

**LAPORAN PROJECT**  
**MOBILE APPLICATION DEVELOPMENT**  
**Perancangan Aplikasi Mobile “Hongstar Fast Food”**  
**Menggunakan Software Android Studio**



**Disusun Oleh :**

Dava Virgio Kertawijaya	(00000056848)
Michael Owen Kohar	(00000056755)
Nigel Andrian	(00000055946)
Samuel Andrew	(00000056975)
Westley Wijaya	(00000056414)

**Program:** Information Systems  
**Matakuliah:** Mobile Application Development

**UNIVERSITAS MULTIMEDIA NUSANTARA**  
**TANGERANG**  
**GANJIL 2021/2022**

## HALAMAN PENGESAHAN PROPOSAL

Disusun oleh tim *Mobile Application Development*

NIM	Nama	Program Studi	No HP
00000056848	Dava Virgio Kertawijaya	Sistem Informasi	089527193296
00000056755	Michael Owen Kohar	Sistem Informasi	08981040213
00000055946	Nigel Andrian	Sistem Informasi	082179588619
00000056975	Samuel Andrew	Sistem Informasi	088809623099
00000056414	Westley Wijaya	Sistem Informasi	081299026742

<b>Topik</b>	:	Perancangan Aplikasi Mobile “Hongstar Fast Food” Menggunakan Software Android Studio
<b>Judul Proposal (Indonesia)</b>	:	Perancangan Aplikasi Mobile “Hongstar Fast Food” Menggunakan Software Android Studio
<b>Judul Proposal (Inggris)</b>	:	Designing "Hongstar Fast Food" Mobile Application Using Android Studio Software
<b>Usulan Pembimbing</b>	:	1. L00079 – Agus Sulaiman

## HALAMAN PENILAIAN REVIEWER

Topik Final (Diisi oleh Dosen):

Hasil Review Proposal Skripsi (Diisi oleh Dosen):

<b>Tanggal Review</b>		
<b>Keputusan</b>		Diterima / Revisi
<b>Paraf Reviewer</b>		

# **BAB I**

## **PENDAHULUAN**

### **1. LATAR BELAKANG**

Fast food / makanan cepat saji telah dikenal oleh masyarakat sebagai alternatif makanan yang dapat dibeli dan dikonsumsi dengan cepat dan mudah. Kata makanan cepat saji itu sendiri berarti makanan yang cepat saji dimana konsumen tidak memerlukan waktu yang lama untuk makanan yang mereka pesan untuk dapat langsung dikonsumsi karena prosesnya mudah. Konsep dari makanan cepat saji itu sendiri berawal mula pada Romawi Kuno yang menjual roti yang telah dicelup dengan minuman anggur, sebagai makanan yang cepat untuk dikonsumsi sebagai sarapan. Makanan cepat saji diciptakan untuk para insan yang tidak memiliki banyak waktu karena disibukkan oleh kegiatan maupun pekerjaan sehingga mereka memilih makanan yang cepat saji sebagai alternatif terbaik sebagai asupan tenaga ketika lapar dan harus buru-buru. Namun sejak tahun 1912 makanan cepat saji berubah haluan menjadi makanan yang telah diproses dahulu agar proses penyajian lebih cepat dan juga bukan hanya dikonsumsi bagi orang-orang yang sibuk saja namun kepada semua lapisan masyarakat terutama anak-anak muda.

Kemajuan dunia teknologi yang sangat pesat dalam informasi menuntut para pelaku usaha untuk dapat memahami serta memanfaatkannya dalam kegiatan bisnis sebagai upaya untuk meningkatkan daya saing. Meningkatnya jumlah pemakai internet secara drastis setiap tahunnya, serta perilaku konsumen yang menginginkan informasi cepat dan transaksi yang dapat dilakukan selama 24 jam, disinyalir menjadi faktor pendorong berkembangnya bisnis berbasis teknologi informasi menjadikan para pebisnis bersaing menciptakan inovasi terbaru serta melihat dari fenomena yang terjadi dimasyarakat. Saat ini internet sangat berpengaruh dalam kehidupan masyarakat. Maraknya pengguna smartphone, baik android maupun ios menjadikan masyarakat bergantung pada handphone serta internet. Apalagi pada saat ini para pengusaha kuliner menggunakan kecanggihan teknologi untuk memasarkan dagangan atau makanannya melalui internet.

Menurut, Salim dan Ihalauw (2017) Inovasi model bisnis yang terus berkembang merupakan suatu langkah strategis bagi perusahaan, yang menjadi penentu langkah sukses atau tidaknya suatu perusahaan dalam persaingan bisnis yang saat ini menjadi semakin ketat untuk dapat tumbuh dan lestari. Model bisnis yang baru muncul mendesak model bisnis lama untuk berubah untuk mengikuti perkembangan zaman atau tergerus jaman. Penemuan internet dan inovasi teknologi, khususnya teknologi informasi dan digital mendorong perubahan yang pesat pada cara manusia berinteraksi dan berkomunikasi, demikian juga pada cara manusia berbisnis.

Hal itu menunjukkan alasan kenapa begitu banyak perusahaan berinvestasi pada teknologi informasi yang cukup tinggi dan sistem kompleks yang saling tersambung dengan jaringan komputer, mengingat perusahaan sedang mencari konsumennya lewat internet. Karena kebutuhan konsumen berkembang sangat cepat, perusahaan berharap dapat berkomunikasi dengan konsumennya lewat internet. Sehingga, internet menjadi salah satu strategi dan memberi kesempatan kepada perusahaan besar dan kecil untuk menawarkan secara cepat dengan biaya murah, menanggapi produk dan pelayanan

dengan kualitas tinggi sesuai dengan kebutuhan konsumen. Internet membuat wajah baru untuk komunikasi interaktif antara konsumen, penjual, dan rekan bisnis lainnya. Hal ini memungkinkan bahwa perusahaan mampu berinteraksi dan bekerja sama secara terus menerus dalam pengembangan produk, pemasaran, pengiriman, pelayanan, dan dukungan teknik. Salah satunya dengan menggunakan layanan yang telah kami buat bernama “Hongstar Fast Food” untuk mengantar dan memesan makanan yang berbasis online.

## **2. RUMUSAN MASALAH**

Berdasarkan latar belakang yang telah dipaparkan, terdapat beberapa permasalahan yang ditemukan dalam perkembangan sistem aplikasi yang masih ramai digunakan agar solusi dapat segera diaplikasikan dan dirumuskan, tanpa permasalahan yang berlebih saat sudah ditujukan kepada *end-users*. Permasalahan tersebut antara lain:

- Sistem “order makanan” yang sudah ada dirasa masih kurang
- Masih kurang ada nya sebuah aplikasi yang dapat mengorder makanan
- Bagaimana pengaruh harga terhadap kepuasan pelanggan

## **3. TUJUAN DAN MANFAAT**

Tujuan :

1. Memberikan kemudahan kepada konsumen dalam memilih pesanan yang ada
2. Memberikan pelayanan kepada konsumen yang praktis dan fleksibel
3. Membantu kasir dalam menginput data makanan yang telah di pesan
3. Mempermudah konsumen dalam memesan makan

Manfaat :

