

IJCIT

(Indonesian Journal on Computer and Information Technology)

Journal Homepage: <http://ejournal.bsi.ac.id/ejurnal/index.php/ijcit>

Migrasi Aplikasi Stock Opname Platform Desktop Ke Android Menggunakan Kivy Framework (Studi Kasus Di PT Sumber Alfaria Trijaya Tbk)

Prihantoro Manahan Tobing¹, Magdalena Ariance Ineke Pakereng²

Program Studi Teknik Informatika, Universitas Kristen Satya Wacana
Salatiga, Indonesia

e-mail: puitobing7@gmail.com¹, ineke.pakereng@uksw.edu²

ABSTRAK

Sistem pengelolaan barang sangat dibutuhkan bagi salah satu perusahaan retail terbesar di Indonesia, yaitu PT Sumber Alfaria Trijaya Tbk atau lebih dikenal dengan nama Alfamart. Dalam menunjang kemajuan bisnisnya, Alfamart selalu *up to date* dalam hal teknologi, agar tidak kalah bersaing dengan perusahaan retail lain dan tetap mempertimbangkan biaya pengeluaran. Untuk mengatasi permasalahan teknologi yang tertinggal dilakukan dengan membangun sistem yang bernama *Stock Opname* menggunakan teknologi *Kivy Framework*. *Kivy* merupakan suatu *library open-source* yang dapat digunakan untuk membangun aplikasi *desktop* dan atau *mobile*. Penelitian ini menggunakan metode penelitian Reayasa *Forward Engineering* dengan metode pengembangan sistem *SDLC* model proses *waterfall*. Penelitian ini menghasilkan aplikasi program komputer bernama *Stock Opname* yang telah teruji dapat berjalan secara *cross-platform*.

Kata Kunci: *aplikasi mobile, cross-platfom, kivy framework, PT sumber alfaria trijaya, retail*

ABSTRACTS

As one of the largest retail companies in Indonesia, PT Sumber Alfaria Trijaya Tbk or widely known as Alfamart is required to have a reliable goods management system. In supporting its business progress, the company also needs to be always up to date in terms of technology, so that it is not being outclassed by competitors while keeping reasonable expenses. To overcome lagging technology problems, building a system called Stock Opname using Kivy Framework technology can be useful. Kivy is an open-source library that can be used to build desktop and or mobile applications. This study uses the Forward Engineering research method with the SDLC system development approach of the waterfall process model. It develops a computer application program called Stock Hosting that has been proved to be able to run across platforms.

Keywords: *cross-platfom, kivy framework, mobile aplcation, retail, PT sumber alfaria trijaya*

1. PENDAHULUAN

Kemajuan di bidang teknologi informasi terus berkembang seiring dengan perkembangan zaman, baik berupa *software* atau aplikasi program komputer dengan berbagai macam *platform*, sistem operasi

maupun teknologi *hardware*. PT Sumber Alfaria Trijaya Tbk adalah perusahaan retail terkemuka di Indonesia yang dikenal dengan nama Alfamart. Perusahaan ini memiliki beragam aplikasi program komputer yang ada pada setiap toko Alfamart. Contohnya adalah *Store Information System (SIS)*, *Point Of Sales (POS)*



dan *Stock Opname* (SO) yang mendukung kinerja setiap toko Alfamart.

Stock Opname adalah aplikasi program komputer yang berguna untuk melakukan verifikasi data barang, agar data fisik yang ada pada toko sesuai dengan data yang ada dalam *database* komputer. *stock opname* berfungsi sebagai sistem pengendalian intern yang bertujuan untuk mengetahui keakuratan catatan pembukuan (Apulina Sembiring, 2019).

Saat ini PT Sumber Alfaria Trijaya Tbk memiliki lebih dari 14.000 gerai / toko beroperasi dan lebih dari 4.000.000 transaksi per harinya. Berdasarkan hasil wawancara secara langsung dengan salah satu staf IT Divisi *Business Solution* sebagai *Store Development Specialist*, Alfamart secara berkala dalam kurun waktu tertentu selalu melakukan migrasi aplikasi program komputer yang dimiliki untuk menyesuaikan teknologi informasi yang digunakan saat ini dengan teknologi informasi terbaru agar tidak kalah bersaing dengan perusahaan retail lain. Dalam upaya mengembangkan teknologi informasi yang digunakan, Alfamart juga mempertimbangkan dari segi biaya dan keefektifan. Maka tujuan penelitian ini adalah untuk membangun dan memperbaiki aplikasi program komputer khususnya *Stock Opname* milik Alfamart menggunakan *Kivy Framework* agar dapat berjalan pada berbagai *platform* maupun sistem operasi hanya dalam satu kali pembuatan kode program.

Dari pembahasan yang telah diuraikan, diperoleh beberapa rumusan masalah yang akan diselesaikan dalam penelitian ini. Rumusan masalah tersebut terdiri dari: (1) Apa itu *Kivy Framework*? (2) Bagaimana alur proses migrasi aplikasi *Stock Opname* dari teknologi lama ke teknologi baru? (3) Bagaimana alur proses perancangan dan pembangunan aplikasi program komputer *Stock Opname* menggunakan *Kivy Framework*?

