

# **APLIKASI MUDAH ALIH SISTEM PENGURUSAN MAKANAN PELAJAR SEKOLAH (KANTEEN!)**

Anis Shamimi Loo

Hasimi bin Sallehuddin

*Fakulti Teknologi dan Sains Maklumat, Universiti Kebangsaan Malaysia*

## **ABSTRAK**

Aplikasi Mudah Alih Sistem Pengurusan Makanan Pelajar Sekolah (KANTEEN!) ini adalah sebuah platform yang akan memudahkan pelajar memenuhi selera makanan mereka dengan nutrisi yang cukup. Pelajar perlu memasukkan kriteria tertentu dan membuat pesanan. Pesanan pelajar akan diproses akan dipaparkan kepada pembekal. Pelajar juga dapat memilih menu makanan yang mereka inginkan mengikut penyediaan yang menu makanan yang di senaraikan oleh pembekal sekolah masing-masing. Aplikasi ini akan dibangunkan dengan menggunakan Android Studio dan memerlukan database untuk menyimpan data pengguna. Android Studio adalah salah satu IDE atau peralatan yang diberi keutamaan oleh Google selain IDE lama iaitu Eclipse. Database seperti Google Firebase memainkan peranan yang sangat penting untuk sebuah sistem yang berjaya supaya data pelanggan dapat disimpan dan diproses kemudian untuk mengetahui trend penggunaan aplikasi para pengguna. Kesimpulannya, aplikasi mudah alih ini bertujuan untuk memudahkan pelajar mengatasi masalah menu pemakanan mereka.

## **1. PENGENALAN**

Aplikasi mudah alih merupakan sebuah perisian aplikasi yang direka untuk digunakan di peranti mudah alih seperti telefon pintar dan komputer tablet. Aplikasi mudah alih berbeza dengan aplikasi desktop yang digunakan di komputer desktop. Aplikasi mudah alih biasanya boleh didapati di platform seperti Apple App Store, Google Play, Windows Telefon Store dan Blackberry App World yang mula muncul pada tahun 2008. Sesetengah aplikasi adalah percuma manakala yang lain perlu dibeli.

Aplikasi mudah alih KANTEEN! merupakan aplikasi yang menyediakan perkhidmatan kepada ibubapa dan penjaga pelajar untuk memilih menu makanan yang bersesuaian kepada anak-anak mereka di sekolah.. . Pelajar juga boleh membuat pilihan menu yang mereka inginkan untuk

dijadikan santapan pada waktu rehat mereka setiap hari. Peranan aplikasi mudah alih ini adalah untuk memenuhi segala keperluan makanan yang sihat dan bersesuaian dengan pelajar ketika mereka di sekolah.

## **2. PENYATAAN MASALAH**

Kebanyakan pelajar sering mengadu bahawa mereka sering kelaparan apabila pulang ke rumah dengan alasan makanan di sekolah mereka sudah habis ataupun tidak mengikut keperluan selera mereka. . Kebanyakan kantin atau kafeteria sekolah menyediakan dan menjual makanan yang tidak sihat kepada pelajar sekolah. Antaranya ialah makanan segera seperti sosej, nugget, makanan ringan, dan makanan-makanan tidak berkhasiat yang lain. Makanan tersebut tidak sesuai dengan keperluan nutrisi pelajar. Manakala ibubapa atau penjaga pelajar pula tidak dapat mengawal makanan yang dibeli dan dimakan oleh anak-anak mereka disekolah selain daripada membekalkan sendiri makanan dari rumah. Bagaimanapun, tidak semua ibubapa atau penjaga mempunyai masa dan peluang untuk menyediakan bekalan makanan kepada anak-anak mereka ke sekolah. Kekurangan nutrisi makanan kepada pelajar akan menyebabkan pelajar tidak bermaya semasa pembelajaran dan juga terdedah kepada pelbagai penyakit seperti diabetes walaupun masih pelajar sekolah. Selain itu, pengusaha kantin atau kafeteria sekolah juga tiada maklumat bersepadu tentang keperluan makanan yang digemari dan bersesuaian oleh pelajar. Ini menyebabkan sesetengah makanan yang disediakan tiada sambutan dan tidak dibeli oleh pelajar dan seterusnya berlaku pembaziran makanan.

## **3. OBJEKTIF KAJIAN**

Objektif projek ini adalah seperti berikut :-

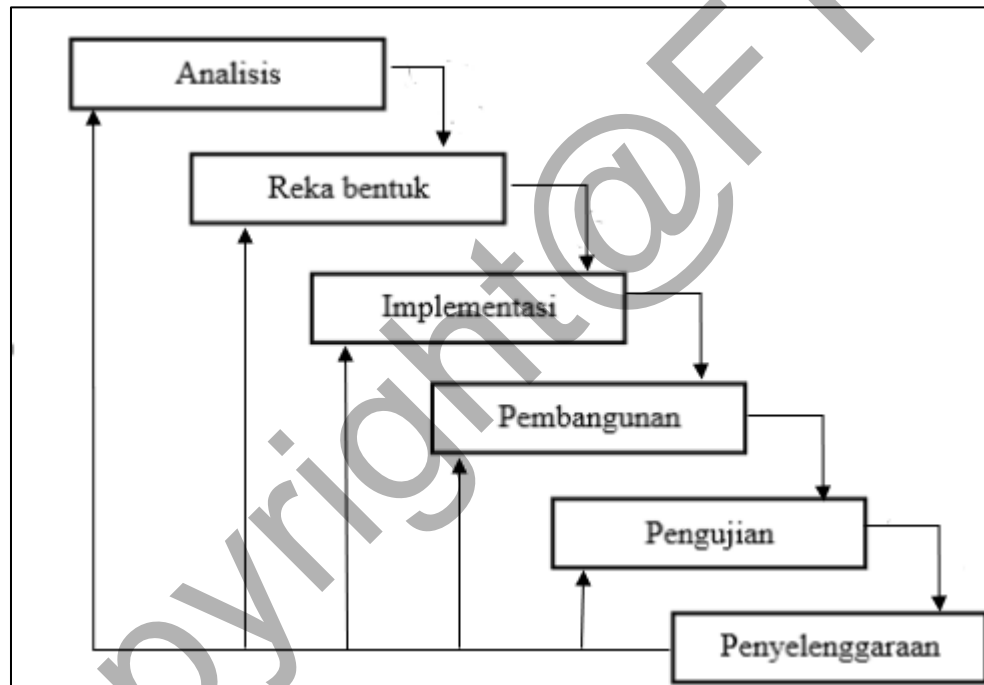
- i. Menyediakan satu platform untuk ibubapa, penjaga dan pelajar sekolah memesan makanan mengikut kesesuaian dan memenuhi nutrisi pelajar.
- ii. Menghasilkan satu aplikasi untuk memenuhi keperluan ibubapa, penjaga dan pelajar sekolah berdasarkan pemakanan mereka.
- iii. Menyediakan satu platform kepada ibubapa, penjaga atau pelajar sekolah kebolehan memilih pelbagai variasi makanan yang bersesuaian.