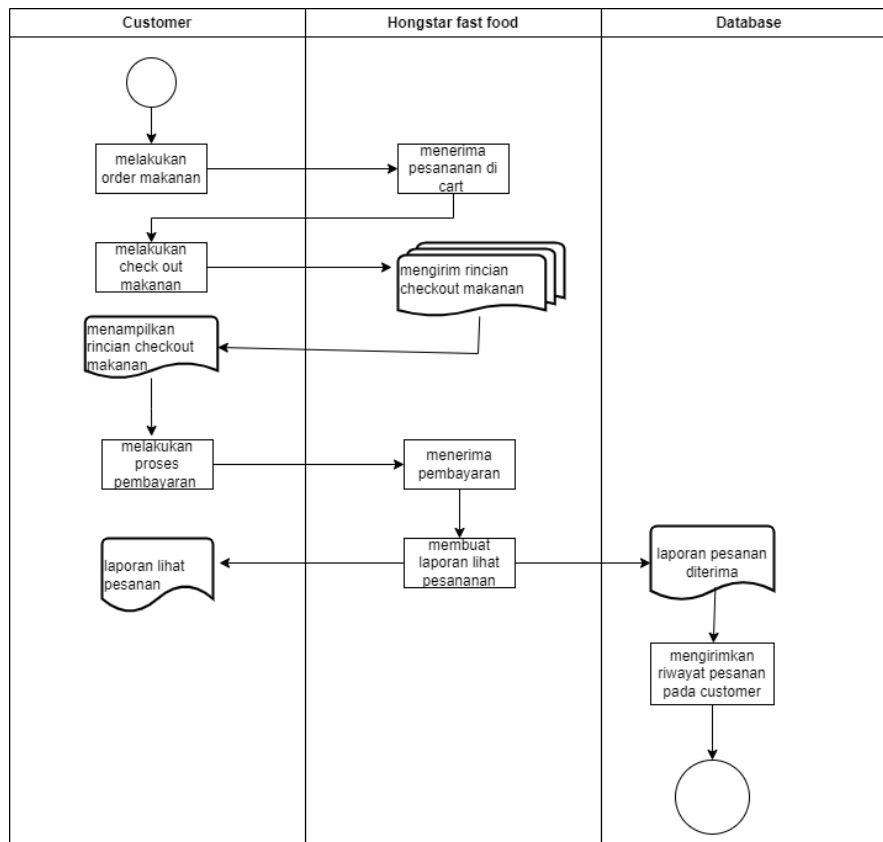
1. Mempermudah masyarakat dalam menentukan pesanan yang ingin di beli
2. Mengurangi human error di dalam pengorderan yang dilakukan oleh kasir
3. Menambah tingkat penjualan dengan cara yang lebih praktis

## **4. PROFIL PERUSAHAAN**

Hongstar Fast Food merupakan aplikasi mobile dari Indonesia yang bentuk oleh sekelompok mahasiswa UMN. Dalam aplikasi tersebut akan menampilkan berbagai macam menu dari Hongstar Fast Food yang kemudian akan kami terima untuk melakukan pemesanan makanan tersebut.

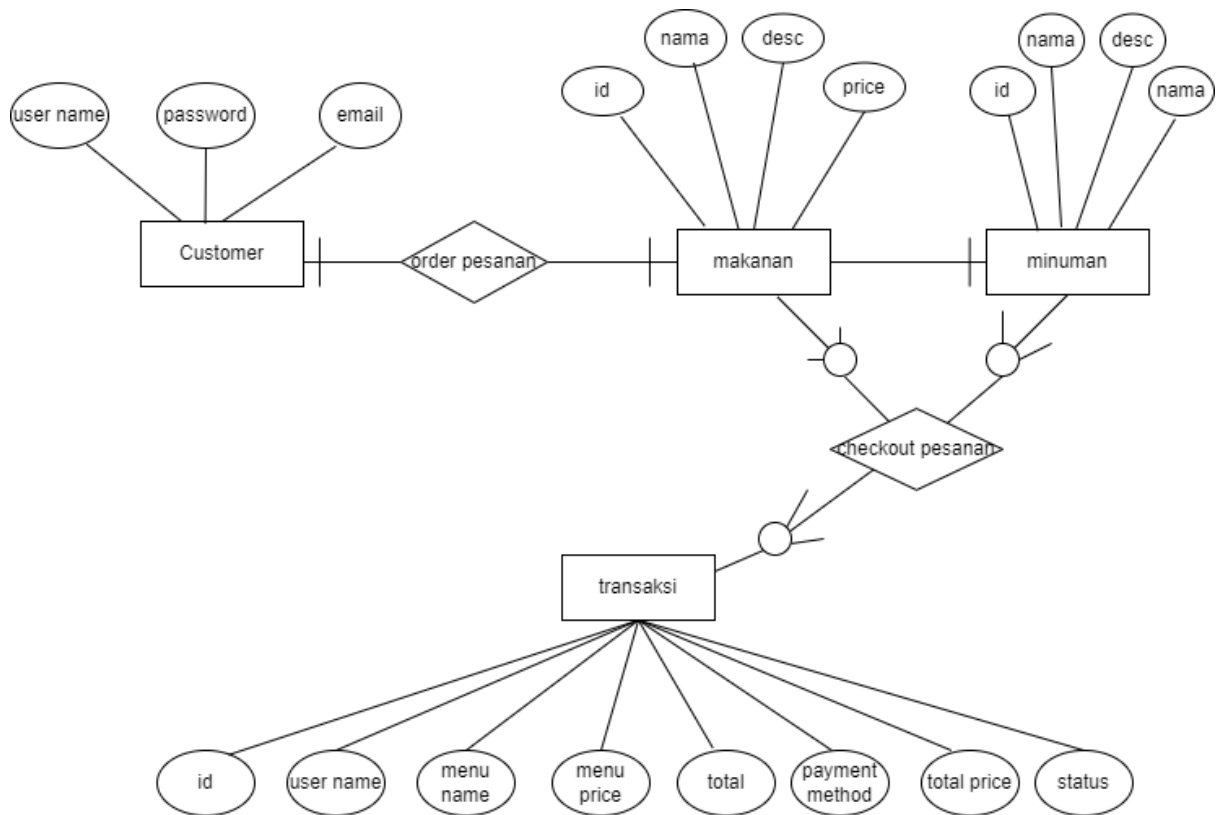
## BAB II ISI

### 2.1 Proses Bisnis



Proses bisnis merupakan setiap rangkaian aktivitas yang saling terkait dan saling bekerja bersama di seluruh organisasi, untuk mencapai beberapa tujuan organisasi yang telah ditentukan sebelumnya, yang biasanya didefinisikan seputar pemenuhan kebutuhan pelanggan. Proses bisnis yang baik adalah proses yang terdiri dari aktivitas yang berurut sesuai dengan runtutan waktu dan terstruktur dengan jelas. Gambar di atas merupakan proses bisnis yang diterapkan pada aplikasi Hongstar Fast Food. Pertama, user dapat mengunduh aplikasi Hongstar Fast Food terlebih dahulu agar dapat menggunakan aplikasi tersebut, kemudian user dapat melakukan register akun terlebih dahulu untuk melakukan login. Setelah user masuk ke dalam aplikasi maka user dapat memilih menu yang telah kami sediakan. Menu yang telah user pilih akan masuk ke dalam cart di dalam aplikasi kami, setelah itu user dapat mengakses menu cart untuk mengedit menu yang dipilih jika ingin menambahkan notes dan lain lain. Kemudian, user dapat melakukan check out makanan lalu user memilih metode pembayaran yang ingin dilakukan. Setelah itu admin akan menerima pesanan dari user dan akan membuat rincian transaksi dan makanan yang akan dikirimkan ke user. Kemudian, user dapat melihat rincian pesanan makanan yang telah dibayar, kemudian admin akan mengupdate status makanan jika makanan yang di pesan telah selesai di buat, user juga bisa melihat riwayat pesanan dan transaksi.

