

Pengembangan Sistem Informasi Pengelolaan Persediaan Barang Dengan *ReactJS* Dan *React Native* Menggunakan *Prototype* (Studi Kasus : Toko Uda Fajri)

Faisal Fajar Nursaid¹, Adam Hendra Brata², Agi Putra Kharisma³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya

Email: ¹faisalfns14@gmail.com, ²adam@ub.ac.id, ³agi@ub.ac.id

Abstrak

Toko Uda Fajri merupakan toko yang menyediakan bahan dalam membuat kue, yang terletak di pasar pejuang, bekasi barat. Pada toko ini pengelolaan barang yang dilakukan masih manual dan belum terjangkau teknologi. Dari segi pencatatan barang masuk dan saat melakukan transaksi. Lalu untuk mengatasi masalah tersebut dikembangkan "Sistem Informasi Pengelolaan Persediaan Barang dengan *ReactJs* dan *React Native* Menggunakan *Prototype*". Pada pengembangan sistem ini menggunakan *framework react js* untuk mengembangkan *website* dan *framework react native* untuk mengembangkan aplikasi *mobile*. Pengembangan sistem ini menggunakan *SDLC prototyping*. Pengembangan sistem ini dimulai dengan tahapan menggali studi literatur, analisis kebutuhan, perancangan, implementasi, pengujian dan pengambilan kesimpulan. Dari hasil analisis kebutuhan diperoleh 2 aktor yang terlibat didalam sistem, yaitu admin dan operator. Lalu pada tahapan analisis kebutuhan dan perancangan juga dibangun *Unified Modeling Language* atau UML berupa *use case diagram* yang didalamnya terdapat *use case scenario*, *sequence diagram*, dan *class diagram*. Selanjutnya pada tahapan implementasi sistem dilakukan dengan menggunakan kerangka kerja *reactjs* dan *react native*. Setelah implementasi selesai system yang dikembangkan kemudian diuji dengan pengujian *whitebox* untuk pengujian *unit*, pada pengujian *unit* ini yang diuji hanya 4 *unit* utama saja dan menghasilkan hasil yang valid. Sedangkan pada pengujian *blackbox* digunakan metode pengujian validasi dan menghasilkan 100% data valid dari 15 kasus uji. Dari pengujian yang sudah dilakukan menggambarkan bahwa sistem dapat berjalan dengan baik dan sesuai dengan kebutuhan yang telah ditentukan sebelumnya. Lalu pada pengujian *User Acceptance Testing* (UAT) menggunakan instrumen kuesioner dan metode skala *likert*, yang mana terdapat 4 tahap pengujian UAT dari tahap pengujian reaksi pengguna terhadap sistem yang memiliki persentase 80%, reaksi pengguna terhadap antarmuka memiliki persentase 68%, reaksi pengguna saat mengoperasikan sistem memiliki persentase 96%, dan reaksi pengguna terhadap fungsionalitas sistem memiliki persentase 76%.

Kata kunci: *ReactJs*, *React Native*, *Prototyping*, UML, UAT

Abstract

Uda Fajri store is a shop that provides ingredients in making cakes, which is located in the warrior market, west bekasi. In this shop the management of goods carried out is still manual and not yet affordable by technology. In terms of recording incoming goods and when making transactions. Then to overcome this problem the development of "Inventory Management Information System with ReactJs and React Native Using Prototype". In the development of this system uses the react reacts framework to develop websites and native react frameworks to develop mobile applications. The development of this system uses SDLC prototyping. The development of this system begins with the stages of exploring the study of literature, needs analysis, design, implementation, testing and conclusion. From the results of the needs analysis, 1 actor involved in the system is obtained, namely the admin. Then in the needs analysis and design stages also built a Unified Modeling Language or UML in the form of use case diagrams in which there are use case scenarios, sequence diagrams, and class diagrams. Then the system implementation stage is carried out using the reactjs and react native frameworks. After the implementation is complete the developed system is then tested with whitebox testing for unit testing, in this unit testing only 4 main units are tested and produce valid results. Whereas the blackbox testing method is used for validation testing and produces 100% valid data from 15 test cases. From the tests

that have been done illustrate that the system can run properly and in accordance with predetermined needs. Then in the User Acceptance Testing (UAT) test using a questionnaire instrument and Likert scale method, where there are 4 stages of UAT testing from the stage of testing the user's reaction to the system that has a percentage of 80%, the user's reaction to the interface has a percentage of 68%, the user's reaction when operating the system has a percentage of 96%, and user reactions to system functionality have a percentage of 76%.

Keywords : *ReactJs, React Native, Prototype, UML, UAT*

1. PENDAHULUAN

Pada beberapa tahun terakhir ini sistem informasi sudah mulai berkembang pesat beserta perkembangan teknologi informasi (Liang & Chen, 2003). Menurut (Zamroni, 2017) perkembangan teknologi informasi pada saat ini berawal dari majunya dibidang komputerisasi. Penggunaan komputer pada masa yang mana hanya sekedar menulis, menyimpan data, dan menggambar sudah berkembang menjadi alat komunikasi yang didalamnya terdapat jaringan dan perangkat lunak yang dapat mencakup seluruh dunia. Selain itu menurut (Lee, 2013) perkembangan sistem informasi dalam teknologi informasi memiliki efek yang luas bagi masyarakat, baik dalam sektor komunikasi, keuangan, dan perdagangan.

