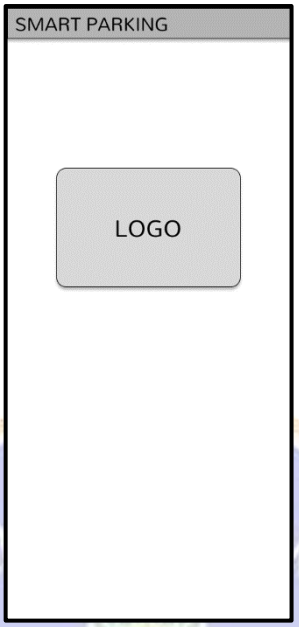
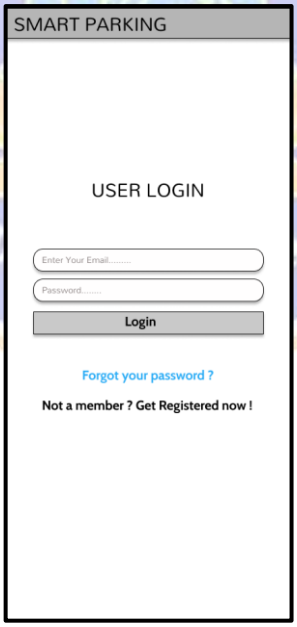
LAMPIRAN

Lampiran 1. Rancangan *Interface* Aplikasi Dan Website *Smart Parking*

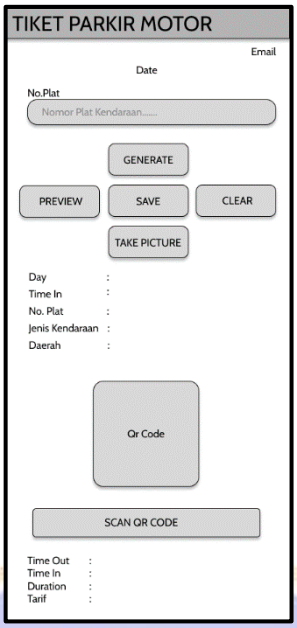
1. Gambar Rancangan Halaman *Splash Screen* pada Aplikasi *Smart Parking*



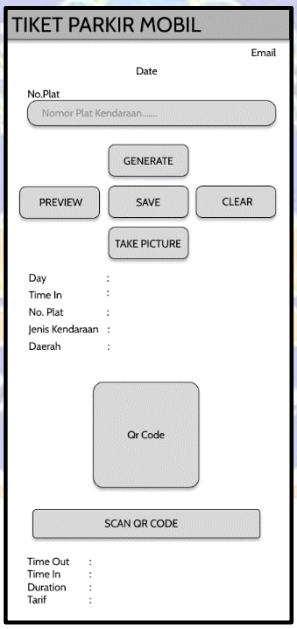
2. Gambar Rancangan Halaman Login pada Aplikasi *Smart Parking*



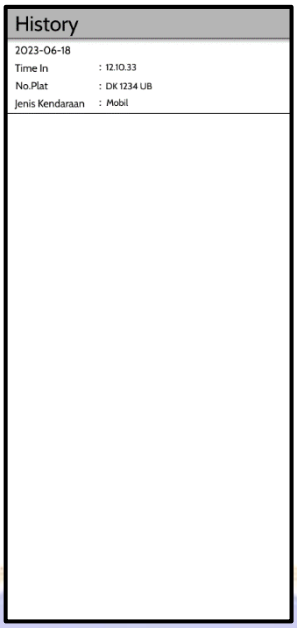
3. Gambar Rancangan Halaman Parkir Motor pada Aplikasi *Smart Parking*



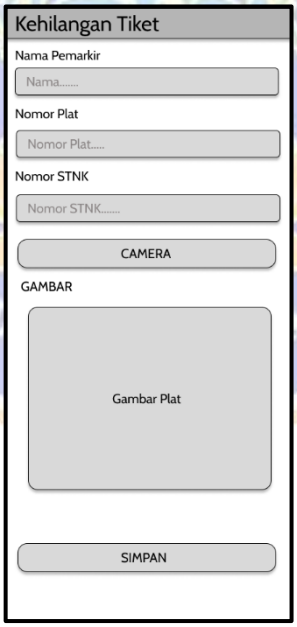
4. Gambar Rancangan Halaman Parkir Mobil pada Aplikasi *Smart Parking*



