DAFTAR PUSTAKA

Andaru, A. (2018). Pengertian Database Secara Umum. Fakultas Komputer Section Class Content.

Henda, A. (2022, June 14). APA ITU JAVA ?, PENGERTIAN, SEJARAH, DAN BAGAIMANA CARA KERJANYA. https://if.unpas.ac.id/berita/apa-itu-java-pengertian-sejarah-dan-bagaimana-cara-kerjanya/

Hidayat, R., Zainuddin, M., & Selamat, M. B. (2016). Desain sistem informasi perikanan pole and line berbasis online di perairan Teluk Bone. Jurnal IPTEKS PSP, 3(6), 484-500.

Huda, I. (2011). Pengembangan aplikasi P3K berbasis smartphone Android (Skripsi, Universitas Islam Negeri Syarif Hidayatullah Jakarta). Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah Jakarta.

Huda, I. (2014). Implementasi Metode Algoritma Genetika dan Particle Swarm Optimization untuk Optimasi Pengaturan Parameter Kontrol Pada Pemodelan Population Growth. UIN Syarif Hidayatullah Jakarta. Diakses 15 Juni 2024, dari https://repository.uinjkt.ac.id/dspace/bitstream/123456789/3074/1/IMAMUL%20HUDA-FST.pdf

Kurnianti, A., Angguningtyas, & Isnanda, R. G. (2017). Perancangan database pada sistem assessment dan pemetaan hasil assessment berbasis tag sebagai pembantu penyusunan strategi pembelajaran. Jurnal Ilmiah Semesta Teknika, 20(2), 106-115.

Mustarim, Mustarim, et al. "Front Termal dan Pengaruhnya Terhadap Kelimpahan Ikan Pelagis di Perairan Selat Malaka." Jurnal Ilmu Kelautan 19.1 (2020): 1-10.

Nurhidayati, & Nur, A. M. (2021). Pemanfaatan Aplikasi Android Dalam Rancang Bangun Sistem Informasi Persebaran Indekos Di Wilayah Pancor Kabupaten Lombok Timur, 4.

Rahmawati, P. R. (2017). Rancang Bangun Aplikasi Travel Online Berbasis Android. Skripsi. Jurusan Teknik Informatika Fakultas Teknologi Informasi. Institut Teknologi Sepuluh Nopember. Surabaya.

Rahmayani, I. (2015, October 2). Indonesia Raksasa Teknologi Digital Asia. Retrieved October 12, 2023.

Safitri, L., & Basuki, S. (2020). ANALISA DAN PERANCANGAN SISTEM INFORMASI TEXT CHATTING BERBASIS ANDROID WEB VIEW, 8.

Sukatmi, & Pitri, E. S. (2018). APLIKASI ABSENSI SISWA BERBASIS WEB DENGAN DUKUNGAN SMS GATEWAY PADA SMK KRIDAWISATA BANDAR LAMPUNG, 6.

Sukmawati, Bustari, & Nofrizal. (2021). PRODUKTIVITAS DAN KELAYAKAN USAHA PENANGKAPAN DENGAN ALAT TANGKAP BAGAN PERAHU DI PELABUHAN PERIKANAN SAMUDERA BUNGUS SUMATERA BARAT. https://jom.unri.ac.id

VhMuzini. (2023, May 31). Google Maps Icons. GitHub. https://gist.github.com/VhMuzini/eae2c8b4324b368c1308dcf66ed4d3b4

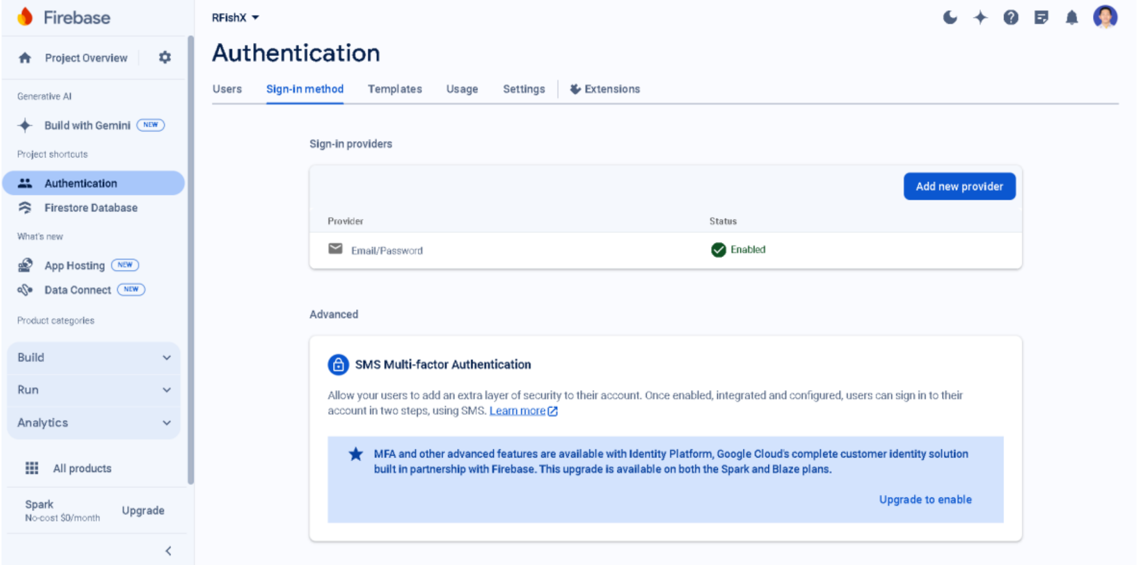
Widodo, P. A. (2017). Transformasi dokumen XML: Sebuah studi kasus pada aplikasi konversi buku elektronik. Jurnal Teknologi Informasi dan Komputer (JTIK), 5(2), 117-129.

LAMPIRAN

Lampiran 1. *Tahapan Pembuatan Database*

Beberapa hal yang harus disiapkan dan harus dilakukan dalam untuk menjadikan *database* sebagai alat validasi pengguna, sebagai berikut :

1. Siapkan akun *Google*
2. Daftarkan akun tersebut kedalam ”https://console.firebase.google.com”
3. Klik New Project, lalu masukkan nama projek sesuai dengan nama aplikasi/projek yang anda buat.
4. Klik next, lalu beralih ke halaman tab ”*Authentication*”.
5. Pada bagian menu “Authentication”, klik menu “Sign-in Method”. Lalu tambahkan email sebagai validasi.



