
 Melakukan perawatan pada perangkat lunak dan melakukan perubahan jika merasa ada sesuatu yang masih ingin dikembangkan pada perangkat lunak.

III. HASIL DAN PEMBAHASAN

Dari penelitian yang telah dilakukan maka didapat hasil analisa kebutuhan dan implementasi sistem informasi berbasis *webseite* yang akan dibahas sebagai berikut :

A. Analisa Kebutuhan Software

Analisa kebutuhan terfokus pada kebutuhan sistem yang akan dibuat untuk Apotek, khususnya sistem persediaan. Kebutuhan pengguna akan terbagi menjadi 3 (Pemilik, Persediaan, dan Penjualan) level akses, adapun masing-masing akses dapat melakukan sebagai berikut:

Tabel 1. Analisa Kebutuhan Software

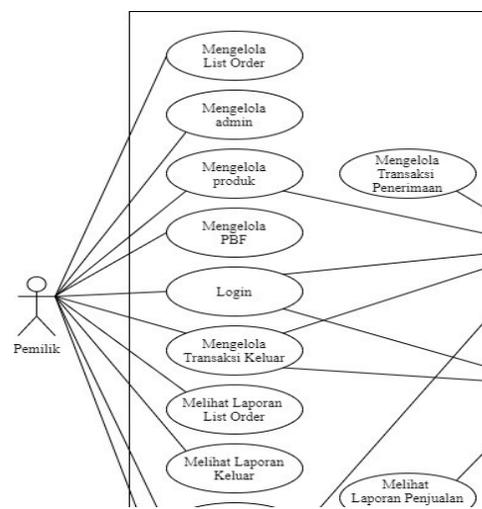
No	Pengguna	Kebutuhan
1	Pemilik	<ul style="list-style-type: none"> • Mengelola admin • Mengelola produk • Mengelola mengelola pedagang besar farmasi (PBF) • Mengelola list order • Mengelola transaksi keluar • Melihat laporan transaksi list order • Melihat laporan transaksi keluar • Melihat laporan jurnal • Melihat persediaan produk
2	Bagian Persediaan	<ul style="list-style-type: none"> • Mengelola produk • Mengelola transaksi penerimaan list order • Mengelola transaksi keluar • Melihat laporan persediaan produk
3	Bagian Penjualan	<ul style="list-style-type: none"> • Mengelola transaksi keluar, • Melihat laporan penjualan harian.

B. Mengembangkan Sistem Berorientasi Objek

Pengembangan sistem pada penelitian ini dimulai dengan pembuatan UML, berikut adalah beberapa komponen UML yang telah penulis buat.

1. Use Case Diagram

Perancangan model dengan *use case diagram* menggambarkan perilaku sistem yang dibuat. Berikut ini adalah hasil rancangan *use case diagram* yang terdiri dari 3 aktor.

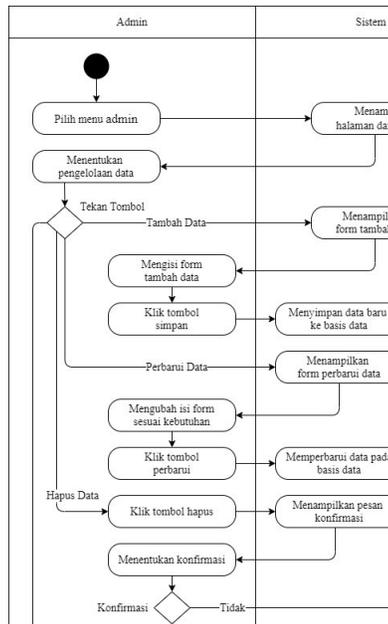


Gambar 1. Use Case Diagram Sistem Persediaan

Use case diagram pada gambar 1 terdiri dari 3 aktor, masing-masing aktor memiliki akses yang berbeda berdasarkan kebutuhan dan tanggung jawab masing-masing.

