

PEMANFAATAN TEKNOLOGI FIREBASE DALAM PENGEMBANGAN APLIKASI PENGELOLAAN STOK BARANG BERBASIS *MOBILE* PADA RUMAH MAKAN NAKAMSE MALANG

Ryan Leonardo^{*1}, Issa Arwani², Dian Eka Ratnawati³

^{1,2,3}Program Studi Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹ryanleonardo@student.ub.ac.id, ²issa.arwani@ub.ac.id, ³dian_ilkom@ub.ac.id

*Penulis Korespondensi

(Naskah masuk : 01 Juli 2020, diterima untuk diterbitkan : 30 Juli 2020)

Abstrak

Nakamse merupakan salah satu rumah makan yang menawarkan konsep unik dan menarik kepada konsumennya. Oleh sebab itu, Nakamse menjadi salah satu usaha kuliner yang kini sedang tumbuh sangat pesat di kota Malang. Hal tersebut dapat dilihat dari bertambahnya jumlah *store* yang dimiliki dalam kurun waktu dua tahun terakhir. Seperti halnya rumah makan lain, setiap *store* Nakamse membutuhkan pasokan bahan makanan dari gudang utama yang mereka miliki. Namun dalam proses pendistribusian barang tersebut sering kali terjadi duplikasi data dan kesalahan pencatatan barang, hal tersebut terjadi diakibatkan manajemen gudang yang masih dilakukan dengan cara manual. Oleh karena itu, diperlukan sebuah perangkat lunak yang dapat melakukan pengelolaan terhadap stok barang yang ada di gudang. Perangkat lunak ini dikembangkan dengan menggunakan metode waterfall dan memanfaatkan Firebase *realtime database* sebagai tempat penyimpanan data. Dari hasil observasi dan wawancara, didapatkan beberapa kebutuhan fungsional yang kemudian dimodelkan dalam bentuk diagram UML. Kemudian dilanjutkan dengan melakukan implementasi kode program menggunakan bahasa pemrograman Java. Setelah tahap perancangan dan implementasi selesai dilakukan maka selanjutnya dilakukan pengujian dengan menggunakan metode *blackbox* dan *usability testing*. Pada pengujian *blackbox* mendapatkan hasil 100% valid pada setiap fungsionalnya. Sedangkan pada pengujian *usability* yang dilakukan dengan metode *system usability scale* mendapatkan skor sebesar 78,5. Skor tersebut mencerminkan bahwa aplikasi pengelolaan stok barang masuk dalam predikat C (*Good*) dan rentang penerimaan berada pada kategori *acceptable*.

Kata Kunci: Pengelolaan stok barang, metode waterfall, Firebase *realtime database*, *unified modelling language (UML)*, *blackbox testing*, *system usability scale (SUS)*

THE UTILIZATION OF FIREBASE TECHNOLOGY IN DEVELOPING MOBILE BASED STOCK MANAGEMENT APPLICATIONS AT NAKAMSE RESTO MALANG

Abstract

Nakamse is one of the restaurants that offers unique and interesting concept to its customers. Therefore, Nakamse has become one of the culinary businesses that is now growing rapidly in Malang. This can be seen from the increasing number of stores owned in the past two years. Like other restaurants, every Nakamse's store needs supply of food from their main warehouse. However, in the process of distributing the goods, some duplication data and error in recording the goods may often happen. These such errors could happen due to warehouse management that is still done manually. Thus, a software that can perform some tasks to manage the goods in the warehouse is needed. This

software was developed using the waterfall method and utilizes Firebase real-time database as a data storage. The observation and interview results some functional requirements which are then modelled in UML diagram, then proceeded to implement the source code using Java. After the design system and implementation were completed, then the software is tested using blackbox and usability testing. The blackbox testing was 100% valid on each functional while the usability testing using System Usability Scale (SUS) scored 78,5. The score reflects that this software is categorized as C (Good) and acceptability range is in the acceptable category.

Keywords: *Stock inventory management, metode waterfall, Firebase realtime database, Unified Modelling Language (UML), blackbox testing, system usability scale (SUS)*

1. PENDAHULUAN

Jenis usaha kuliner saat ini sedang tumbuh sangat pesat, tak terkecuali di kota Malang. Dapat dilihat bahwa, kota Malang merupakan spot kuliner bagi berbagai jenis rumah makan dan kafe. Rumah makan tersebut menyuguhkan berbagai jenis dan rasa yang tersaji dan menawarkan konsep unik mulai dari kawasan perumahan hingga tempat-tempat *hits* dan kekinian di kota Malang.