## 2.2 ER Diagram



ERD adalah singkatan dari Entity Relationship Diagram, ERD sering juga disebut sebagai diagram ER atau model ER. Sederhananya, ERD adalah jenis diagram struktur yang sering digunakan saat mendesain database atau rencana bisnis.

ERD terdiri dari banyak komponen, seperti simbol, tetapi juga konektor yang berbeda. Komponen ini menampilkan dua informasi penting. Informasi pertama adalah entitas utama yang berada dalam ruang lingkup sistem. Untuk menggambarkan atau membuat model atau struktur suatu *database* dengan diagram yang sederhana sehingga memudahkan dalam membuat sebuah *database*, maka buatlah ER diagram di atas. Pada ER diagram kami terdiri dari 4 entitas, yaitu *Customer*, makanan, minuman dan transaksi.

Dapat kita lihat pada ER Diagram di atas dimulai dari entitas customer yang memiliki atribut username, password, email. Pada atribut ini, akan digunakan sebagai data yang akan masuk ke dalam database kami untuk mendata orderan pesanan yang akan dilakukan oleh customer. Kemudian dilanjutkan dengan entitas makanan dan minuman yang dimana di dalam entitas tersebut terdapat beberapa atribut, yaitu id, nama, desc, dan price. *id* merupakan kode yang diberikan kepada makanan, *Nama* merupakan jenis makanan yang dipilih oleh customer kemudian *Desc* menggambarkan seperti apa

makanan yang dipesan oleh customer lalu price yang akan menampilkan harga dari makanan yang ingin dibeli oleh customer. Setelah entitas makanan dan minuman, terdapat juga entitas transaksi. Pada transaksi tersebut terdapat atribut id, username, menu name, menu price, total, payment method, total price, dan status. *id* pada transaksi merupakan kode yang diberikan saat kita memesan makanan dan akan digunakan pada pembayaran. Kemudian, ada username yang dimana username tersebut akan memastikan nama customer ketika memesan makanan lalu ada *menu name* merupakan setiap menu makanan yang telah di order oleh customer kemudian *menu price* harga dari setiap menu yang di pesan lalu *total* yang merupakan total dari makanan yang telah kita pesan lalu *payment method* yang dimana kita bisa memilih metode pembayaran yang ingin dilakukan. Setelah itu, ada *total price* yang menampilkan total seluruh pesanan yang telah di order oleh kita, total price tersebut bisa digunakan di payment method untuk melakukan pembayaran, dan yang terakhir *status* di dalam status akan menampilkan pesanan yang telah dipesan lalu menampilkan pembayaran yang telah berhasil.

1. Tabel Customer

Atribut: username

Atribut: password

Atribut: email

2. Tabel makanan

Atribut: id

Atribut: nama

Atribut: desc

Atribut: price

3. Tabel minuman

Atribut: id

Atribut: nama

Atribut: desc

Atribut: price

4. Tabel transaksi

Atribut: id

Atribut: username

Atribut: menu name

Atribut: menu price

Atribut: total

Atribut: payment method

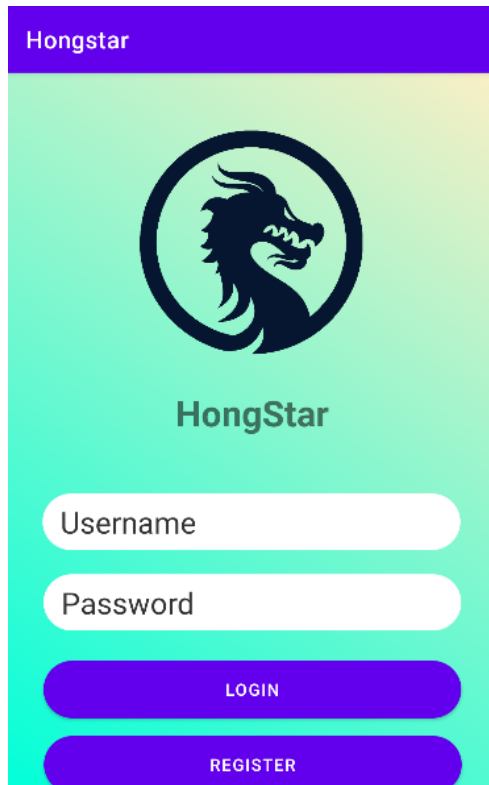
Atribut: total price

Atribut: status

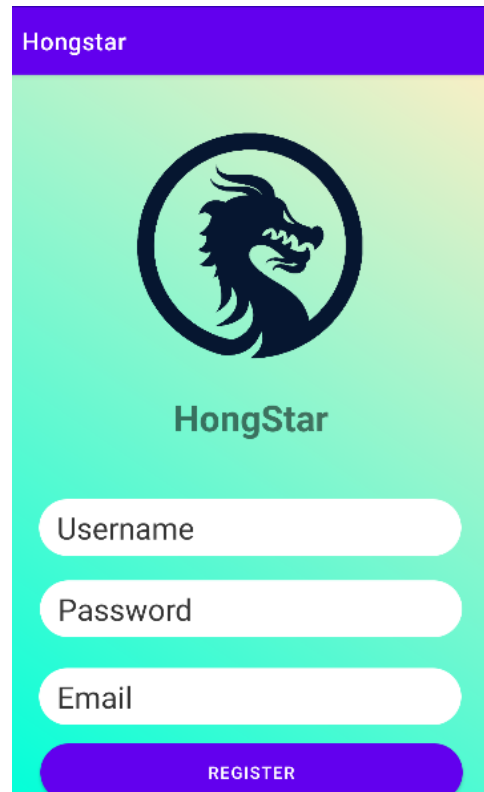


## 2.3 Layout Aplikasi

### *login dan register*



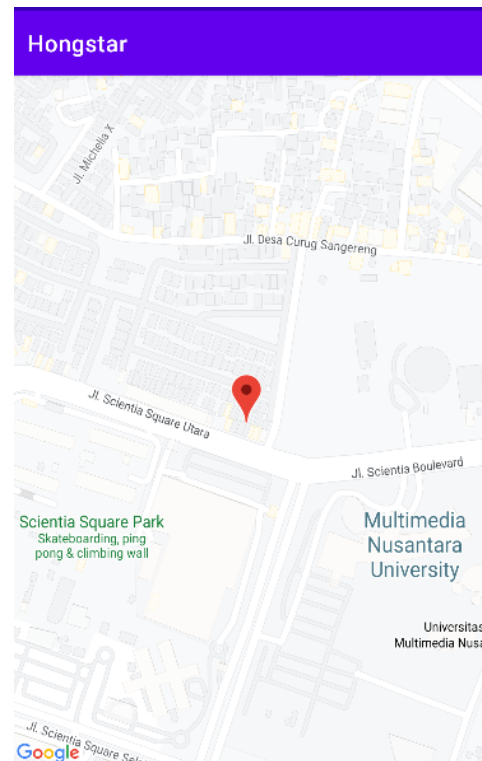
The screenshot shows the login interface of the HongStar application. At the top, there is a purple header with the text "Hongstar". Below the header is a light green background with a circular logo featuring a black silhouette of a dragon's head. Under the logo, the text "HongStar" is displayed in a bold, black font. Below the logo and text, there are two white input fields with rounded corners. The first field is labeled "Username" and the second field is labeled "Password". Below these fields are two purple buttons with white text. The top button is labeled "LOGIN" and the bottom button is labeled "REGISTER".