Sistem informasi menurut (Christanto, 2009) merupakan alat bantu dalam menampilkan informasi kepada semua orang yang membutuhkannya dan dapat mempermudah dalam pengelolaan data maupun informasi. Dan sistem informasi pengelolaan merupakan susunan dari berbagai komponen data maupun informasi. Lalu menurut (Listiani & Wahyuningsih, 2019), bahwa pengelolaan barang merupakan suatu upaya untuk menentukan banyaknya atau tidaknya tingkat persediaan barang serta mengendalikan secara efektif dan efisien. Jadi dapat ditarik kesimpulan dari penelitian terdahulu bahwa sistem informasi pengelolaan dapat memberikan informasi kepada pengguna ketika informasi tersebut sesuai dengan kriteria yang dipilih secara individual oleh pengguna dan membebani pengguna hanya untuk informasi yang dipilih. Sistem informasi pengelolaan barang sangat membantu pengguna dalam menyimpan data dan menyeleksi data.

Dan pada saat ini sudah ditemukannya sebuah masalah dalam mengelola barang dagangan yang ada pada toko UDA FAJRI.

Menurut hasil observasi yang dilakukan pada tanggal 15 Maret 2019 pengelolaan barang dagang yang ada di toko tersebut masih manual dan belum menggunakan komputerisasi, baik itu dari segi transaksi dan pencatatan pemasukan dan pengeluaran barang dengan buku catatan yang harus selalu dibawa. Masalah ini sangat disayangkan dan memiliki resiko, yang mana resiko dalam menggunakan pencatatan kertas tersebut yaitu, catatan dapat hilang kapan saja, sobek dan dapatnya data untuk dimanipulasi.

Untuk mengatasi masalah tersebut akan dikembangkannya sebuah sistem informasi pengelolaan barang untuk toko UDA FAJRI. Tujuan sistem informasi pengelolaan barang ini dikembangkan agar dapat mempermudah pedagang yang memiliki toko tersebut tanpa harus mencatat barang masuk dan keluar dengan menggunakan kertas lagi, selain itu juga dapat melakukan transaksi barang dengan menggunakan telepon genggam yang sudah terpasang aplikasi sistem pengelolaan barang. Lalu data juga akan lebih aman tersimpan didalam database. Pada pengembangan sistem informasi pengelolaan barang ini diharapkan sistem dapat digunakan melalui sebuah pc untuk admin atau operator dalam menambah barang, memasukan jumlah barang, mengubah data barang, dan mengurangi jumlah barang keluar melalui cara pembelian yang ada didalam aplikasi Android, dan juga menampilkan rekap data barang masuk dan keluar. Jadi dengan adanya sistem ini pedagang dapat memantau barang masuk dan keluar.

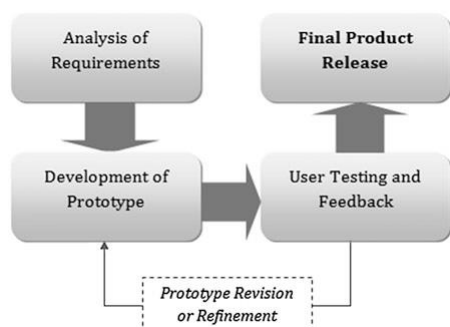
Lalu sistem ini akan dikembangkan dengan arsitektur *MVVM (Model View ViewModel)* menggunakan platform Reactjs untuk web admin dan platform React Native untuk aplikasi yang dijalankan pada device Android, ini dikarenakan *React* memberikan kemudahan dan kesederhanan dalam mengembangkan maupun pembaruan

sistem(Kumar & Singh, 2016). Selain itu pengembangan sistem ini menggunakan *SDLC (Software Development Life Cycle) Prototype*, alasan digunakan menggunakan *prototyping* ini karena dapat mengeksplorasi kebutuhan pengguna untuk memperoleh atau mengklarifikasi persyaratan pengguna dan sangat membantu pengembang dalam mengembangkan sistem dalam mendapatkan wawasan tentang tugas, masalah yang ada pada pengguna dan juga membantu memperjelas persyaratan dan kebutuhan pengguna yang masih terlihat samar-samar (Carr & Verner, 1997).

2. LANDASAN KEPUSTAKAAN

2.1. SDLC Prototyping

Pada penelitian ini menggunakan *SDLC prototyping*. *Prototyping* adalah suatu model didalam *SDLC* dimana cara kerja pada model ini mengalami perulangan dan pendekatannya lebih gesit untuk pengembangan perangkat lunak. *Prototype* ini sangat lah dinamis, efisien, konsisten, dan lebih responsif terhadap kebutuhan klien, serta kurang beresiko. Pada model ini penekanannya ditempatkan pada penciptaan perangkat lunak dan kurang memperhatikan dokumentasi. *Prototyping* juga merupakan model yang pendekatannya berfokus terhadap kebutuhan pengguna karena umpan balik pengguna merupakan hal yang sangat mendasar untuk



mengembangkan perangkat lunak (Isaias & Issa, 2015).

Gambar 1. SDLC Prototyping

Sumber: (Carr & Verner, 1997)

2.2. UML

Unified Modeling Language (UML) merupakan sarana dalam pemodelan sistem perangkat lunak berbasis komponen, standar,

dan berorientasi objek. *UML* sendiri digunakan untuk memberikan gambaran deskripsi visual pada model sistem perangkat lunak. Bukan hanya untuk memberikan gambaran visual, *UML* juga dapat membantu menyelesaikan permasalahan dimana dapat mengembangkan perangkat lunak berkualitas tinggi dalam jangka waktu yang rasional. Pada saat yang sama juga klien memiliki persyaratan yang tinggi untuk desain, pemeliharaan, dan pengelolaan pengembangan terhadap perangkat lunak yang meliputi pemendekan siklus pengembangan perangkat lunak, peningkatan kualitas perangkat lunak, dan perlindungan pengembangan perangkat lunak, maka dari itu *UML* sangat lah membantu untuk mengatasi semua masalah tersebut (Zheng, Feng, & Zhao, 2014).