Provider	Status
Email/Password	Enabled

SMS Multi-factor Authentication

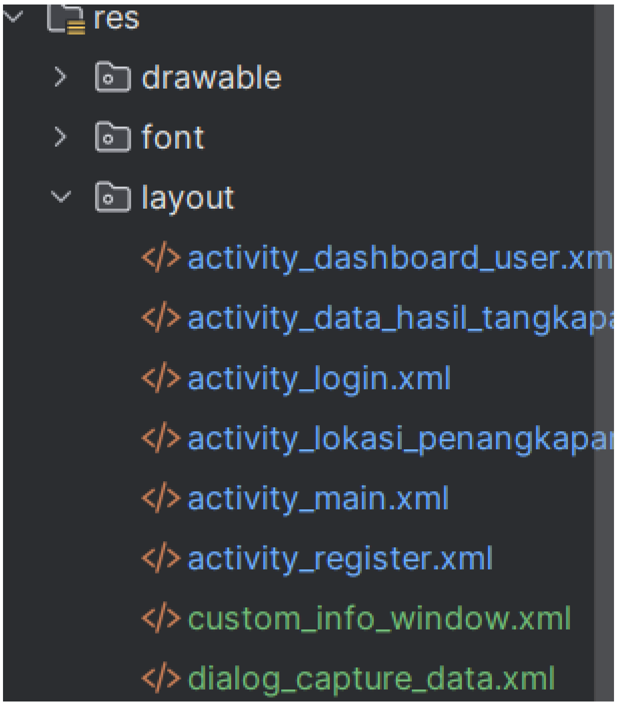
Allow your users to add an extra layer of security to their account. Once enabled, integrated and configured, users can sign in to their account in two steps, using SMS. Learn more

MFA and other advanced features are available with Identity Platform, Google Cloud's complete customer identity solution built in partnership with Firebase. This upgrade is available on both the Spark and Blaze plans.

Lampiran 2. *Tahapan Pembuatan Desain Aplikasi*

Dalam pembuatan desain aplikasi hal-hal yang harus disiapkan yaitu :

1. Buat Projek baru di *Android Studio*
2. Berikan nama aplikasi yang ingin dibuat
3. Pilih minimal Versi *Android* yang dibutuhkan.
4. Klik Next, Kemudian biarkan gradle selesai memuat.
5. Setelah selesai, Buka bagian *Layout* di *directory Res.*
6. Masukkan Kode sesuai dengan yang anda inginkan.



```
res
    drawable
    font
    layout
        activity_dashboard_user.xm
        activity_data_hasil_tangkapa
        activity_login.xml
        activity_lokasi_penangkapa
        activity_main.xml
        activity_register.xml
        custom_info_window.xml
        dialog_capture_data.xml
```

Lampiran 3. *Source Code* pada *XML Layout Main Activity*

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:orientation="vertical"
    android:gravity="center"
    android:background="@drawable/background_gradient">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="348dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.069"
        app:srcCompat="@drawable/ic_rfish" />

    <TextView
        android:id="@+id/welcomeTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Selamat datang di RFISH"
        android:fontFamily="@font/montserrat"
        android:textSize="24sp"
        android:textStyle="bold"
        android:textColor="#FFFFFF"
        android:layout_marginBottom="8dp"/>

    <TextView
        android:id="@+id/descriptionTextView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="RFISH adalah aplikasi inovatif yang membantu
para nelayan dan pemancing untuk menemukan titik-titik lokasi
penangkapan ikan yang optimal. Kami menyediakan prediksi untuk
memastikan Anda mendapatkan hasil tangkapan terbaik."
        android:textAlignment="center"
        android:textColor="#CCCCCC"
        android:fontFamily="@font/montserrat"
        android:layout_marginBottom="32dp"
        android:paddingStart="16dp"
        android:textStyle="bold"
        android:paddingEnd="16dp"/>

    <Button
        android:id="@+id/btn_login"
        android:layout_width="300dp"
        android:layout_height="78dp"
        android:layout_marginBottom="15dp"
        android:background="@drawable/rounded_btn"
        android:text="LOGIN"
        android:textStyle="bold"
        style="@style/Button"
        android:fontFamily="@font/montserrat"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView"
        app:layout_constraintVertical_bias="0.0" />

    <Button
        android:id="@+id/btn_register"
        android:layout_width="300dp"
        android:layout_height="78dp"
        android:background="@drawable/rounded_btn"
        android:text="REGISTER"
        android:textStyle="bold"
        style="@style/Button"
        android:fontFamily="@font/montserrat"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        android:layout_marginBottom="90dp"
        app:layout_constraintTop_toBottomOf="@+id/btn_login" />
</LinearLayout>
```

Lampiran 4. *Source Code* pada *XML Layout Login Activity*

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    tools:context=".LoginActivity"
    android:background="@drawable/background_gradient2">

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="30dp"
        app:cardCornerRadius="30dp"
        app:cardElevation="20dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:orientation="vertical"
            android:padding="24dp">

            <TextView
                android:id="@+id/loginText"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="LOGIN"
                android:textAlignment="center"
                android:textColor="@color/bluelight"
                android:textSize="36sp"
                android:textStyle="bold" />

            <EditText
                android:id="@+id/email1"
                android:layout_width="match_parent"
                android:layout_height="50dp"
                android:layout_marginTop="40dp"
                android:background="@drawable/custom_edittext"
                android:drawableLeft="@drawable/ic_email"
                android:drawablePadding="8dp"
                android:hint="Email"
                android:fontFamily="@font/montserrat"
                android:padding="8dp"
                android:textColor="@color/black"

android:textColorHighlight="@color/cardview_dark_background" />

            <com.google.android.material.textfield.TextInputLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="15dp"
                app:endIconMode="password_toggle"
                app:hintEnabled="false"> <!-- Disable floating label
-->


<com.google.android.material.textfield.TextInputEditText
                    android:id="@+id/password"
                    android:layout_width="match_parent"
                    android:layout_height="50dp"
                    android:hint="Password"
                    android:background="@drawable/custom_edittext"
                    android:drawableLeft="@drawable/ic_lock"
                    android:drawablePadding="8dp"
                    android:fontFamily="@font/montserrat"
                    android:inputType="textPassword"
                    android:padding="8dp"
                    android:textColor="@color/black"

android:textColorHighlight="@color/cardview_dark_background" />
            </com.google.android.material.textfield.TextInputLayout>

            <Button
                android:id="@+id/loginbutton"
                android:layout_width="match_parent"
                android:layout_height="60dp"
                android:layout_marginTop="30dp"
                android:backgroundTint="@color/bluedark"
                android:textStyle="bold"
                android:background="@drawable/rounded_btn"
                android:textColor="@color/white"
                android:fontFamily="@font/montserrat"
                android:text="LOGIN"
                android:textSize="18sp"
                app:cornerRadius="20dp" />

            <TextView
                android:id="@+id/signupText"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="20dp"
                android:layout_marginBottom="20dp"
                android:fontFamily="@font/montserrat"
                android:padding="8dp"
                android:text="@string/login_subtitle"
                android:textAlignment="center"
                android:textColor="@color/black"
                android:textSize="14sp" />
        </LinearLayout>
    </androidx.cardview.widget.CardView>
</LinearLayout>
```

Lampiran 5. *Source Code* pada *XML Layout Register Activity*

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    tools:context=".RegisterActivity"
    android:background="@drawable/background_gradient2">