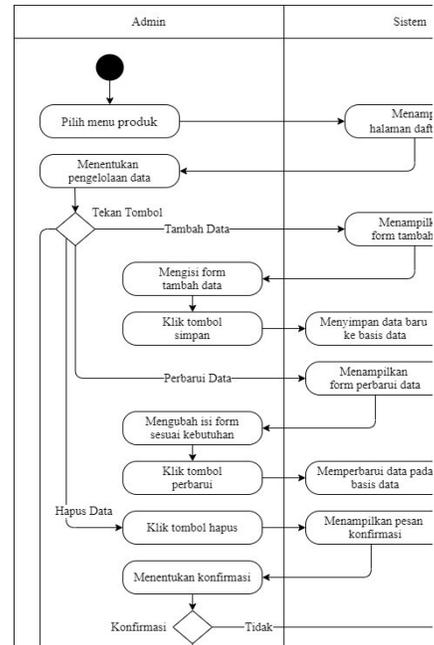
2. Activity Diagram

Untuk memberikan gambaran visual berkaitan dengan alur sistem yang dibangun, maka pada subbab ini akan digambarkan dengan activity diagram. Berikut ini adalah hasil rancangan activity diagram pada sistem persediaan yang dikembangkan untuk Apotek.



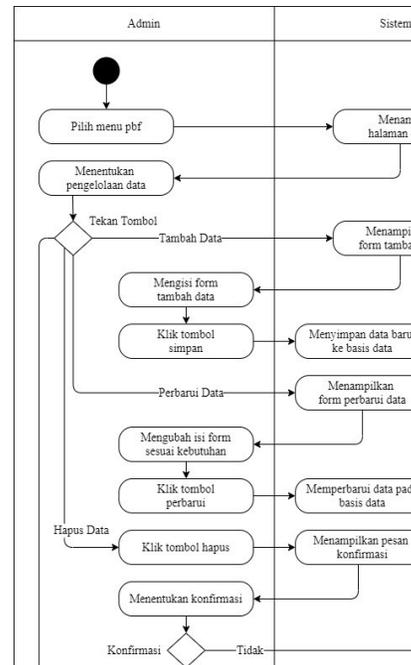
Gambar 2. Activity Diagram Mengelola Admin

Dari gambar 2 di atas Activity diagram mengelola admin menjelaskan proses pemilik saat menambahkan data baru atau melakukan perubahan data karyawan yang diberikan akses masuk kedalam sistem.



Gambar 3. Activity Diagram Mengelola Produk

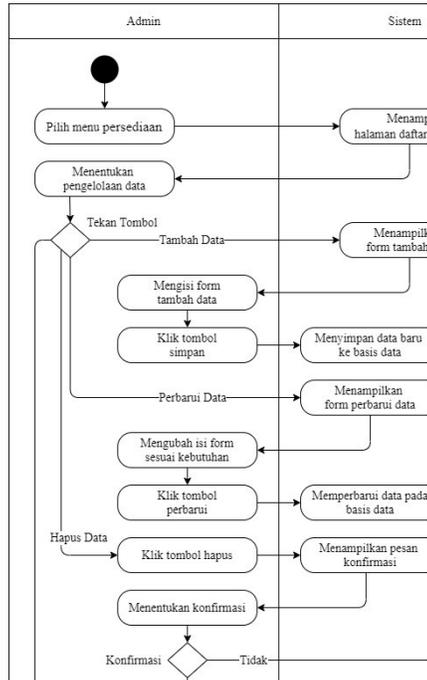
Dari gambar 3 di atas Activity diagram mengelola produk menjelaskan proses aktor yang mempunyai akses untuk menambahkan, mengubah maupun menghapus data produk yang ada didalam sistem.