- iv. Menyediakan satu platform bagi memudahkan ibu bapa atau penjaga pelajar sekolah untuk memantau pemakanan anak mereka.

Copyright@FTSM

#### 4. METOD KAJIAN

Kaedah air terjun atau yang lebih dikenali sebagai kaedah "waterfall" sering digunakan untuk menerangkan perkembangan peringkat aktiviti di dalam pembangunan perisian. Kaedah ini membolehkan pasukan memecahkan projek ke fasa yang dapat difahami dan dijelaskan dengan hasil yang jelas. Kaedah ini juga hanya akan ke fasa seterusnya apabila fasa sebelumnya selesai dan semuanya disiapkan. Aplikasi KANTEEN! akan menggunakan kaedah ini untuk menyiapkan projek ini. Kaedah ini mengadungi 6 fasa iaitu Fasa Analisis, Fasa Rekabentuk, Fasa Implementasi, Fasa Pembangunan, Fasa Pengujian dan Fasa Penyelenggaraan



Rajah 1.1 Kaedah Waterfall

##### 4.1 Fasa Analisis Keperluan (*Requirement specification*)

Fasa ini adalah untuk mengumpul maklumat berkaitan projek ini untuk membangunkan aplikasi Kanteen! Fasa ini juga akan membincangkan kajian yang akan dilakukan terhadap pengguna untuk membangunkan perisian yang mengikut kehendak pengguna.

##### 4.2 Fasa Reka Bentuk (*Design*)

Dalam fasa ini, semua kajian yang dibuat akan diterjemahkan menjadi reka bentuk secara bertulis. Contohnya seperti menyenaraikan objektif projek, perisian pembangunan aplikasi dan sebagainya.

#### 4.3 Fasa Implementasi (*Implementation*)

Dalam fasa ini, pembangunan aplikasi Kanteen! akan dibina mengikut reka bentuk yang telah dibincangkan dari fasa yang terdahulu.

#### 4.4 Fasa Pembangunan (*Integration*)

Fasa ini merupakan pengaturcaraan aplikasi Kanteen! dibangunkan mengikut spesifikasi yang dibincangkan.

#### 4.5 Fasa Pengujian (*Testing and debugging*)

Aplikasi Kanteen! akan diuji dan dinilai untuk memastikan ia berfungsi dengan baik, memenuhi reka bentuk yang telah dibuat, mencapai objektif yang telah ditetapkan dan sebagainya.

#### 4.6 Fasa Penyelenggaraan (*Maintenance*)

Fasa terakhir dalam model ini merupakan peringkat di mana aplikasi ini akan diselenggarakan sebelum digunakan oleh pengguna secara lebih berkesan.

## 5. HASIL KAJIAN

### 5.1 PEMBANGUNAN ANTARAMUKA APLIKASI

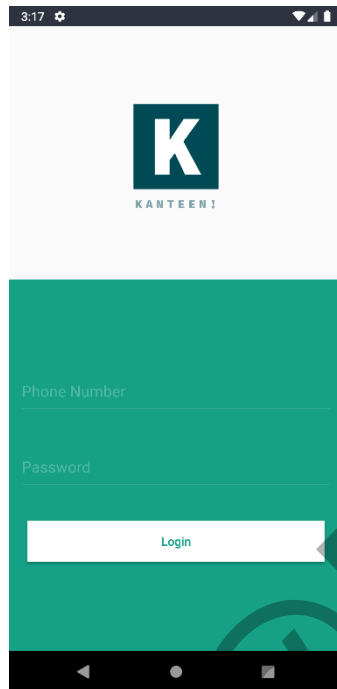
Pembangunan aplikasi KANTEEN! Mementingkan rekabentuk antara muka. Hal ini demikian kerana rekabentuk antara muka memainkan peranan yang penting untuk memastikan kemudahan untuk pengguna mengakses aplikasi KANTEEN!. Bagi membangunkan rekabentuk antara muka ini, perisian yang digunakan ialah Android Studio.

#### 5.1.1 Halaman Permulaan

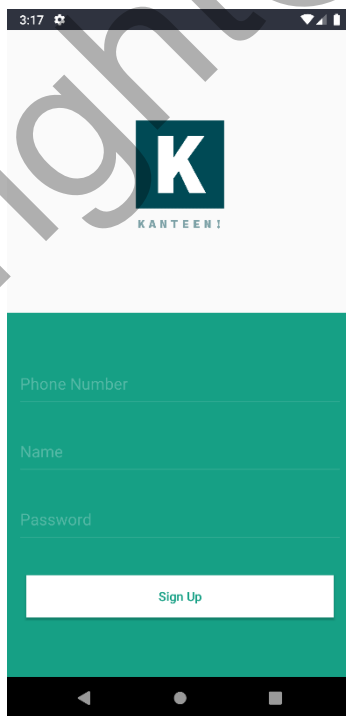
Rajah 5.1 merupakan halaman pertama yang dipaparkan kepada pengguna setelah mengakses aplikasi ini. Pada halaman ini, terdapat dua butang iaitu Sign Up dan Sign In. Bagi pengguna yang sudah mempunyai akaun, mereka boleh terus menekan butang Sign In untuk ke Rajah 5.2 iaitu antara muka log masuk aplikasi. Pengguna yang belum mempunyai akaun perlu mendaftar dengan menekan butang Sign Up terlebih dahulu dan akan dibawa kepada Rajah 5.3 iaitu antara muka Sign Up untuk pengguna mendaftarkan akaun yang baru.



(Rajah 5.1 Splash Screen)



(Rajah 5.2 Sign In)



(Rajah 5.3 Sign Up)

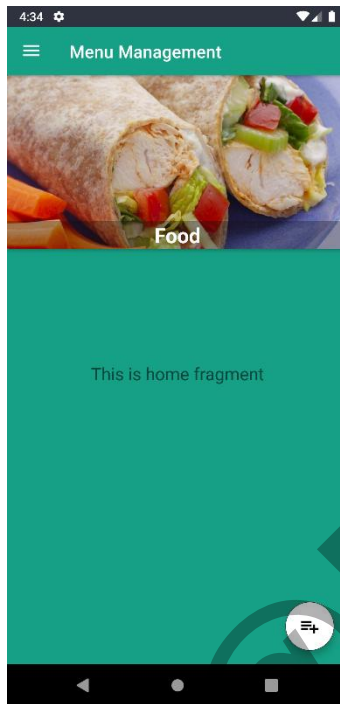
### 5.1.2 Halaman Utama

Rajah 5.4 menunjukkan halaman utama aplikasi KANTEEN! setelah pengguna berjaya mengelog masuk. Dalam antara muka halaman utama ini, terdapat butang navigasi dan dua butang yang mempunyai fungsinya tersendiri. Dua butang tersebut ialah List of Category dan Cart yang akan membawa pengguna kepada fungsi masing-masing. Namun, fungsi antara muka tersebut hanyalah untuk pengguna yang mendaftar diri mereka sebagai pelanggan. Sekiranya, pengguna tersebut ialah pekerja kantin, dua fungsi butang yang terdapat di halaman utama ialah List of Category dan Add New Category seperti dalam Rajah 5.5. Rajah 5.6 menunjukkan kandungan dalam butang navigasi untuk pelanggan dan Rajah 5.7 kandungan dalam butang navigasi untuk pekerja.