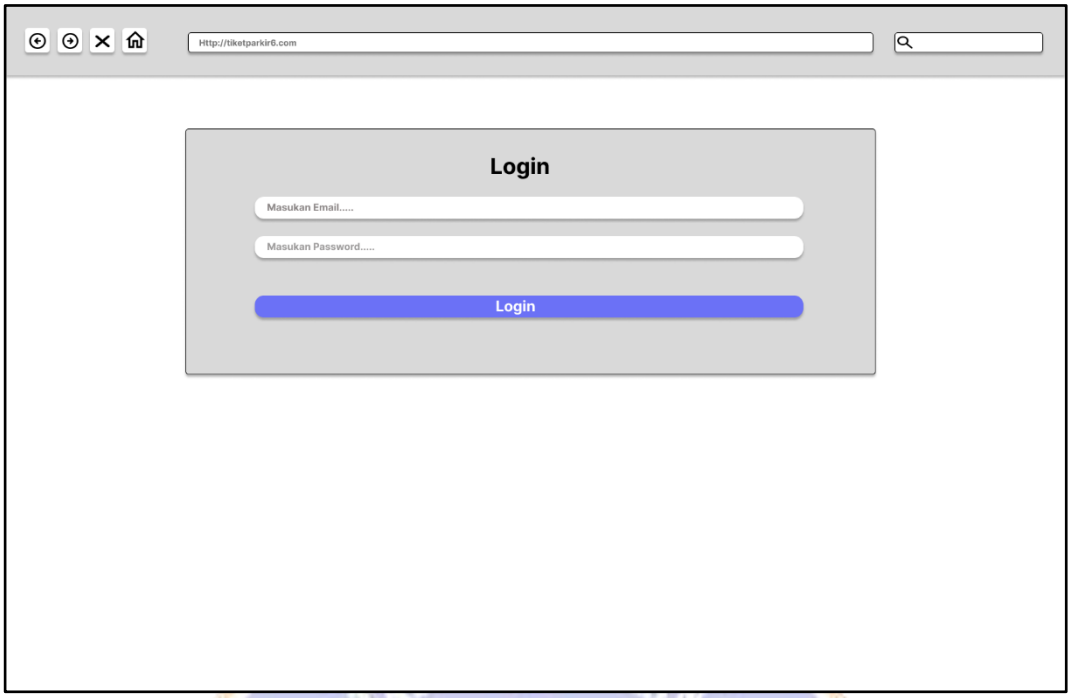
5. Gambar Rancangan Halaman *History* pada Aplikasi *Smart Parking*



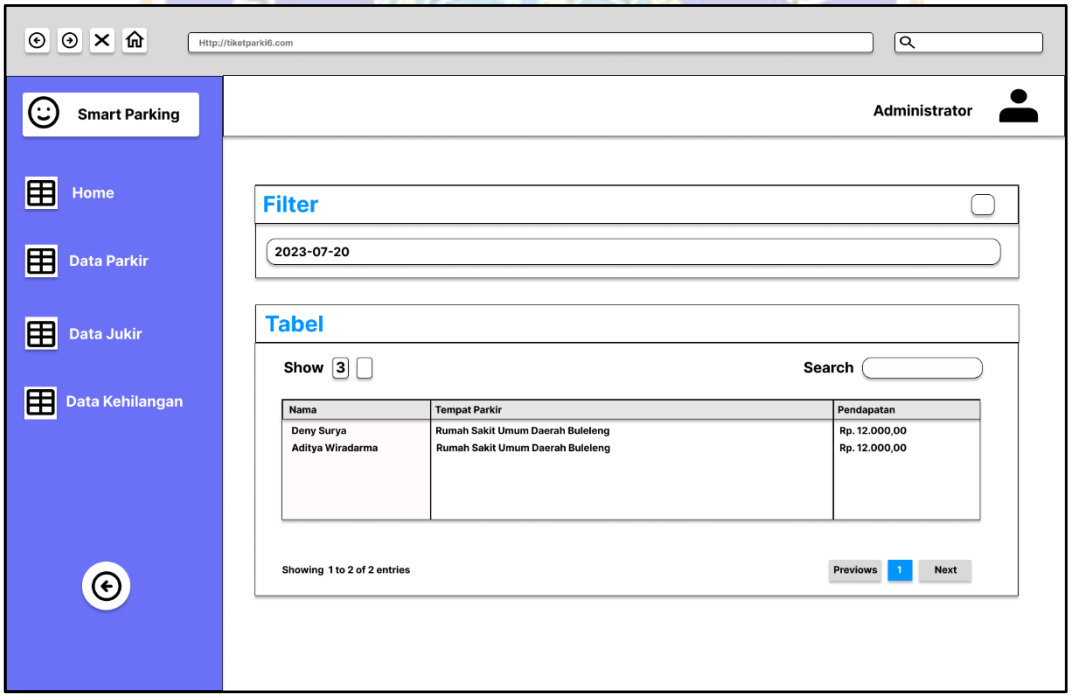
6. Gambar Rancangan Halaman Kehilangan Tiket pada Aplikasi *Smart Parking*