Gambar 4. Activity Diagram Pedagang Besar Farmasi (PBF)

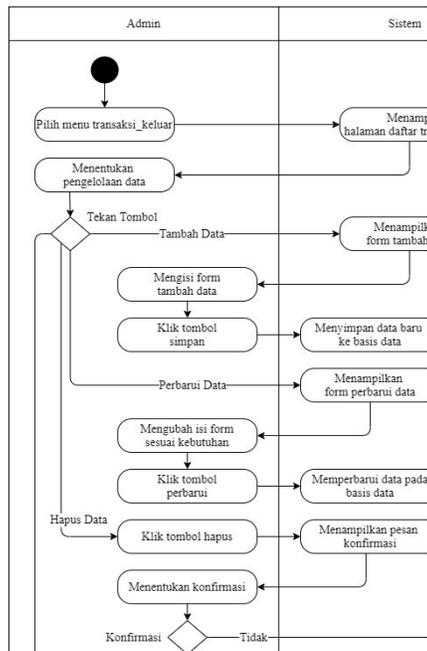
Dari gambar 4 di atas Activity diagram Pedagang Besar Farmasi (PBF) menjelaskan proses aktor yang mempunyai akses untuk menambahkan, mengubah

maupun menghapus data PBF yang ada didalam sistem.



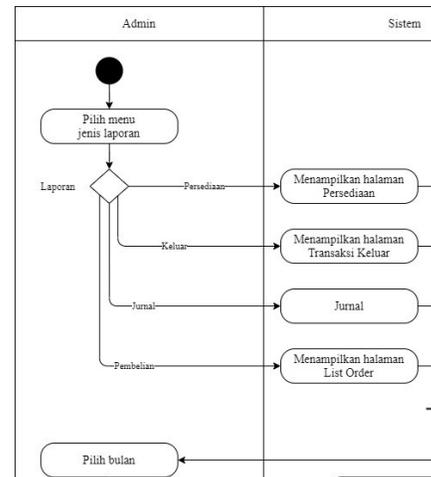
Gambar 5. Activity Diagram Mengelola Persediaan

Dari gambar 5 di atas Activity diagram mengelola persediaan menjelaskan proses program dalam melihat maupun mengelola produk yang dijual.



Gambar 6. Activity Diagram Mengelola Transaksi

Dari gambar 6 di atas Activity diagram mengelola transaksi menjelaskan proses program dalam mengelola transaksi jual beli.

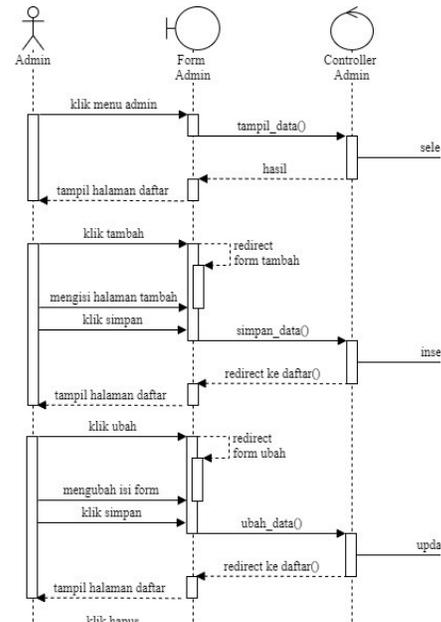


Gambar 7. Activity Diagram Melihat Laporan

Dari gambar 7 di atas Activity diagram melihat laporan menjelaskan proses bagaimana aktor yang mempunyai akses dapat melihat laporan transaksi berdasarkan periode tertentu.

3. Sequence Diagram

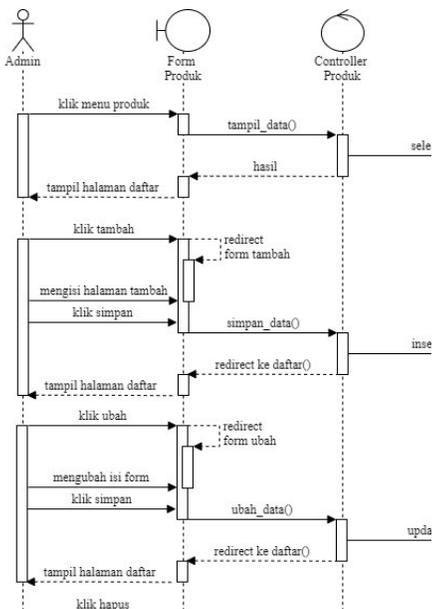
Berikut ini adalah hasil sequence diagram yang dibuat untuk memberikan gambaran tentang interaksi objek dan menunjukkan (memberi tanda atau petunjuk) komunikasi diantara objek-objek tersebut.