Tujuan dan manfaat yang didapat setelah melakukan penelitian ini diantaranya: (1) Mengetahui dan memahami konsep dari teknologi *Kivy Framework*; (2) Mengetahui alur proses yang dikerjakan dalam melakukan migrasi aplikasi dari teknologi lama ke teknologi baru; (3) Mengetahui alur proses perancangan dan pembangunan aplikasi program komputer *Stock Opname* menggunakan *Kivy Framework*.

Penelitian yang berjudul "*Application Development using Kivy Framework*" memfokuskan pada fitur-fitur yang dimiliki oleh

Kivy Framework. *Kivy Framework* membangun suatu aplikasi yang mampu berjalan di berbagai sistem operasi seperti *Linux*, *Windows*, *Ios* maupun *Android*. Karena kelebihan ini lah *Kivy Framework* layak disebut sebagai *Framework* yang *powerful* (Bhoyarkar et al., 2019). Pada penelitian lain yang berjudul "*Sistem Informasi Logistik Stock Opname Menggunakan Quick Response Code Berbasis Android*" meneliti perbedaan sebelum dan setelah pembangunan aplikasi *Stock Opname* dibuat. Perhitungan jumlah barang yang sebelumnya dilakukan manual kini dapat dilakukan secara cepat *Quick Response* (Ipan A'shary Wulung et al., 2014).

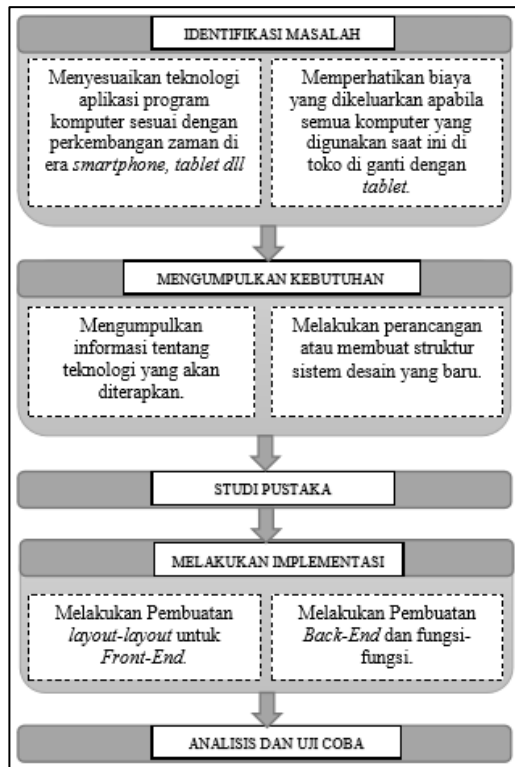
Penelitian ini dilakukan untuk migrasi aplikasi *Stock Opname* dari *platform desktop* ke *Android*. Sebelumnya, aplikasi *Stock Opname* dibangun dengan menggunakan *VB.NET*. *VB.NET* adalah bahasa pemrograman *object-oriented* yang dikembangkan oleh *Microsoft*, yang mendukung fitur *encapsulation*, *polymorphism*, *abstraction*, dan *inheritance* (Ruli, 2017). *VB.NET* sendiri merupakan bahasa pemrograman *Visual Basic* (VB) yang digunakan untuk membangun aplikasi yang berjalan pada *Dot NET Framework*. Salah satu kelebihan dari *VB.NET* adalah memudahkan *developer* dalam mendesain *rancangan interfacenya*, dikarenakan adanya fitur *drag and drop*, sehingga dapat mempersingkat dalam pembuatan kode program pada bagian desainnya.

Kivy Framework adalah *cross-platform Framework* dari *library Python open-source* yang digunakan untuk membangun *multi-touch* aplikasi program komputer *platform desktop* dan atau *mobile* dengan *Natural User Interface* (NUI), sehingga memudahkan *developer* untuk mengembangkan *User Interface* (UI). Selain itu, *Kivy Framework* dapat berjalan pada berbagai sistem operasi yang umum digunakan seperti *Windows*, *Linux*, *Mac OS*, *Ios*, dan *android* (Qutsiah et al., 2016).

Setelah membaca dan melihat penelitian sebelumnya yang memberikan informasi bahwa penelitian ini lebih unggul dikarenakan aplikasi *Stock Opname* yang dibangun dengan *Kivy Framework* dapat membantu perusahaan retail PT Sumber Alfaria Trijaya Tbk dalam menjalankan bisnisnya, serta mampu berjalan pada berbagai sistem operasi yang umum digunakan, sehingga efisiensi sumber daya semakin baik.

2. METODE PENELITIAN

Metode penelitian yang digunakan adalah metode penelitian Rekayasa *Forward Engineering*. Meliputi Identifikasi Masalah, Pengumpulan Kebutuhan, Studi Pustaka, Melakukan Implementasi serta Analisis Dan Uji Coba seperti yang terlihat pada Gambar 1 (Weeks, 2015).



Gambar 1. Bagan Metodologi Penelitian Yang Dirancang

Tahap-tahap penelitian yang terdapat pada Gambar 1 akan dijelaskan sebagai berikut:

1) Identifikasi Masalah

Tahap Identifikasi Masalah menemukan masalah, yaitu ingin memperbaiki aplikasi program komputer *Stock Opname* yang masih menggunakan teknologi lama. Memperhitungkan biaya yang dikeluarkan jika ada pergantian *hardware* maupun penggunaan *hardware* lain yang akan digunakan pada toko.