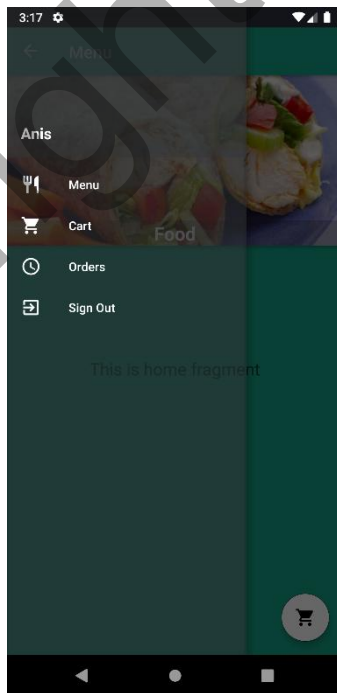


(Rajah 5.4 Halaman Utama untuk Pelanggan)

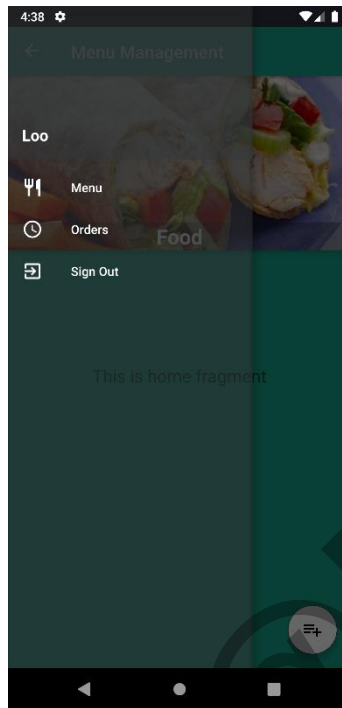




(Rajah 5.5 Halaman Utama untuk Pekerja Kantin)



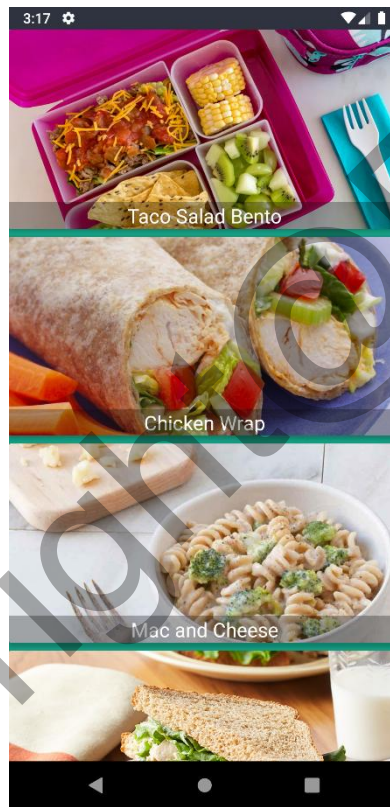
(Rajah 5.6 Navigation Drawer untuk Pelanggan)



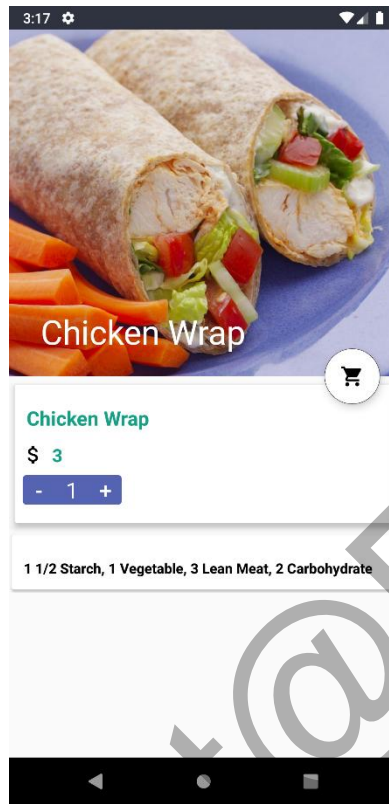
(Rajah 5.7 Navigation Drawer untuk Pekerja Kantin)

### 5.1.3 List of Menu

Rajah 5.8 menunjukkan senarai menu yang telah diupload oleh pekerja ke dalam sistem. Kepada pelanggan mereka boleh menekan menu mana yang mereka inginkan dan ianya akan membawa kepada fungsi seterusnya iaitu description of menu seperti Rajah 5.9. Di antara muka tersebut juga, pelanggan boleh memilih kuantiti menu yang mereka ingin memesan kepada anak mereka dan terus add to cart.



(Rajah 5.8 List of Menu)



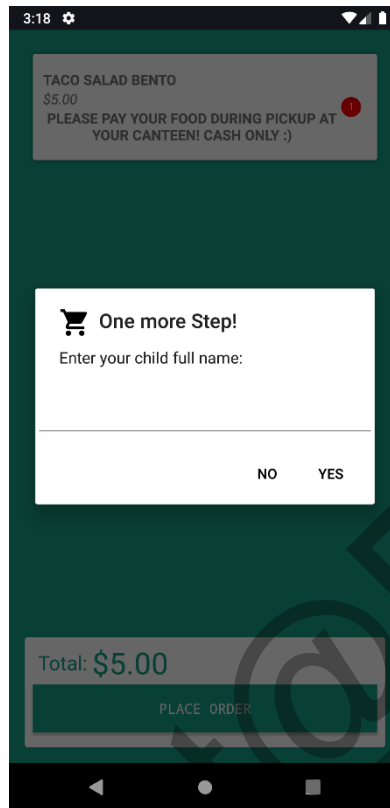
(Rajah 5.9 Description of food)

### 5.1.4 Add to Cart

Rajah 5.10 adalah antara muka bagi fungsi Add to Cart. Selepas pelanggan menekan butang add to cart selepas memilih makanan bagi anak mereka, menu tersebut akan muncul di add to cart berserta deskripsi menu yang dipilih. Pembayaran hanya boleh dilakukan melalui tunai, dimana ianya perlu dibayar ketika makanan diambil. Setelah pelanggan sudah selesai memilih menu, pelanggan perlu menekan butang 'PLACE ORDER' untuk mengesahkan pesanan mereka dan mereka perlu memasukkan nama penuh anak mereka selepas menekan butang tersebut seperti Rajah 5.11.