Gambar 8. Sequence Diagram Mengelola Admin

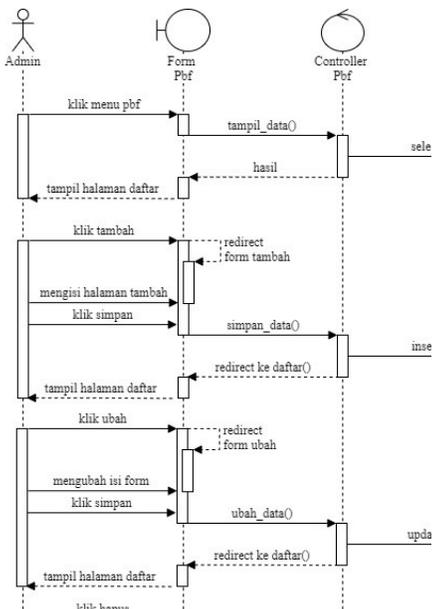
Dari gambar 8 di atas dapat kita lihat proses dari halaman admin. Adapun beberapa fungsi yang telah

disiapkan antara lain menambah data admin baru, mengubah data admin yang sudah terdaftar bahkan menghapus data admin yang sudah tidak terpakai lagi.



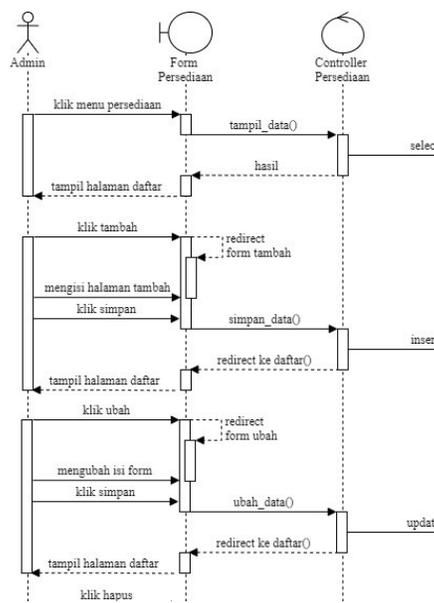
Gambar 9. Sequence Diagram Mengelola Produk

Dari gambar 9 dapat kita lihat bahwa pada halaman produk ini bisa kita manfaatkan untuk mengolah seluruh data produk yang dijual oleh apotek.



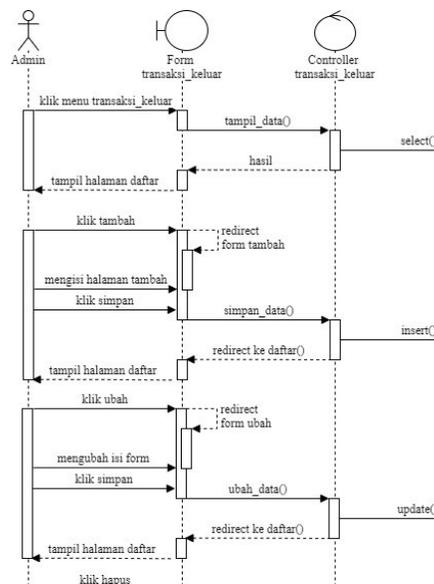
Gambar 10. Sequence Diagram Pedagang Besar Farmasi (PBF)

Dari gambar 10 di atas dapat kita lihat sistem ini sudah mempersiapkan laman untuk menampung data Pedagang Besar Farmasi (PBF) atau bias kita bilang *supplier* produk yang bergerak dibidang kesehatan.