2) Mengumpulkan Kebutuhan

Tahap mengumpulkan kebutuhan adalah mencari informasi tentang teknologi yang akan digunakan sekaligus menyusun sistem desain baru. Kebutuhan yang dimaksud adalah kebutuhan *software* diantaranya *database*, bahasa pemrograman hingga kebutuhan *hardware* apa yang akan digunakan.

3) Studi Pustaka

Tahap Studi Pustaka adalah mendalami kebutuhan-kebutuhan yang telah dikumpulkan dan melaporkan setiap hasil dari pendalaman.

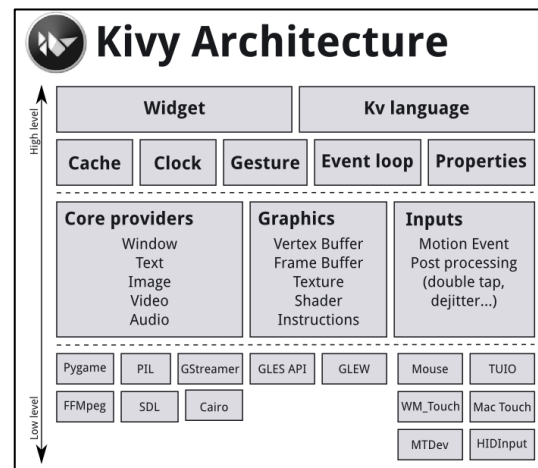
4) Melakukan Implementasi

Tahap Implementasi adalah serangkaian proses mulai dari percobaan sampai layak coba.

5) Analisis dan Uji Coba

Tahap analisis dan uji coba adalah tahap penyempurnaan dari tahap implementasi.

Penelitian ini menggunakan *Kivy Framework* dalam pembangunan aplikasi *Stock Opname*. Untuk lebih mempermudah alur penelitian dan pemahaman tentang teknologi *Kivy Framework*, berikut *arsitektur Kivy* yang terlihat pada Gambar 2.



Gambar 2. Arsitektur Kivy

Pada Gambar 2 memberikan gambaran arsitektur dari teknologi *Kivy Framework*. Dimana level tertinggi terdapat *Widget* dan *Kv language*. *Widget* adalah komponen-komponen pada *User Interface* yang ditampilkan aplikasi kepada pengguna aplikasi. Contoh dari *widget* adalah *button*, *textfield*, *image* dan lain-lain. *Kv language* adalah bahasa pemrograman pada *Kivy Framework* yang digunakan untuk mengatur tampilan atau setiap *Widget* dalam *Kivy Framework*. Kode program *Kv language* sendiri dapat terpisah dari *file* kode program utama atau jadi satu *file* dengan kode program utama. Level dibawah *Widget* dan *Kv language*, yaitu *Cache*, *Clock*, *Gesture*, *Event loop*, dan *Properties* bertindak untuk menangani proses yang berjalan pada *background*. Contohnya *Clock* untuk membuat suatu *timer* penjadwalan, *Cache* untuk penyimpanan pada *memory cache*, *Gesture* untuk mendeteksi suatu pola gerakan seperti lingkaran atau persegi, *Event loop* untuk

mendeteksi suatu *event* yang digunakan, dan *Properties* untuk menghubungkan *Widget code* dengan deskripsi *User Interface*. Level selanjutnya *Core providers*, *Graphics*, dan *Input* yang menangani tugas utama pada sistem aplikasi yang sudah terintegrasi lintas *platform*. Contohnya *Core providers* bertugas untuk membuka *window*, menampilkan *text*, gambar, *video*, dan *Audio*. *Graphics* untuk menangani grafik, dan *Inputs* untuk menangani berbagai macam jenis *input*. Level terendah mendefinisikan modul atau kumpulan *library* dari level di atasnya (*Architectural Overview – Kivy 1.11.1 Documentation*, n.d.).

Metode yang digunakan untuk membangun aplikasi program komputer *Stock Opname* menggunakan *Software Development Life Cycle* (SDLC) dengan model proses *Waterfall*. Menurut (Rani Susanto & Andriana, 2016) Model *waterfall* lebih cocok untuk pengembangan sistem bersifat generik, dimana kebutuhan sistem dapat diidentifikasi dari awal dengan spesifikasi yang umum. Model proses *Waterfall* bersifat sistematis yang terdiri dari lima tahapan, yaitu Analisis (*Analysis*), Desain (*Design*), Kode (*Coding*), Testing (*Testing*), dan Implementasi (*Implementation*). Model proses *Waterfall* yang digunakan pada penelitian ini dapat dilihat pada Gambar 3.

Pada tahap *analysis*, peneliti melakukan wawancara secara langsung dengan dua staf IT Divisi *Business Solution* sebagai *Store Development Specialist* dan *Store Application Programmer Analyst* untuk mengetahui

kebutuhan fungsional serta tampilan *User Interface* aplikasi *Stock Opname*.