7. Gambar Rancangan Halaman Login pada Website *Smart Parking*



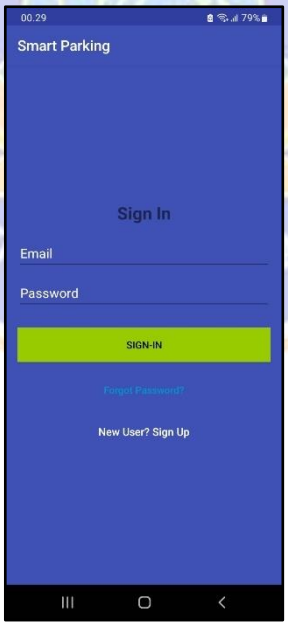
8. Gambar Rancangan Halaman Data Parkir pada Website *Smart Parking*



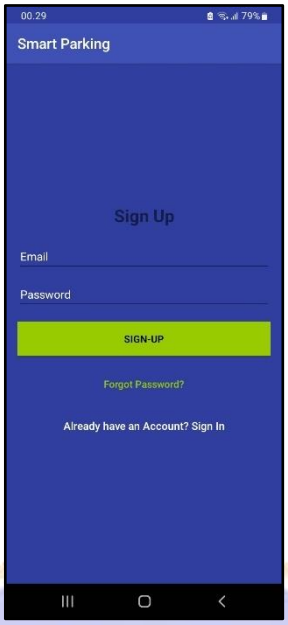
Lampiran 2. Hasil Implementasi Aplikasi Dan Website *Smart Parking*