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="30dp"
        app:cardCornerRadius="30dp"
        app:cardElevation="20dp"
        android:background="@drawable/custom_edittext">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:padding="24dp">

            <TextView
                android:id="@+id/registerText"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="@string/create_account"
                android:textAlignment="center"
                android:textColor="@color/bluelight"
                android:textSize="36sp"
                android:textStyle="bold"
                android:layout_marginBottom="20dp"/>

            <EditText
                android:id="@+id/email"
                android:layout_width="match_parent"
                android:layout_height="50dp"
                android:layout_marginTop="20dp"
                android:layout_marginBottom="10dp"
                android:background="@drawable/custom_edittext"
                android:drawableLeft="@drawable/ic_email"
                android:drawablePadding="8dp"
                android:fontFamily="@font/montserrat"
                android:hint="@string/email"
                android:inputType="textEmailAddress"
                android:padding="16dp"
                android:textColor="@color/black"

android:textColorHighlight="@color/cardview_dark_background" />

            <com.google.android.material.textfield.TextInputLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="10dp"
                app:endIconMode="password_toggle"
                app:hintEnabled="false">


<com.google.android.material.textfield.TextInputEditText
                    android:id="@+id/Password"
                    android:layout_width="match_parent"
                    android:layout_height="50dp"
                    android:layout_marginTop="20dp"
                    android:background="@drawable/custom_edittext"
                    android:drawableLeft="@drawable/ic_lock"
                    android:drawablePadding="8dp"
                    android:hint="@string/password"
                    android:fontFamily="@font/montserrat"
                    android:inputType="textPassword"
                    android:padding="16dp"
                    android:textColor="@color/black"

android:textColorHighlight="@color/cardview_dark_background" />
            </com.google.android.material.textfield.TextInputLayout>

            <com.google.android.material.textfield.TextInputLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="10dp"
                app:endIconMode="password_toggle"
                app:hintEnabled="false">


<com.google.android.material.textfield.TextInputEditText
                    android:id="@+id/Password2"
                    android:layout_width="match_parent"
                    android:layout_height="50dp"
                    android:layout_marginTop="20dp"
                    android:background="@drawable/custom_edittext"
                    android:drawableLeft="@drawable/ic_lock"
                    android:drawablePadding="8dp"
                    android:hint="@string/confirm_password"
                    android:fontFamily="@font/montserrat"
                    android:inputType="textPassword"
                    android:padding="16dp"
                    android:textColor="@color/black"

android:textColorHighlight="@color/cardview_dark_background" />
            </com.google.android.material.textfield.TextInputLayout>

            <Button
                android:id="@+id/btn_register2"
                android:layout_width="match_parent"
                android:layout_height="60dp"
                android:layout_marginTop="30dp"
                android:backgroundTint="@color/bluedark"
                android:textStyle="bold"
                android:fontFamily="@font/montserrat"
                style="@style/Button"
                android:text="@string/register"
                android:textColor="@color/white"
                android:textSize="18sp"
                app:cornerRadius="35dp" />

            <TextView
                android:id="@+id/signupText"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="20dp"
                android:layout_marginBottom="20dp"
                android:fontFamily="@font/montserrat"
                android:padding="8dp"
                android:text="@string/SignUpSubtitle"
                android:textAlignment="center"
                android:textColor="@color/black"
                android:textSize="14sp" />
        </LinearLayout>
    </androidx.cardview.widget.CardView>
</LinearLayout>
```

Lampiran 6. *Source Code* pada *XML Layout DashboardActivity*

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:background="@drawable/background_gradient2"
    tools:context=".DashboardUser">

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="16dp"
        android:padding="16dp"
        app:cardCornerRadius="16dp"
        app:cardElevation="8dp"
        app:cardBackgroundColor="@android:color/white">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:gravity="center_vertical"
            android:padding="16dp">

            <ImageView
                android:id="@+id/weatherImageView"
                android:layout_width="50dp"
                android:layout_height="50dp"
                android:layout_marginEnd="16dp"
              android:contentDescription="@string/weather_icon"
/>

            <LinearLayout
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:orientation="vertical">

                <TextView
                    android:id="@+id/weatherTextView"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Loading weather..."
                    android:textSize="18sp"
                    android:textColor="@android:color/black"
                    android:fontFamily="@font/montserrat"
                    android:gravity="left" />

                <TextView
                    android:id="@+id/welcomeTextView"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Loading..."
                    android:textSize="18sp"
                    android:textColor="@android:color/black"
                    android:fontFamily="@font/montserrat"
                    android:textStyle="bold"
                    android:gravity="left" />

                <TextView
                    android:id="@+id/timeTextView"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Time"
                    android:textSize="18sp"
                    android:textColor="@android:color/black"
                    android:fontFamily="@font/montserrat"
                    android:textStyle="italic"
                    android:gravity="left" />
            </LinearLayout>
        </LinearLayout>
    </androidx.cardview.widget.CardView>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:orientation="vertical"
        android:padding="24dp">

        <Button
            android:id="@+id/btn_LokasiPenangkapan"
            android:layout_width="match_parent"
            android:layout_height="60dp"
            android:layout_marginTop="30dp"
            android:backgroundTint="@color/bluedark"
            android:textStyle="bold"
            android:textColor="@android:color/white"
            android:fontFamily="@font/montserrat"
            android:text="@string/lokasi_penangkapan"
            android:textSize="18sp"
            style="@style/Button"
            app:cornerRadius="20dp"
            android:gravity="center" />

        <Button
            android:id="@+id/btn_DataHasilTangkapan"
            android:layout_width="match_parent"
            android:layout_height="60dp"
            android:layout_marginTop="30dp"
            android:backgroundTint="@color/bluedark"
            android:textStyle="bold"
            android:textColor="@android:color/white"
            android:fontFamily="@font/montserrat"
            android:text="@string/data_hasil_tangkapan"
            android:textSize="18sp"
            style="@style/Button"
            app:cornerRadius="20dp"
            android:gravity="center" />

        <TextView
            android:id="@+id/versionTextView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Version 1.1"
            android:textSize="14sp"
```

```
            android:textStyle="bold"
            android:textColor="@android:color/white"
            android:layout_gravity="center_horizontal"
            android:layout_marginTop="25dp"
            android:fontFamily="@font/montserrat" />

        <TextView
            android:id="@+id/otherInfoTextView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="© 2024 RFISH"
            android:textSize="14sp"
            android:textStyle="bold"
            android:textColor="@android:color/white"
            android:layout_gravity="center_horizontal"
            android:fontFamily="@font/montserrat"
            android:layout_marginTop="4dp" />
    </LinearLayout>
</LinearLayout>
```

