

Sistem Inventori Dan Manajemen Kebutuhan Dalam Berbelanja

Yeremia Danang Priambodo¹, Oscar Wongso²

Program Studi Teknik Informatika Universitas Kristen Maranatha
Jl. Surya Sumantri No.65, Bandung

¹priambododo@outlook.com

²oscar.wongso@itmaranatha.org

Abstract — *Many times we found foods in storage that has expired for days or even months. We also saved foods and forgot where the storage is. This thing can make foods become uneatable and can be expired if we forgot to use them before it's expired date. Therefore, an application for food's inventory is made using Xamarin.Android and Web API. With this application, users can manage storages in their home so storage become well-organized and can display inventories data that will or have expired for reducing foods that stale or wasted.*

Keywords— *Inventory, Storages, Android, Web Service, Expired Foods*

I. PENDAHULUAN

Berbelanja adalah kegiatan yang dilakukan oleh setiap orang untuk memenuhi kebutuhannya. Pada saat berbelanja terdapat beberapa kendala yang ditemukan seperti, proses untuk membandingkan harga, dan proses menyimpan barang belanjaan ke tempat penyimpanan jika dilakukan secara manual.

Saat ini aktifitas berbelanja masih dilakukan secara manual, mulai dari pendataan lokasi penyimpanan barang, pengecekan tanggal kedaluwarsa barang, dan pencatatan pembelian serta penggunaan barang. Pencatatan yang masih dilakukan manual ini menyebabkan kurang efisiennya pengolahan data pada proses pembelian, penggunaan, dan penyimpanan barang.

Oleh karena itu, maka diperlukan sebuah aplikasi mobile yang diharapkan dapat membantu pengguna dalam pencatatan pembelian barang, data lokasi penyimpanan barang, dan laporan per periode kegiatan berbelanja secara lengkap. Sehingga pengaksesan data barang dapat diproses lebih cepat dan mudah. Pengguna juga dapat menambahkan pengguna lain untuk menggunakan penyimpanan yang sama dalam sistem yang membuat pengguna dapat memakai multiple device. Selain itu, aplikasi juga dapat memberikan notifikasi kepada pengguna mengenai barang – barang yang akan mendekati tanggal kedaluwarsa. Dengan adanya fitur ini diharapkan dapat mengurangi pembuangan barang – barang yang seharusnya masih dapat digunakan sebelum basi atau rusak.

II. KAJIAN LITERATUR

A. Xamarin Android

Xamarin merupakan android development tool yang menggunakan bahasa pemrograman C#. Melalui Xamarin pengguna dapat menggunakan kode Java, frameworks dan custom control yang sudah ada melalui automatic binding generator Xamarin. Xamarin memiliki dua IDE yang terdiri dari Xamarin terintegrasi Microsoft Visual Studio dan Xamarin Studio [1]. Pengembang dapat mengembangkan aplikasi secara multiplatform baik itu Windows Phone, Android, dan iOS dengan menggunakan kode pemrograman C# yang sama [2].

B. Web Service

Web Service adalah layanan yang tersedia di Internet. *Web Service* menggunakan format standar *XML* untuk pengiriman pesannya. *Web Services* juga tidak terikat kepada bahasa pemrograman atau sistem operasi tertentu. *Web Services* adalah antar muka yang mendeskripsikan koleksi yang dapat diakses dalam jaringan menggunakan format standar *XML* untuk pertukaran pesan. *Web Services* mengerjakan tugas yang spesifik. *Web Services* dideskripsikan menggunakan format standar notasi *XML* yang disebut *services description* [3].

C. ASP.Net MVC

ASP.Net MVC (Model-View-Controller) adalah pola arsitektur berbasis ASP.Net framework. MVC sendiri membagi sebuah aplikasi menjadi tiga bagian utama yaitu *model*, *view* dan *controller* [4]:

1. *Model*: Merupakan kelas yang berisi data. Kelas ini dapat secara praktis menjadi kelas apapun yang bisa menyediakan data. *Model* bisa menjadi *entity framework* yang dihasilkan oleh entitas, *collection*, *generic* atau *generic collection*.

2. *Controllers*: merupakan kelas yang akan dipanggil ketika pengguna meminta akses. Tugas utama controllers adalah untuk menghasilkan objek dari kelas model dan membuatnya menjadi beberapa tampilan. *Controller* memberi tahu tampilan untuk menghasilkan markup dan merender markup tersebut di browser pengguna.