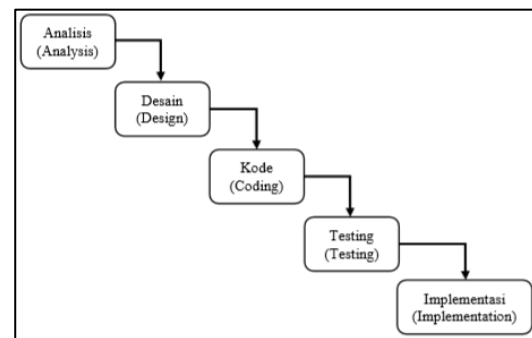
Pada tahap *Design*, peneliti membagi menjadi dua bagian yang terdiri dari:

a. *Design database*

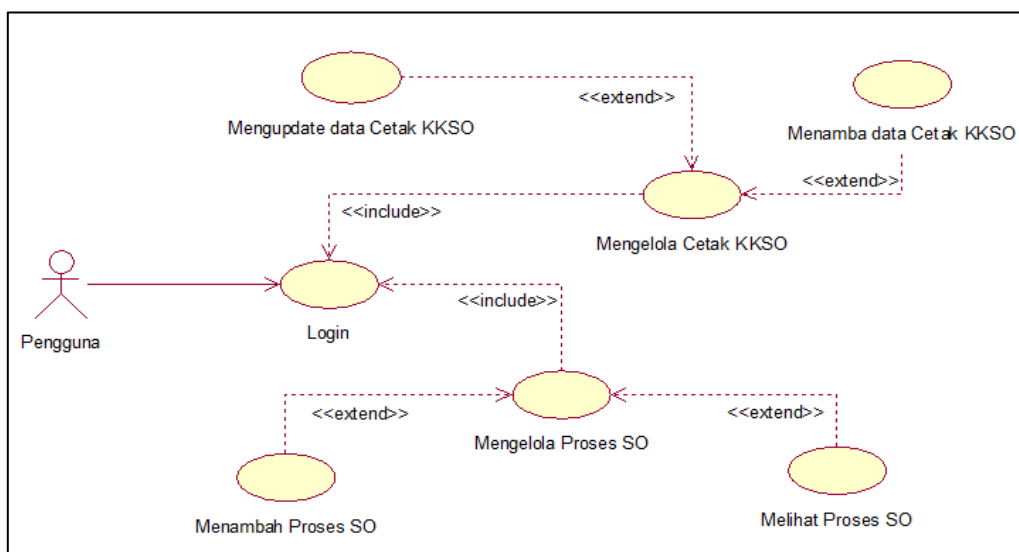
Pada tahap ini peneliti menggunakan *PostgreSQL* sebagai *database* utama dalam aplikasi *Stock Opname*. Tabel yang dibuat antara lain adalah tabel pengguna, tabel barang, tabel *Stock Opname*, dan tabel proses *Stock Opname*.

b. *Design Arsitektur Sistem*

Pada tahap ini berisikan tentang perancangan sistem dengan *Unified Modeling Language* (UML) yang terdiri dari *Use Case Diagram*, *Class Diagram*, dan *Activity Diagram*. UML merupakan standar penulisan untuk menggambarkan konsep proses bisnis, penjabaran kelas-kelas, skema *database* serta komponen-komponen yang diperlukan dalam membangun sebuah sistem informasi (Mubarak et al., 2019).



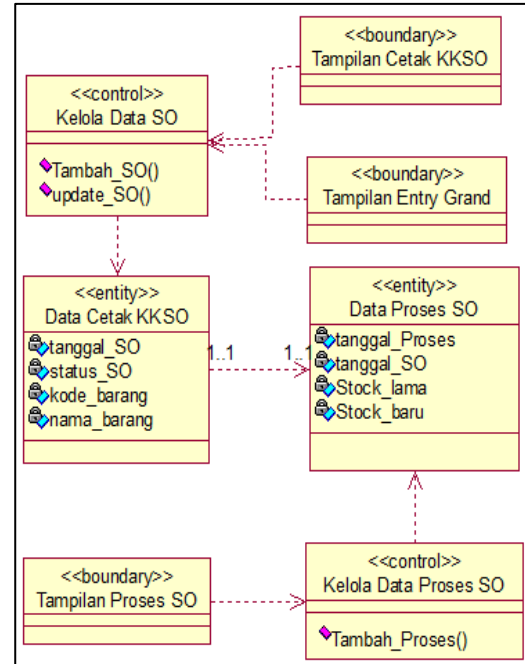
Gambar 3. Tahapan Model Proses *Waterfall*