Lampiran 7. *Source Code* pada *XML Layout* LokasiPenangkapan

XML yang saya gunakan pada *Layout* LokasiPenangkapan berbeda dengan *XML* lainnya, dikarenakan saya menggunakan *Layout ”SupportMapFragment”* untuk memunculkan tampilan *Google Maps*.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.fragment.app.FragmentContainerView
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LokasiPenangkapan"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".LokasiPenangkapan"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    />
```

Lampiran 8. *Source Code* pada *XML Layout* DataHasilTangkapan

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@drawable/bg_home_wave"
    android:padding="16dp">

    <EditText
        android:id="@+id/fishingbaseEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Titik Fishing Base" />

    <EditText
        android:id="@+id/shipNameEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Nama Kapal" />

    <EditText
        android:id="@+id/fishNameEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Nama Ikan" />

    <EditText
        android:id="@+id/captureDateEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Tanggal Penangkapan" />

    <EditText
        android:id="@+id/fishWeightEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Berat Ikan" />

    <EditText
        android:id="@+id/captureLocationEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Titik Penangkapan" />

    <EditText
        android:id="@+id/jarakEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Jarak" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:padding="16dp">
        <Button
            android:id="@+id/saveButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Save"
            android:layout_marginTop="10dp"
            android:layout_marginLeft="75dp"
            android:textStyle="bold"
            style="@style/Button"/>

        <Button
            android:id="@+id/loadButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="20dp"
            android:text="Load"
            android:textStyle="bold"
            android:layout_marginTop="10dp"
            style="@style/Button"/>

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:padding="16dp">

        <Button
            android:id="@+id/lokasifishingbase"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="FISHING BASE"
            android:layout_marginRight="15dp"
            style="@style/Button"/>

        <Button
            android:id="@+id/lokasifishingground"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="FISHING GROUND"
            style="@style/Button"/>

    </LinearLayout>
    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:layout_marginTop="15dp">
        <TextView
            android:id="@+id/displayTextView"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textStyle="bold"
            android:text="Captured data will appear here" />
    </ScrollView>
</LinearLayout>
```

Lampiran 9. *Source Code* pada *Class MainActivity*

```
package com.example.rfishx;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;


public class MainActivity extends AppCompatActivity {

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

        Button btn_login = findViewById(R.id.btn_login);
        Button btn_register = findViewById(R.id.btn_register);

        btn_login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, 
LoginActivity.class);
                startActivity(intent);

        btn_register.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, 
RegisterActivity.class);
                startActivity(intent);
            }
        });
            }
        }
        )
        ;
    }
}
```

Lampiran 10. *Source Code* pada *Class RegisterActivity*

```
package com.example.rfishx;

import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Patterns;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.android.material.textfield.TextInputEditText;

public class RegisterActivity extends AppCompatActivity {
    Button btn_register2;
    EditText et_email;
    TextInputEditText et_password, et_confirmPassword;
    private FirebaseAuth mAuth;
    private DatabaseReference mDatabase;

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

        mAuth = FirebaseAuth.getInstance();
        mDatabase =
FirebaseDatabase.getInstance().getReference("Users");

        btn_register2 = findViewById(R.id.btn_register2);
        et_email = findViewById(R.id.email);
        et_password = findViewById(R.id.Password);
        et_confirmPassword = findViewById(R.id.Password2);

        btn_register2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (validateInput()) {
                    registerUser();
                }
            }
        });
    }

    private boolean validateInput() {
        String email = et_email.getText().toString().trim();
        String password = et_password.getText().toString().trim();
        String confirmPassword =
et_confirmPassword.getText().toString().trim();

        if (TextUtils.isEmpty(email) ||
!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
            et_email.setError("Masukkan Email yang Valid");
            return false;
        }

        if (TextUtils.isEmpty(password)) {
            et_password.setError("Masukkan Password");
            return false;
        }

        if (password.length() < 6) {
            et_password.setError("Password harus terdiri dari 6
karakter");
            return false;
        }

        if (!password.equals(confirmPassword)) {
            et_confirmPassword.setError("Password tidak sesuai");
            return false;
        }

        return true;
    }

    private void registerUser() {
        String email = et_email.getText().toString().trim();
        String password = et_password.getText().toString().trim();

        mAuth.createUserWithEmailAndPassword(email,
password).addOnCompleteListener(this, new
OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful()) {
                    // Registration successful, send email
verification
                    FirebaseUser user = mAuth.getCurrentUser();
                    if (user != null) {

user.sendEmailVerification().addOnCompleteListener(new
OnCompleteListener<Void>() {
                            @Override
                            public void onComplete(@NonNull
Task<Void> task) {
                                if (task.isSuccessful()) {

Toast.makeText(RegisterActivity.this, "Registrasi Berhasil.
Verifikasi email telah dikirim.", Toast.LENGTH_SHORT).show();
                                    // Save user info to the database
                                    String userId = user.getUid();
                                    User userInfo = new User(email);

mDatabase.child(userId).setValue(userInfo).addOnCompleteListener(new
OnCompleteListener<Void>() {
@Override
public void onComplete(@NonNull Task<Void> task) {
if (task.isSuccessful()) { // Redirect to login activity
Intent intent = new Intent(RegisterActivity.this,
LoginActivity.class);startActivity(intent);
finish(); // Finish current activity (registration)} else {
Toast.makeText(RegisterActivity.this, "Gagal Membuat Akun",
Toast.LENGTH_SHORT).show();
                                            }
                                        }
                                    });
                                } else {

Toast.makeText(RegisterActivity.this, "Gagal mengirim email
verifikasi.", Toast.LENGTH_SHORT).show();
                                }
                            }
                        });
                    }
                } else {
                    Toast.makeText(RegisterActivity.this, "Registrasi
Gagal: " + task.getException().getMessage(),
Toast.LENGTH_SHORT).show();
                }
            }
        });
    }

    // User class to create user objects
    public static class User {
        public String email;

        public User() {
            // Default constructor required for calls to
DataSnapshot.getValue(User.class)
        }

        public User(String email) {
            this.email = email;
        }
    }
}
```