1. Gambar Tampilan Halaman *Splash Screen* pada Aplikasi *Smart Parking*

2. Gambar Tampilan Halaman Login pada Aplikasi *Smart Parking*



3. Gambar Tampilan Halaman *Sign Up* pada Aplikasi *Smart Parking*

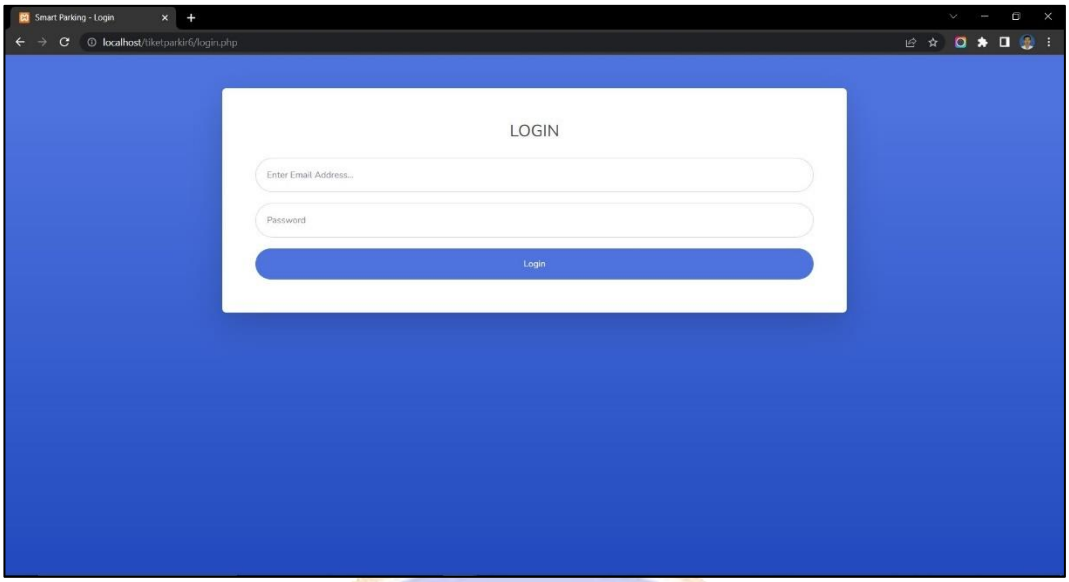


4. Gambar Tampilan Halaman *Forgot Password* pada Aplikasi *Smart Parking*

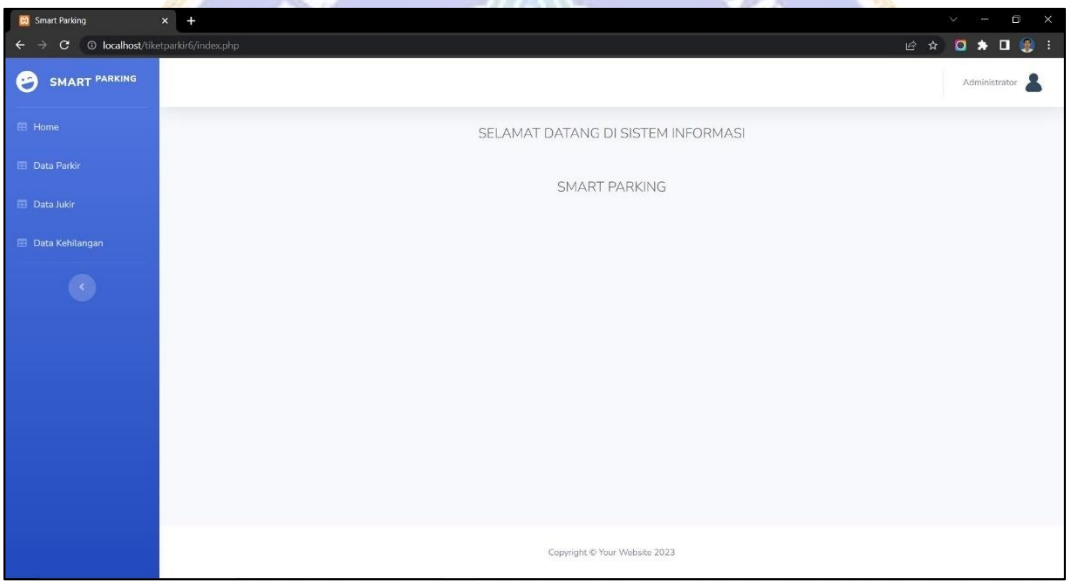
5. Gambar Tampilan Halaman *History* pada Aplikasi *Smart Parking*

6. Gambar Tampilan Halaman Kehilangan Tiket pada Aplikasi *Smart Parking*

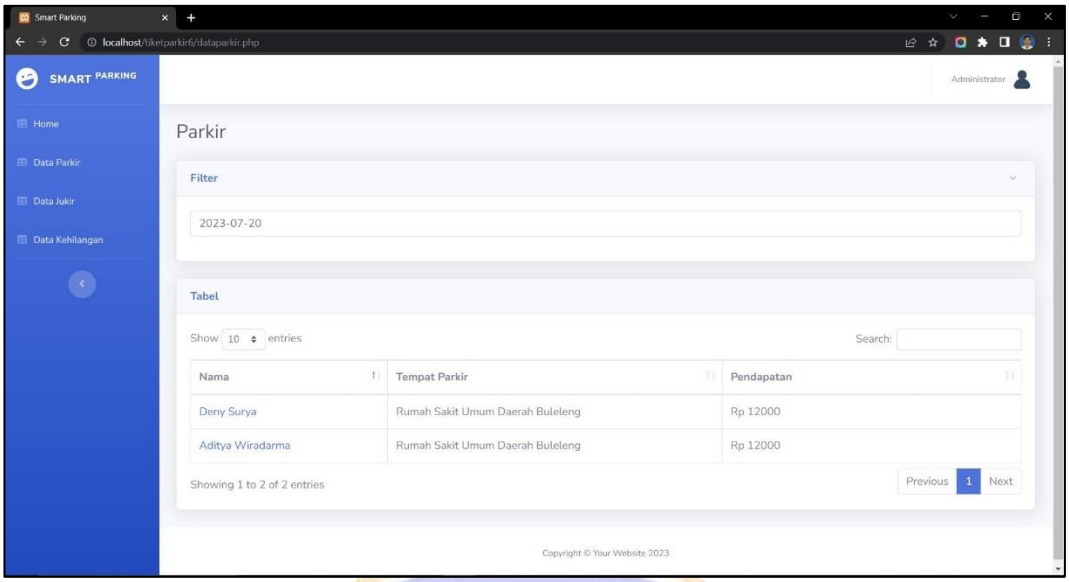
7. Gambar Tampilan Halaman Login pada Website *Smart Parking*