2.2.2 Class Diagram

Class diagram merupakan diagram yang memiliki hubungan antar kelas dan penjelasan detail dari setiap kelasnya dalam model desain suatu sistem. *Class Diagram* juga menunjukkan atribut dan constraint yang berhubungan dengan objek yang dikoneksikan. Lalu *Class Diagram* ini memiliki jenis relasi yang berbeda dimana terdapat agregasi, asosiasi, komposisi (Hendini, 2016). *Class Diagram* sendiri sangatlah berguna dalam semua bentuk pemrograman berorientasi objek(Souri, 2011).

2.2.3 Sequence Diagram

Sequence diagram merupakan suatu diagram yang menggambarkan bagaimana objek, atau kelompok objek berinteraksi didalam suatu sistem. Objek yang berinteraksi dapat berupa kelas atau komponen program (Poranen, Erkki, & Nummenmaa, 2015). *Sequence* diagram berfokus pada urutan waktu pengiriman pesan. Pada *sequence* diagram ini ditekankan terhadap apa yang terjadi pada tahap pertama, kedua, dan seterusnya yang mewakili jalannya waktu secara grafis (Song, 2013).

2.3. ReactJS

ReactJS merupakan kerangka kerja open source yang menggunakan *library javascript* untuk membuat user interface dan *React* biasa digunakan untuk menangani pengembangan pada aplikasi *single-page* dan aplikasi *mobile*. *ReactJS* memiliki keunggulan dimana kerangka kerja ini memberikan kecepatan, *simplicity*, dan *scalability*(Tung, 2018). *React* yang dikembangkan oleh facebook untuk

memfasilitasi pengembang dalam membuat komponen *UI* yang lebih interaktif, *stateful*, & *reusable* (Kumar & Singh, 2016). Dalam kaidah *MVC (Model View Control)* *react* hanya merepresentasikan pada bagian *View* saja dan ini merupakan bagian terbaik dalam penyederhanaan.

2.3. React Native

React Native merupakan kerangka kerja *javascript* yang digunakan untuk membangun aplikasi *mobile* *Android* maupun *iOS*. *React Native* ini memiliki dasar dari *React* dan *library javascript* dalam membangun antarmuka. *React Native* ini ditulis dengan campuran *javascript* dan *JSX*, lalu *React Native* ini juga memaparkan antarmuka *javascript* untuk *platform API* dimana pengembang dalam membangun aplikasi ini dapat mengakses fitur-fitur seperti kamera, lokasi, dll yang ada pada ponsel (Eisenman, 2018).

2.4. Pengujian Whitebox

Whitebox testing merupakan teknik yang melakukan penyelidikan secara detail dari logika internal dan struktur dari kode tersebut. Didalam melakukan pengujian *Whitebox* penguji diperlukan pemahaman penuh terhadap source code yang dikembangkan. Didalam melakukan pengujian ini penguji dapat mengungkapkan kesalahan manajemen yang buruk dengan menganalisis cara kerja internal sistem dan struktur sistem (Khan & Khan, 2012).

Pada penelitian ini peneliti menggunakan teknik dari *Whitebox* yang berupa teknik pengujian unit yang mana didalamnya terdapat basis path testing dan *cyclomatic complexity*. Dimana *basis path testing* mungkin untuk membantu penguji dalam memahami dan menentukan ukuran kompleksitas dari sebuah sistem. Dengan ukuran kompleksitas yang logis penguji dapat mendesain serangkaian jalur eksekusi yang dijamin untuk mengeksekusi setiap pernyataan dalam program setidaknya satu kali dalam tahap pengujian (Zapata, Akundi, Pineda, & Smith, 2013).

2.4. Pengujian Blackbox

Blackbox merupakan bagian yang penting dari pengujian perangkat lunak tetapi dalam *blackbox* idenya tidak hanya sebatas menguji kebenaran saja. Pengujian *Blackbox* sangat berkaitan dengan teknik yang ada pada