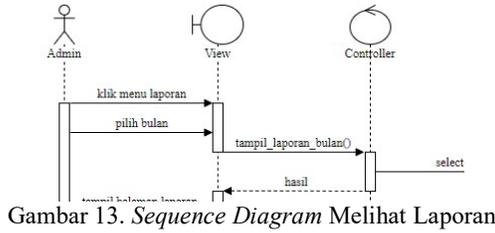
Gambar 11. Sequence Diagram Mengelola Persediaan

Dari gambar 11 di atas adalah fungsi yang dipersiapkan untuk melihat seluruh transaksi produk baik itu informasi stok produk, produk yang terjual, produk yang rusak maupun produk yang diretur.



Gambar 12. Sequence Diagram Mengelola Transaksi

Dari gambar 12 ini dapat dilihat bahwa sistem sudah dipersiapkan hingga proses transaksi penjualan. Harapannya agar petugas apotek tidak perlu repot lagi mencari data produk yang dijual masih tersedia atau sudah habis, karena dengan adanya fitur transaksi ini sistem dapat memberikan informasi secara langsung apakah obat yang diinginkan sudah habis atau masih tersedia.

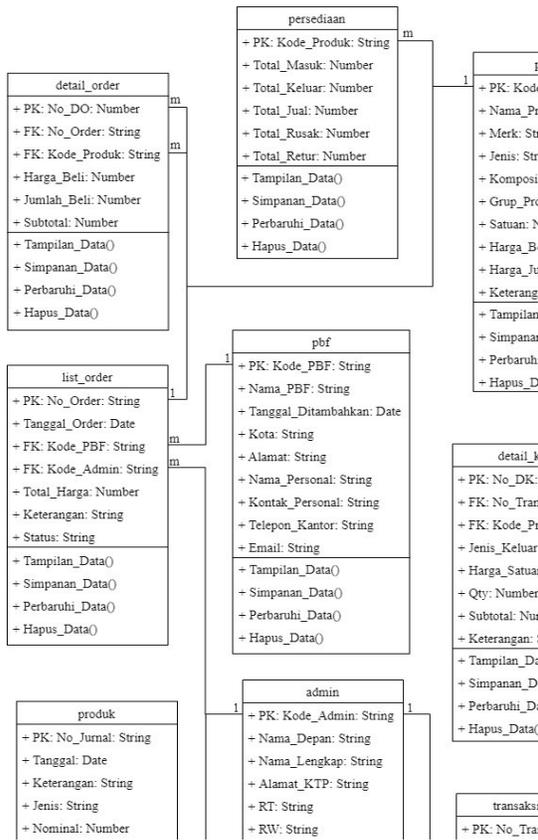


Gambar 13. Sequence Diagram Melihat Laporan

Tentu saja tujuan akhir dari kebutuhan pengguna pada sistem informasi adalah terdapatnya laporan yang bisa dihasilkan secara otomatis, cepat dan akurat. Dapat dilihat pada gambar 13 bahwa sistem ini sudah tersedia fitur mencetak laporan otomatis.

4. Class Diagram

Berikut ini adalah hasil class diagram yang dibuat berdasarkan analisa yang dilakukan pada tahapan sebelumnya, terdapat 9 class yang berbeda dan saling berhubungan.



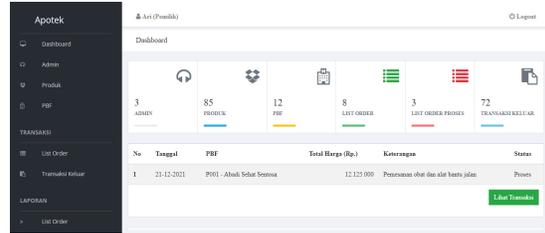
Gambar 14. Class Diagram

C. Implementasi Pada Aplikasi

Dalam implementasi pada aplikasi dapat dilihat sebagai berikut.