3. *Views*: adalah halaman sederhana berisi kode *HTML* dan *C#* yang akan menggunakan objek dari *server side (model)* untuk mengekstrak data, menyesuaikan markup *HTML* dan dikirim ke *browser* pengguna.

D. Newtonsoft JSON.Net

JSON (JavaScript Object Notation) adalah format pertukaran data yang sederhana. Sangat mudah bagi manusia untuk membaca dan menulis. Sangat mudah untuk mesin untuk mengurai dan menghasilkan [5]. Dalam pengembangan aplikasi ini akan digunakan *JSON.Net Newtonsoft* yang bertipe *Request* dan *Response*.

E. ZXing.Net Mobile

ZXing (dibaca “zebra crossing”) Net Mobile merupakan sebuah *library open-source* berbasis *C#/.Net*. *ZXing* menggunakan kamera built-in pada ponsel untuk memindai dan melakukan *decode barcode* pada perangkat, tanpa berkomunikasi dengan server menggunakan *Xamarin.iOS*, *Xamarin.Android* dan *Windows Phone* [6]. *ZXing* dapat melakukan *decode barcode* dalam berbagai format *barcode* meliputi: EAN, UPC, Code128, ITF-14, Code39, QR-Code, DataMatrix Code, Aztec, dll.

F. Firebase Cloud Messaging

Firebase Cloud Messaging (FCM) adalah suatu *service* yang menyediakan pengiriman, *routing* dan *queueing* pesan antar aplikasi mobile dengan aplikasi server. *FCM* bertindak sebagai perantara antara pengirim pesan dengan *clients*. Melalui *FCM* pengguna dapat mengirimkan pesan langsung ke *client apps* melalui *Firebase Console Notifications GUI* [7].

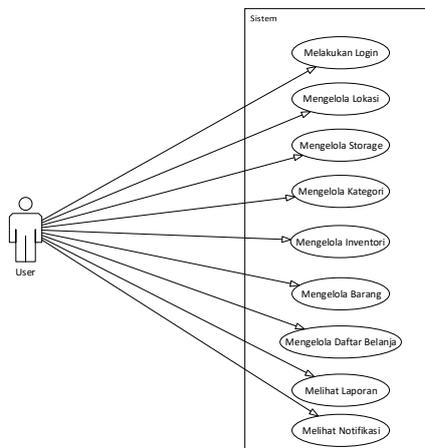
III. ANALISIS DAN PEMODELAN

A. Arsitektur Sistem

Berikut ini adalah arsitektur sistem yang meliputi: *use case diagram*, *activity diagram*, dan *aplikasi*.

1. Use Case Diagram

Rancangan *Use Case Diagram* secara menyeluruh dapat dilihat pada Gambar 1. Hak akses di dalam sistem ini ada *admin* dan pengguna yang mengunduh aplikasi ini dan sudah mendaftarkan akunnya. *Admin* dapat melakukan *login*, mengelola kategori dasar pada aplikasi, melihat laporan dan juga mengelola *user*. *User* yang sudah mendaftarkan akunnya dapat melakukan *login*, mengelola lokasi, mengelola *storage*, mengelola *category*, mengelola barang, mengelola daftar belanja, melihat laporan dan melihat notifikasi.



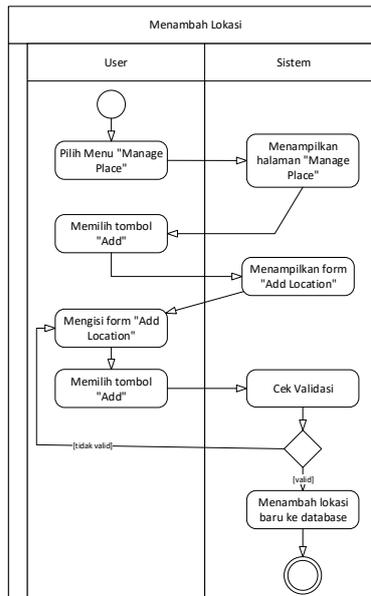
Gambar 1 Use Case Pengembangan Aplikasi

2. Activity Diagram

Activity Diagram untuk aplikasi dapat dilihat dalam Gambar 2-5. Penjelasan detail hanya diberikan untuk bagian utama yang berhubungan dengan sistem penambahan *location*, *storage*, *inventory*, dan menggunakan.