Salah satu rumah makan yang kini sedang populer dan mulai diminati ialah Nakamse, tempat makan ini mulai terkenal karena mengadaptasi konsep makanan *take away* yang sedang *trend* di Indonesia. Kini Nakamse telah memiliki tiga cabang di kota Malang dan setiap *store* yang ada membutuhkan stok barang dari gudang utama yang mereka miliki. Pasokan barang yang ada di gudang akan didistribusikan setiap hari. Namun dalam melakukan proses pendistribusian barang tersebut terdapat suatu permasalahan yang *real* terjadi, yaitu proses manajemen gudang yang mereka lakukan masih menggunakan cara yang manual karena setiap pengeluaran barang yang dilakukan dari gudang hanya akan ditulis pada sebuah buku sehingga masih sering terjadi duplikasi data dan kesalahan pencatatan barang.

Oleh sebab itu, solusi yang dinilai sangat tepat bagi rumah makan nakamse saat ini adalah sebuah sistem perangkat lunak yang dapat melakukan fungsi monitoring terhadap stok barang dan mengganti proses manajemen gudang yang lama dengan proses manajemen gudang berbasis *mobile apps*. Aplikasi berbasis *mobile* dipilih karena dirasa efektif dan tepat sasaran untuk lingkungan kerja yang ada di Nakamse. Selain itu *platform mobile* dipilih karena sifatnya yang portabel sehingga proses pengawasan terhadap stok barang dan bahan makanan dapat dilakukan dari mana saja selama karyawan membawa dan menggunakan *smartphone*.

Dengan adanya pengelolaan stok barang berbasis aplikasi *mobile* ini maka alur pencatatan dan pengeluaran barang maupun bahan makanan akan *ter-record* dengan baik, sehingga perusahaan dapat melakukan perhitungan modal secara lebih detail dan akurat. Kelebihan lain dari adanya aplikasi *mobile* ini adalah proses pengawasan berkala terhadap stok barang akan semakin baik.

Untuk penyimpanan data, aplikasi ini memanfaatkan teknologi *realtime database* yang dimiliki oleh Firebase. Firebase dipilih karena memiliki beberapa kelebihan yang dirasa cocok dengan kondisi Nakamse saat ini, salah satu kelebihan utama dari Firebase adalah memiliki kemampuan untuk tetap bisa diakses meskipun dalam keadaan *offline* yang tentunya akan sangat bermanfaat bagi karyawan yang ada di lapangan jika sedang terkendala dengan koneksi internet. Fitur lain yang dimiliki oleh Firebase ialah telah terintegrasi sebagai API (*Application Programming Interface*) dengan Android Studio.

2. LANDASAN PUSTAKA

2.1 Nakamse

Nakamse berdiri sejak tahun 2017 silam. Rumah makan ini memiliki konsep yang trendi karena menawarkan sajian makanan dalam bentuk ricebox sehingga mudah untuk dibawa kemana saja dan dimakan kapan saja. Nakamse menawarkan berbagai jenis varian makanan dan minuman yang memiliki tampilan dan packaging serta sungguh menawan di mata. Berbagai pilihan menu yang ada disesuaikan dengan lidah masyarakat. Sehingga, nakamse tidak hanya menawarkan rasa pedas saja namun mulai dari asin hingga gurih juga tersedia. Harga dari makanan ini pun cukup bersahabat di kantong pelajar dan mahasiswa, karena sejak awal target pasar dari tempat makan ini adalah para anak muda yang ada di kota Malang. Kini sejak 3 tahun berdiri, nakamse telah memiliki 3 store di Malang dan 1 store di Surabaya.

2.2 *Mobile Apps*

Aplikasi mobile merupakan aplikasi yang dibuat khusus untuk perangkat bergerak, pada dasarnya aplikasi mobile dibangun menggunakan bahasa pemrograman tertentu seperti Java, Kotlin, dan berbagai bahasa pemrograman lainnya. Dalam setiap perangkat bergerak (mobile device), wajib memiliki software yang berfungsi untuk mendukung penggunaan dari device tersebut. Aplikasi mobile memiliki beberapa perbedaan mendasar jika dibandingkan dengan aplikasi web dan desktop yakni terdapat pada tampilan aplikasi (User Interface) yang sederhana dan mudah untuk digunakan. Perbedaan lain terdapat pada mobilitas yang dimiliki, yakni pengguna dari aplikasi mobile dapat mengakses aplikasi dari mana saja. Aplikasi mobile kini telah memiliki fungsi yang sangat penting dalam kehidupan sehari-hari, karena tanpa disadari sebagian pekerjaan yang dilakukan sangat terbantu dengan adanya aplikasi mobile. Dalam pengembangan aplikasi mobile terdapat beberapa pendekatan yang dapat dilakukan, antara lain: *mobile native*, *mobile web*, *mobile hybrid* (Information, 2015). Aplikasi mobile pada umumnya dibagi menjadi beberapa kategori antara lain: aplikasi komunikasi, pendidikan, berita, permainan dan jejaring sosial. Menurut (B'Far, 2005) aplikasi mobile merupakan suatu system yang dapat digunakan serta melakukan fungsi yang fleksibel karena sifatnya yang mudah untuk dibawa dan dipindahkan.