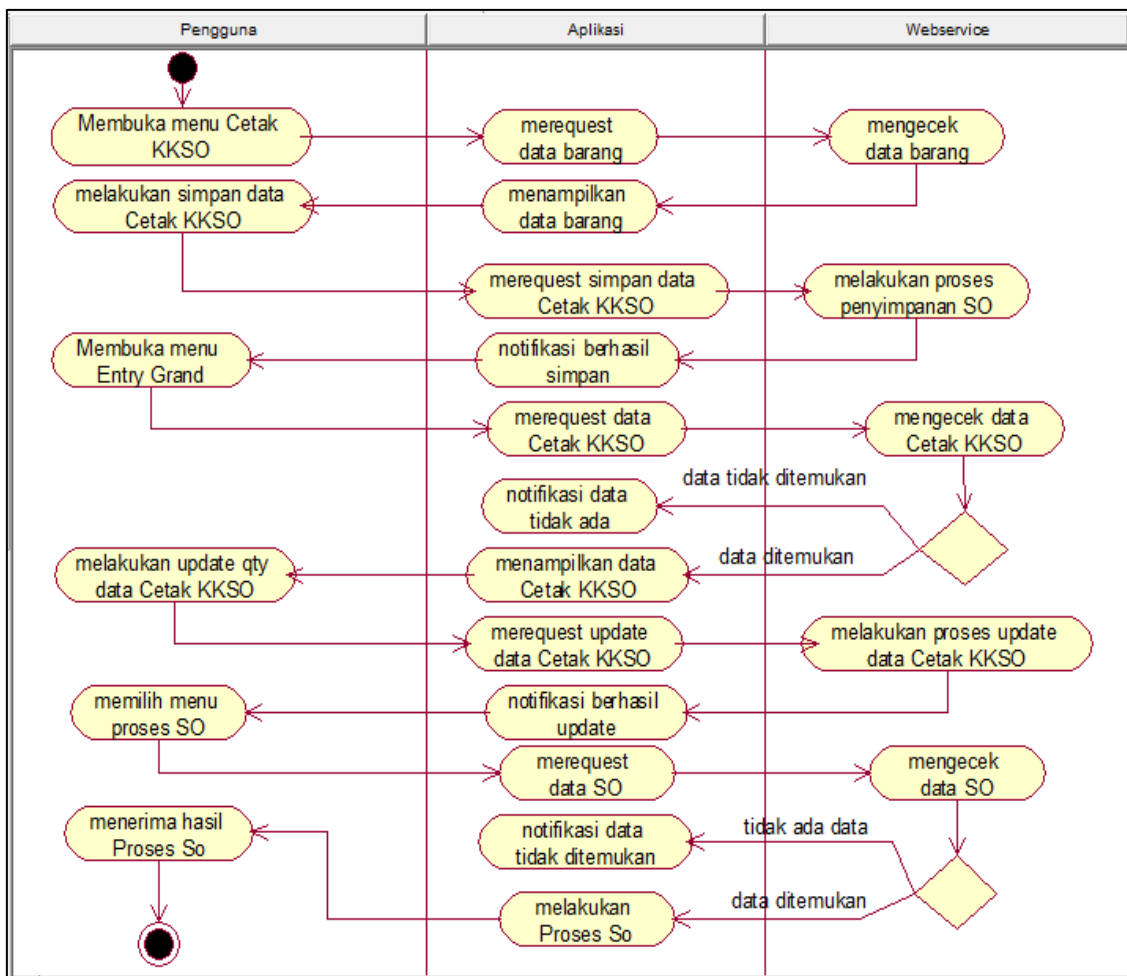
Gambar 4. *Use Case Diagram* Sistem *Stock Opname*

Gambar 4 Menjelaskan bahwa terdapat aktor bernama pengguna yang bertugas menjalankan sistem *Stock Opname*. Pengguna dapat mengelola data Cetak KKSO dan mengelola Proses *Stock Opname*. Pengguna akan menambahkan data Cetak KKSO setiap kali ingin melakukan Proses SO baru. Berikutnya adalah *Class Diagram* pada Gambar 5.

Gambar 5 menjabarkan *Class* yang ada dalam aplikasi *Stock Opname*. *Class Entity* digunakan untuk mewakili menu-menu pada aplikasi yang terhubung dengan *database* meliputi Cetak KKSO dan Proses SO. *Class Boundary* digunakan untuk mewakili tampilan *Graphic User Interface (GUI)* dalam suatu aplikasi program komputer. Pada Gambar 5 *Class Diagram* memiliki beberapa *Boundary* antara lain adalah tampilan Cetak KKSO, *Entry Grand*, dan tampilan Proses SO. Berikutnya *Class Controller* digunakan untuk mewakili suatu fungsi yang akan mengelola data dari *Entity* yang akan ditampilkan.



Gambar 5. *Class Diagram* Sistem *Stock Opname*



Gambar 6. *Activity Diagram* Sistem *Stock Opname*

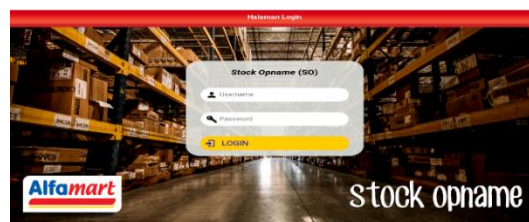
Gambar 6 Menjelaskan *Activity Diagram* pada sistem SO. Aktivitas untuk memulai *Stock Opname* dimulai dengan memasukkan data SO yang dibutuhkan. Data SO diperoleh dari data barang yang ada pada *database*. Setelah data SO baru dibuat, tahap selanjutnya adalah mencatat jumlah data barang tersedia pada kolom *Qty* dalam menu *Entry Grand* kemudian disimpan ke *database*. Terakhir adalah pengguna melakukan rekap data *Stock Opname* pada menu Proses SO, jika data yang dimasukkan oleh pengguna valid, maka Proses SO dapat dilakukan dan sebaliknya apabila tidak valid. Dari hasil perhitungan Proses SO inilah akan terlihat bilamana terdapat selisih jumlah data fisik dengan data yang ada pada *database*.

3. HASIL DAN PEMBAHASAN

Berdasarkan perancangan sistem yang telah dibuat, maka penelitian menghasilkan sebuah sistem bernama *Stock Opname* (SO) yang dibangun menggunakan *Kivy Framework* yang memungkinkan untuk dijalankan pada beragam sistem operasi seperti *Windows, Mac OS X, Linux, Ios, Android* hingga *Raspberry Pi* (Putra, 2015). Berikut adalah tampilan halaman *login* dari aplikasi program komputer *Stock Opname* pada Gambar 7.