The screenshot shows the register interface of the HongStar application. At the top, there is a purple header with the text "Hongstar". Below the header is a light green background with a circular logo featuring a black silhouette of a dragon's head. Under the logo, the text "HongStar" is displayed in a bold, black font. Below the logo and text, there are three white input fields with rounded corners. The first field is labeled "Username", the second field is labeled "Password", and the third field is labeled "Email". Below these fields is a single purple button with white text labeled "REGISTER".

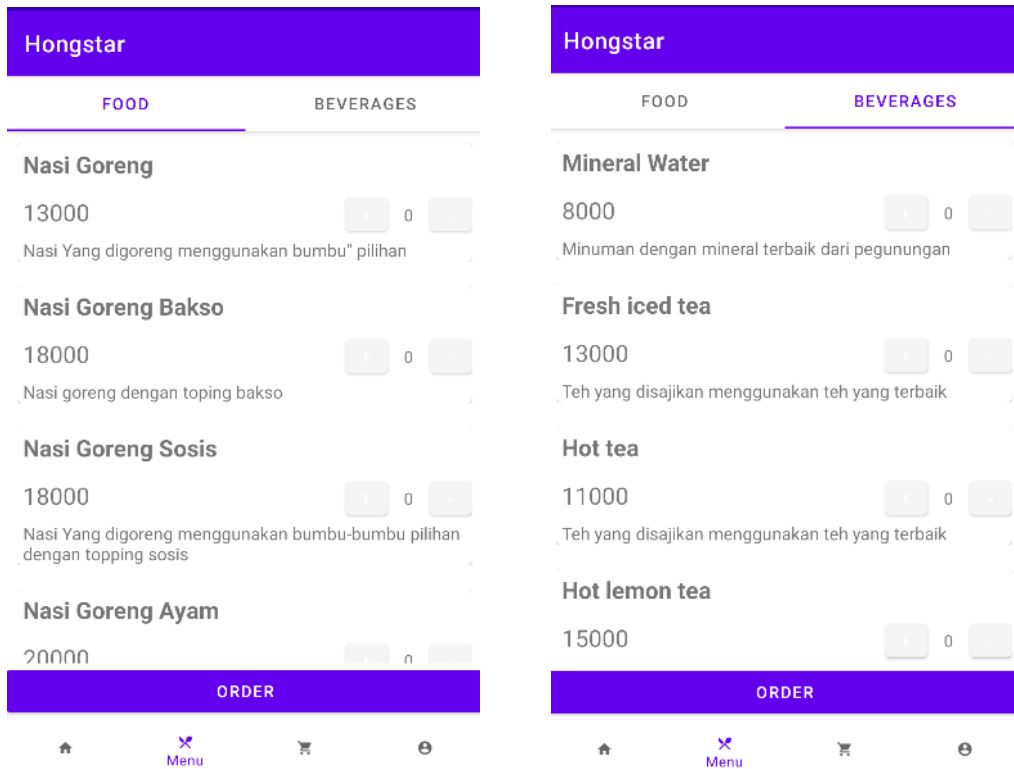
Di atas merupakan tampilan *user interface* login dan register, pada tampilan ini terdapat 2 button login dan register, pada button "*register*" dapat berfungsi untuk melakukan pendaftaran akun untuk menjadi user pada aplikasi kami, di button register tersebut kami meminta untuk memasukan username, password, dan email yang berguna untuk pendataan kami, kemudian button "*login*" digunakan jika user sudah membuat akun atau telah memiliki akun sebelum nya, maka user dapat memasukan username dan password yang telah user buat untuk masuk ke dalam aplikasi kami

## home



Di atas merupakan tampilan *user interface* dari home, apabila user sudah melakukan login pada halaman pertama kami, maka menu dari home akan muncul, pada menu “home” ini akan menampilkan pesanan yang sebelum nya masih berada di cart dan belum melakukan proses pembayaran, tampilan tersebut akan muncul secara otomatis jika user tidak menyelesaikan proses pesanan nya kemudian di dalam home tersebut, kami juga menyediakan fitur peta, yang dapat berguna untuk mencari tahu lokasi.

## menu



Gambar di atas merupakan tampilan dari UI menu makanan, user dapat melakukan pemesanan makanan maupun minuman pada menu ini. Pada tampilan ini sudah terdapat harga pada setiap makanan maupun detail penjelasan dari masing masing makanan tersebut.

### *make order*

The screenshot shows a mobile application interface for a restaurant named 'Hongstar'. It displays a list of food items with their prices and quantity controls. The items are: Nasi Goreng (13000), Nasi Goreng Bakso (18000), Nasi Goreng Sosis (18000), Mineral Water (8000), and Fresh iced tea (12000). Each item has a minus button, a quantity display (currently 1), and a plus button. At the bottom, there is a red button labeled 'MAKE ORDER'.

Item	Price	Quantity
Nasi Goreng	13000	1
Nasi Goreng Bakso	18000	1
Nasi Goreng Sosis	18000	1
Mineral Water	8000	1
Fresh iced tea	12000	1

Gambar di atas merupakan tampilan jika kita sudah selesai memilih menu makanan yang sudah kita pesan sebelum nya. Pada tampilan ini, user dapat mengedit jumlah makanan atau minuman yang akan dipesan atau juga bisa membatalkan pesanan yang sudah di pesan sebelum nya. Setelah itu, user dapat melakukan order makanan untuk segera diproses dan data dari orderan tersebut akan disimpan di dalam database.

*cart*

Hongstar	
DALAM PROSES	RIWAYAT PESANAN
31	
15000	
paid	
33	
39000	
paid	
34	
39000	
paid	
35	
39000	
paid	
36	

Hongstar	
DALAM PROSES	RIWAYAT PESANAN
30	
15000	
DONE	
32	
105000	
DONE	
37	
52000	
DONE	

Gambar di atas merupakan *User Interface* menu cart/keranjang. Pada menu ini user dapat melihat proses pemesanan yang sudah di order apakah sudah terbayar atau belum. Kemudian, user juga dapat melihat riwayat pemesanan atau transaksi yang sudah selesai sebelumnya.