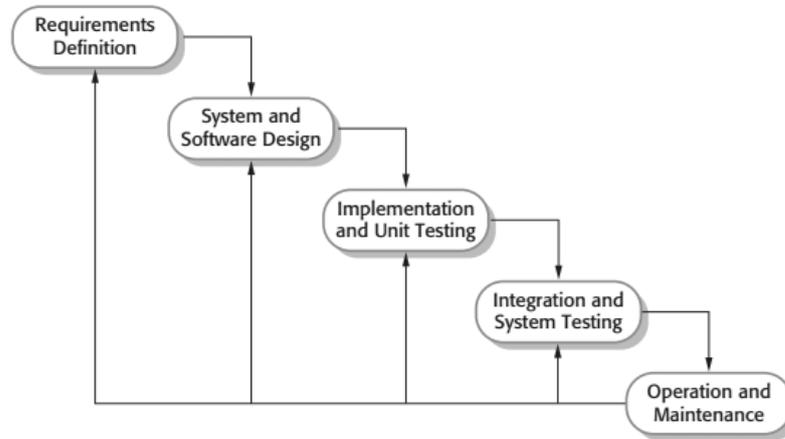
2.3 *Android*

Android merupakan suatu sistem operasi yang terdapat dalam aplikasi mobile. Sistem operasi ini dikembangkan oleh Android, Inc menggunakan teknologi touchscreen yang berbasis Linux (Alliance, 2012). Sistem operasi android pertama kali dirilis pada bulan November 2007 dan terus berkembang hingga saat ini (Elgin, 2005). Sistem operasi ini sangat terkenal dikalangan pengembang aplikasi mobile karena sifatnya yang open source sehingga dapat dimodifikasi dan memiliki pendistribusian yang bebas.

2.4 *System Development Life Cycle (SDLC)*

SDLC atau System Development Life Cycle merupakan serangkaian aktifitas yang dilakukan secara terus menerus dan berulang-ulang dalam pengembangan suatu perangkat lunak. Hal ini dilakukan agar pengembangan perangkat lunak lebih mudah dan terstruktur agar perangkat lunak yang dihasilkan bisa lebih baik. Serangkaian aktivitas pada SDLC meliputi penambahan, mengubah atau mengadopsi sistem yang telah ada dan mencampurkan kedalam sistem yang baru (Sommerville, 2011). Metode SDLC merupakan suatu metode yang sering digunakan dalam pengembangan suatu sistem berbasis komputer. Salah satu pendekatan yang paling sering digunakan dalam pengembangan perangkat lunak adalah metode waterfall. Pada metode waterfall setiap tahap dilakukan secara bertahap berdasarkan urutan pengerjaan yang

telah dirancang. Pada metode waterfall tahap awal yang perlu dilakukan adalah pendefinisian dari sistem yang akan dibangun, kemudian disusul oleh analisis sistem dan desain lalu menuju tahap implementasi dan pengujian, setelah itu aplikasi yang dibangun harus melakukan proses perawatan. Urutan tahapan-tahapan dalam metode waterfall dapat dilihat pada Gambar 1.



Gambar 1. *Waterfall Metode*
Sumber: (Sommerville, 2011)

2.5 Unified Modeling Language (UML)

UML atau *Unified Modeling Language* merupakan permodelan yang memiliki standar khusus yang harus diikuti (Herlawati, 2011). UML dimodelkan dengan cara membuat grafis atau gambar yang bertujuan untuk melakukan pemetaan atau mendeskripsi perancangan dari suatu perangkat lunak. Dalam permodelan UML terdapat standarmodel yang digunakan yakni *usecase diagram*, *activity diagram*, *sequence diagram* dan *class diagram*.

a. *Usecase diagram*

Use case merupakan sebuah cara yang digunakan dalam pengembangan sebuah *software* atau sistem informasi untuk menangkap dan menganalisa kebutuhan fungsional dari sistem yang akan dibangun, *Use case* menjelaskan interaksi yang terjadi antara pengguna dan sistem dan mengilustrasikannya kedalam bentuk diagram.

b. *Activity diagram*

Aktivitas yang terjadi pada sebuah sistem pada dasarnya dapat digambarkan kedalam bentuk diagram yang nantinya diagram tersebut berupa aliran aktivitas dari awal terjadinya suatu proses, kemungkinan-kemungkinan yang dapat terjadi, bahkan berjalannya suatu proses secara parallel pada beberapa kasus pun dapat digambarkan, hingga bagaimana suatu proses tersebut nantinya berakhir.

c. *Sequence diagram*

Sequence diagram menggambarkan bagaimana bagian-bagian yang berbeda pada satu sistem berinteraksi satu sama lain untuk menjalankan suatu fungsi dan urutan interaksi tersebut terjadi ketika *use case* tertentu di eksekusi.

d. *Class diagram*

Class diagram adalah dasar dalam permodelan berorientasi objek. *Class diagram* digunakan untuk menunjukkan objek, atribut, operasi dan hubungan diantara objek-objek tersebut.