▪ Menambah Lokasi Baru

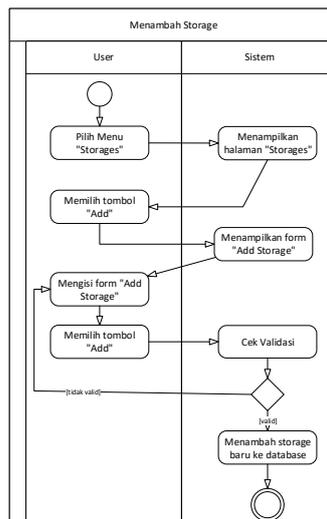
Pengguna diharuskan menambah minimal satu lokasi sebelum lanjut menambahkan *storage*. Untuk menambah lokasi baru, pengguna harus masuk ke menu “Manage Place” yang muncul setelah menekan tombol “3 strips drawer” pada halaman utama. Lalu aplikasi akan menunjukkan halaman “Manage Place”. Setelah itu pengguna harus menekan tombol “Add” dan mengisi form “Add Location”. Jika pengguna menekan tombol “Add” pada halaman form “Add Location” dan data form sudah valid, maka lokasi baru tersedia dan ditampilkan di jendela list halaman “Manage Place”.



Gambar 2 Activity Diagram Menambah Lokasi

▪ Menambah Storage Baru

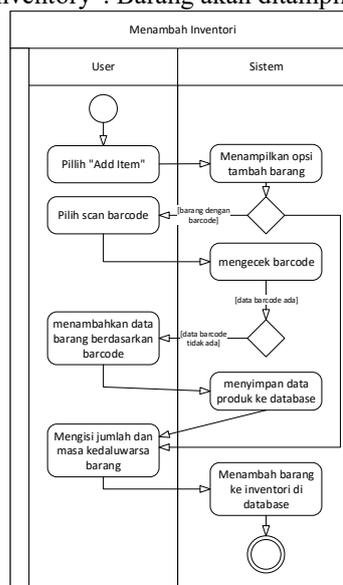
Pengguna diharuskan menambah minimal satu *storage* sebelum lanjut menambahkan barang ke inventori. Untuk menambah *storage* baru, pengguna harus masuk ke menu “Storages” yang muncul setelah menekan tombol “3 strips drawer” pada halaman utama. Lalu aplikasi akan menampilkan halaman “Storages”. Setelah itu pengguna harus menekan tombol “Add” dan mengisi form “Add Storage”. Jika pengguna menekan tombol “Add” pada halaman form “Add Storage” dan data form sudah valid, maka *storage* baru tersedia dan ditampilkan di jendela list halaman “Storages”.



Gambar 3 Activity Diagram Menambah Storage

▪ *Menambah Inventory Baru*

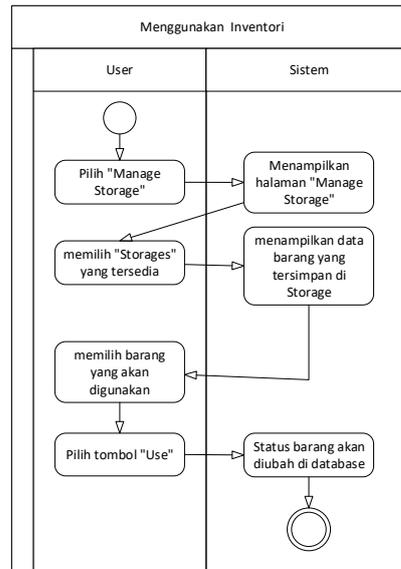
Setelah barang dibeli dan akan dimasukkan ke penyimpanan, pengguna dapat menambahkan barang dengan melakukan scan terhadap barang yang memiliki barcode atau menggunakan cara manual. Setelah mengisi form, memilih tempat penyimpanan, dan menekan tombol “Add to Inventory”. Barang akan ditampilkan di halaman “Use Item”.