(Rajah 5.10 Add to Cart)



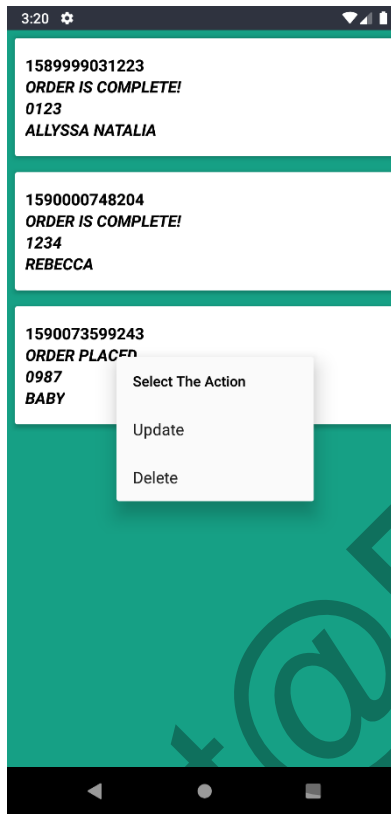
(Rajah 5.11 Enter your child full name)

### 5.1.5 Order Status

Rajah 5.12 menunjukkan Order Status setelah pelanggan membuat pengesahan terhadap pesanan mereka. Pelanggan boleh membatalkan pesanan mereka hanya ketika status order tersebut ialah “Order placed”, sekiranya status itu sudah bertukar, pelanggan tidak dapat membatalkan pesanan tersebut lagi. Rajah 5.13 juga menunjukkan antara muka yang akan disampaikan kepada pekerja kantin dan boleh dikemaskini oleh mereka dengan menukarkan status pesanan bersesuaian dengan situasi pesanan tersebut.



(Rajah 5.12 Order Status kepada Pelanggan)

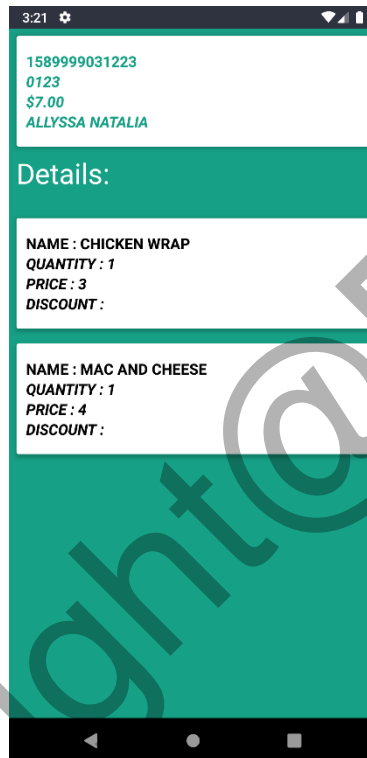


(Rajah 5.13 Order Status kepada Pekerja Kantin)



### 5.1.6 Order Details

Rajah 5.14 adalah antara muka bagi fungsi Order Details. Antara muka ini akan memaparkan deskripsi pesanan yang telah dibuat oleh pelanggan. Antara muka ini juga akan memaparkan antara muka yang serupa kepada pekerja kantin.

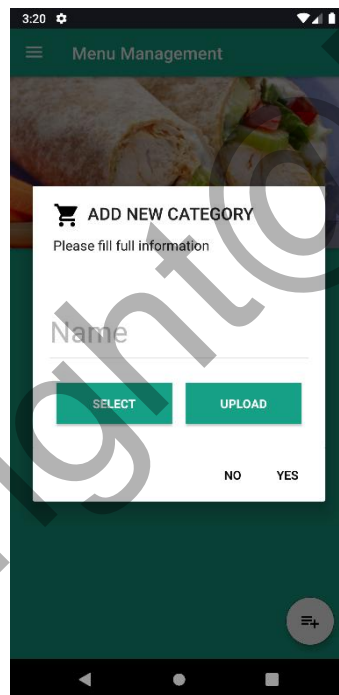


(Rajah 5.14 Order Details)

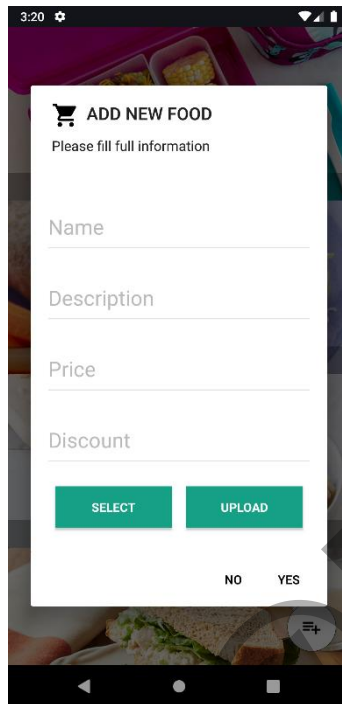
## 5.2 PEMBANGUNAN SISTEM BELAKANG

Sistem belakang ialah sistem yang penting dalam aplikasi KANTEEN!. Sistem belakang membolehkan pentadbir menambah, mengemaskini dan membuang data dan informasi yang disimpan dalam pangkalan data Firebase untuk kegunaan setiap fungsi. Pentadbir bagi sistem belakang aplikasi KANTEEN! ialah pemilik kantin. Fungsi-fungsi yang memerlukan sistem belakang ialah daftar akaun, log masuk, kemaskini kategori menu dan kemaskini menu.

Rajah 5.2.1 dan Rajah 5.2.2 menunjukkan sistem belakang bagi fungsi kemaskini kategori menu untuk menambah dan mengemaskini rekod data.



(Rajah 5.2.1 Add New Category)