2.6 Firebase

Firestore adalah sebuah *platform* yang menyediakan layanan *database* dan backend (BaaS) yang bersifat *realtime*. Firestore memiliki library yang dapat digunakan untuk

pengembangan aplikasi web maupun *mobile*. Firebase telah menyediakan layanan API (*Application Programming Interface*) yang berfungsi untuk mengintegrasikan antara aplikasi dengan *database* atau dengan aplikasi lainnya.

Firestore *Realtime database* merupakan basis data yang digunakan sebagai media penyimpanan data dari suatu aplikasi. Data yang ada pada *database* Firestore, akan disimpan sebagai bit dalam bentuk JSON (*JavaScript Object Notation*) pada cloud dan akan disinkronisasi secara *realtime* ke setiap client yang terhubung. *Database* milik Firestore merupakan *database* yang bersifat non-relational atau NoSQL, dimana *database* ini merupakan jenis *database* yang tidak menggunakan sistem tabel dalam implementasinya. Salah satu kelebihan yang ditawarkan oleh Firestore adalah, *database* ini menyimpan data secara lokal ketika suatu perangkat tidak terhubung dengan akses internet. Kemudian secara otomatis akan melakukan sinkronisasi data setelah perangkat tersebut terhubung kembali dengan internet (Octavianus, 2016).

2.7 Bahasa Pemrograman Java

Java merupakan suatu bahasa pemrograman yang digunakan dalam pengembangan suatu perangkat lunak. Bahasa pemrograman java dikembangkan dengan tujuan untuk memenuhi kebutuhan akan sebuah *platform* yang dapat digunakan untuk membangun sebuah perangkat lunak (*software*). Pada penggunaannya java tersusun atas kelas-kelas yang memiliki method didalamnya, yang berfungsi untuk menjalankan dan mengeksekusi suatu perintah atau mengembalikan informasi setelah melakukan perintah tersebut. Salah satu kelebihan utama yang dimiliki oleh java adalah bahasa pemrograman ini dapat digunakan dalam berbagai *platform* seperti aplikasi *mobile*, desktop dan web. Menurut (Indrajani, 2007) bahasa pemrograman java memiliki ciri antara lain: berbasis Objek (OOP), *Multithreading*, *Garbage Collector Support*, *Statically Type* dan *Multiplatform*.

2.8 Web Service

Web service merupakan suatu metode yang digunakan untuk bertukar data tanpa harus memperhatikan jenis *database*, bahasa pemrograman dan *platform* yang digunakan. *Web service* pada dasarnya dibangun agar sebuah aplikasi dapat saling bekerja sama. Beberapa kelebihan yang dimiliki *Web service* yakni: *Web service* dapat berjalan dalam lintas *platform* dan menjembatani antara *database* dengan aplikasi. Sehingga dengan adanya *Web service* pertukaran data dan penggunaan kembali komponen aplikasi akan lebih mudah (Lucky, 2008). Salah satu *Web service* yang kini banyak digunakan adalah *Representational State Transfer (REST)*.

2.9 JavaScript Object Notation (JSON)

JavaScript Object Notation merupakan sebuah format penulisan yang mudah dibuat dan dibaca oleh manusia. JSON dibuat dengan tujuan agar pengiriman dan pengambilan data dapat dilakukan dengan mudah tanpa harus bergantung pada bahasa pemrograman yang digunakan. Struktur data yang dihasilkan oleh JSON akan berupa JSON tree. Menurut (Indrajani, 2007) bahasa pemrograman java memiliki ciri antara lain: berbasis Objek (OOP), *Multithreading*, *Garbage Collector Support*, *Statically Type* dan *Multiplatform*.

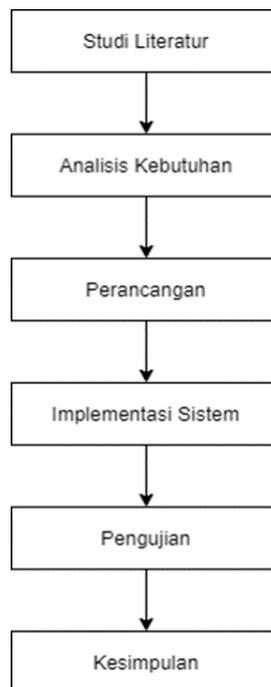
2.10 Pengujian

Pengujian perangkat lunak merupakan langkah yang harus dilakukan saat telah melalui tahap implementasi, hal ini diperlukan untuk mengetahui kesalahan atau permasalahan yang terjadi pada perangkat lunak yang dibangun. Tahap pengujian dilakukan untuk menguji kesesuaian antara rancangan kebutuhan dengan hasil implementasi dan output dari perangkat

lunak tersebut. Pada proses pengujian, terdapat beberapa bagian yakni *software verification* dan *software validation* (Sommerville, 2011). Pengujian berguna untuk mengetahui kualitas dari suatu perangkat lunak dari hasil respons yang diberikan oleh penggunaannya, oleh sebab itu pengujian perangkat lunak memiliki banyak metode yang dapat digunakan salah satunya *blackbox testing* dan *usability testing*.