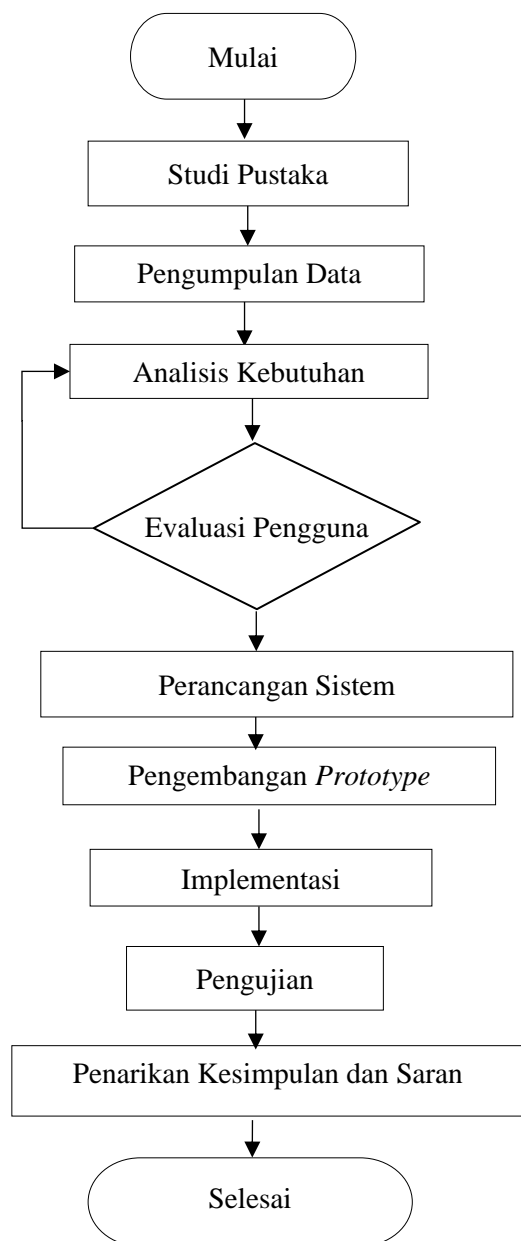
pengujian *whitebox* dan cenderung mengungkap kesalahan kelas yang berbeda dengan metode pengujian *whitebox*. Didalam pengujian *blackbox* penguji hanya tau bagian input (proses suatu sistem) dan output yang diperlakukan atau dengan kata lain penguji tidak harus mengetahui kerja internal suatu sistem (Khan, 2011). Pada penelitian ini peneliti menggunakan teknik pengujian validasi.

2.5. User Acceptance Testing

Selain menggunakan *whitebox* dan *blackbox* peneliti menggunakan pengujian *User Acceptance Testing (UAT)*. Pengujian ini tujuan menggunakan pengujian ini peneliti ingin membuktikan kepada pengguna bahwa sistem bekerja sesuai dengan pemahaman pengguna dan sesuai dengan kebutuhan pengguna. Menurut (Kusiak, 2007) pengujian ini merupakan satu-satunya yang menjadi kesempatan untuk pengguna dalam menguji sistem dengan kepuasan pengguna sebelum mereka menyatakan untuk menggunakan sistem tersebut. Untuk mendapatkan hasil dari pengujian *UAT* ini dilakukan pengumpulan data dengan instrumen yang digunakan, yaitu kuesioner. Menurut (Riduwan, 2010) tujuan penggunaan kuesioner dapat mencari informasi yang lebih lengkap dan responden tidak merasa khawatir jika memberikan jawaban yang tidak sesuai ekspektasi dalam pengisian daftar pertanyaan. Kuesioner yang digunakan pada pengujian *UAT* bersumber (Singapore Management University, 2012) yang dipublikasikan oleh wiki.smu.edu.sg dan menggunakan perhitungan skala *likert*. Dan menurut (Riduwan, 2010) bahwa metode skala *likert* dapat digunakan untuk mengukur pendapat atau persepsi seseorang.

3. METODOLOGI

Metode penelitian Sistem Informasi Pengelolaan Persediaan Barang ini menerapkan metode *Prototyping*. Terdapat diagram blok metodologi yang ditunjukkan pada Gambar 2



Gambar 2. Diagram alir proses penelitian

Pada tahapan ini peneliti mencari studi pustaka yang menjadi referensi pendukung untuk penelitian ini. Selain untuk menjadi referensi pendukung pada tahapan ini berfungsi sebagai pembelajaran untuk peneliti agar dapat memahami mengenai konsep atau objek yang akan diteliti serta membantu peneliti dalam proses analisis kebutuhan, perancangan sistem, implementasi, dan pengujian sistem. Peneliti mengambil banyak sumber dari jurnal ilmiah, e-book, laporan ilmiah, dll.

Pengumpulan Data yang didapat oleh peneliti mengenai permasalahan yang terjadi pada Toko UDA FAJRI tersebut dengan melakukan metode observasi dan wawancara kepada pemilik toko yang bernama bapak

Afrizal. Selain menjadi pemilik dari toko tersebut, beliau juga yang mengatur jalan masuknya barang ke dalam toko tersebut. Pada fase ini peneliti menggunakan instrumen pengumpulan data dengan memberikan pertanyaan-pertanyaan mengenai bagaimana alur distribusi barang dan manajemen barang didalam toko tersebut. Metode wawancara yang digunakan oleh peneliti ini menggunakan wawancara terpimpin dimana daftar pertanyaan yang diajukan sudah disusun oleh peneliti. Dari metode pengumpulan data yang dilakukan akan didapatkan data secara riil untuk dilanjutkannya ke dalam pengembangan sistem informasi distribusi barang ditoko tersebut.

Pada fase analisis kebutuhan, dimana kebutuhan dan fitur dari pengembangan sistem ini didapatkan oleh peneliti berasal dari hasil wawancara yang dilakukan terhadap pemilik toko. Dalam analisis kebutuhan yang dilakukan akan terdapat beberapa fase dimana fase tersebut mengidentifikasi aktor, mengidentifikasi setiap kebutuhan, mendefinisikan kebutuhan fungsional & non-fungsional, membuat *use case* diagram, dan membuat *use case* skenario. Dari hasil identifikasi tersebut akan dikembangkan lah sebuah sistem distribusi barang yang berbentuk web admin, didalam web admin ini pengguna dapat memasukan stok barang dan melihat rekap data barang apa saja yang akan dijual nantinya. Dan peneliti menemukan analisis kebutuhan lainnya yaitu dengan membuat satu aplikasi berbasis Android dimana aplikasi ini terdapat jumlah stok barang yang tersedia didalam web admin sekaligus dapat melakukan pembayaran dan pengurangan barang secara otomatis nantinya.