3:20

**ADD NEW FOOD**

Please fill full information

Name

Description

Price

Discount

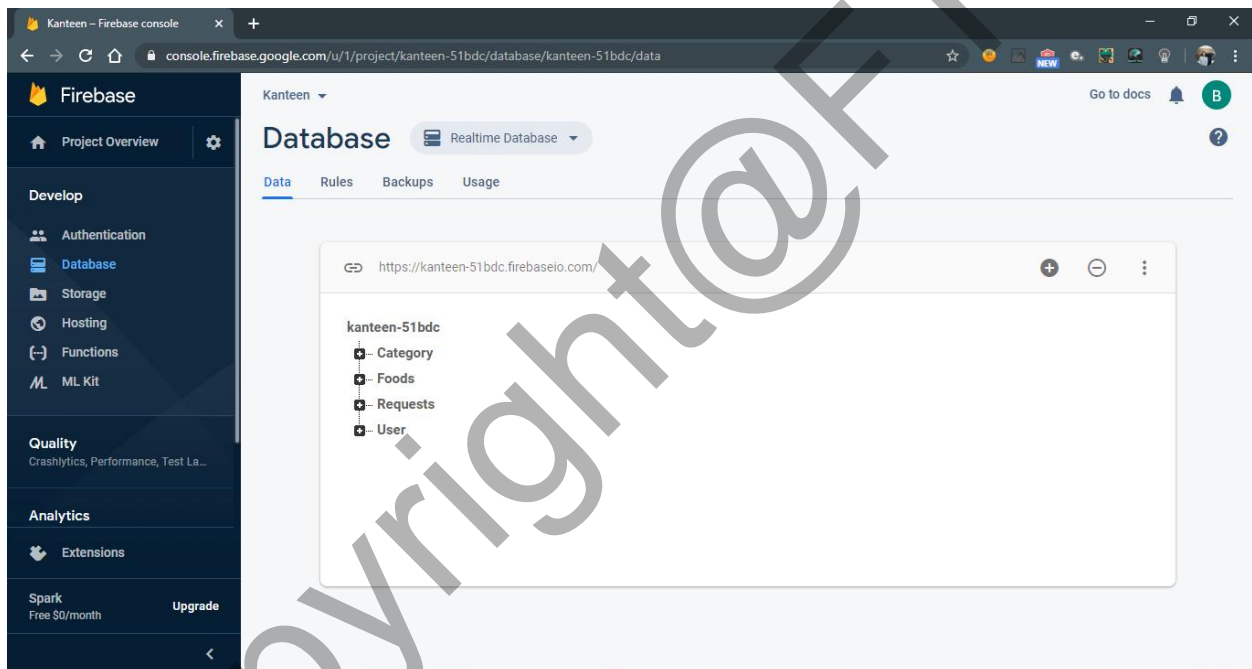
**SELECT** **UPLOAD**

NO YES

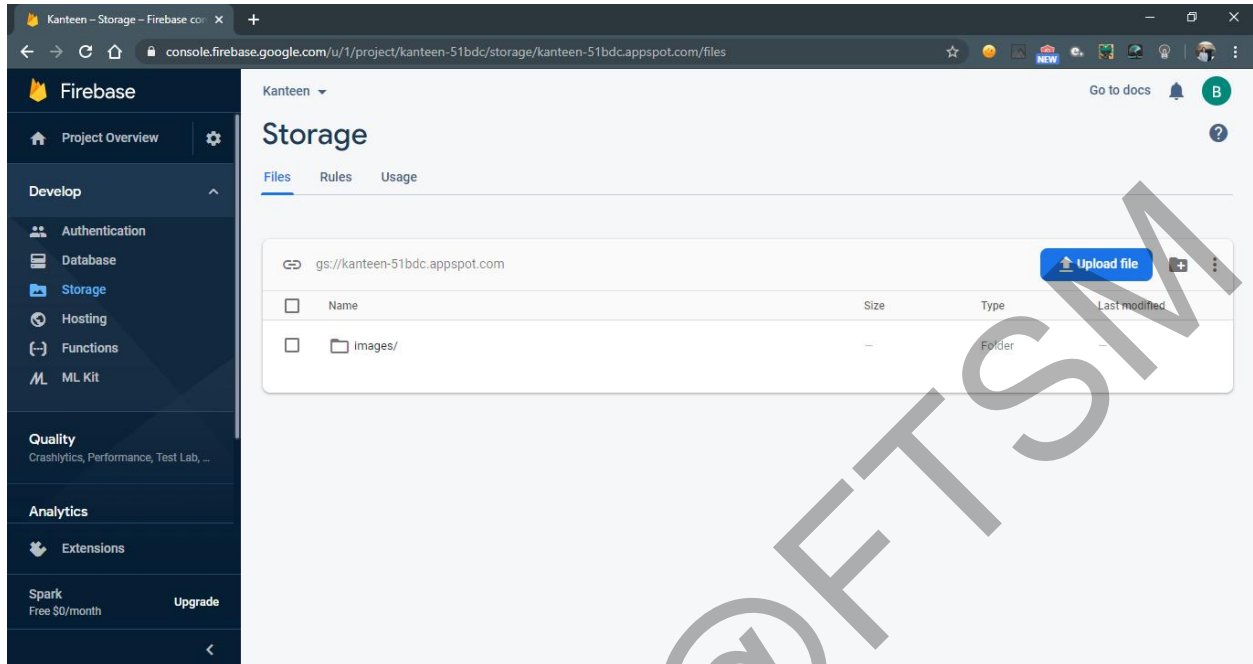
(Rajah 5.2.2 Add New Food)

### 5.3 PANGKALAN DATA

Pangkalan data yang digunakan dalam aplikasi ini ialah Firebase. Pangkalan data ini berfungsi untuk menyimpan dan merekod segala data yang diperlukan oleh aplikasi KANTEEN!. Rajah 5.3.1 menunjukkan pangkalan data projek yang telah disimpan dalam Firebase Database. Servis Firebase Database ini digunakan untuk menyimpan maklumat bagi akaun pengguna, maklumat kategori menu, maklumat menu, dan pesanan. Servis Firebase Storage juga digunakan untuk menyimpan gambar yang dimuatnaik dalam pangkalan data. Maklumat ini dicapai di konsol Firebase menggunakan pelayar web.



(Rajah 5.3.1 Konsol Firebase Database)



(Rajah 5.3.2 Konsol Firebase Storage)

## 5.4 SEGMENT KOD KRITIKAL

Berikut merupakan segmen kod pengaturcaraan bagi fungsi-fungsi aplikasi KANTEEN!. Kod kritikal ini adalah kod pengaturcaraan bagi daftar akaun, log masuk, menambah kategori menu dan kemaskini menu. Kod-kod pengaturcaraan ini menggunakan Bahasa pengaturcaraan Java.

### i. Kod Kritikal Daftar Akaun

```

29  edtName = (MaterialEditText)findViewById( R.id.edtName );
30  edtPhone = (MaterialEditText)findViewById( R.id.edtPhone );
31  edtPassword = (MaterialEditText)findViewById( R.id.edtPassword );
32  btnSignUp = (Button)findViewById( R.id.btnSignUp );
33
34
35  //Init Firebase
36  final FirebaseDatabase db = FirebaseDatabase.getInstance();
37  final DatabaseReference users = db.getReference( path: "User" );
38
39
40
41
42  btnSignUp.setOnClickListener( v ) -> {
43    final ProgressDialog mDialog = new ProgressDialog( context: SignUp.this );
44    mDialog.setMessage( "Please wait ..." );
45    mDialog.show();
46
47
48    users.addListenerForSingleValueEvent( new ValueEventListener() {
49      @Override
50      public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
51        //Check if the user already exists
52        if(dataSnapshot.child( edtPhone.getText().toString() ).exists()){
53          mDialog.dismiss();
54          Toast.makeText( context: SignUp.this, text: edtPhone.getText().toString()+"Phone Number Exists !",Toast.LENGTH_LONG ).show();
55        }
56        else{
57          mDialog.dismiss();
58          User user = new User( edtPhone.getText().toString(), edtName.getText().toString(),edtPassword.getText().toString() );
59          users.child( edtPhone.getText().toString() ).setValue( user );
60          Toast.makeText( context: SignUp.this, text:"SignUp Successful",Toast.LENGTH_LONG ).show();
61        }
62      }
63    }
64  }