3. METODOLOGI

Dalam penelitian ini akan menggunakan metode SDLC dengan pendekatan *waterfall*. Metode ini dipilih karena dianggap relevan dengan urutan pengembangan aplikasi yang akan dibuat. Tahapan metodologi meliputi studi literatur, perancangan sistem, analisis kebutuhan, desain sistem, implementasi, pengujian serta kesimpulan.



Gambar 2. Metodologi

Studi literatur yang digunakan untuk pengumpulan data dan informasi didapat melalui media cetak (buku) maupun media elektronik (*e-book, paper, jurnal*).

Analisis kebutuhan dilakukan dengan cara observasi, wawancara dan studi literature. Kebutuhan yang telah ditentukan dan disepakati akan digambarkan kedalam bentuk *usecase diagram, usecase scenario dan activity diagram*.

Dalam perancangan, peneliti akan menentukan lingkup sistem sesuai dengan kebutuhan yang telah ditentukan sebelumnya. Setelah itu, memulai perancangan dengan membuat diagram, membuat permodelan, perancangan interface dan menganalisa kebutuhan yang akan digunakan untuk membuat aplikasi pengelolaan stok barang.

Pada tahap implementasi, semua perancangan aplikasi maupun interface diimplementasikan dalam bentuk kode program dengan menggunakan *platform android* pada *software* Android Studio yang menggunakan bahasa pemrograman Java. Untuk pengolahan *database* dan API sendiri akan dibangun dengan memanfaatkan teknologi Firebase.

Setelah implementasi telah dilakukan, selanjutnya aplikasi akan diuji dan dianalisis kembali apakah sesuai dengan kebutuhan pengguna, Pengujian sendiri akan menggunakan metode *black box testing* dan *usability testing*.

Penutup berisi bagian kesimpulan dan saran, yang mana pada bagian kesimpulan akan menjelaskan kesimpulan hasil akhir yang didapatkan melalui penelitian yang dilakukan dan saran sebagai pengembangan dari kekurangan penelitian saat ini untuk dilanjutkan pada penelitian selanjutnya.

4. ANALISA KEBUTUHAN DAN PERANCANGAN

Pada sub bab ini akan membahas mengenai beberapa tahapan yang akan dilakukan meliputi: gambaran umum aplikasi, identifikasi aktor, kebutuhan fungsional dan kebutuhan non fungsional.

4.1 Identifikasi Aktor

Dalam pengembangan aplikasi pengelolaan stok barang ini terdapat 2 jenis aktor, yang perannya dijelaskan pada Tabel 1. Penamaan Aktor.

Tabel 1. Penamaan Aktor

Nama Aktor	Deskripsi
Admin Gudang	Aktor yang akan melakukan pendataan terhadap stok barang dan membuat pelaporan dari jumlah barang yang digunakan.
Karyawan	Aktor yang melakukan request untuk pengambilan barang dari gudang.

4.2 Analisis Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan utama dari sistem yang menggambarkan layanan dari perangkat lunak yang dibangun. Pada aplikasi pengelolaan stok barang ini terdapat beberapa kebutuhan yang akan dijelaskan lebih rinci pada Tabel 2.

Tabel 2. Kebutuhan Fungsional

Nama Usecase	Kebutuhan Fungsional
<i>Login</i>	Sistem mampu menampilkan halaman <i>login</i> dan melakukan akses ke halaman utama.
<i>Check stock</i>	Sistem mampu menampilkan jumlah stok barang yang tersedia di dalam gudang.
<i>Add stock</i>	Sistem mampu menjalankan fungsi untuk menambahkan data barang kedalam <i>database</i> gudang, yang kemudian data tersebut dapat dilihat oleh karyawan sebagai <i>stock</i> barang.
<i>Edit Stock</i>	Sistem mampu menjalankan fungsi perubahan jumlah <i>stock</i> barang yang ada pada <i>database</i> gudang.
<i>Record Pengeluaran Barang</i>	Sistem mampu melakukan pendataan dari setiap proses transaksi barang yang terjadi di gudang baik berupa data masuk dan data keluar.
<i>Request Barang</i>	Sistem mampu melakukan proses <i>request</i> barang dan menyimpan data hasil request kedalam fungsi tracking.
<i>Tracking</i>	Sistem mampu melakukan proses tracking untuk mengupdate setiap progres dari barang yang telah di request.
<i>Logout</i>	Sistem mampu mengeksekusi function signout dan mengirim <i>user</i> kembali ke halaman <i>login</i> .