1. Halaman Dashboard

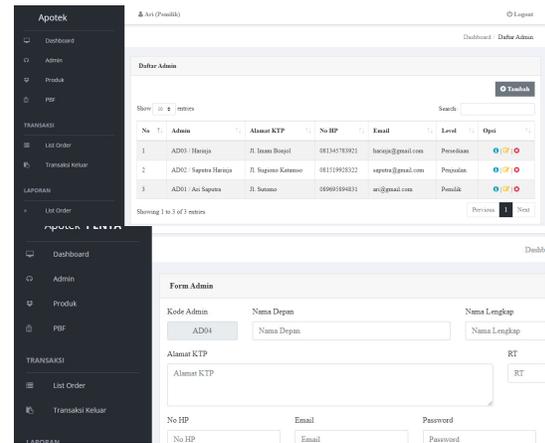
Ini merupakan halaman utama setelah kita berhasil masuk kedalam sistem.



Gambar 15. User Interface Dashboard

2. Halaman Admin

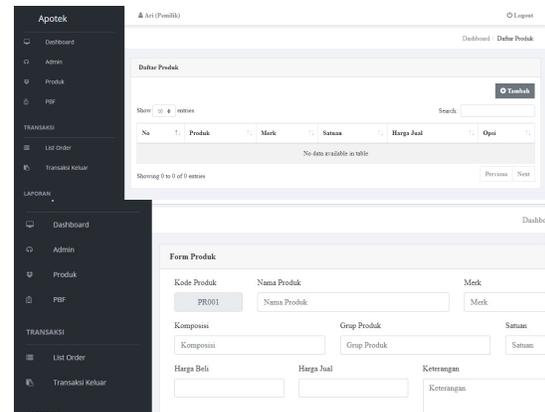
Halaman admin digunakan untuk mengelola siapa saja yang dapat masuk kedalam sistem, halaman admin hanya dapat diakses oleh pimpinan.



Gambar 16. User Interface Admin

3. Halaman Produk

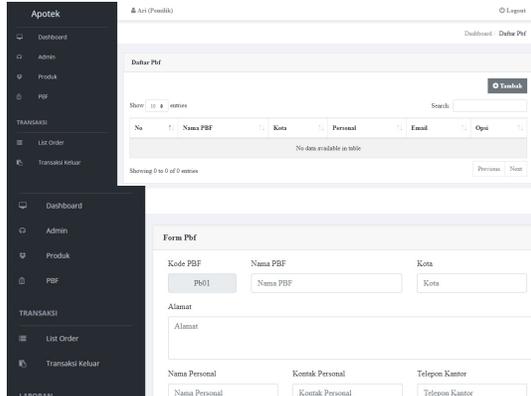
Halaman produk digunakan untuk mengelola produk apa saja yang dijual oleh apotek.



Gambar 17. User Interface Produk

4. Halaman Pedagang Besar Farmasi (PBF)

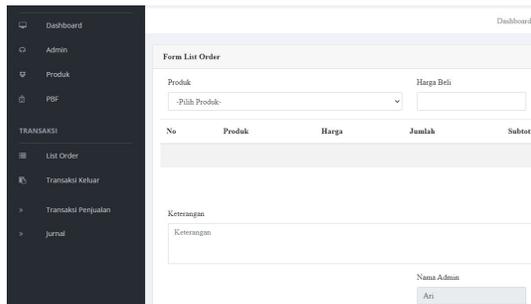
Halaman ini digunakan untuk mengelola data supplier yang berkerjasama dengan apotek.



Gambar 18. User Interface Pedagang Besar Farmasi (PBF)

5. Halaman List Order

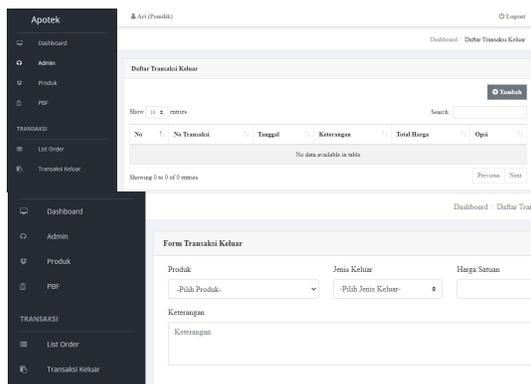
Halaman list order dimanfaatkan untuk membuat data kebutuhan obat yang akan ditambah kembali.