```

### ii. Kod Kritikal Log Masuk

```

41 btnSignIn = (Button)findViewById( R.id.btnSignIn );
42
43 //Init Firebase
44 db = FirebaseDatabase.getInstance();
45 users = db.getReference( "path:User" );
46
47 btnSignIn.setOnClickListener( (v) -> {
48     signInUser(edtPhone.getText().toString(),edtPassword.getText().toString());
49 } );
50
51 private void signInUser(final String phone, String password) {
52     final ProgressDialog mDialog = new ProgressDialog( context,SignIn.this );
53     mDialog.setMessage( "Please Wait ..." );
54     mDialog.show();
55
56     final String localPhone = phone;
57     final String localPassword = password;
58     users.addValueEventListener( new ValueEventListener() {
59         @Override
60         public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
61             if(dataSnapshot.child( localPhone ).exists()){
62                 mDialog.dismiss();
63                 User user = dataSnapshot.child( localPhone ).getValue( User.class );
64                 user.setPhone( localPhone );
65                 if( Boolean.parseBoolean( user.getIsStaff() ) ){ //If isStaff == true
66                     if( user.getPassword().equals( localPassword ) ){
67                         Toast.makeText( context, SignIn.this, text: "Staff Sign In Successful", Toast.LENGTH_LONG ).show();
68                         Intent i = new Intent( packageContext, SignIn.this, Home.class );
69                     }
70                 }
71             }
72         }
73     } );
74 }

```

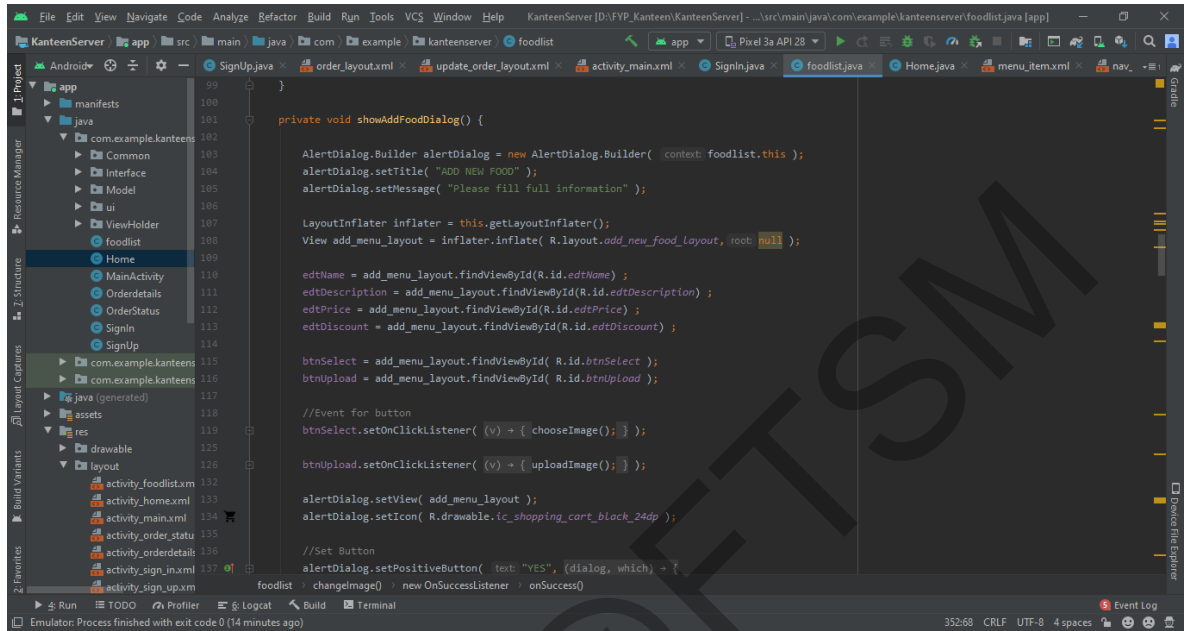
iii. Kod Kritikal Kemaskini Kategori Menu

```

146 private void showDialog() {
147     AlertDialog.Builder alertDialog = new AlertDialog.Builder( context, Home.this );
148     alertDialog.setTitle( "ADD NEW CATEGORY" );
149     alertDialog.setMessage( "Please fill full information" );
150
151     LayoutInflater inflater = this.getLayoutInflater();
152     View add_menu_layout = inflater.inflate( R.layout.add_new_menu, root: null );
153
154     edtName = add_menu_layout.findViewById( R.id.edtName );
155     btnSelect = add_menu_layout.findViewById( R.id.btnSelect );
156     btnUpload = add_menu_layout.findViewById( R.id.btnUpload );
157
158     //Event for button
159     btnSelect.setOnClickListener( (v) -> { chooseImage(); } );
160
161     btnUpload.setOnClickListener( (v) -> { uploadImage(); } );
162
163     alertDialog.setView( add_menu_layout );
164     alertDialog.setIcon( R.drawable.ic_shopping_cart_black_24dp );
165
166     //Set Button
167     alertDialog.setPositiveButton( text: "YES", (dialog, which) -> {
168         dialog.dismiss();
169
170         if( newCategory != null ){
171             categories.push().setValue( newCategory );
172
173             Snackbar.make( drawer, text: "New Category "+newCategory.getName()+" was added", Snackbar.LENGTH_SHORT ).show();
174         }
175     } );
176 }

```

iv. Kod Kritikal Kemaskini Menu



```
private void showAddFoodDialog() {  
  
    AlertDialog.Builder alertDialog = new AlertDialog.Builder( context: foodlist.this );  
    alertDialog.setTitle( "ADD NEW FOOD" );  
    alertDialog.setMessage( "Please fill full information" );  
  
    LayoutInflater inflater = this.getLayoutInflater();  
    View add_menu_layout = inflater.inflate( R.layout.add_new_food_layout, root: null );  
  
    edtName = add_menu_layout.findViewById( R.id.edtName );  
    edtDescription = add_menu_layout.findViewById( R.id.edtDescription );  
    edtPrice = add_menu_layout.findViewById( R.id.edtPrice );  
    edtDiscount = add_menu_layout.findViewById( R.id.edtDiscount );  
  
    btnSelect = add_menu_layout.findViewById( R.id.btnSelect );  
    btnUpload = add_menu_layout.findViewById( R.id.btnUpload );  
  
    //Event for button  
    btnSelect.setOnClickListener( ( v ) -> { chooseImage(); } );  
  
    btnUpload.setOnClickListener( ( v ) -> { uploadImage(); } );  
  
    alertDialog.setView( add_menu_layout );  
    alertDialog.setIcon( R.drawable.ic_shopping_cart_black_24dp );  
  
    //Set Button  
    alertDialog.setPositiveButton( text: "YES", ( dialog, which ) -> {  
        foodlist :> changeImage(); new onSuccessListener onSuccess() )
```

Copyright@FTSM



## 5.5 PENGUJIAN APLIKASI

Tujuan utama pengujian aplikasi adalah untuk mengenalpasti jika berlakunya masalah atau ralat pada fungsi setiap modul dalam aplikasi. Bagi membolehkan aplikasi berfungsi dengan baik dan mencapai objektif, setiap ralat pada aplikasi perlu dibaiki sebelum dipasarkan pada pengguna sebenar.