*detail pesanan*

Hongstar

**DETAIL PESANAN**

TRX ID : 32

Nama Customer : NAIJEL

Waktu : 2022.12.12 AD at 16:49:00 GMT+07:00

Location : Ruko Newton No 32 Timur

Detail Pesanan :

**Nasi Goreng**

15000 1x

**Nasi Goreng Sosis**

18000 2x

**Nasi Goreng Ayam**

20000 1x

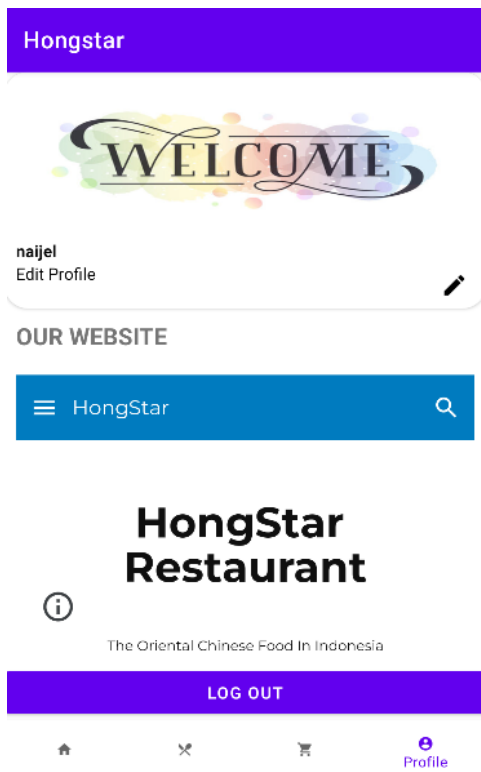
Metode Pembayaran : SHOPEEPAY

**TOTAL :** Rp 105000

**Status :** DONE

Pada gambar di atas merupakan gambar jika kita sudah selesai melakukan pembayaran. Bisa dilihat bahwa tampilan makanan yang sebelumnya telah di pesan akan masuk ke dalam detail pesanan kemudian total harga dari makanan yang pesan akan terhitung dan status pembayaran dari pesanan kita bisa kita lihat *done* atau pembayaran telah berhasil dilakukan.

### *profile*



Gambar di atas merupakan **User Interface** menu *Profil*. Pada tampilan ini, user dapat mengakses profile user. Selain itu, kami juga menyediakan website yang jika ditekan maka akan otomatis menuju ke link website. Setelah itu, user juga dapat melakukan logout jika user ingin mengganti akun ke akun yang lain.

### *edit profile*

**Edit Profile**

Username

naijel

Password

.....

Email

naijel@gmail.com

**SIMPAN**

Gambar di atas merupakan *User Interface* pada menu edit profile. User dapat mengubah/mengedit profile mereka sesuai yang mereka inginkan, seperti mengubah username, password maupun email.



## 2.4 Coding Aplikasi Android Studio

### *Coding aplikasi register*

```
public class register extends AppCompatActivity {

    private static final String TAG_USER = "data";
    private Button btncreateacc;
    private EditText edtusername,edtpassword,edtemail;
    String username,password,email;

    private final static String url_register = "http://192.168.0.102/hongstarvs/adduser.php";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);

        edtusername = (EditText) findViewById(R.id.txtusername);
        edtpassword = (EditText) findViewById(R.id.txtpassword);
        edtemail = (EditText) findViewById(R.id.txtemail);
        btncreateacc = (Button) findViewById(R.id.btncreateacc);

        btncreateacc.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                username = edtusername.getText().toString();
                password = edtpassword.getText().toString();
                email = edtemail.getText().toString();

                RequestQueue queue = Volley.newRequestQueue( context.register.this);
```

```
                btncreateacc.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {

                        username = edtusername.getText().toString();
                        password = edtpassword.getText().toString();
                        email = edtemail.getText().toString();

                        RequestQueue queue = Volley.newRequestQueue( context.register.this);

                        StringRequest stringRequest = new StringRequest(Request.Method.POST, url_register, new Response.Listener<String>() {
                            @Override
                            public void onResponse(String response) {
                                try {
                                    JSONObject jsonObj = new JSONObject(response);
                                    int sukses = jsonObj.getInt( "name: "success");
                                    if (sukses == 1) {
                                        Toast.makeText( context.register.this, text: "User Berhasil ditambah", Toast.LENGTH_SHORT).show();
                                        Intent intent = new Intent(getApplicationContext(), MainActivity.class);
                                        startActivity(intent);
                                    } else if (sukses == 0) {
                                        Toast.makeText( context.register.this, text: "User gagal ditambah", Toast.LENGTH_SHORT).show();
                                    } else if (sukses == -1){
                                        Toast.makeText( context.register.this, text: "Tolong Isi semuanya dengan benar", Toast.LENGTH_SHORT).show();
                                    } else{
                                        Toast.makeText( context.register.this, text: "Username Sudah Ada", Toast.LENGTH_SHORT).show();
                                    }
                                } catch (Exception ex) {
                                    Log.e( tag: "Error", ex.toString());
                                }
                            }
                        });
```

```
                    }
                }, new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        Toast.makeText( context.register.this, text: "Fail to get response = " + error, Toast.LENGTH_SHORT).show();
                    }
                }) {
                    @Override
                    protected Map<String, String> getParams() {
                        Map<String, String> params = new HashMap<>();
                        params.put("username", username);
                        params.put("password", password);
                        params.put("email", email);
                        return params;
                    }

                    @Override
                    public Map<String, String> getHeaders() throws AuthFailureError {
                        Map<String, String> params = new HashMap<>();
                        params.put("Content-Type", "application/x-www-form-urlencoded");
                        return params;
                    }
                });
                queue.getCache().clear();
                stringRequest.setRetryPolicy(new DefaultRetryPolicy( initialTimeoutMs: 20 * 1000, maxNumRetries: 1, backoffMultiplier: 1.0f));
                queue.add(stringRequest);
            }
        });
    }
}
```

Gambar di atas merupakan code yang telah kami buat di dalam android studio untuk dapat menjalankan proses register user. Pada codingan tersebut, kami menggunakan database agar dapat menampung data sign up yang telah dilakukan oleh user. Bukan hanya itu, kami juga melakukan hosting dengan menggunakan php di dalam localhost. Dalam code tersebut, kami juga menambahkan library volley yang akan berguna untuk melakukan input data ketika user melakukan sign up, yang dimana jika user telah berhasil melakukan register maka akan muncul toast *"user berhasil ditambah"*. Kemudian, jika user mengisi data yang telah ada di database, maka akan muncul juga toast *"username sudah ada"*.

## Maps

```
public class Map extends Fragment {
    private GoogleMap mMap;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view=inflater.inflate(R.layout.fragment_map, container, attachToRoot: false);

        // Initialize map fragment
        SupportMapFragment supportMapFragment=(SupportMapFragment)
            getChildFragmentManager().findFragmentById(R.id.google_map);

        // Async map
        supportMapFragment.getMapAsync(new OnMapReadyCallback() {
            @Override
            public void onMapReady(GoogleMap googleMap) {
                mMap = googleMap;

                // Add a marker in Sydney and move the camera
                LatLng hongstar = new LatLng(-6.255755, 106.616926);
                mMap.addMarker(new MarkerOptions().position(hongstar).title("Hongstar"));
                mMap.moveCamera(CameraUpdateFactory.newLatLng(hongstar));
                mMap.animateCamera(CameraUpdateFactory.zoomTo( 17.0f ));
            }
        });
        // Return view
        return view;
    }
}
```

```
        // Return view
        return view;
    }

    private BitmapDescriptor BitmapFromVector(Context context, int vectorResId) {
        // below line is use to generate a drawable.
        Drawable vectorDrawable = ContextCompat.getDrawable(context, vectorResId);

        // below line is use to set bounds to our vector drawable.
        vectorDrawable.setBounds(0, 0, vectorDrawable.getIntrinsicWidth(), vectorDrawable.getIntrinsicHeight());

        // below line is use to create a bitmap for our
        // drawable which we have added.
        Bitmap bitmap = Bitmap.createBitmap(vectorDrawable.getIntrinsicWidth(), vectorDrawable.getIntrinsicHeight(), Bitmap.Config.ARGB_8888);

        // below line is use to add bitmap in our canvas.
        Canvas canvas = new Canvas(bitmap);

        // below line is use to draw our
        // vector drawable in canvas.
        vectorDrawable.draw(canvas);

        // after generating our bitmap we are returning our bitmap.
        return BitmapDescriptorFactory.fromBitmap(bitmap);
    }
}
```

Gambar di atas merupakan *code* dari ‘Maps’. Pada saat customer masuk ke dalam aplikasi, maka user dapat melihat lokasi yang berada di tempat user tersebut. Lalu, user dapat menentukan lokasi tempat yang ingin dituju dalam pengorderan makanan. Setelah user mengorder makanan, user juga bisa melihat tempat di dalam peta yang telah ditampilkan oleh admin.