Gambar 4 Activity Diagram Menambah Inventori

▪ *Menggunakan Inventory*

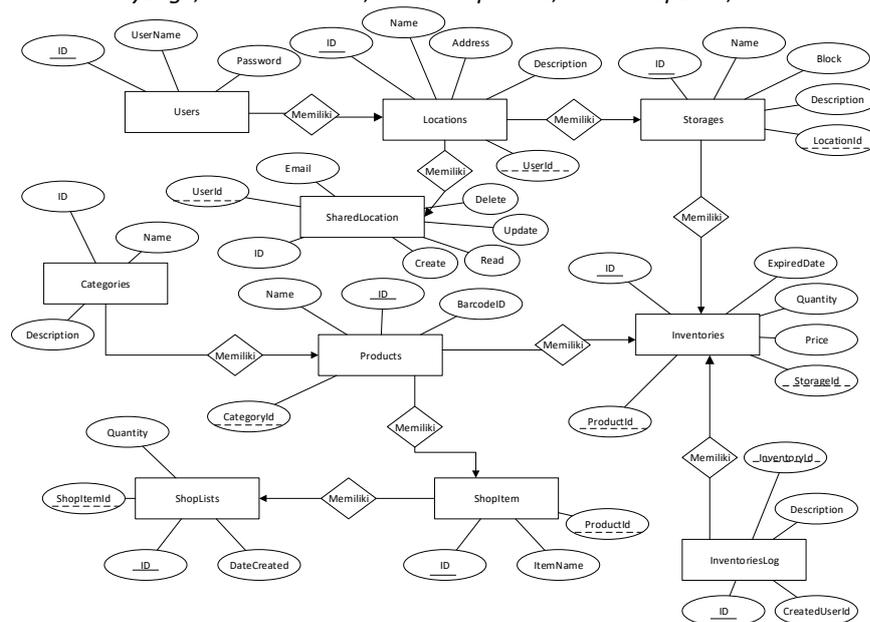
Untuk menggunakan barang, pengguna memilih tombol “Manage Storage” untuk masuk ke halaman “Manage Storage” dan memilih salah satu storage yang ada di daftar. Jika sudah ditampilkan, pengguna dapat memilih salah satu barang di list dan menekan tombol “Use”.



Gambar 5 Activity Diagram Menggunakan Inventori

3. Entity Relationship Diagram

Gambar 6 adalah penggambaran *database* Aplikasi Android ShopDiary App. Pada pembuatan *website* ini terdapat 10 tabel data, yaitu tabel *Users*, tabel *Locations*, tabel *SharedLocations* tabel *Storages*, tabel *Inventories*, tabel *Inventories Logs*, tabel *InventoryLogs*, tabel *Products*, tabel *ShopItems*, tabel *ShopLists*, dan tabel *Categories*.



Gambar 6 .EntityRelationship Diagram

B. Analisis Proses Bisnis

Berikut ini adalah proses bisnis penggunaan aplikasi. Proses bisnis meliputi proses penambahan lokasi dan penyimpanan, penambahan barang ke inventori, menggunakan barang, dan membuat daftar belanja.

1. Proses Bisnis Penambahan Lokasi dan Penyimpanan

Setelah melakukan login, pengguna diharuskan menambahkan data lokasi berupa bangunan tempat penyimpanan akan ditambahkan. Setelah lokasi ditambahkan, pengguna diharuskan menambahkan data lokasi penyimpanan (storage) berdasarkan lokasi yang sudah ditambahkan sebelumnya. Setelah itu pengguna dapat menambahkan data barang.

2. Proses Bisnis Penambahan Inventory

Setelah pengguna menambah lokasi penyimpanan, pengguna dapat melanjutkan untuk menambah inventory. Akan ada 2 tipe barang yang dapat ditambahkan ke lokasi penyimpanan, yaitu barang yang memiliki *barcode* dan tidak memiliki *barcode*. Untuk barang yang memiliki *barcode* pengguna dapat menggunakan *barcode scanner*, sedangkan barang yang tidak memiliki *barcode* dapat mengisi form secara manual.

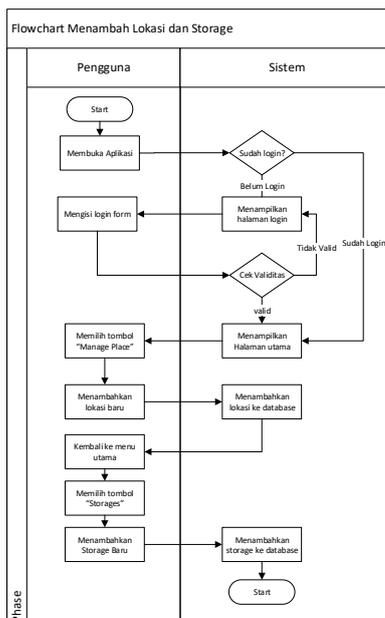
IV. IMPLEMENTASI

A. Implementasi Proses Bisnis

Pada bagian ini akan dijelaskan tentang implementasi proses bisnis pada sistem aplikasi yang dibagi menjadi 2 bagian, yaitu Penambahan Lokasi Penyimpanan dan Inventory.