Gambar 7 merupakan halaman *login* aplikasi program komputer *Stock Opname*. Dalam halaman *login* pengguna diharuskan memasukkan *username* dan *password* ke dalam sistem agar dapat masuk ke dalam aplikasi *Stock Opname*. Jika pengguna memasukkan *input data* yang salah, maka akan muncul notifikasi dari sistem agar pengguna memasukkan *input data* yang benar. Setelah pengguna berhasil *login*

pengguna akan masuk ke menu Cetak KKS0 yang terlihat pada gambar 8.



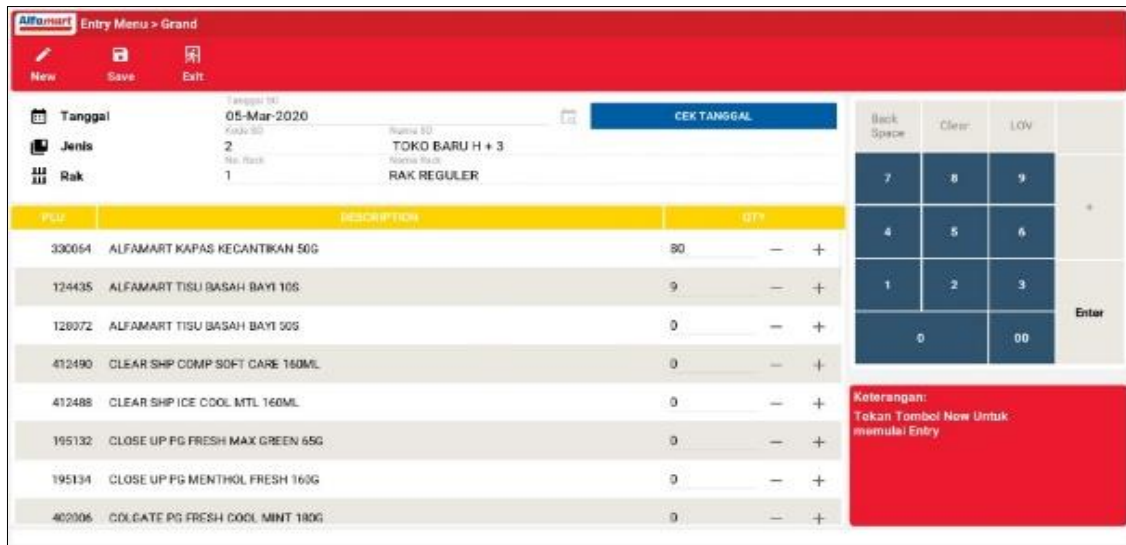
Gambar 7. Halaman *Login Stock Opname*

Gambar 8 merupakan menu Cetak KKS0. Menu Cetak KKS0 adalah menu yang digunakan untuk memasukkan data SO baru sebelum dilakukan Proses SO. Pada menu Cetak KKS0 pengguna diharuskan memasukkan tanggal SO, Jenis SO lalu rak SO. Jika data-data yang dimasukkan valid, maka data Cetak KKS0 akan ditampilkan, setelah itu data SO akan disimpan ke *database* untuk proses selanjutnya. Proses selanjutnya adalah mencatat jumlah barang yang tersedia pada kolom *Qty* di menu *Entry Grand* yang terlihat pada gambar 9.

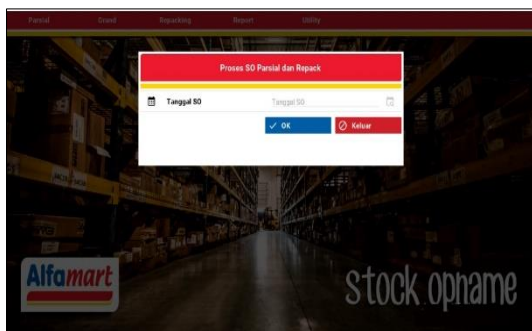
Gambar 9 merupakan menu *Entry Grand*. Menu *Entry Grand* adalah menu yang digunakan untuk memperbarui data jumlah (*quantity*) setelah pengguna memasukkan data SO pada menu Cetak KKS0. Pengguna akan melakukan perhitungan jumlah barang terlebih dahulu yang ada pada toko kemudian memasukkannya ke sistem SO. Setelah pengguna mengisi semua jumlah barang pada menu *Entry Grand*, maka pengguna akan melakukan penyimpanan data ke *database*. Jika sudah, akan dilanjutkan ke menu selanjutnya yaitu Proses SO yang terlihat pada Gambar 10.

PLU	DESCRIPTION	Qty	Qty
330064	ALFAMART KAPAS KECANTIKAN 50G	1	96
124435	ALFAMART TISU BASAH BAYI 10S	1	120
128072	ALFAMART TISU BASAH BAYI 50S	1	40
412490	CLEAR SHP ODMF SDFT CARE 160ML	1	36
412488	CLEAR SHP ICE COOL MTL 160ML	1	36
195132	CLOSE UP PG FRESH MAX GREEN 65G	4	48
195134	CLOSE UP PG MENTHOL FRESH 160G	4	48
402006	COLGATE PG FRESH COOL MINT 180G	1	72

Gambar 8. Menu Cetak KKS0 *Stock Opname*



Gambar 9. Menu Entry Grand Stock Opname



Gambar 10. Menu Proses SO Stock Opname

Gambar 10 Merupakan menu Proses SO. menu Proses SO digunakan untuk melakukan perhitungan *Stock Opname* setelah pengguna memasukkan data SO pada menu Cetak KKSO. Pengguna diminta memasukkan data tanggal SO kemudian sistem akan melakukan perhitungan dengan berbagai perintah *Query Structure Query Language (SQL)* yang di-request melalui *Web Service Flask* ke *database*. *Flask* merupakan *microframework* untuk bahasa pemrograman *python* yang dapat membantu dalam pembuatan situs dengan cepat meskipun dengan library yang sederhana (Samudera et al., 2015). Dengan *library Python*, memungkinkan untuk mengelola *HTTP request*.