## Main Activity

```
3 import ...
33
34 public class MainActivity extends AppCompatActivity {
35
36     private static final String TAG_USER = "data";
37     private Button btnlogin, btnregister;
38     private EditText edtusername, edtpassword, edtemail;
39     String username, password, email;
40     int i=0;
41
42     private final static String url_login = "http://192.168.0.102/hongstarvs/login.php";
43
44     @Override
45     protected void onCreate(Bundle savedInstanceState) {
46         super.onCreate(savedInstanceState);
47         setContentView(R.layout.activity_main);
48
49         btnlogin = (Button) findViewById(R.id.btnlogin);
50         btnregister = (Button) findViewById(R.id.btnregister);
51         edtusername = (EditText) findViewById(R.id.txtusername);
52         edtpassword = (EditText) findViewById(R.id.txtpassword);
53         edtemail = (EditText) findViewById(R.id.txtemail);
54
55
56         btnlogin.setOnClickListener(new View.OnClickListener() {
57             @Override
58             public void onClick(View v) {
59                 if (edtpassword.getText().toString().equals("") && edtusername.getText().toString().equals("")){
60                     Toast.makeText(getApplicationContext(),
61                         text: "Isi Username dan Password",
```

```
btnlogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (edtpassword.getText().toString().equals("") && edtusername.getText().toString().equals("")){
            Toast.makeText(getApplicationContext(),
                text: "Isi Username dan Password",
                Toast.LENGTH_LONG).show();
            return;
        }
        username = edtusername.getText().toString();
        password = edtpassword.getText().toString();

        RequestQueue queue = Volley.newRequestQueue( context: MainActivity.this);

        StringRequest stringRequest = new StringRequest(Request.Method.POST, url_login, new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                try {
                    JSONObject jsonObj = new JSONObject(response);
                    int sukses = jsonObj.getInt( name: "success");
                    if (sukses == 1) {
                        Toast.makeText( context: MainActivity.this, text: "Selamat Datang " + username, Toast.LENGTH_SHORT).show();
                        Intent intent = new Intent(getApplicationContext(), HomeActivity.class);
                        intent.putExtra( name: "username", username);
                        startActivity(intent);
                    } else if( sukses == 2 ) {
                        Toast.makeText( context: MainActivity.this, text: "Selamat Datang Admin", Toast.LENGTH_SHORT).show();
                        Intent intent = new Intent(getApplicationContext(), HomeAdmin.class);
                        intent.putExtra( name: "username", username);
                    }
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        });
        queue.add(stringRequest);
    }
});
```

```
int sukses = jobJ.getIntent().getIntExtra("sukses", 0);
if (sukses == 1) {
    Toast.makeText(context MainActivity.this, text "Selamat Datang " + username, Toast.LENGTH_SHORT).show();
    Intent intent = new Intent(getApplicationContext(), HomeActivity.class);
    intent.putExtra(name: "username", username);
    startActivity(intent);
} else if (sukses == 2) {
    Toast.makeText(context MainActivity.this, text "Selamat Datang Admin", Toast.LENGTH_SHORT).show();
    Intent intent = new Intent(getApplicationContext(), HomeAdmin.class);
    intent.putExtra(name: "username", username);
    startActivity(intent);
} else {
    Toast.makeText(context MainActivity.this, text "User Id tidak ditemukan", Toast.LENGTH_SHORT).show();
}
} catch (Exception ex) {
    Log.e(tag: "Error", ex.toString());
}
}
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        Toast.makeText(context MainActivity.this, text "Fail to get response = " + error, Toast.LENGTH_SHORT).show();
    }
}) {
    @Override
    protected Map<String, String> getParams() {
        Map<String, String> params = new HashMap<>();
        params.put("username", username);
        params.put("password", password);
        return params;
    }
}
```

```
@Override
protected Map<String, String> getParams() {
    Map<String, String> params = new HashMap<>();
    params.put("username", username);
    params.put("password", password);
    return params;
}

@Override
public Map<String, String> getHeaders() throws AuthFailureError {
    Map<String, String> params = new HashMap<>();
    params.put("Content-Type", "application/x-www-form-urlencoded");
    return params;
}
};
queue.getCache().clear();
stringRequest.setRetryPolicy(new DefaultRetryPolicy(initialTimeoutMs: 20 * 1000, maxNumRetries: 1, backoffMultiplier: 1.0f));
queue.add(stringRequest);
}
});

btnregister.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(getApplicationContext(), register.class);
        startActivity(intent);
    }
});
});
```

Pada ‘Main Activity’, terdapat fitur login dan register yang dimana jika user masuk ke dalam aplikasi, maka akan muncul button login dan register. User dapat melakukan register terlebih dahulu jika tidak memiliki akun. Lalu, dapat melakukan login jika sudah memiliki akun. Bisa kita lihat dalam codingan di atas jika user telah berhasil login maka akan muncul toast “selamat datang user”.

## Menu

```
public class Menu extends Fragment {
    private ViewPagerAdapter viewPagerAdapter;
    private ViewPager viewPager;
    private TabLayout tabLayout;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_menu, container, attachToRoot: false);
    }

    public void onStart() {
        super.onStart();
        viewPager = getActivity().findViewById(R.id.viewpagermenu);
        tabLayout = getActivity().findViewById(R.id.tab_layout);
        Button btnorder = getActivity().findViewById(R.id.btnorder);

        String username = getArguments().getString(key: "username");

        btnorder.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getActivity(), PembayaranActivity.class);
                intent.putExtra(name: "username", username);
                startActivity(intent);
            }
        });
    }
}
```

```
tabLayout = getActivity().findViewById(R.id.tab_layout);
Button btnorder = getActivity().findViewById(R.id.btnorder);

String username = getArguments().getString(key: "username");

btnorder.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(getActivity(), PembayaranActivity.class);
        intent.putExtra(name: "username", username);
        startActivity(intent);
    }
});

viewPagerAdapter = new ViewPagerAdapter(getChildFragmentManager());
viewPagerAdapter.setData(username);

viewPager.setAdapter(viewPagerAdapter);
tabLayout.setupWithViewPager(viewPager);
}
```

Code di atas merupakan code untuk *menu*. Code tersebut berguna untuk menampilkan menu-menu makanan yang ada di dalam aplikasi. Pada bagian ini, kami juga menambahkan fungsi intent yang akan menampilkan *webview* ke dalam user sehingga dapat menampilkan menu yang ada.