1. Penambahan Lokasi dan Penyimpanan

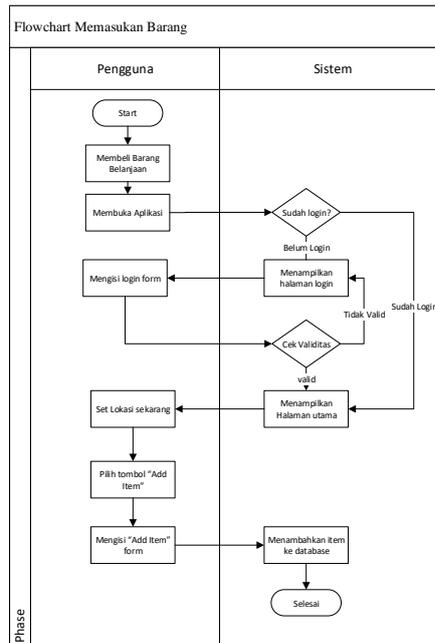
Proses dimulai dengan pengguna yang melakukan login. Jika pengguna sudah berhasil login, maka hak akses untuk menggunakan aplikasi diberikan. Sebelum dapat menambahkan item ke aplikasi, Pengguna diharuskan menambah minimal 1 lokasi dan minimal 1 storage untuk setiap lokasi. Setelah halaman utama ditampilkan, pengguna harus memilih tombol “Manage Place” setelah itu melakukan penambah lokasi. Jika lokasi sudah ditambahkan, pengguna harus kembali ke halaman utama dan memilih tombol ‘Storage’. Setelah itu, storage dapat ditambahkan oleh pengguna.



Gambar 7 Implementasi Proses Bisnis Penambahan Lokasi dan Penyimpanan

2. Menggunakan Inventory

Proses ini dimulai setelah lokasi dan storage sudah ditambahkan serta barang belanjaan yang akan di simpan sudah ada. Pengguna dapat menambahkan item dengan atau tanpa barcode. Dalam form “Add Item” pengguna harus memilih storage tempat item akan disimpan.

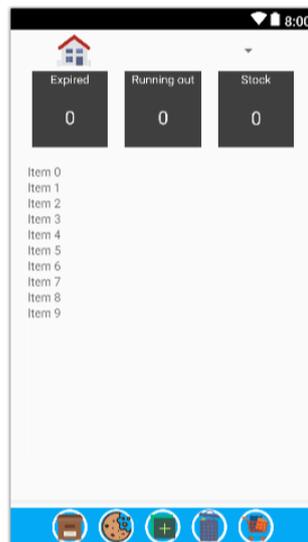


Gambar 8 Implementasi Proses Bisnis Menggunakan Inventory

B. Implementasi Program Aplikasi

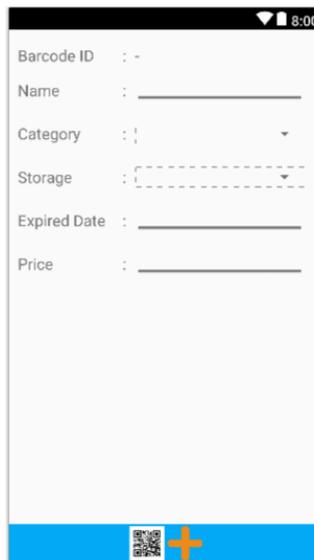
Implementasi keseluruhan sistem dilakukan dalam bahasa pemrograman C#, berbentuk aplikasi Android. Pada pembuatan aplikasi, digunakan Visual Studio 2017 dengan minimal API 24 untuk Client (Front End), sedangkan untuk Server (Back End) memanfaatkan sistem basis data SqlServer dengan memanfaatkan ASP.Net Web Api sebagai fungsi API untuk Web Service dan memanfaatkan JSON untuk mentransmisikan data yang diolah oleh Server kepada Client.