Setelah program aplikasi *Stock Opname* selesai dibuat, maka langkah selanjutnya adalah melakukan *testing* agar aplikasi *Stock Opname* dapat dipastikan telah berjalan sesuai dengan rancangan. Untuk menguji kesesuaian aplikasi *Stock Opname* dengan rancangan di awal beserta dengan fungsionalitasnya, digunakan pengujian *Black Box Testing*. *Black Box Testing* adalah sebuah metode pengujian untuk

mengetahui apakah semua fungsi aplikasi telah berjalan sesuai dengan kebutuhan fungsional (Rouf, 2012). Dengan metode *blackbox testing*, kita dapat mengetahui permasalahan pada aplikasi yang telah dibuat (Saifudin, 2020). Adapun hasil pengujian *Black Box Testing* dari menu Cetak KKSO, *Entry Grand* dan *Proses SO* dapat dilihat pada tabel 2, tabel 3 dan tabel 4.

Tabel 1. Rancangan Pengujian *Black Box Testing*.

Item Uji	Jenis Pengujian
Menu Cetak KKSO	<i>Black Box</i>
Menu <i>Entry Grand</i>	<i>Black Box</i>
Menu Proses <i>Stock Opname</i>	<i>Black Box</i>

Tabel 2 dan tabel 3, memberikan informasi bahwa fungsi-fungsi yang terdapat pada menu Cetak KKSO sudah sesuai dengan rancangan di awal dan kebutuhan aplikasi *Stock Opname*. Hal ini dapat dilihat dari munculnya pesan bahwa data Cetak KKSO telah berhasil disimpan ketika pengguna selesai mengisi *form* sesuai dengan ketentuan dan melakukan simpan data.

Tabel 4 dan tabel 5, memberikan informasi bahwa fungsi-fungsi yang terdapat pada menu *Entry Grand* sudah sesuai dengan rancangan di awal dan kebutuhan aplikasi *Stock Opname*. Hal ini dapat dilihat dari muncul pesan bahwa data Cetak KKSO berhasil diperbarui ketika pengguna selesai mengisi *form* sesuai dengan ketentuan dan melakukan *update* data.

Tabel 6 dan tabel 7, menggambarkan bahwa fungsi-fungsi yang terdapat pada menu *Proses SO* sudah sesuai dengan rancangan di

awal dan kebutuhan aplikasi *Stock Opname*. Hal ini dapat dilihat dari munculnya pesan bahwa proses SO telah selesai dilakukan ketika pengguna selesai melakukan proses SO.

Dari hasil pengujian dengan *Black Box Testing* pada aplikasi program komputer *Stock Opname*, dapat diketahui bahwa setiap fungsi dan menu telah berjalan sesuai dengan rancangan di awal karena tidak ditemukan *error* atau *bug*.

Tabel 2. Pengujian Menu Cetak KKSO - Kasus dan Hasil Uji (Data Benar)

Data yang dimasukkan	Yang Diharapkan	Pengamatan	Kesimpulan
<i>input data</i> sesuai dengan ketentuan <i>form</i>	Data Cetak KKSO berhasil disimpan.	Muncul pesan data Cetak KKSO berhasil disimpan	(✓) diterima () ditolak

Tabel 3. Pengujian Menu Cetak KKSO - Kasus dan Hasil Uji (Data Salah)

Data yang dimasukkan	Yang Diharapkan	Pengamatan	Kesimpulan
<i>input data</i> tidak sesuai dengan ketentuan <i>form</i> .	Data Cetak KKSO tidak berhasil disimpan	Muncul pesan bahwa data Cetak KKSO tidak berhasil disimpan	(✓) diterima () ditolak

Tabel 4. Pengujian Menu *Entry Grand* - Kasus dan Hasil Uji (Data Benar)

Data yang dimasukkan	Yang Diharapkan	Pengamatan	Kesimpulan
<i>input data</i> sesuai ketentuan <i>form</i> .	Data Cetak KKSO berhasil <i>update quantity</i>	Muncul Pesan Data <i>quantity</i> Cetak KKSO berhasil diperbarui.	(✓) diterima () ditolak

Tabel 5. Pengujian Menu *Entry Grand* - Kasus dan Hasil Uji (Data Salah)

Data yang dimasukkan	Yang Diharapkan	Pengamatan	Kesimpulan
<i>input data</i> tidak sesuai dengan ketentuan <i>form</i> .	Data Cetak KKSO tidak berhasil <i>update quantity</i>	Muncul Pesan bahwa Data <i>quantity</i> Cetak KKSO tidak berhasil diperbarui .	(✓) diterima () ditolak

Tabel 6. Pengujian Menu Proses SO - Kasus dan Hasil Uji (Data Benar)