Pada fase evaluasi pengguna peneliti akan menganalisis ulang kebutuhan yang akan diinginkan pemilik toko. Untuk mencari kebutuhan tambahan tersebut peneliti menggunakan metode wawancara dan observasi secara langsung. Peneliti menggunakan 1 sampel populasi homogen yaitu hanya pemilik toko yang memiliki kriteria dimana kriteria tersebut pengguna yang hanya tahu dan mengerti bagaimana alur pengelolaan barang masuk dan keluar ditoko tersebut. Wawancara ini dilakukan dengan bertanya secara langsung kepada responden sekiranya apa yang kurang dalam kebutuhan tersebut.

Perancangan sistem ini dilakukan ketika analisis kebutuhan sudah didapatkan. Pada fase perancang sistem merupakan acuan dalam melakukan pengembangan sistem ini. Dalam perancangan sistem ini peneliti menggunakan

model perancangan *UML* dengan menggunakan kebutuhan yang sudah didapatkan. Untuk memberikan bagaimana sistem ini berjalan penileliti menggunakan *sequence* diagram, menggunakan *class* diagram untuk merancang kelas yang digunakan sistem, perancangan database, dan perancangan antarmuka. Selain menggunakan perancangan menggunakan *UML* peneliti menggunakan arsitektur perancangan *MVVM* yang akan diterapkan ke dalam pengembangan sistem tersebut. Didalam fase ini peneliti melakukan perancangan sistem sebelum dikembangkannya *prototype*. Perancangan yang dibuat peneliti mencakup dari perancangan sistem yang belum ditambahkannya kebutuhan fungsional dan sudah ditambahkannya kebutuhan fungsional.

Setelah fase analisis kebutuhan dan perancangan sistem dilakukan, maka peneliti melakukan pengembangan sistem. Sistem yang dikembangkan masih berupa *prototype* dengan menggunakan metode *SDLC prototyping*, ini dikarenakan untuk mengetahui kebutuhan tambahan yang diinginkan pengguna untuk mengetahui kebutuhan tambahan tersebut pada fase ini akan lanjut kepada fase analisis pengguna. Lalu peneliti menggunakan jenis *prototyping Exploratory Prototyping* dengan alasan karena peneliti tidak ingin kebutuhan yang diinginkan oleh pengguna terlihat samar-samar. Didalam fase ini peneliti mendapatkan dua iterasi, pada iterasi yang pertama peneliti mengajukan sistem web admin yang masih sangat sederhana kepada pengguna. Dan sistem hanya dapat mengeluarkan barang dan memasukan barang dari web admin. Lalu pada iterasi kedua yang didapatkan dari evaluasi pengguna peneli mengembangkan sistem web admin yang dipadukan dengan *device* Android yang mana sistem yang dikembangkan tersebut dapat melakukan transaksi barang, scan kode barcode pada barang.

Pada fase implementasi ini merupakan tahap akhir didalam pengembangan sistem sebelum masuk ke dalam pengujian. Pada ini sistem diharapkan sudah memenuhi apa yang dibutuhkan pengguna dan siap untuk masuk ke dalam pengujian. Dalam implementasi ini peneliti menggunakan bahasa pemrograman

javascript dan CSS dengan bantuan framework *ReactJs* dan *Native*, selain itu peneliti juga menggunakan *Postgresql* sebagai database tempat penyimpanan data. Pada tahap ini peneliti membutuhkan perangkat keras berupa laptop dengan spesifikasi yang laptop tersebut memiliki *chipset AMD A6, HDD 500 GB, OS Windows 8*, dan berupa *Android device* dengan spesifikasi yang harus dimiliki adalah *chipset qualcomm snapdragon 625/660 dan OS Android 7.0 – 9.0*.

Pengujian ini dilakukan ketika implementasi sudah selesai dilakukan dan sesuai dengan kebutuhan pengguna. Proses pengujian ini untuk mengetahui error atau bug yang ada didalam sistem yang dikembangkan. Selain untuk mencari bug atau error pengujian dilakukan untuk mengetahui ketepatan dalam kebutuhan fungsional dan kebutuhan non-fungsional. Peneliti melakukan pengujian fungsional dengan teknik *whitebox testing* dan *blackbox testing*. Pada *whitebox testing* pengembang melakukan pengujian dengan tipe pengujian *unit*, dimana pada tipe pengujian ini akan dilakukan pengujian terhadap *sourcecode* sistem yang sudah dikembangkan dengan *scope* tidak semua *class* yang diuji. Selain menggunakan *unit* pengembang menggunakan integrasi dimana pada tipe ini terdapat *whitebox* dan *blackbox testin*. Pada integrasi ini dilakukan pengujian terhap *sourcecode* yang sudah dikembangkan dengan *scope* dimana *class* individu digabungkan menjadi sebuah kelompok. Dan pengujian non-fungsional akan dilakukan dengan menggunakan *user acceptance testing*.

Penarikan kesimpulan dapat dilakukan setelah tahapan penelitian yang dilakukan selesai. Kesimpulan merupakan hasil dari penelitian yang dilakukan oleh peneliti yang berisi hasil rancangan sistem dan pengujian yang dilakukan. Saran bertujuan untuk memberikan arahan kepada peniliti yang ingin melanjutkan penelitian ini.

4. ANALISIS KEBUTUHAN

4.1. Identifikasi Aktor

Aktor yang dapat menggunakan sistem ini akan dipaparkan pada tabel 1 dibawah ini.

Tabel 1. Identifikasi Aktor

Aktor	Deskripsi
Pemilik	Pemilik merupakan aktor yang melakukan

pengelolaan barang dari jalan masuknya barang hingga keluarnya barang pada sistem.