Lampiran 11. *Source Code* pada *Class LoginActivity*

```
package com.example.rfishx;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class LoginActivity extends AppCompatActivity {

    private FirebaseAuth mAuth;
    private EditText emailEditText;
    private EditText passwordEditText;

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

        // Initialize Firebase Authentication
        mAuth = FirebaseAuth.getInstance();

        emailEditText = findViewById(R.id.email1);
        passwordEditText = findViewById(R.id.password);
        Button loginButton = findViewById(R.id.loginbutton);
        TextView signupText = findViewById(R.id.signupText);


        // Handle login button click
        loginButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String email = emailEditText.getText().toString();
                String password =
passwordEditText.getText().toString();

                if (!email.isEmpty() && !password.isEmpty()) {
                    signInWithEmailAndPassword(email, password);
                } else {
                    Toast.makeText(LoginActivity.this, “Isi bidang
yang masih kosong”, Toast.LENGTH_SHORT).show();
                }
            }
        });

        // Handle sign up text click
        signupText.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                // Navigate to sign up page
                // Implement navigation logic according to your app’s
requirements
            }
        });
    }
    private void signInWithEmailAndPassword(String email, String
password) {
        mAuth.signInWithEmailAndPassword(email, password)
                .addOnCompleteListener(this, new
OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult>
task) {
                        if (task.isSuccessful()) {
                            // Sign in success, update UI with the
signed-in user’s information
                            FirebaseUser user =
mAuth.getCurrentUser();
                            String username = user.getDisplayName();
// Ambil nama pengguna dari Firebase

saveUsernameToSharedPreferences(username); // Simpan nama pengguna ke
SharedPreferences
                            Toast.makeText(LoginActivity.this, “Login
Berhasil.”,
                                    Toast.LENGTH_SHORT).show();
                            Intent intent = new
Intent(LoginActivity.this, DashboardUser.class);
                            startActivity(intent);
                            finish();
                        } else {
                            // If sign in fails, display a message to
the user.
                            Toast.makeText(LoginActivity.this, “Login
Gagal.”,
                                    Toast.LENGTH_SHORT).show();
                        }
                    }
                });
    }
    private void saveUsernameToSharedPreferences(String username) {
        SharedPreferences sharedPreferences =
getSharedPreferences(“MyPrefs”, MODE_PRIVATE);
        SharedPreferences.Editor editor = sharedPreferences.edit();
        editor.putString(“username”, username);
        editor.apply();
    }
}
```

Lampiran 12. *Source Code* pada *Class Dashboard*

```
package com.example.rfishx;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.os.AsyncTask;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;
import android.util.Log;
import android.view.Gravity;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationCallback;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationResult;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.tasks.OnSuccessListener;

import org.json.JSONArray;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.HashMap;
import java.util.List;
import java.util.Locale;
import java.util.Map;
import java.util.TimeZone;

public class DashboardUser extends AppCompatActivity {

    private static final String TAG = "DashboardUser";

    private Button btn_LokasiPenangkapan;
    private Button btn_DataHasilTangkapan;
    private TextView welcomeTextView;
    private TextView timeTextView;
    private TextView weatherTextView;
    private ImageView weatherImageView;
    private Handler handler;
    private Runnable runnable;

    private FusedLocationProviderClient fusedLocationClient;
    private LocationCallback locationCallback;
    private boolean isCityNameSet = false;
    private String currentCityName = "";

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

        // Initialize views
        btn_LokasiPenangkapan = 
findViewById(R.id.btn_LokasiPenangkapan);
        btn_DataHasilTangkapan = 
findViewById(R.id.btn_DataHasilTangkapan);
        welcomeTextView = findViewById(R.id.welcomeTextView);
        timeTextView = findViewById(R.id.timeTextView);
        weatherTextView = findViewById(R.id.weatherTextView);
        weatherImageView = findViewById(R.id.weatherImageView);

        // Set text alignment to left
        weatherTextView.setGravity(Gravity.LEFT);
        welcomeTextView.setGravity(Gravity.LEFT);
        timeTextView.setGravity(Gravity.LEFT);

        // Initialize FusedLocationProviderClient
        fusedLocationClient = 
LocationServices.getFusedLocationProviderClient(this);

        // Check permissions
        checkPermissions();

        // Set welcome message
        setWelcomeMessage();

        // Button click listeners
        btn_LokasiPenangkapan.setOnClickListener(new 
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(DashboardUser.this, 
LokasiPenangkapan.class);
                startActivity(i);
            }
        });
        btn_DataHasilTangkapan.setOnClickListener(new 
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(DashboardUser.this, 
DataHasilTangkapan.class);
                startActivity(i);
            }
        });

        // Initialize handler and runnable for updating time
        handler = new Handler();
        runnable = new Runnable() {
            @Override
            public void run() {
                updateCurrentTime();
                handler.postDelayed(this, 1000);
            }
        };
        handler.post(runnable);

        // Initialize LocationCallback
        locationCallback = new LocationCallback() {
            @Override
            public void onLocationResult(LocationResult
locationResult) {
                if (locationResult == null) {
                    return;
                }
                for (Location location :
locationResult.getLocations()) {
                    if (location != null) {
                        double latitude = location.getLatitude();
                        double longitude = location.getLongitude();
                        if (!isCityNameSet) {
                            fetchCityName(latitude, longitude);
                        }
                        fetchWeatherData(latitude, longitude);
                    }
                }
            }
        };

        // Start location updates
        startLocationUpdates();
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        handler.removeCallbacks(runnable);
        if (fusedLocationClient != null && locationCallback != null)
{

fusedLocationClient.removeLocationUpdates(locationCallback);
        }
    }

    private void checkPermissions() {
        String[] permissions = {
                Manifest.permission.ACCESS_FINE_LOCATION,
                Manifest.permission.ACCESS_COARSE_LOCATION
        };
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M &&

(checkSelfPermission(Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED ||

checkSelfPermission(Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED)) {
            requestPermissions(permissions, 1234);
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        if (requestCode == 1234) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                startLocationUpdates();
            }
        }
    }

    private void setWelcomeMessage() {
        String username = getIntent().getStringExtra("username");
        if (username != null && !username.isEmpty()) {
            String message = "Selamat Datang, " + username;
            welcomeTextView.setText(message);
        } else {
            welcomeTextView.setText("Selamat datang!");
        }
    }

    private void updateCurrentTime() {
        SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss",
Locale.getDefault());
        sdf.setTimeZone(TimeZone.getDefault());
        String currentTime = sdf.format(new Date());
        timeTextView.setText(currentTime);
    }

    private void startLocationUpdates() {
        LocationRequest locationRequest = LocationRequest.create();
        locationRequest.setInterval(10000); // 10 seconds
        locationRequest.setFastestInterval(5000); // 5 seconds

locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);

        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
                ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            return;
        }
        fusedLocationClient.requestLocationUpdates(locationRequest,
locationCallback, Looper.getMainLooper());
    }

    private void fetchWeatherData(double latitude, double longitude)
{
        String apiKey = "aa89acdf1a722dba61c381ab3bdfcade";
        String urlString =
"https://api.openweathermap.org/data/2.5/weather?lat=" + latitude +
"&lon=" + longitude + "&appid=" + apiKey + "&units=metric";
```

```
        new FetchWeatherTask().execute(urlString);
    }

    private class FetchWeatherTask extends AsyncTask<String, Void,
String> {
        @Override
        protected String doInBackground(String... params) {
            StringBuilder result = new StringBuilder();
            try {
                URL url = new URL(params[0]);
                HttpURLConnection conn = (HttpURLConnection)
url.openConnection();
                BufferedReader reader = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
                String line;
                while ((line = reader.readLine()) != null) {
                    result.append(line);
                }
                reader.close();
            } catch (Exception e) {
                e.printStackTrace();
            }
            return result.toString();
        }

        @Override
        protected void onPostExecute(String result) {
            try {
                JSONObject jsonObject = new JSONObject(result);
                JSONObject main = jsonObject.getJSONObject("main");
                JSONArray weatherArray =
jsonObject.getJSONArray("weather");
                JSONObject weatherObject =
weatherArray.getJSONObject(0);
                double temperature = main.getDouble("temp");
                String description =
weatherObject.getString("description");

                String translatedDescription =
translateWeatherDescription(description);

                if (!currentCityName.isEmpty()) {
                    weatherTextView.setText(currentCityName + "\n" +
translatedDescription + "\nTemperature: " + temperature + "°C");
                    int weatherIconResId =
getWeatherIconResource(description);

weatherImageView.setImageResource(weatherIconResId);
                    weatherTextView.setGravity(Gravity.LEFT);
                }
            } catch (Exception e) {
                e.printStackTrace();
                weatherTextView.setText("Failed to load weather
data");
                weatherTextView.setGravity(Gravity.LEFT);
            }
        }
    }
    private int getWeatherIconResource(String description) {
        Map<String, Integer> weatherIconMap = new HashMap<>();
        weatherIconMap.put("clear sky", R.drawable.ic_clear_sky);
        weatherIconMap.put("few clouds", R.drawable.ic_few_clouds);
        weatherIconMap.put("shower rain", R.drawable.ic_shower_rain);
        weatherIconMap.put("rain", R.drawable.ic_rain);
        weatherIconMap.put("thunderstorm",
R.drawable.ic_thunderstorm);
        weatherIconMap.put("snow", R.drawable.ic_snow);
        weatherIconMap.put("mist", R.drawable.ic_mist);

        Integer resourceId = weatherIconMap.get(description);
        return resourceId != null ? resourceId :
R.drawable.ic_unknown; // default icon
    }

    private void fetchCityName(double latitude, double longitude) {
        Geocoder geocoder = new Geocoder(this, Locale.getDefault());
        try {
            List<Address> addresses =
geocoder.getFromLocation(latitude, longitude, 1);
            if (addresses != null && !addresses.isEmpty()) {
                currentCityName = addresses.get(0).getLocality();
                isCityNameSet = true;
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private String translateWeatherDescription(String description) {
        Map<String, String> translationMap = new HashMap<>();
        translationMap.put("clear sky", "Cerah");
        translationMap.put("few clouds", "Berawan");
        translationMap.put("shower rain", "Hujan Deras");
        translationMap.put("rain", "Hujan Ringan");
        translationMap.put("thunderstorm", "Badai Petir");
        translationMap.put("snow", "Salju");
        translationMap.put("mist", "Kabut");

        return translationMap.getOrDefault(description, description);
    }
}
```