Pada halaman utama di Gambar 9, ditampilkan *Expired Item*, *Running Out* dan *Stock counter* yang menghitung data item secara keseluruhan. Pada halaman ini juga terdapat *recycler view* yang menampilkan *quick view* dari item-item yang akan / sudah kedaluwarsa. Dibagian bawah *recycler view* di buat lima *shortcut menu* "Storage", "User Item", "Add Item", "ShopList", dan "RunOut List".



Gambar 9 Tampilan Main Aplikasi

Pada halaman “Add Inventory” pada Gambar 10, pengguna dapat menambah produk yang akan ditambah ke storage dengan memilih storage tempat item akan disimpan terlebih dahulu.



Gambar 10 Tampilan Halaman Add Inventory

V. PENGUJIAN

Pada bab ini akan membahas uji coba hasil penelitian. Uji coba hasil penelitian menggunakan metode pengukuran kualitatif melalui survei berdasarkan rancangan pertanyaan yang dibuat. Survei diberikan kepada responden yang sudah menggunakan aplikasi ShopDiary App. Responden secara acak diambil untuk mengamati aplikasi ShopDiary App lalu mengisi survei melalui web.

Survei dilakukan pada lingkungan Universitas Kristen Maranatha dengan jumlah responden sebanyak 20 (dua puluh) orang. Berikut adalah hasil dari survei:

TABEL I
 HASIL SURVEI 1

Pertanyaan	Hasil (rata-rata)
Apakah pendaftaran akun dan login mudah dilakukan?	4.3
Apakah penambahan lokasi mudah dilakukan?	4.3
Apakah fitur storage sharing membantu dalam pendataan?	4.2
Apakah penambahan storage mudah dilakukan?	4.3
Apakah penambahan barang ke penyimpanan mudah dilakukan?	4.2
Apakah penggunaan fungsi daftar belanja mudah dilakukan?	4
Apakah laporan kegiatan manajemen inventori informatif?	4
Apakah pemberitahuan adanya barang kedaluwarsa informatif?	4
Apakah secara keseluruhan aplikasi ShopDiaryApp mudah digunakan	4.2

Lalu, untuk kolom opini hasil yang telah dirangkum adalah sebagai berikut.

TABEL III
 HASIL SURVEI 2

Pertanyaan	Hasil (rangkuman)
Saran	Fitur laporan lebih dibuat lengkap jangan terlalu general. Gunakan warna yang Cerah. Tambahin dong fitur laporannya, sama keranjang belanja diberesin.. butuh :) Tampilan barang-barang bedain warna dong buat yang udah expired ato belom.. Jangan terlalu banyak pindah halaman, jadi rumit. Harus ada shortcut tiap halaman

Pertanyaan	Hasil (rangkuman)
	Bagus :) tambahin push notification benerin validasinya tolong fitur laporannya, bagus kalo pake graph
Komentar	Warnanya kurang semangat Fitur lprn kurang lengkap. Warna tampilan sedih Warnanya Asik :) sprtinya bgus kalau fitur shoplistnya ada.. jangan ada di menu aja tapi gabisa dipake Pilih tema tampilan bagus, tapi jadi gelap Bagus :) kayaknya bagus buat mama fitur barcode membantu, cuma terlalu banyak halaman

VI. KESIMPULAN DAN SARAN

A. Kesimpulan

Kesimpulan dari hasil analisis, pembuatan dan implementasi aplikasi ini adalah sebagai berikut:

1. Melalui aplikasi ShopDiaryApp, pengguna dapat mengelola data penyimpanan barang berdasarkan lokasi yang ditambahkan dengan mudah. Pengguna juga dibantu dengan adanya fitur *storage sharing*.
2. Fitur *barcode scanner* mempermudah pengguna dalam memasukkan data barang yang memiliki barcode ke aplikasi.
3. Pengguna dapat melihat daftar data barang yang sudah/akan kedaluwarsa dalam tampilan berupa daftar dan notifikasi. Notifikasi yang ditampilkan adalah pesan yang menunjukkan bahwa ada barang yang mendekati masa kedaluwarsa.