Operator Operator adalah aktor yang dapat melakukan transaksi pembelian barang yang dilakukan pembeli.

4.2. Kebutuhan Fungsional

Kebutuhan fungsional merupakan suatu fitur atau layanan yang harus dan dapat digunakan oleh pengguna yang telah menentukan kebutuhan seperti apa yang diinginkannya. Kebutuhan fungsional ini diperoleh dari hasil wawancara dan observasi. Pada sistem ini terdapat 15 kebutuhan fungsional. Berikut salah satu kebutuhan yang ditunjukkan pada tabel 2.

Tabel 2. Kebutuhan Fungsional Menambah Barang

Kode	Aktor	Nama Fungsi	Spesifikasi/Deskripsi
TUF_F_03	Pemilik	Menambah Barang	Sistem dapat memberikan fungsi input barang kepada pengguna. Spesifikasi : 1. Menambah barang dilakukan dengan cara menginputkan nama, jumlah, harga, dan foto

4.3. Kebutuhan Non Fungsional

Kebutuhan non-fungsional berfokus pada reaksi pengguna terhadap sistem yang sedang dikembangkan dan termasuk kebutuhan yang diperlukan pengguna sistem ini. Kebutuhan non-fungsional sistem ini yaitu *Acceptance* dijelaskan pada tabel 3.

Tabel 3. Kebutuhan Non-Fungsional

Kode	Aktor	Nama Fungsi
TUF_NF_01	<i>Acceptance</i>	Apakah aktor dapat menerima sistem informasi pengelolaan barang dengan baik dan dapat dimengerti.

4.4. Use Case Scenario

Use case scenario memberikan penjelasan mengenai kebutuhan fungsional. Salah satu *use case scenario* adalah seperti pada tabel 4 berikut.

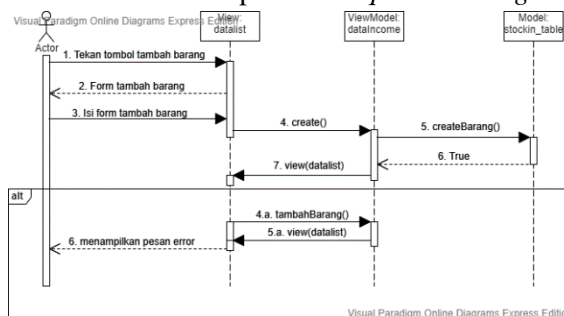
Tabel 4. *Use Case Scenario* Menambah Barang

Aktor	Pemilik
Objective	Aktor dapat menambah barang.
Pre-Condition	Aktor berada dihalaman daftar barang.
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol tambah barang. 2. Sistem menampilkan form tambah barang dengan input kode, nama, jumlah, harga, foto. 3. Sistem menambahkan user ke dalam database. 4. Sistem menampilkan pesan sukses.
Alternative Flow	<ol style="list-style-type: none"> 1. Aktor tidak memasukan salah satu identitas sistem akan menampilkan pesan “
Post-Condition	Aktor berhasil terdaftar didalam database.

5. PERANCANGAN & IMPLEMENTASI

5.1. Pemodelan *Sequence Diagram*

Pada *sequence* diagram akan dijelaskan proses jalannya program yang terjadi untuk mencapai suatu kebutuhan sistem. Objek yang terdapat pada *sequence* diagram merupakan hasil identifikasi dari spesifikasi kebutuhan dan use case scenario. *Sequence* diagram akan menggambarkan setiap use case yang ada. Gambar 3 merupakan *sequence diagram*



menambah barang.

Gambar 3. *Sequence diagram*

5.2. Perancangan *Class Diagram*

Pada perancangan *class* diagram ini terdapat 3 bagian diantaranya terdapat, *view*, *viewmodel*, dan *model*. Guna dari perancangan *class* diagram ini agar dapat memetakan struktur sistem dari aplikasi yang akan dikembangkan.

5.3. Perancangan Basis Data

Perancangan basis data berisi gambaran dari data yang dibutuhkan pada sistem. Perancangan basis data akan dijelaskan dengan menggunakan *Entity Relationship Diagram* (ERD) dan *Relational Model*.

5.4. Perancangan Pseudocode

Pada bagian ini akan dijelaskan algoritme *pseudocode*. Tabel 4 merupakan tabel *pseudocode* menambah barang,

Tabel 4. *Pseudocode* Menambah Barang

```

Arrow function handleCreate =
mendeklarasikan fungsi untuk melakukan
pengiriman data

Begin
Form = mendeklarasikan bahwa function
bagian dari form

validateField = melakukan validasi
seleksi kondisi if = jika inputan
kosong

akan menolak inputan

seleksi kondisi else = jika inputan
terisi

menyimpan data

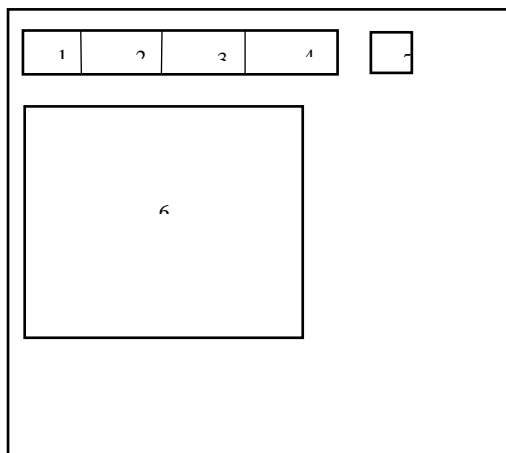
menampilkan halaman daftar barang

end

```

5.4. Perancangan Antarmuka

Perancangan antarmuka ini berisi beberapa rancangan halaman antarmuka dari sistem informasi pengelolaan persediaan barang dari segi *website*.