Lampiran 13. *Source Code* pada *Class* LokasiPenangkapan

```
package com.example.rfishx;

import android.os.Bundle;
import androidx.fragment.app.FragmentActivity;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.CircleOptions;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

public class LokasiPenangkapan extends FragmentActivity implements
OnMapReadyCallback {
    private GoogleMap mMap;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_lokasi_penangkapan);
        SupportMapFragment mapFragment = (SupportMapFragment)
                getSupportFragmentManager()
                        .findFragmentById(R.id.LokasiPenangkapan);
        if (mapFragment != null) {
            mapFragment.getMapAsync(this);
        } else {
            mapFragment = new SupportMapFragment();
            getSupportFragmentManager().beginTransaction()
                    .replace(R.id.LokasiPenangkapan,
mapFragment).commit();
            mapFragment.getMapAsync(this);
        }
    }

    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;

        mMap.setInfoWindowAdapter(new CustomInfoWindowAdapter(this));

        LatLng location1 = new LatLng(-4.4129857, 119.5366135);
        LatLng location2 = new LatLng(-5.100526, 119.300611);
        LatLng location3 = new LatLng(-2.097516,116.941838);
        LatLng location4 = new LatLng( -4.971904,117.027499);
        LatLng location5 = new LatLng(-2.107033,116.951356);
        LatLng location6 = new LatLng(-5.761886,119.654424);
        LatLng location7 = new LatLng(-4.952869,116.741963);
        LatLng location8 = new LatLng(-2.107033,116.941838);
        LatLng location9 = new LatLng(-5.999832,119.492620);
        LatLng location10 = new LatLng( -3.905906,116.637267);

        MarkerOptions options1 = new MarkerOptions()
                .position(location1)
                .title("Lokasi 1")
                .snippet("Informasi tambahan untuk Lokasi 1");
        MarkerOptions options2 = new MarkerOptions()
                .position(location2)
                .title("Lokasi 2")
                .snippet("Informasi tambahan untuk Lokasi 2");
        MarkerOptions options3 = new MarkerOptions()
                .position(location3)
                .title("Lokasi 3")
                .snippet("Informasi tambahan untuk Lokasi 3");
        MarkerOptions options4 = new MarkerOptions()
                .position(location4)
                .title("Lokasi 4")
                .snippet("Informasi tambahan untuk Lokasi 4");
        MarkerOptions options5 = new MarkerOptions()
                .position(location5)
                .title("Lokasi 5")
                .snippet("Informasi tambahan untuk Lokasi 5");
        MarkerOptions options6 = new MarkerOptions()
                .position(location6)
                .title("Lokasi 6")
                .snippet("Informasi tambahan untuk Lokasi 6");
        MarkerOptions options7 = new MarkerOptions()
                .position(location7)
                .title("Lokasi 7")
                .snippet("Informasi tambahan untuk Lokasi 7");
        MarkerOptions options8 = new MarkerOptions()
                .position(location8)
                .title("Lokasi 8")
                .snippet("Informasi tambahan untuk Lokasi 8");
        MarkerOptions options9 = new MarkerOptions()
                .position(location9)
                .title("Lokasi 9")
                .snippet("Informasi tambahan untuk Lokasi 9");
        MarkerOptions options10 = new MarkerOptions()
                .position(location10)
                .title("Lokasi 10")
                .snippet("Informasi tambahan untuk Lokasi 10");

        mMap.addMarker(options1);
        mMap.addMarker(options2);
        mMap.addMarker(options3);
        mMap.addMarker(options4);
        mMap.addMarker(options5);
        mMap.addMarker(options6);
        mMap.addMarker(options7);
        mMap.addMarker(options8);
        mMap.addMarker(options9);
        mMap.addMarker(options10);

        mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(location1,
10));
        mMap.getUiSettings().setZoomControlsEnabled(true);
        mMap.getUiSettings().setCompassEnabled(true);

        // Radius Penangkapan
        mMap.addCircle(new CircleOptions()
                .center(location1)
                .radius(500) // radius dalam meter
                .strokeColor(0xFF0000FF) // warna garis tepi (biru)
                .fillColor(0x220000FF) // warna isian dengan
transparansi (biru)
                .strokeWidth(5)); // lebar garis tepi
        mMap.addCircle(new CircleOptions()
                .center(location2)
                .radius(500) // radius dalam meter
                .strokeColor(0xFF0000FF) // warna garis tepi (biru)
                .fillColor(0x220000FF) // warna isian dengan
transparansi (biru)
                .strokeWidth(5)); // lebar garis tepi
    }
}
```