Gambar 19. User Interface List Order

6. Halaman Transaksi

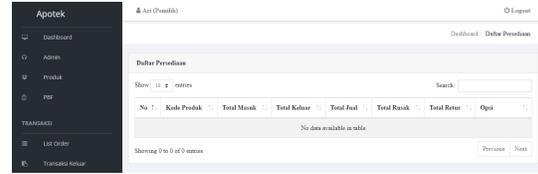
Halaman transaksi akan sering digunakan karena transaksi penjualan produk akan terpusat pada halaman ini.



Gambar 20. User Interface Transaksi

7. Halaman Persediaan

Halaman persediaan dapat digunakan untuk melihat total produk yang ada, produk yang terjual, produk yang rusak dan lain sebagainya.



Gambar 21. User Interface Persediaan

D. Pengujian Sistem

Untuk mencari tahu tentang kesalahan terhadap fungsional aplikasi yang dibuat, maka penulis menguji aplikasi ini dengan menggunakan pengujian metode *blackbox testing*, dimana halaman yang diuji untuk halaman transaksi inti saja, seperti halaman login, halaman admin dan halaman transaksi keluar.

1. Pengujian Terhadap Form Login

Tabel 1. Hasil *Blackbox Testing* Halaman Login

Skenario pengujian	Hasil yang diharapkan	Hasil pengujian
Email dan Password tidak diisi kemudian klik tombol login	Sistem akan menolak akses pengguna dan menampilkan pesan untuk mengisi Email dan Password	Sesuai harapan
Mengetikkan Email dan Password tidak diisi kemudian klik tombol login	Sistem akan menolak akses pengguna dan menampilkan "Masukan Password"	Sesuai harapan
Email tidak diisi dan Password diisi kemudian klik tombol login	Sistem akan menolak akses pengguna dan menampilkan "Masukan Email"	Sesuai harapan
Mengetikkan salah satu kondisi salah pada Email atau Password kemudian klik tombol login	Sistem akan menolak akses pengguna dan menampilkan "Email atau Password yang anda masukan salah"	Sesuai harapan

2. Pengujian Terhadap Form Admin

Tabel 2. Hasil *Blackbox Testing* Halaman Admin

Skenario pengujian	Hasil yang diharapkan	Hasil pengujian
Mengisi salah satu form kemudian klik tombol simpan.	Tombol Simpan di klik maka akan muncul pesan untuk mengisi data yang masih kosong.	Sesuai harapan
Mengisi semua form kemudian klik tombol simpan.	Data bisa disimpan, dan muncul notifikasi "Berhasil simpan data".	Sesuai harapan

3. Pengujian Terhadap Form Transaksi Keluar
Tabel 3. Hasil *Blackbox Testing* Halaman Transaksi Keluar

Skenario pengujian	Hasil yang diharapkan	Hasil pengujian
Mengisi salah satu form kemudian klik tombol simpan.	Tombol simpan di klik maka akan muncul pesan untuk mengisi data yang masih kosong.	Sesuai harapan
Mengisi semua form kemudian klik tombol simpan.	Data bisa disimpan, dan muncul notifikasi "Berhasil Simpan Data".	Sesuai harapan

IV. KESIMPULAN

Berdasarkan penelitian, implementasi dan pengujian, maka dapat diambil kesimpulan sebagai berikut :