Gambar 4. Perancangan antarmuka Dashboard

5.5. Implementasi Basis Data

Hasil implementasi basis data dari sistem ini adalah terdapat 4 tabel *database* yaitu tabel *user*, *listitem_table*, *stockin_table*, dan *stockout_table*.

5.6. Implementasi Source Code

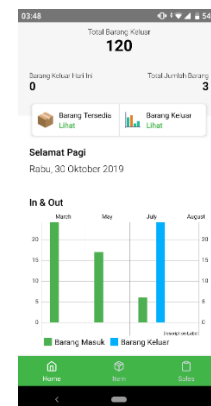
Implementasi *source code* adalah hasil dari implementasi yang dilakukan sesuai kebutuhan fungsional dan perancangan system kedalam bentuk *source code*.

5.7. Implementasi Antarmuka

Implementasi Antarmuka ini dikembangkan sesuai perancangan antarmuka yang telah dibentuk atau dirancang. Antarmuka merupakan sebagai penghubung antara pengguna dengan sistem. Gambar 5 dan 6 merupakan salah satu contoh dari implementasi antarmuka dashboard pada segi *website* dan *home* dari segi Android.



Gambar 5. Implementasi Antarmuka Dashboard



Gambar 6. Implementasi Antarmuka Home

6. PENGUJIAN

6.1. Pengujian Whitebox

Teknik pengujian unit dilakukan dengan metode *whitebox testing* yang berdasarkan pada implementasi dari *source code*. Pada bagian pengujian whitebox ini digunakan metode *basis path testing* dan *Cyclomatic Complexity*. Dari kedua metode tersebut dihasilkan jalur independen yang akan dijadikan sebagai acuan dalam melakukan pengujian dengan *test case*.

6.2. Pengujian Validasi

Teknik pengujian validasi dilakukan dengan menggunakan metode *blackbox testing* yang dilakukan untuk menguji fungsionalitas pada sistem tanpa melihat implementasi kode sistem. Pada pengujian *blackbox* ini akan digunakan *use case* sebagai tolak ukur dalam pengujian ini.

6.3. Pengujian User Acceptance Testing

User Acceptance Testing dilakukan dengan metode kuesioner yang memiliki 4 tahapan, yaitu reaksi pengguna terhadap sistem, reaksi pengguna terhadap antarmuka sistem, reaksi pengguna dalam mengoperasikan sistem, dan reaksi pengguna terhadap fungsionalitas sistem. Kuesioner ini diberikan kepada 5 responden dan dilakukan perhitungan pencarian hasil dengan menggunakan skala *likert*. Berikut tabel 5 yang merupakan tabel hasil reaksi pengguna terhadap sistem.

Indikator / Poin	Jumlah Responden	Jumlah Skor
Sangat Setuju / 5	1	5
Setuju / 4	3	12
Netral / 3	1	3
Total Skor		20

Rumus : $TS = \sum (PI \times JR)$

Keterangan :

- TS = Total Skor
- PI = Poin Indikator
- JR = Jumlah Responden

Lalu persentase pada tabel reaksi pengguna terhadap sistem dapat diketahui dengan perhitungan, yaitu $20/25 \times 100\% = 80\%$.

7. KESIMPULAN

Pada hasil analisis kebutuhan yang dilakukan pada pengembangan sistem ini dengan menggunakan metode wawancara dan observasi secara langsung didapatkan kebutuhan fungsional mulai dari kebutuhan fungsional pada aplikasi berbasis *website* dan aplikasi berbasis Android. Pada aplikasi *website* yang diperuntukan kepada pemilik terdapat 10 kebutuhan fungsional. Lalu untuk aplikasi berbasis Android memiliki 5 kebutuhan fungsional yang diperuntukan kepada operator. Pada analisis kebutuhan fungsional terdapat 2 kali iterasi dengan *prototyping* yang mana untuk memastikan kembali kebutuhan fungsional yang benar-benar diinginkan oleh pengguna.

Pada merancang sistem ini terdapat perancangan *sequence diagram*, *class diagram*, perancangan basis data, perancangan antarmuka, dan perancangan *pseudocode*. Pada perancangan

basis data terdapat 2 perancangan, yaitu perancangan *Entity Relationship Diagram* dan perancangan *Relational Model*.

Pada tahapan implementasi menggunakan *framework React* dan Native dengan *library* dan bahasa pemrograman *java script* dan media penyimpanan basis menggunakan *Postgresql*. Didalam tahapan ini terdapat spesifikasi sistem, implementasi basis data, implementasi *source code*, dan implementasi antarmuka.