Lampiran 14. *Source Code* pada *Class* DataHasilTangkapan

```
package com.example.rfishx;

import android.Manifest;
import android.content.pm.PackageManager;
import android.location.Location;
import android.os.Bundle;
import android.util.Log;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.firestore.QuerySnapshot;

import java.util.List;

public class DataHasilTangkapan extends AppCompatActivity {
    private static final int REQUEST_LOCATION_PERMISSION = 1;
    private static final String TAG = "DataHasilTangkapan";

    private FirebaseAuth mAuth;
    private CaptureRepository captureRepository;
    private EditText fishingbaseEditText;
    private EditText shipNameEditText;
    private EditText fishNameEditText;
    private EditText captureDateEditText;
    private EditText fishWeightEditText;
    private EditText captureLocationEditText;
    private EditText jarakEditText;
    private TextView displayTextView;
    private FusedLocationProviderClient fusedLocationClient;

    private Location fishingBaseLocation;
    private Location fishingGroundLocation;

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

        mAuth = FirebaseAuth.getInstance();
        captureRepository = new CaptureRepository();
        fishingbaseEditText = findViewById(R.id.fishingbaseEditText);
        shipNameEditText = findViewById(R.id.shipNameEditText);
        fishNameEditText = findViewById(R.id.fishNameEditText);
        captureDateEditText = findViewById(R.id.captureDateEditText);
        fishWeightEditText = findViewById(R.id.fishWeightEditText);
        captureLocationEditText =
findViewById(R.id.captureLocationEditText);
        jarakEditText = findViewById(R.id.jarakEditText);
        displayTextView = findViewById(R.id.displayTextView);

        fusedLocationClient =
LocationServices.getFusedLocationProviderClient(this);

        Button saveButton = findViewById(R.id.saveButton);
        saveButton.setOnClickListener(v -> saveCaptureData());

        Button loadButton = findViewById(R.id.loadButton);
        loadButton.setOnClickListener(v -> loadCaptureData());

        Button lokasifishingbaseButton =
findViewById(R.id.lokasifishingbase);
        lokasifishingbaseButton.setOnClickListener(v ->
checkLocationPermission(true));

        Button lokasifishinggroundButton =
findViewById(R.id.lokasifishingground);
        lokasifishinggroundButton.setOnClickListener(v ->
checkLocationPermission(false));
    }

    private void checkLocationPermission(boolean isFishingBase) {
        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
                ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION,
Manifest.permission.ACCESS_COARSE_LOCATION},
REQUEST_LOCATION_PERMISSION);
        } else {
            if (isFishingBase) {
                setFishingBaseLocation();
            } else {
                setFishingGroundLocation();
            }
        }
    }

    private void saveCaptureData() {
        FirebaseUser currentUser = mAuth.getCurrentUser();
        if (currentUser != null) {
            String userId = currentUser.getUid();
            String shipName =
shipNameEditText.getText().toString().trim();
            String fishName =
fishNameEditText.getText().toString().trim();
            String captureDate =
captureDateEditText.getText().toString().trim();
            String fishWeightStr =
fishWeightEditText.getText().toString().trim();
            String captureLocation =
captureLocationEditText.getText().toString().trim();
            String fishingBase =
fishingbaseEditText.getText().toString().trim();
            String jarakStr =
jarakEditText.getText().toString().trim();

            if (shipName.isEmpty() || fishName.isEmpty() ||
captureDate.isEmpty() || fishWeightStr.isEmpty() ||
captureLocation.isEmpty() || fishingBase.isEmpty() ||
jarakStr.isEmpty()) {
                Toast.makeText(this, "Please fill in all fields",
Toast.LENGTH_SHORT).show();
                return;
            }

            double fishWeight = Double.parseDouble(fishWeightStr);
            double jarak = Double.parseDouble(jarakStr);

            // Parse the fishing base and fishing ground coordinates
            double fishingBaseLatitude = 0.0;
            double fishingBaseLongitude = 0.0;
            double fishingGroundLatitude = 0.0;
            double fishingGroundLongitude = 0.0;

            try {
                String[] fishingBaseCoords = fishingBase.split(", ");
                fishingBaseLatitude =
Double.parseDouble(fishingBaseCoords[0]);
                fishingBaseLongitude =
Double.parseDouble(fishingBaseCoords[1]);

                String[] fishingGroundCoords =
captureLocation.split(", ");
                fishingGroundLatitude =
Double.parseDouble(fishingGroundCoords[0]);
                fishingGroundLongitude =
Double.parseDouble(fishingGroundCoords[1]);
            } catch (Exception e) {
                Toast.makeText(this, "Invalid coordinate format.
Please use 'latitude, longitude'", Toast.LENGTH_SHORT).show();
                return;
            }

            Capture capture = new Capture(userId, shipName, fishName,
captureDate, fishWeight, captureLocation, fishingBase, jarak,
                    fishingBaseLatitude, fishingBaseLongitude,
fishingGroundLatitude, fishingGroundLongitude);

            captureRepository.addCapture(capture, task -> {
                if (task.isSuccessful()) {
                    Toast.makeText(DataHasilTangkapan.this, "Data
saved", Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(DataHasilTangkapan.this, "Error
saving data", Toast.LENGTH_SHORT).show();
                }
            });
        } else {
            Toast.makeText(this, "User not logged in",
Toast.LENGTH_SHORT).show();
        }
    }


    private void loadCaptureData() {
        FirebaseUser currentUser = mAuth.getCurrentUser();
        if (currentUser != null) {

captureRepository.getCapturesByUserId(currentUser.getUid(), task -> {
                if (task.isSuccessful()) {
                    StringBuilder displayText = new StringBuilder();
                    List<Capture> captureList =
task.getResult().toObjects(Capture.class);
                    for (Capture capture : captureList) {
                        displayText.append("Ship Name:
").append(capture.getShipName()).append("\n")
                                .append("Fish Name:
").append(capture.getFishName()).append("\n")
                                .append("Capture Date:
").append(capture.getCaptureDate()).append("\n")
                                .append("Fish Weight:
").append(capture.getFishWeight()).append("\n")
                                .append("Capture Location:
").append(capture.getCaptureLocation()).append("\n")
                                .append("Fishing Base:
").append(capture.getFishingBase()).append("\n")
                                .append("Distance:
").append(capture.getJarak()).append("\n")
                                .append("\n");
                    }
                    displayTextView.setText(displayText.toString());
                } else {
                    Toast.makeText(this, "Error getting data",
Toast.LENGTH_SHORT).show();
                }
            });
        } else {
            Toast.makeText(this, "User not logged in",
Toast.LENGTH_SHORT).show();
        }
    }

    private void setFishingBaseLocation() {
        getCurrentLocation(task -> {
            if (task.isSuccessful() && task.getResult() != null) {
                Location location = task.getResult();
                fishingBaseLocation = location;
                String fishingBase = location.getLatitude() + ", " +
location.getLongitude();
                fishingbaseEditText.setText(fishingBase);
            } else {
                Log.e(TAG, "Unable to get current location");
                Toast.makeText(this, "Unable to get current
location", Toast.LENGTH_SHORT).show();
            }
        });
    }
    private void setFishingGroundLocation() {
        getCurrentLocation(task -> {
            if (task.isSuccessful() && task.getResult() != null) {
                Location location = task.getResult();
                fishingGroundLocation = location;
                String fishingGround = location.getLatitude() + ", "
+ location.getLongitude();
                captureLocationEditText.setText(fishingGround);
                calculateAndSetDistance();
            } else {
                Log.e(TAG, "Unable to get current location");
                Toast.makeText(this, "Unable to get current
location", Toast.LENGTH_SHORT).show();
            }
        });
    }

    private void calculateAndSetDistance() {
        if (fishingBaseLocation != null && fishingGroundLocation !=
null) {
            double distance =
calculateDistance(fishingBaseLocation.getLatitude(),
fishingBaseLocation.getLongitude(),
                    fishingGroundLocation.getLatitude(),
fishingGroundLocation.getLongitude());
            jarakEditText.setText(String.format("%.2f", distance));
        } else {
            Toast.makeText(this, "Unable to calculate distance.
Ensure both locations are set.", Toast.LENGTH_SHORT).show();
        }
    }

    private void getCurrentLocation(OnCompleteListener<Location>
onCompleteListener) {
        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
                ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION,
Manifest.permission.ACCESS_COARSE_LOCATION},
REQUEST_LOCATION_PERMISSION);
            return;
        }

        LocationRequest locationRequest = LocationRequest.create();

locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);


fusedLocationClient.getCurrentLocation(locationRequest.getPriority(),
null)
                .addOnCompleteListener(onCompleteListener)
                .addOnFailureListener(e -> {
                    Log.e(TAG, "Failed to get location: " +
e.getMessage());
                    Toast.makeText(this, "Failed to get location: " +
e.getMessage(), Toast.LENGTH_SHORT).show();
                });
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        if (requestCode == REQUEST_LOCATION_PERMISSION) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                Toast.makeText(this, "Location permission granted",
Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(this, "Location permission denied",
Toast.LENGTH_SHORT).show();
            }
        }
    }

    private double calculateDistance(double lat1, double lon1, double
lat2, double lon2) {
        final int R = 6371; // Radius of the earth in kilometers
        double latDistance = Math.toRadians(lat2 - lat1);
        double lonDistance = Math.toRadians(lon2 - lon1);
        double a = Math.sin(latDistance / 2) * Math.sin(latDistance /
2)
                + Math.cos(Math.toRadians(lat1)) *
Math.cos(Math.toRadians(lat2))
                * Math.sin(lonDistance / 2) * Math.sin(lonDistance /
2);
        double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
        double distance = R * c;
        return distance;
    }
}
```