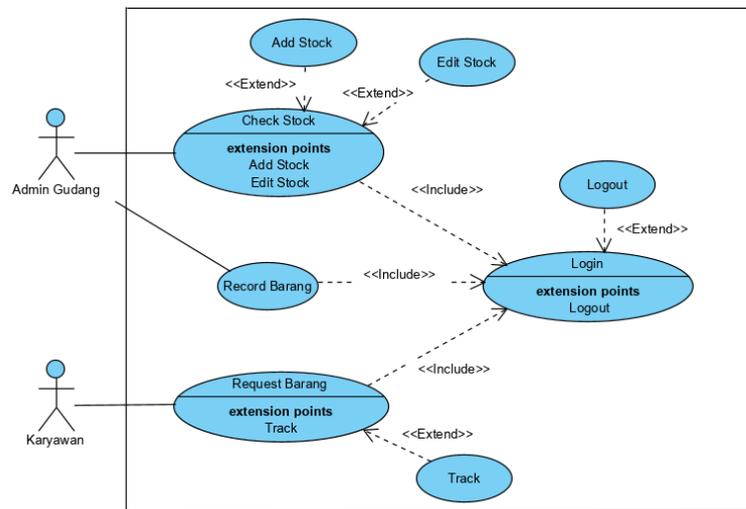
Kebutuhan non fungsional merupakan kebutuhan tambahan yang dibutuhkan dalam performa aplikasi ataupun tahan daya sistem dalam suatu kondisi. Kebutuhan pada aplikasi ini akan dijelaskan pada Tabel 3. Kebutuhan Non Fungsional.

Tabel 3. Kebutuhan Non Fungsional

Nama Kebutuhan	Deskripsi
Usability	Karyawan dan admin gudang nakamse mampu untuk menjalankan, mengingat dan mempelajari setiap langkah-langkah dalam pengoperasikan Aplikasi pengelolaan stok barang Berbasis <i>Mobile</i> .

4.3 Diagram Usecase

Dalam *use case* tersebut terdapat kebutuhan fungsional yang akan dijalankan oleh aktor. *Use case diagram* pada penelitian ini akan ditampilkan pada Gambar 3.



Gambar 3. Usecase diagram

4.4 Activity diagram

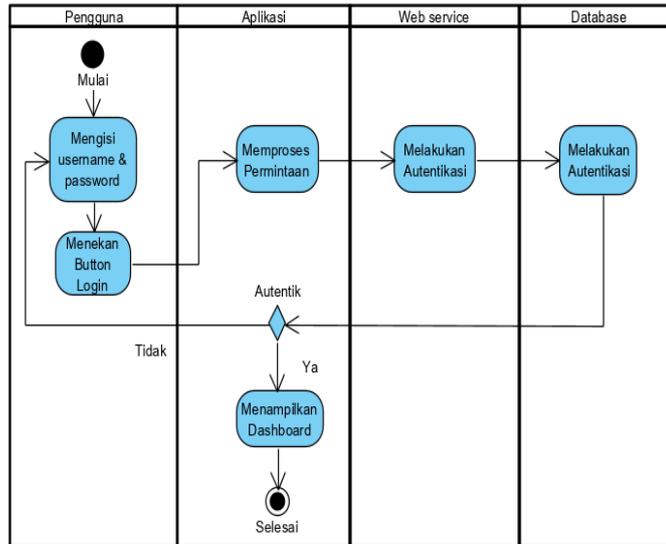
Pada tahap ini akan membahas mengenai aliran kerja yang digambarkan dalam bentuk *activity diagram*. Pada *activity diagram* ini akan menjelaskan secara lebih detail mengenai tahapan-tahapan aktivitas yang ada didalam sistem.

Pada Gambar 4. menjelaskan mengenai alur *login* yang ada pada sistem. Alur dimulai dari pengguna menginputkan data *username* dan *password* yang nantinya akan digunakan sebagai bahan autentikasi.

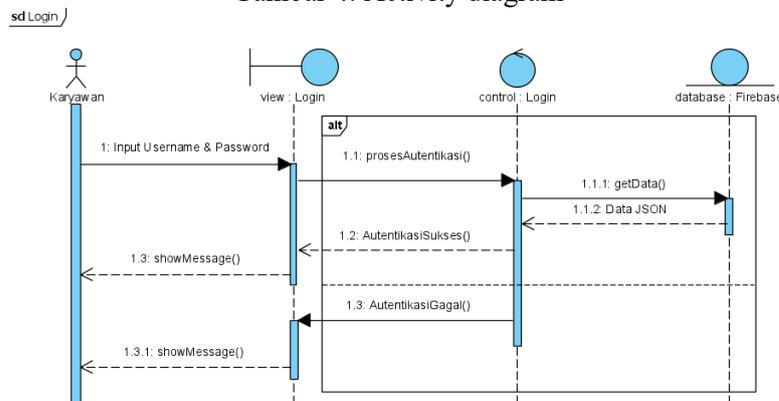
4.5 Sequence diagram

Permodelan *sequence diagram* ini menggambarkan urutan eksekusi yang terjadi sesuai dengan urutan waktu.