## home

```
27 public class Home extends Fragment {
28     private MyDBHandler dbHandler;
29     private ArrayList<notif> values;
30     private ListView List;
31
32     public Home(){
33         // require a empty public constructor
34     }
35
36     @Override
37     public View onCreateView(LayoutInflater inflater, ViewGroup Container, Bundle savedInstanceState) {
38
39         return inflater.inflate(R.layout.fragment_home, Container, attachToRoot: false);
40     }
41
42     public void onStart() {
43         super.onStart();
44
45         dbHandler = new MyDBHandler(getActivity());
46
47         try {
48             dbHandler.open();
49         } catch (SQLException e) {
50             e.printStackTrace();
51         }
52
53         values = dbHandler.getAllBarang();
54
55         ArrayAdapter<notif> adapter = new ArrayAdapter <=>(getActivity(), android.R.layout.simple_list_item_1, values);
56
57
```

```
public void onStart() {
    super.onStart();

    dbHandler = new MyDBHandler(getActivity());

    try {
        dbHandler.open();
    } catch (SQLException e) {
        e.printStackTrace();
    }

    values = dbHandler.getAllBarang();

    ArrayAdapter<notif> adapter = new ArrayAdapter <=>(getActivity(), android.R.layout.simple_list_item_1, va

    List = (ListView) getActivity().findViewById(android.R.id.List);
    List.setAdapter(adapter);

    TextView txt = (TextView) getActivity().findViewById(R.id.selamatdatang);
    String username = getArguments().getString( key: "username");
    txt.setText("Selamat Datang " + username);

    ImageView map = (ImageView) getActivity().findViewById(R.id.imgmap);
    map.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(getActivity(), MapActivity.class);
            startActivity(intent);
        }
    });
};
```

Codingan di atas terdapat menu home yang dimana jika user mengakses menu home, user dapat melihat pilihan menu makanan yang diminati oleh orang lain dan lain sebagainya. Di menu home, kami juga membuat untuk admin yang berguna untuk melihat menu yang di pesan oleh user.

### *view pager (cart)*

```
public class Cart extends Fragment {
    private ViewPagerAdapterCart viewPagerAdapter;
    private ViewPager viewPager;
    private TabLayout tabLayout;

    public Cart() {
        // require a empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_cart, container, attachToRoot: false);
    }

    public void onStart() {
        super.onStart();
        viewPager = getActivity().findViewById(R.id.viewpagercart);
        tabLayout = getActivity().findViewById(R.id.tab_layoutcart);

        String username = getArguments().getString(key: "username");

        viewPagerAdapter = new ViewPagerAdapterCart(getChildFragmentManager());
        viewPagerAdapter.setData(username);

        viewPager.setAdapter(viewPagerAdapter);
        tabLayout.setupWithViewPager(viewPager);
    }
}
```

Code di atas merupakan tampilan cart, yang dimana user yang telah memilih makanan yang di order dapat melihat makanan nya lalu user dapat mengedit cart tersebut jika memerlukan notes pada makanan yang ingin dipesan. Dalam codingan, kami juga menambahkan intent *view pager* yang akan digunakan pada halaman pesanan makanan.



## SQLite(pesanan sementara)

```
public class MyDBHandler extends SQLiteOpenHelper {

    private static final int DATABASE_VERSION = 1;
    private static final String DATABASE_NAME = "pesanan.db";
    private static final String TABLE_NAME = "pesanan";

    private static final String COLUMN_ID = "id";
    private static final String COLUMN_USERNAME = "username";
    private static final String COLUMN_MENUNAME = "menuname";

    public MyDBHandler(Context context) { super(context, DATABASE_NAME, null, DATABASE_VERSION); }

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase) {
        String CREATE_TABLE_BARANG = "CREATE TABLE " + TABLE_NAME + "(" +
            COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
            COLUMN_USERNAME + " VARCHAR (50) NOT NULL, " +
            COLUMN_MENUNAME + " VARCHAR(50) NOT NULL)";

        sqLiteDatabase.execSQL(CREATE_TABLE_BARANG);
    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {
        sqLiteDatabase.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(sqLiteDatabase);
    }
}
```

```
@Override
public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {
    sqLiteDatabase.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
    onCreate(sqLiteDatabase);
}

private SQLiteDatabase database;

public void open() throws SQLException {
    database = this.getWritableDatabase();
}

private String[] allColumns =
    {COLUMN_ID, COLUMN_USERNAME, COLUMN_MENUNAME};

private notif cursortonotif(Cursor cursor){
    notif notif = new notif();

    notif.setId((int) cursor.getLong( columnIndex: 0));
    notif.setUsername(cursor.getString( columnIndex: 1));
    notif.setMenuname(cursor.getString( columnIndex: 2));

    return notif;
}

public void createnotif(String nama, String menuname) {
    ContentValues values = new ContentValues();
    values.put(COLUMN_USERNAME, nama);
    values.put(COLUMN_MENUNAME, menuname);
    database.insert(TABLE_NAME, null, values);
}
```

```

public void createnotif(String nama, String menuname) {
    ContentValues values = new ContentValues();
    values.put(COLUMN_USERNAME, nama);
    values.put(COLUMN_MENUNAME, menuname);

    database.insert(TABLE_NAME, null, values);
}

public notif getnotif(long id) {
    notif notif = new notif();

    Cursor cursor =
        database.query(TABLE_NAME, allColumns, selection: "id=" + id, (String[])null, (String)null, (String)null, (String)null);
    cursor.moveToFirst();
    notif notif = cursortonotif( cursor);
    cursor.close();
    return notif;
}

public ArrayList<notif> getAllBarang() {
    ArrayList<notif> daftarBarang = new ArrayList<>();
    Cursor cursor =
        database.query(TABLE_NAME, allColumns, (String) null, (String[]) null, (String) null, (String) null, (String) null);
    cursor.moveToFirst();

    while(!cursor.isAfterLast()) {
        notif notif = cursortonotif(cursor);
        daftarBarang.add(notif);
        cursor.moveToNext();
    }
}

```

```

ArrayList<notif> daftarBarang = new ArrayList<>();
Cursor cursor =
    database.query(TABLE_NAME, allColumns, (String) null, (String[]) null, (String) null, (String) null, (String) null);
cursor.moveToFirst();

while(!cursor.isAfterLast()) {
    notif notif = cursortonotif(cursor);
    daftarBarang.add(notif);
    cursor.moveToNext();
}

cursor.close();
return daftarBarang;
}

public void deletenotif(long id) {
    String filter = "id=" + id;
    database.delete(TABLE_NAME, whereClause: "", (String[])null);
}

public void deletenotifbyname(String menuname) {
    database.delete(TABLE_NAME, whereClause: "", (String[])null);
}
}

```

Selain menggunakan fungsi *web view* dan *view page*, kami juga menambahkan fitur *pesanan sementara* yang menggunakan SQLite. Fungsi ini dapat berperan sebagai pengingat pesanan yang belum di order untuk diingatkan ke user jika user belum membuat orderan. Orderan yang sudah masuk kedalam cart akan ditampung sementara ke dalam database sqlite.

## 2.4.2 Coding PHP

## conn.php

```
1 <?php
2
3 $server = "localhost";
4 $user = "root";
5 $password = "";
6 $database = "hongstar";
7
8 $connect = mysqli_connect($server, $user, $password) or die ("Koneksi gagal!");
9 mysqli_select_db($connect, $database) or die ("Database belum siap!");
10
11 ?>
```

Tampilan di atas merupakan codingan yang berfungsi untuk menghubungkan database yang telah dibuat ke dalam phpmyadmin.