Lampiran 15. *Source Code* pada *Class Capture*

```
package com.example.rfishx;

public class Capture {
    private String userId;
    private String shipName;
    private String fishName;
    private String captureDate;
    private double fishWeight;
    private String captureLocation;
    private String fishingBase;
    private double jarak;
    private double fishingBaseLatitude;
    private double fishingBaseLongitude;
    private double fishingGroundLatitude;
    private double fishingGroundLongitude;

    public Capture() {
        // Diperlukan untuk Firestore
    }

    public Capture(String userId, String shipName, String fishName,
String captureDate, double fishWeight, String captureLocation,
                   String fishingBase, double jarak, double
fishingBaseLatitude, double fishingBaseLongitude,
                   double fishingGroundLatitude, double
fishingGroundLongitude) {
        this.userId = userId;
        this.shipName = shipName;
        this.fishName = fishName;
        this.captureDate = captureDate;
        this.fishWeight = fishWeight;
        this.captureLocation = captureLocation;
        this.fishingBase = fishingBase;
        this.jarak = jarak;
        this.fishingBaseLatitude = fishingBaseLatitude;
        this.fishingBaseLongitude = fishingBaseLongitude;
        this.fishingGroundLatitude = fishingGroundLatitude;
        this.fishingGroundLongitude = fishingGroundLongitude;
    }

    // Getter dan Setter untuk semua atribut
    public String getUserId() { return userId; }
    public void setUserId(String userId) { this.userId = userId; }

    public String getShipName() { return shipName; }
    public void setShipName(String shipName) { this.shipName =
shipName; }

    public String getFishName() { return fishName; }
    public void setFishName(String fishName) { this.fishName =
fishName; }

    public String getCaptureDate() { return captureDate; }
    public void setCaptureDate(String captureDate) { this.captureDate
= captureDate; }

    public double getFishWeight() { return fishWeight; }
    public void setFishWeight(double fishWeight) { this.fishWeight =
fishWeight; }

    public String getCaptureLocation() { return captureLocation; }
    public void setCaptureLocation(String captureLocation) {
this.captureLocation = captureLocation; }

    public String getFishingBase() { return fishingBase; }
    public void setFishingBase(String fishingBase) { this.fishingBase
= fishingBase; }

    public double getJarak() { return jarak; }
    public void setJarak(double jarak) { this.jarak = jarak; }

    public double getFishingBaseLatitude() { return
fishingBaseLatitude; }
    public void setFishingBaseLatitude(double fishingBaseLatitude) {
this.fishingBaseLatitude = fishingBaseLatitude; }

    public double getFishingBaseLongitude() { return
fishingBaseLongitude; }
    public void setFishingBaseLongitude(double fishingBaseLongitude)
{ this.fishingBaseLongitude = fishingBaseLongitude; }

    public double getFishingGroundLatitude() { return
fishingGroundLatitude; }
    public void setFishingGroundLatitude(double
fishingGroundLatitude) { this.fishingGroundLatitude =
fishingGroundLatitude; }

    public double getFishingGroundLongitude() { return
fishingGroundLongitude; }
    public void setFishingGroundLongitude(double
fishingGroundLongitude) { this.fishingGroundLongitude =
fishingGroundLongitude; }
}
```

Lampiran 16. *Source Code* pada *Class CaptureRepository*

```
package com.example.rfishx;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.firestore.CollectionReference;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.QuerySnapshot;

public class CaptureRepository {
    private FirebaseFirestore db;
    private CollectionReference capturesRef;

    public CaptureRepository() {
        db = FirebaseFirestore.getInstance();
        capturesRef = db.collection("captures");
    }

    public void addCapture(Capture capture,
OnCompleteListener<DocumentReference> onCompleteListener) {

capturesRef.add(capture).addOnCompleteListener(onCompleteListener);
    }

    public void getCapturesByUserId(String userId,
OnCompleteListener<QuerySnapshot> onCompleteListener) {
        capturesRef.whereEqualTo("userId",
userId).get().addOnCompleteListener(onCompleteListener);
    }
}
```