### 5.5.1 Pengujian Unit

Pengujian unit adalah pengujian yang dijalankan terhadap setiap elemen yang terdapat dalam aplikasi. Hal ini bagi memastikan tiada ralat syntax ataupun logik pengaturcaraan dalam setiap fungsi elemen. Pengujian dilakukan mengikut halaman agar lebih sistematik dan lengkap.

#### a. Pengujian Unit pada Halaman Permulaan

Jadual 5.1 Prosedur pengujian unit pada halaman permulaan

Kod Pengujian	Prosedur pengujian	Keputusan pengujian (Lulus/Gagal)
U101	Butang “Sign In” bertindak balas dengan memaparkan halaman “Login”.	Lulus
U102	Butang “Sign Up” bertindak balas dengan memaparkan halaman mendaftar akaun.	Lulus
U103	Butang “Login” bertindak balas dengan memaparkan paparan.	Lulus

## b. Pengujian Unit pada Halaman Utama

Jadual 5.2.1 Prosedur pengujian unit pada halaman utama bagi pengguna

Kod Pengujian	Prosedur pengujian	Keputusan pengujian (Lulus/Gagal)
U201	Elemen tabHost “List of Category” berfungsi dengan baik iaitu boleh memaparkan kepada halaman seterusnya.	Lulus
U202	Elemen tabHost “List of Menu” berfungsi dengan baik iaitu boleh memaparkan halaman kepada menu yang dipilih.	Lulus
U203	Paparan “Description of Food” dapat dipaparkan dengan baik.	Lulus
U204	Butang “Quantity” berfungsi dengan baik apabila diklik.	Lulus
U205	Butang “Add to Cart” berfungsi dengan baik apabila diklik.	Lulus
U206	Butang “Navigation” berfungsi dengan baik apabila diklik.	Lulus
U207	Butang “Cart” berfungsi dengan baik apabila diklik.	Lulus
U208	Paparan “Add to Cart” dapat memaparkan menu pilihan pelanggan dan maklumat menu tersebut dengan baik.	Lulus
U209	Butang “Place Order” dapat berfungsi dengan baik apabila diklik.	Lulus
U210	“Pop up message” dipaparkan dengan baik. Lulus	
U211	Butang “Yes” atau “No” berfungsi dengan baik apabila diklik selepas pelanggan mengisi nama anak mereka.	Lulus
U212	Paparan “Order Status” dapat dipaparkan dengan baik.	Lulus
U213	Butang “Delete Order” berfungsi dengan baik apabila diklik.	Lulus

Jadual 5.2.2 Prosedur pengujian unit pada halaman utama bagi pekerja kantin

Kod Pengujian	Prosedur Pengujian	Keputusan Pengujian (Lulus/Gagal)
U214	Elemen tabHost “List of Category” berfungsi dengan baik iaitu boleh memaparkan kepada halaman seterusnya.	Lulus
U215	Paparan “List of Menu” dapat dipaparkan dengan baik.	Lulus
U216	Butang “Navigation” berfungsi dengan baik apabila diklik.	Lulus
U217	Butang “Orders” berfungsi dengan baik apabila diklik.	Lulus
U218	Paparan “Order Status” dapat dipaparkan dengan baik.	Lulus
U219	Pengguna boleh menekan butang spinner bagi “Update Order Status” dan klik “Update” atau “Delete”.	Lulus
U230	Pengguna boleh “Update Order Status” mengikut status penyediaan menu.	Lulus
U231	Pengguna akan dapat melihat status berubah selepas dikemaskini.	Lulus

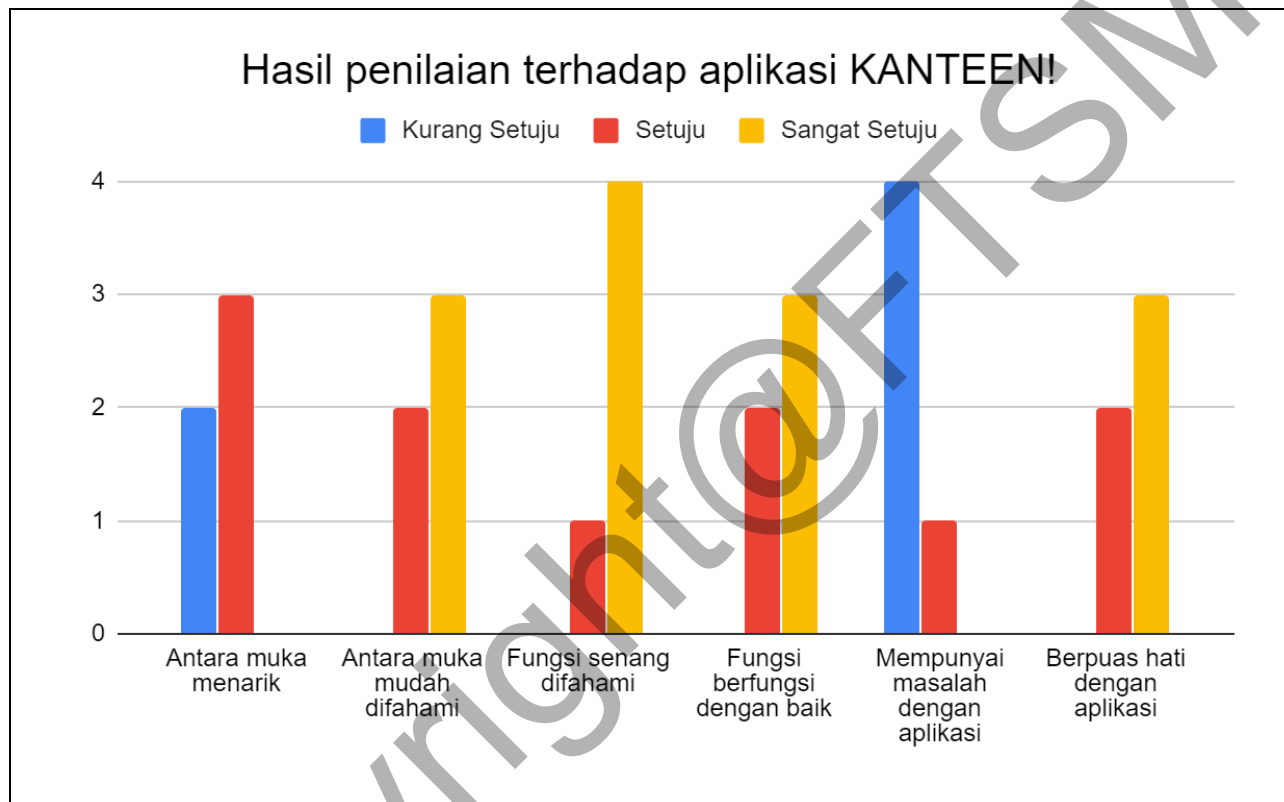
## c. Pengujian Unit pada Pembangunan Sistem Belakang