Data yang dimasukkan	Yang Diharapkan	Pengamatan	Kesimpulan
Tanggal SO yang ada pada <i>database</i> .	Proses perhitungan <i>Stock Opname</i> berhasil.	Muncul Pesan Proses SO telah berhasil.	(✓) diterima () ditolak

Tabel 7. Pengujian Menu Proses SO - Kasus dan Hasil Uji (Data Salah)

Data yang dimasukkan	Yang Diharapkan	Pengamatan	Kesimpulan
Tanggal SO yang tidak ada pada <i>database</i> .	Proses Perhitungan <i>Stock Opname</i> tidak berhasil.	Muncul Pesan bahwa Proses <i>Stock Opname</i> tidak berhasil.	(✓) diterima () ditolak

4. KESIMPULAN

Pada proses pembuatan *User Interface* dengan menggunakan *Kivy Framework* cukup rumit dan kompleks namun mudah untuk dipelajari, dibutuhkan ketelitian dan logika pemrograman untuk membuat *layout* aplikasi. Selain itu, *Kivy Framework* juga tidak

memerlukan spesifikasi komputer yang tinggi untuk menjalankannya. Kelebihannya juga adalah dapat berjalan *multiplatform* pada *desktop* maupun *mobile*, dan hampir di semua sistem operasi yang umum digunakan seperti *Windows*, *Linux*, *Mac OS X*, *ios*, *Android* dan *Raspberry Pi*. Aplikasi program komputer *Stock Opname* yang sudah dibuat dengan *Kivy*

Framework terbukti dapat berjalan secara *cross-platform* yakni pada *platform desktop* dan *Android* hanya dengan satu kali pembuatan kode program. Dengan hasil ini, maka toko yang menggunakan perangkat komputer tidak perlu mengganti perangkat dan toko yang baru akan dibuka dapat menggunakan perangkat *Tablet*

5. REFERENSI

- Apulina Sembiring, E. (2019). Pengaruh Metode Pencatatan Persediaan Dengan Sistem Periodik Dan Perpetual Berbasis SIA Terhadap Stock Opname Pada Perusahaan Dagang Di PT Jasum Jaya. *Accumulated Journal (Accounting and Management Research Edition)*, 1(1), 69–77.
<https://doi.org/10.22303/ACCUMULATED.1.1.2019.69-77>
- Architectural Overview — Kivy 1.11.1 documentation*. (n.d.).
- Bhoiyarkar, A., Solanki, A., & Balbudhe, A. (2019). Application Development using Kivy Framework. *Ijarccce (International Journal of Advanced Research in Computer and Communication Engineering)*, 8(2), 53–58.
<https://doi.org/10.17148/ijarcce.2019.82.09>
- Ipan A'shary Wulung, Ekojono, & tiqah Nurul Asri. (2014). Sistem Informasi Logistik Stock Opnamemenggunakan Quick Response Code Berbasis Android. *Jurnal Informatika Polinema*, 1(1 November), 23–29.
- Mubarak, A., Metro, J. J., & Selatan, K. T. (2019). Rancang Bangun Aplikasi Web Sekolah Menggunakan UML (Unified Modeling Language) Dan Bahasa Pemrograman Php (Php Hypertext Preprocessor) Berorientasi Objek. *JIKO (Jurnal Informatika Dan Komputer)*, 2(1), 19–25.
<https://doi.org/10.33387/JIKO.V2I1.1052>
- Putra, A. W. (2015). *Kivy - Framework Untuk Kembangkan Aplikasi Mobile Multiplatform Menggunakan Bahasa Pemrograman Python | TeknoJurnal*.
<https://teknojurnal.com/kivy-framework-python/>
- Qutsiah, S. A., Sophan, M. K., & Hendrawan, Y. F. (2016). Aplikasi Pembelajaran Matematika Dasar Bangun Datar Menggunakan Python Pada Perangkat Bergerak. *SCAN - Jurnal Teknologi Informasi Dan Komunikasi*, XI(3), 13–22.
- Rani Susanto, & Andriana, A. D. (2016). Perbandingan Model Waterfall Dan Prototyping Untuk Pengembangan Sistem Informasi. *Majalah Ilmiah UNIKOM*, 14(1), 41–46.
- Rouf, A. (2012). Pengujian Perangkat Lunak Dengan Menggunakan Metode White Box Dan Black Box. *Journal Himsyatech*, 8(1).
<http://ojs.stmik-himsya.ac.id/index.php/JHT/article/view/28>
- Ruli, A. R. (2017). Implementasi Aplikasi Pendaftaran Dan Pembayaran Kontrakan Ahmad Rais Berbasis Desktop VB .NET Dan Microsoft Access. *Paradigma - Jurnal Komputer Dan Informatika*, 19(1), 9–19.
<https://doi.org/10.31294/P.V19I1.996>
- Saifudin, A. (2020). Penerapan Teknik Boundary Value Analysis untuk Pengujian Aplikasi Penjualan Menggunakan Metode Black Box Testing. *Jurnal Informatika Universitas Pamulang*, 5(2), 129–135.
- Samudera, N. A., Mulyana, A., & Rakhman, E. (2015). Perancangan Sistem Keamanan Ruang Menggunakan Raspberry Pi(bagian: Aplikasi). *EProceedings of Engineering*, 2(2).
<https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/1342>
- Weeks, D. P. C. L. E. Y. N. to K. in 20. (2015). 濟無No Title No Title. *Dk*, 53(9), 1689–1699.
<https://doi.org/10.1017/CBO9781107415324.004>