Pada tahapan pengujian ini digunakan 3 metode pengujian diantaranya, *whitebox testing*, *blackbox testing*, dan *user acceptance testing*. Yang mana dari setiap metode pengujian tersebut memiliki hasil tersendiri. Mulai dari pengujian *whitebox* ini menggunakan teknik pengujian *unit* hanya menguji 4 fungsi utama dalam pengembangan sistem ini, yaitu pengujian *unit* terhadap tambah barang, ubah data barang, unduh laporan penjualan, dan *scan qr code* yang mana hasil keseluruhan pada 4 fungsi utama memiliki hasil valid. Lalu pada pengujian *blackbox* ini menguji 15 kebutuhan fungsional dengan menggunakan teknik pengujian validasi yang mana hasil dari ke 15 kebutuhan fungsional tersebut memiliki hasil yang valid atau sesuai ekspektasi pengguna. Dan yang terakhir dari pengujian *user acceptance testing* menggunakan instrumen kuesioner dan menggunakan perhitungan skala *likert* ini memiliki 4 tahapan pengujian yang memiliki hasil masing-masing, mulai dari reaksi pengguna terhadap sistem, reaksi pengguna terhadap antarmuka sistem, reaksi pengguna dalam mengoperasikan sistem, dan reaksi pengguna terhadap fungsionalitas sistem. Pada reaksi pengguna terhadap sistem ini memiliki persentase 80% dengan kategori “Setuju”, reaksi pengguna terhadap antarmuka sistem memiliki persentase 68% dengan kategori “Cukup”, reaksi pengguna dalam mengoperasikan sistem memiliki persentase 96% dengan kategori “Sangat Mudah”, dan reaksi pengguna terhadap fungsionalitas sistem memiliki persentase 76% dengan kategori “Setuju”.

8. SARAN

Saran yang diberikan dari penelitian ini dengan tujuan untuk mengembangkan kembali atau memperbaiki sistem ini untuk kedepannya, yaitu :

1. Dapat ditambahkan fitur agar sistem yang dikembangkan dapat lebih bermanfaat.
2. Dapat dikembangkannya kembali antarmuka sistem agar lebih menarik.

9. DAFTAR PUSTAKA

- Aljamea, M., & Alkandari, M. (2018). MMVMi : A Validation Model for MVC and MVVM Design Patterns in iOS Applications, (August).
- Barjtya, S., Sharma, A., & Rani, U. (2017). A detailed study of Software Development Life Cycle (SDLC) Models, 6(7), 22097–22100.
- Berisha-Shaqiri, A. (2014). Management Information System and Decision-Making. *E-ISSN 2281-4612*, 3, 19–23.
- Boell, S. K., & Cecez-Kecmanovic, D. (2015). What is an information system? *Proceedings of the Annual Hawaii International Conference on System Sciences, 2015-March*(January), 4959–4968.
- Carr, M., & Verner, J. (1997). Prototyping and Software Development Approaches. *Prototyping and Software Development Approaches*, 2004(3), 1–16.
- Christanto, E. N. (2009). SISTEM INFORMASI MANAJEMEN DISTRIBUSI (Studi kasus : PT . SOLOPOS RAYA).
- Eisenman, B. (2018). *Learning React Native*. (M. Foley, Ed.) (2nd ed.). California: O'Reilly Media, Inc.
- Grgec, M., & Muzar, R. (2007). ROLE OF UML SEQUENCE DIAGRAM CONSTRUCTS IN OBJECT LIFECYCLE CONCEPT. *Journal of Information and Organizational Sciences*, 31(1), 63–74.
- Hendini, A. (2016). Pemodelan UML Sistem Informasi Monitoring Penjualan dan Stok Barang (STUDI KASUS: DISTRO ZHEZHA PONTIANAK), IV(2), 107–116.
- Kusiak, J. (2007). *A Manager 's Guide to User Acceptance Testing* (ABN 56 007 219 589). Melbourne. Retrieved from www.irm.com.au
- Lee, K. R. (2013). Impacts of Information Technology on Society in the new Century, 1–6. Retrieved from
- Liang, T., & Chen, D. (2003). Evolution of Information Systems Research. *7th PACIS, 10-13 July, Adelaide*, (July 2003), 834–842.
- Listiani, A., & Wahyuningsih, S. D. (2019). ANALISIS PENGELOLAAN PERSEDIAN BARANG DAGANG UNTUK MENGOPTIMALKAN LABA. *E-ISSN 2528-2581*, 4(1), 95–103.
- Poranen, T., Erkki, M., & Nummenmaa, J. (2015). How to Draw a Sequence Diagram, 4(Project 51528), 91–104.
- Riduwan. (2010). *Metode Dan Teknik Menyusun Tesis*. (H. AKDON & Z. Arifin, Eds.) (8th ed.). Bandung: ALFABETA. Retrieved from www.cvalfabeta.com
- Singapore Management University. (2012). 2012T1 Fortune Cookies: User Testing. Retrieved from https://wiki.smu.edu.sg/is480/2012T1_Fortune_Cookies%3A_User_Testing#User_Acceptance_Test
- Song, I. (2013). Developing Sequence Diagrams in UML, (September 2001),
- Souri, A. (2011). Formalizing Class Diagram in UML, (May 2014). <https://doi.org/10.1109/ICSESS.2011.5982368>
- Tung, K. (2018). *Developing a frontend application using ReactJS and Redux*. Laurea University of Applied Sciences.
- Waskitho, W., & Baskoro, F. (2002). PENGUJIAN PERANGKAT LUNAK DENGAN MENGGUNAKAN MODEL BEHAVIOUR UML, 1(1), 43–50.
- Zamroni, M. (2017). Perkembangan Teknologi Komunikasi Dan Dampaknya Terhadap Kehidupan. *Jurnal Pendidikan*, X(2), 195–
- Zapata, F., Akundi, A., Pineda, R., & Smith, E. (2013). Basis Path Analysis for Testing Complex System of Systems. *Procedia - Procedia Computer Science*, 20, 256–261.
- Zheng, J., Feng, Y., & Zhao, Y. (2014). A Unified Modeling Language-Based Design and Application for a Library Management Information System, 14, 129–144.