Jadual 5.3 Prosedur pengujian unit pada Pembangunan Sistem Belakang

Kod Pengujian	Prosedur Pengujian	Keputusan Pengujian (Lulus/Gagal)
U301	Elemen dialog “Add New Category” dipaparkan dan pengguna perlu mengisi butiran yang diperlukan.	Lulus
U302	Butang “Select” berfungsi dengan baik apabila diklik.	Lulus
U303	Butang “Upload” berfungsi dengan baik apabila diklik.	Lulus
U304	Butang “Yes” dan “No” berfungsi dengan baik apabila diklik.	Lulus
U305	Elemen dialog “Add New Menu” dipaparkan dan pengguna perlu mengisi butiran yang diperlukan.	Lulus
U306	Butang “Select” berfungsi dengan baik apabila diklik.	Lulus
U307	Butang “Upload” berfungsi dengan baik apabila diklik.	Lulus
U308	Butang “Yes” dan “No” berfungsi dengan baik apabila diklik.	Lulus

### 5.5.2 Pengujian Penerimaan

Kaedah yang digunakan bagi mendapatkan maklumbalas pengguna adalah melalui Google Form. Responden yang dipilih adalah dari golongan ibu bapa dan pekerja kantin. Hal ini, kerana kedua-dua golongan tersebut lebih cenderung dan sesuai untuk menjadi responden.



Rajah 5.5.1 Hasil Penilaian Aplikasi KANTEEN!

Rajah 5.5.1 menunjukkan hasil penilaian yang dijalankan kepada 5 orang pengguna. Merujuk kepada hasil penilaian untuk soalan pertama dan kedua ialah aplikasi ini mempunyai antara muka yang kurang menarik tetapi mudah difahami. Bagi soalan ketiga dan keempat pula, majoriti mengatakan fungsi aplikasi ini senang difahami dan berfungsi dengan baik ketika mereka menggunakannya. Hanya terdapat seorang pengguna yang mengatakan mereka mempunyai masalah ketika menggunakan aplikasi ini. Majoriti hasil penilaian oleh semua pengguna menunjukkan bahawa mereka berpuas hati dengan penggunaan aplikasi KANTEEN! ini.

Copyright@FTSM

## 6. KESIMPULAN

Bab ini akan membincangkan kelebihan aplikasi yang telah dibangunkan setelah melepasi beberapa siri pengujian. Kelemahan sistem juga turut dibincangkan bersama cadangan kajian masa hadapan untuk penambahbaikan aplikasi. Di samping itu, kekangan semasa pembangunan aplikasi juga turut dibincangkan.

### 6.1 KELEBIHAN APLIKASI

Keseluruhannya, sistem ini telah berjaya mencapai kesemua objektif kajian seperti yang dinyatakan dalam bab satu. Lain-lain kelebihan aplikasi adalah seperti berikut:

- i. Pengguna juga mudah memilih menu makanan yang mereka inginkan melalui aplikasi ini.
- ii. Aplikasi ini juga dapat memudahkan pekerja kantin menyediakan makanan mengikut kuantiti yang diinginkan dan mengelak pembaziran berlaku.
- iii. Maklumat yang dipaparkan juga merupakan mesra pengguna. Penggunaan gambar yang mudah difahami dan arahan yang mudah senang difahami oleh pengguna.

## 6.2 KELEMAHAN APLIKASI

Di samping kelebihan aplikasi, terdapat juga beberapa kelemahan dalam aplikasi ini. Antara kelemahan yang dikenalpasti ialah :

- i. Aplikasi KANTEEN! akan mempunyai had pemilihan makanan kerana pemilihan makanan adalah ditetapkan oleh pihak kantin.
- ii. Selain itu, aplikasi ini hanya boleh dimuat turun oleh pengguna yang menggunakan pengeoperasian *Android* sahaja. Hal ini demikian kerana, penggunaan aplikasi pembangunannya iaitu *Android Studio* yang hanya boleh digunakan untuk sistem operasi *Android* sahaja.
- iii. Aplikasi ini hanya mempunyai pembayaran secara tunai sahaja.

## 6.3 PENAMBAIKAN MASA HADAPAN

Banyak penambahbaikan yang boleh dilakukan terhadap aplikasi KANTEEN! Antaranya adalah :

- i. Memperluaskan lagi pilihan menu makanan yang sedia ada. Hal ini kerana mungkin kuota pemilihan makanan tidak dapat mengikut selera pengguna.
- ii. Aplikasi KANTEEN! perlu juga dibolehkan untuk dimuat turun oleh peranti yang menggunakan sistem pengoperasian *iOS*. Hal ini berikutan kerana sebahagian besar pengguna pada masa kini lebih cenderung menggunakan peranti *Apple iOS* berbanding *Android*.
- iii. Aplikasi ini juga perlu mengadakan pembayaran secara atas talian bagi memudahkan lagi pengguna membuat pembayaran memandangkan pengguna hari ini lebih cenderung kepada 'cashless' teknologi.
- iv. Aplikasi ini mungkin boleh dibaiki untuk lebih mesra pengguna seperti mengadakan kupon diskaun kepada pengguna biasa.



#### 6.4 KEKANGAN SEMASA PEMBANGUNAN APLIKASI

Kekangan yang dihadapi semasa membangunkan aplikasi adalah masalah kurang ilmu dalam pembinaan aplikasi. Hal ini kerana tidak mempunyai pembelajaran terhadap subjek tersebut dan tidak berapa senang untuk mempelajari daripada kawan kerana menghadapi pandemik yang memaksa pelajar tidak bertemu secara bersemuka dengan rakan. Hal ini juga menyukarkan pelajar untuk mendapatkan nasihat dan tunjuk ajar daripada penyelia dengan lebih mendalam. Kebanyakan fungsi aplikasi ini kurang difahami secara khusus dalam subjek pengaturcaraan. Bagi mengatasi permasalahan ini, rujukan mengenai pengekodan banyak dilakukan melalui tutorial di Youtube dan contoh-contoh pengekodan yang terdapat di internet.

#### 6.5 KESIMPULAN

Kesimpulannya, aplikasi KANTEEN! ini telah dibangunkan mengikut perancangan yang telah dirancang berpandukan metodologi kajian. Objektif dan penyelesaian masalah telah berjaya diterapkan walaupun masih mempunyai sedikit ralat yang perlu diperbaiki. Diharapkan kewujudan aplikasi ini akan dapat memberi manfaat kepada penggunanya.

#### 7. RUJUKAN

Draw.io , URL: <https://www.draw.io/?libs=general;uml>

Lucid Chart, Diagram Design : <https://www.lucidchart.com/>

MockPlus, Interface Design : <https://www.mockplus.com/>

Pogue, David (November 4, 2009). "A Place to Put Your Apps". New York Times. Dicapai September 28, 2019.

Sequence Diagram, URL: [https://en.wikipedia.org/wiki/Sequence\\_diagram](https://en.wikipedia.org/wiki/Sequence_diagram). [20 November 2019].

UML Use Case Diagrams, URL: <http://www.uml-diagrams.org> [24 Oktober 2017]. "Waterfall Software Development Model". 5 February 2014. Retrieved 11 August 2014.