## login.php

```
in2.php
<?php
header('Content-type:application/json;charset=utf-8');
include "conn.php";

$username = 'Cashier';
$password = 'admin123';

$q=mysqli_query($connect,"SELECT * FROM user WHERE username = '$username'");
$response = array();

if(mysqli_num_rows($q) == 1 ){
    $response["data"] = array();
    $r=mysqli_fetch_array($q);
    $user = array();
    $user["user_id"] = $r["user_id"];
    $user["username"] = $r["username"];
    $user["password"] = $r["password"];
    $decrypt = password_verify($password,$r['password']);
    $user["email"] = $r["email"];
    $user["role"] = $r["role"];
    // array_push($response["data"], $user);

    if($r['username'] == $username && $r['password'] == $decrypt){
        $response["success"] = 1;
        $response["message"] = "Data user berhasil dibaca";
        echo json_encode($response);
    }else {
        $response["success"] = 0;
        $response["message"] = "Tidak ada data";
        echo json_encode($response);
    }
}
```

Codingan login.php di atas dapat berguna untuk menghubungkan database pada aplikasi. Jika user telah berhasil melakukan login, maka bisa diketahui bahwa akun yang telah dimasukkan sesuai dengan database.

## addcart.php

```
1 <?php
2 header('Content-type:application/json;charset=utf-8');
3 include "conn.php";
4 $username = $_POST['username'];
5 $type = $_POST['type'];
6 $menu_name = $_POST['menu_name'];
7 $menu_price = $_POST['menu_price'];
8 $total = $_POST['total'];
9
10 $q=mysqli_query($connect,"SELECT * FROM cart WHERE username = '$username' && menu_name = '$menu_name'");
11 $row = mysqli_fetch_assoc($q);
12 $response = array();
13 $jumlah = 0;
14
15 if($row > 0){
16     $q=mysqli_query($connect,"SELECT * FROM cart WHERE username = '$username' && menu_name = '$menu_name'");
17     $r=mysqli_fetch_array($q);
18     $cart = array();
19     $total = $r['total'] + $total;
20
21     $q = mysqli_query($connect, "UPDATE cart SET total=$total WHERE username='$username' && menu_name = '$menu_name'");
22     $response["success"] = 2;
23     $response["message"] = "Data berhasil diupdate";
24     echo json_encode($response);
25 }else {
26
27     $q=mysqli_query($connect,"INSERT INTO cart(username, type, menu_name, menu_price,total)
28     VALUES('$username','$type','$menu_name','$menu_price',$total)");
29     $response = array();
30     if($q){
31         $response["success"] = 1;
32         $response["message"] = "Data berhasil ditambah";
33         echo json_encode($response);
34     }else{
35         $response["success"] = 0;
36         $response["message"] = "Data gagal ditambah";
37         echo json_encode($response);
38     }
39 }
```

Codingan di atas berguna untuk melakukan penambahan jika ada makanan yang ingin ditambahkan. Data tersebut akan masuk ke dalam database sqlite.

## updatefood.php

```
1 <?php
2 header('Content-type:application/json;charset=utf-8');
3 include "conn.php";
4
5 $foodid = $_POST['food_id'];
6 $foodname = $_POST['food_name'];
7 $fooddesc = $_POST['food_desc'];
8 $foodprice = $_POST['food_price'];
9
10
11 $q = mysqli_query($connect, "UPDATE food SET food_name='$foodname', food_price='$foodprice', food_desc='$fooddesc'
12 WHERE food_id='$foodid'");
13 $response = array();
14
15 if($q) {
16     $response["success"] = 1;
17     $response["message"] = "Data berhasil diupdate";
18     echo json_encode($response);
19 } else {
20     $response["success"] = 0;
21     $response["message"] = "Data gagal diupdate";
22     echo json_encode($response);
23 }
24
25 ?>
```

Gambar di atas merupakan codingan *updatefood.php* yang dapat berguna untuk melakukan update data makanan yang akan masuk ke dalam database pada aplikasi. Data yang diupdate, yaitu *food id*, *food name*, *food desc*, dan *food price*.

## **BAB III**

### **PENUTUP**

#### **3.1 Peran Anggota**

Berikut merupakan peran dari masing-masing anggota dalam kelompok Hongstar Fast Food:

Dava Virgio Kertawijaya - 00000056848

Bertugas dalam membuat database aplikasi, menyusun laporan, pembuatan *file* PHP

Michael Owen Kohar - 00000056755

Bertugas menyusun laporan, mendesain UI, membuat ERD, Proses Bisnis, pembuatan file PHP dan membuat database aplikasi

Samuel Andrew - 00000056975

Bertugas menyusun laporan, membuat ERD, dan Proses Bisnis

Nigel Andrian - 00000055946

Bertugas menyusun laporan, mendesain UI, membuat ERD, dan proses bisnis

Westley Wijaya - 00000056414

Bertugas dalam mendesain UI dan membuat database aplikasi, merapikan laporan.

#### **3.2 Kesimpulan**

Hongstar Fast Food adalah aplikasi pemesanan makanan ataupun minuman berbasis *online* yang dapat membantu user dalam membeli makanan di Restoran Hongstar Fast Food tanpa harus keluar dari rumah. Tujuan pembuatan aplikasi tersebut adalah untuk memudahkan *user* dalam memesan makanan tanpa perlu mengantri saat membeli makanan yang diinginkan. Cukup dengan melalui aplikasi tersebut *user* dapat melakukan pembelian berbagai macam makanan yang tersedia di Restoran Hongstar Fast Food.

Dalam aplikasi Hongstar Fast Food terdapat berbagai macam menu yang tersedia, dimulai dari fitur *login* dan *sign up* untuk mendaftarkan *user* dan memvalidasi akun *user* agar bisa masuk ke aplikasi Hongstar Fast Food. Pada *home* menu, *user* dapat melihat rekomendasi menu, pesanan yang sedang berlangsung. Pada *food menu*, *user* dapat mencari dan memesan makanan yang diinginkan serta memiliki kategori yang disediakan pada menu tersebut seperti food and beverages. Pada menu '*Cart*', *user* dapat melihat *riwayat pemesanan* terkait pemesanan yang telah selesai sebelumnya dan juga dapat melihat keranjang pemesanan yang sedang berlangsung. Lalu pada menu *detail pesanan* user dapat melihat pesanan yang mereka pesan dengan lengkap seperti harga, proses pembayaran. Pada *main menu* yang terakhir, yaitu *profile*, *user* dapat melihat data diri pada akunnya serta menu menu informasi lain web Hongstar, edit profile, log out.

Pada aplikasi Hongstar Fast Food, *database* yang digunakan ada 2, yaitu *database* yang berbasis 'phpMyAdmin' dan yang kedua, yaitu *database* lokal yang berasal dari aplikasi Android Studio, yaitu *SQLite Database*.

## **LAMPIRAN**

Video presentasi project:

[https://drive.google.com/file/d/15AmReQIyPEVK6PObtzKPh8kReqEkD-2u/view?usp=share  
link](https://drive.google.com/file/d/15AmReQIyPEVK6PObtzKPh8kReqEkD-2u/view?usp=share_link)