Gambar 5. merupakan permodelan *sequence diagram* dari fitur *login*. Proses *login* dimulai saat pengguna masuk ke halaman *login* dan menginputkan *username* dan *password* yang telah didaftarkan sebelumnya. Kemudian sistem akan melakukan autentikasi() terhadap *user* tersebut.



Gambar 4. Activity diagram



Gambar 5. Sequence diagram

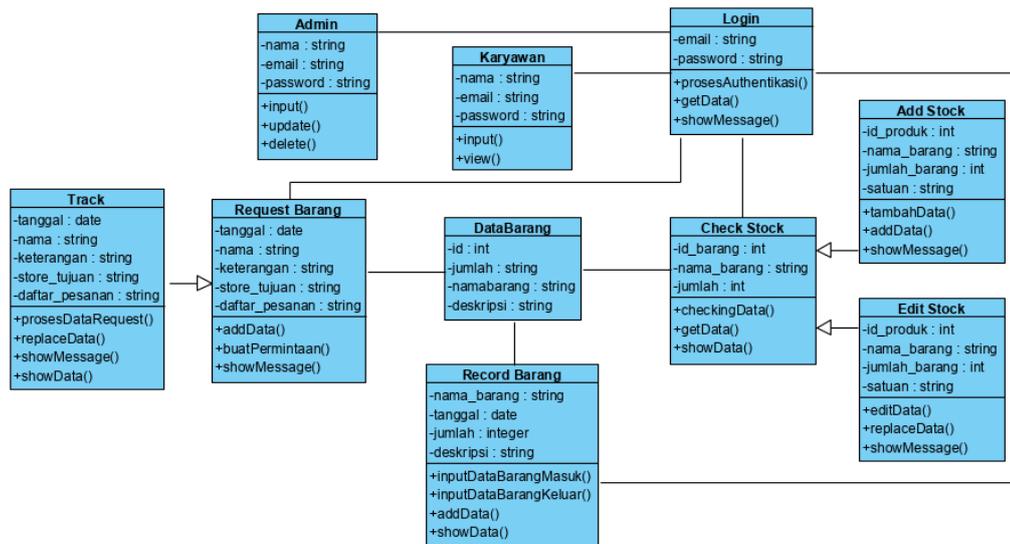
4.6 Class diagram

Pada Gambar 6. terdapat beberapa kelas yang menjadi objek utama dalam pengembangan aplikasi ini. Setiap kelas memiliki fungsi masing-masing, kelas Karyawan berisikan atribut yang mendefinisikan aktor yang berinteraksi dengan sistem. Kemudian kelas Login yang berisi *username* dan *password* dan beberapa *method* untuk menjalankan proses tersebut. Lalu diikuti oleh beberapa kelas utama lainnya yang berisikan *method* dan atribut untuk menjalankan setiap fungsinya. Setiap kelas yang berkaitan satu dengan yang lainnya memiliki relasi asosiasi sedangkan kelas yang menjadi turunan dari kelas lainnya memiliki relasi generalisasi.

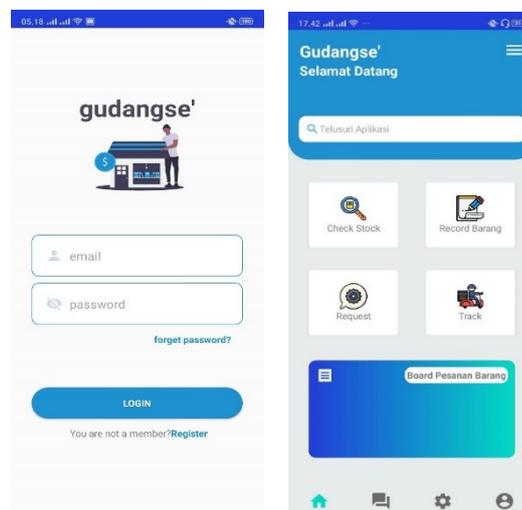
5. IMPLEMENTASI

Pada tahap pengembangan aplikasi pengelolaan stok barang ini, peneliti menggunakan Firebase sebagai tempat penyimpanan basis data. Firebase memiliki fitur bernama *realtime database* yang memungkinkan seluruh data dapat diakses secara *realtime*.