B. Saran

Adapun saran untuk pengembangan aplikasi ini:

1. Perlu dibuat *database* lokal sehingga aplikasi dapat diakses dengan atau tanpa internet.
2. Ada kendala dalam proses decode barcode berformat EAN-13 Pengguna harus menyesuaikan permukaan barcode dengan tampilan pada kamera dan kondisi cahaya yang minim membuat proses decode menjadi lebih lama. Tampilan barcode pada aplikasi harus jelas, tidak seperti barcode berformat 2D dikarenakan barcode EAN-13 memiliki batang yang relative lebih kecil sehingga barcode scanner harus mendapatkan hasil scan yang presisi. Pengujian dilakukan dengan cara melakukan scan pada barcode berformat EAN-13 dan 2D yang berdampingan. Hasil scan lebih cepat melakukan decode terhadap barcode 2D.
3. Penambahan fitur dimana pengguna dapat membandingkan harga barang yang akan dibeli berdasarkan harga barang-barang yang sudah dibeli di toko yang berbeda sehingga pengguna mendapatkan harga yang termurah.
4. Penambahan fitur untuk laporan keuangan pengguna berdasarkan barang yang dibeli.
5. Fitur *generate barcode* untuk pengguna yang ingin menambahkan data *barcode* sendiri.
6. Pembuatan aplikasi ShopDiaryApp untuk platform *IOS*, *WindowsPhone* dan juga *Web version*.

DAFTAR PUSTAKA

- [1] Xamarin Microsoft, "All-Guides," Xamarin, [Online]. Available: <https://developer.xamarin.com/guides/>. [Accessed 5 March 2017].
- [2] J. C. Dunn and D. Britch, "An Introduction to Xamarin.Forms," Microsoft, 24 February 2018. [Online]. Available: <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/get-started/introduction-to-xamarin-forms>. [Accessed 10 December 2018].
- [3] C. Darie and W. Barnett, Build Your Own ASP.Net Website Using C# & VB.Net, SitePoint, 2008.
- [4] R. Anderson, A. Scott and A. Pasic, "ASP.NET Web API," Microsoft, 17 January 2012. [Online]. Available: <https://docs.microsoft.com/en-us/aspnet/web-api/>. [Accessed 23 2017 July].
- [5] Newtonsoft, "JSON.Net," Newtonsoft, 2017. [Online]. Available: <http://www.newtonsoft.com/json>. [Accessed 8 June 2017].
- [6] J. Dick, "ZXing.Net.Mobile," Xamarin, 23 January 2014. [Online]. Available: <https://github.com/Redth/ZXing.Net.Mobile>. [Accessed 15 April 2017].
- [7] Google, "Firebase Cloude Messaging," Google, 1 11 2018. [Online]. Available: <https://firebase.google.com/docs/cloud-messaging/>. [Accessed 1 11 2018].
- [8] O. Wongso, "QR Code, Face Recognition, and Google Location as Alternative Solution for Attendances in Private Colouring Studio," JuTISI, vol. II, no. 3, pp. 309-318, 2016.
- [9] M. C. Wibisono, A. Noertjahyana and A. Handoyo, "Pembuatan Aplikasi Pencatatan Stock Dengan Menggunakan Barcode Pada Android".
- [10] F. Wahyutama, F. Samopa and H. Suryotrisongko, "Penggunaan Teknologi Augmented Reality Berbasis Barcode sebagai Sarana Penyampaian Informasi Spesifikasi dan Harga Barang yang Interaktif Berbasis Android, Studi Kasus pada Toko Elektronik ABC Surabaya," Jurnal Teknik Pomits Vol.2, No.3, pp. 481-486, 2013.
- [11] B. Sempf, C. Sphar and S. R. Davis, "C# 5.0 All-in-One for Dummies," 2010.
- [12] Microsoft, Microsoft Application Architecture Guide: Patterns & Practices, Microsoft, 2009.
- [13] I. Matkov, B.-Y. Fu and D., "Push Notifications," Microsoft, 25 10 2018. [Online]. Available: <https://docs.microsoft.com/en-us/appcenter/push/>. [Accessed 1 11 2018].
- [14] H. Ibrahim, "ASP.Net BoilerPlate Documentation," Volosoft, 2013-2017. [Online]. Available: <https://aspnetboilerplate.com/>. [Accessed 28 February 2017].
- [15] Scandit, "Types of Barcodes: Choosing the right barcode: EAN, UPC, Code128, ITF-14 or Code39?," Scandit, 2016. [Online]. Available: <https://www.scandit.com/types-of-barcodes-choosing-the-right-barcode-type-ean-upc-code128-itf-14-or-code39/>. [Accessed 4 August 2017].
- [16] Object Management Group, "Business Process Model and Notation Resource Page," 9 June 2014. [Online]. Available: <http://www.omg.org/bpmn/index.htm>. [Accessed 12 Jan 2016].