1. Dengan memanfaatkan sistem informasi persediaan apotek yang terkomputerisasi maka proses pengelolaan dan penyimpanan data disimpan pada *database* sehingga sistem dapat mengolah data tersebut menjadi laporan yang dapat diakses dengan waktu yang singkat dan akurat.
2. Penerapan model Waterfall dan mengikuti tahapannya dapat memudahkan pengembang mengkomunikasikan tiap tahapannya kepada *stakeholder* dan tim *project* selama proses pengembangan sehingga minimnya kesalahan yang mengurangi resiko kegagalan *project*.
3. Sistem yang dikembangkan berbasis objek dibantu dengan menerapkan framework CodeIgniter, sehingga proses pengkodean menjadi lebih cepat dan bahasa yang digunakan mudah dipahami antar tim karena sudah terstandarkan oleh aturan-aturan CodeIgniter.

DAFTAR PUSTAKA

[1] W. Nugraha And M. Syarif, "Penerapan Metode Prototype Dalam Perancangan Sistem Informasi Penghitungan Volume Dan Cost Penjualan Minuman Berbasis Website," *Jusim (Jurnal Sist. Inf. Musirawas)*, Vol. 3, No. 2, Pp. 94-101, 2018, Doi: 10.32767/Jusim.V3i2.331.

[2] O. Irnawati And G. B. A. Listianto, "Metode Rapid Application Development (Rad) Pada Perancangan Website Inventory Pt. Sarana Abadi Makmur Bersama (S.A.M.B) Jakarta," *Evolusi*, Vol. 6, No. 2, Pp. 57-65, 2018.

[3] E. Y. Anggraeni And R. Irviani, *Pengantar Sistem Informasi*. Yogyakarta: Cv Andi Offset, 2017.

[4] R. A. Sukamto And M. Shalahuddin, *Kolaborasi Rekayasa Perangkat Lunak Terstruktur Dan Berorientasi Objek*. Bandung: Informatika, 2018.

[5] Muharto And A. Ambarita, *Metode Penelitian Sistem Informasi*. Yogyakarta: Deeppublish Publisher, 2016.

[6] Y. Yudhanto And H. A. Prasetio, *Panduan Mudah Belajar Framework Laraval*. Jakarta: Pt. Elex Media Komputindo, 2018.

[7] M. Destiningrum And Q. J. Adrian, "Sistem Informasi Penjadwalan Dokter Berbassis Web Dengan Menggunakan Framework Codeigniter (Studi Kasus: Rumah Sakit Yukum Medical Centre)," *J. Teknoinfo*, Vol. 11, No. 2, P. 30, 2017, Doi: 10.33365/Jti.V11i2.24.

[8] H. Sulistiono, "Cara Mudah Dengan Codeigniter, JQuery, Bootstrap Dan Datatable," 2018.

[9] M. Syarif And W. Nugraha, "Pemodelan Diagram Uml Sistem Pembayaran Tunai Pada Transaksi E-Commerce," *J. Tek. Inform. Kaputama*, Vol. 4, No. 1, P. 70 Halaman, 2020, [Online]. Available: [Http://jurnal.kaputama.ac.id/index.php/jtik/article/view/240](http://jurnal.kaputama.ac.id/index.php/jtik/article/view/240).

[10] Y. Fitriani, R. Pakpahan, And Adrianichsan, "Sistem Informasi Forum Diskusi Online Antar Mahasiswa Pada Smartphone Jisamar (Journal Of Information System , Applied , Management , Accounting And Research) P-Issn : 2598-8700 (Printed) Jisamar (Journal Of Information System , Applied , Management ," Vol. 4, No. 4, Pp. 92-101, 2020.

[11] D. Wahyuningtyas, Solikin, And E. Retnoningsih, "Sistem Informasi Akademik Nilai Berbasis Web Pada Mts Perguruan Islam Nurul Kasysyaf (Pink) 03 Bekasi," *Inf. Syst. Educ. Prof.*, Vol. 2, No. 2, Pp. 191-200, 2018.

[12] Wagiran, *Metode Penelitian Pendidikan (Teori Dan Implementasi)*. Yogyakarta: Deeppublish Publisher, 2019.