Untuk dapat menghubungkan antara Firebase *realtime database* dengan aplikasi *mobile* yang dimiliki oleh pengguna maka dibutuhkan sebuah API (*Application Programming Interface*) sebagai layanan untuk mengirimkan dan mengambil data atau lebih sederhananya dapat dikatakan sebagai penghubung antar kedua *platform* tersebut. Firebase sendiri telah menyediakan API berupa REST sebagai endpointnya, yang kemudian akan disinkronkan secara *realtime* pada setiap client yang terhubung. Data yang disimpan oleh Firebase akan berupa *Json tree*. Berikut pada Gambar 7. merupakan hasil dari implementasi yang telah dilakukan.



Gambar 6. Class diagram



Gambar 7. Implementasi Halaman Login dan Dashboard

6. PENGUJIAN

Pada subbab ini merupakan analisis dari pengujian fungsional dan non fungsional yang telah dilakukan untuk ditarik kesimpulannya. Berikut merupakan kesimpulan dari setiap proses pengujian.

1. Pada tahap pengujian fungsional menggunakan metode *blackbox* mendapatkan hasil yang valid pada setiap fungsionalnya. Sehingga dapat disimpulkan bahwa aplikasi pengelolaan stok barang Nakamse telah sesuai dengan kebutuhan yang ada.
2. Pada pengujian *usability* didapatkan hasil bahwa aplikasi masuk dalam kategori *acceptable* dengan grade C dan rata-rata skor SUS adalah 78,5 (*good*).

7. PENUTUP

Setelah dilakukan penelitian dan implementasi terhadap aplikasi pengelolaan stok barang berbasis *mobile* pada rumah makan nakamse malang maka dapat ditarik kesimpulan sebagai berikut:

1. Hasil analisa kebutuhan dari proses pengembangan aplikasi pengelolaan stok barang berbasis *mobile* terdiri atas pendefinisian kebutuhan fungsional dan kebutuhan non

fungsional. Setiap kebutuhan fungsional tersebut akan dimodelkan kedalam use case diagram, kemudian spesifikasi dari setiap kebutuhan fungsional tersebut akan dijelaskan lebih detail dalam use case scenario. Proses analisa kebutuhan kemudian dilanjutkan ke tahap permodelan activity diagram yang mengacu pada usecase diagram yang telah dimodelkan sebelumnya.

2. Hasil pada perancangan dan implementasi dari pengembangan aplikasi pengelolaan stok barang menghasilkan rancangan dan implementasi, sebagai berikut:
 - a. Proses perancangan aplikasi pengelolaan stok barang diawali dengan perancangan arsitektur sistem dari aplikasi yang dibangun, permodelan *sequence diagram* dan *class diagram* yang berguna untuk memodelkan objek. Kemudian dilanjutkan dengan rancangan struktur data yang akan digunakan didalam Firebase.
 - b. Aplikasi pengelolaan stok barang berbasis *mobile* diimplementasikan menggunakan metode *waterfall* yang menghasilkan sebuah aplikasi berdasarkan kebutuhan dan kemudahan dalam penggunaan aplikasi tersebut. Proses implementasi dilakukan dengan membuat kode program pada *platform* Android Studio dengan memanfaatkan teknologi *Firestore Realtime database* sebagai tempat penyimpanan data.
3. Berdasarkan hasil pengujian yang telah dilakukan menggunakan metode *blackbox testing* dan usability testing maka didapatkan hasil sebagai berikut:
 - a. Pada tahap pengujian fungsional menggunakan metode *blackbox* mendapatkan hasil yang valid pada setiap fungsionalnya.
 - b. Pada pengujian *usability* didapatkan hasil bahwa aplikasi masuk dalam kategori *acceptable* dengan grade C dan rata-rata skor SUS adalah 78,5 (*good*).

DAFTAR PUSTAKA

- Alliance, O. H., 2012. Android Overview Open Handset Alliance. [Online] Available at: http://www.openhandsetalliance.com/android_overview.html
- B'Far, R., 2005. *Mobile Computing Principle*. p.3.
- Elgin, B., 2005. Google Buys Android for Its *Mobile Arsenal* Bloomberg Businessweek Bloomberg. [Online] Available at: <http://www.webcitation.org/5wk7sIvVb>
- Herlawati, W. P. P. d., 2011. *Menggunakan UML*. Bandung: Informatika.
- H, S., 2007. *Java™ The Complete Reference*. United State of America: The McGraw-Hill Companies.
- Indrajani, M., 2007. *Pemrograman Berbasis Objek dengan Bahasa Java*. In: Jakarta: Elex Media.
- Information, O., 2015. *Native, Hybrid, or Mobile Web Application Development*.
- Lucky, 2008. *XML Web services: Aplikasi Desktop, Internet & Handphone*. In: Jakarta: Jasakom.
- Octavianus, B., 2016. *Apa Itu Firebase dan Apa Saja Fiturnya*. [Online] [Accessed 11 Mey 2020].
- Sommerville, 2011. *Software Engineering*. New York: Pearson Education, Inc.