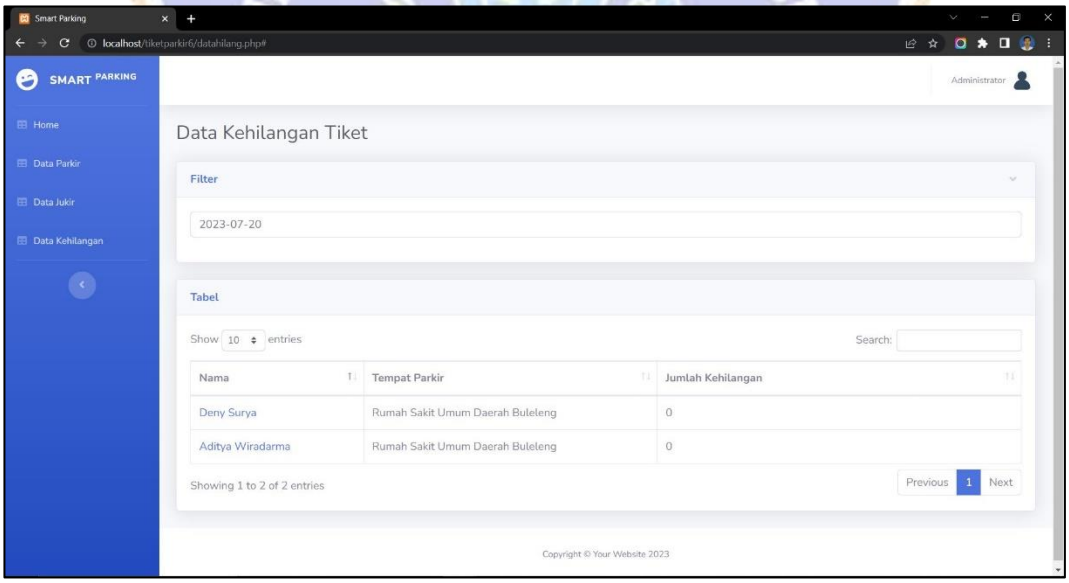
8. Gambar Tampilan Halaman Home pada Website Smart Parking



9. Gambar Tampilan Halaman Data Parkir pada Website Smart Parking



10. Gambar Tampilan Halaman Kehilangan Tiket Parkir pada Website *Smart Parking*



Lampiran 3. *Source Code* Aplikasi *Smart Parking*

1. Tabel *Source Code* Halaman Login pada Aplikasi *Smart Parking*

```
package com.jtd.adit.tiketparkir;

import android.content.Intent;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;

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;

public class LoginActivity extends AppCompatActivity {

    private EditText inputEmail, inputPassword;
    private FirebaseAuth auth;
    private ProgressBar progressBar;
    private Button btnSignup, btnLogin, btnReset;

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

        //Get Firebase auth instance
        auth = FirebaseAuth.getInstance();

        if (auth.getCurrentUser() != null) {
            startActivity(new Intent(LoginActivity.this,
HomeActivity.class));
            finish();
        }

        // set the view now
        setContentView(R.layout.activity_login);


        inputEmail = (EditText) findViewById(R.id.email);
        inputPassword = (EditText)
findViewById(R.id.password);
        progressBar = (ProgressBar)
findViewById(R.id.progressBar);
        btnSignup = (Button)
```

```
findViewById(R.id.btn_signup);
        btnLogin = (Button) findViewById(R.id.btn_login);
        btnReset = (Button)
findViewById(R.id.btn_reset_password);

        //Get Firebase auth instance
        auth = FirebaseAuth.getInstance();

        btnSignup.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new
Intent(LoginActivity.this, SignUpActivity.class));
            }
        });

        btnReset.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new
Intent(LoginActivity.this, ResetPasswordActivity.class));
            }
        });

        btnLogin.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String email =
inputEmail.getText().toString();
                final String password =
inputPassword.getText().toString();

                if (TextUtils.isEmpty(email)) {

Toast.makeText(getApplicationContext(), "Email",
Toast.LENGTH_SHORT).show();
                    return;
                }

                if (TextUtils.isEmpty(password)) {

Toast.makeText(getApplicationContext(), "Password",
Toast.LENGTH_SHORT).show();
                    return;
                }

                progressBar.setVisibility(View.VISIBLE);

                //authenticate user
                auth.signInWithEmailAndPassword(email,
```

```
password)

.addOnCompleteListener(LoginActivity.this, new
OnCompleteListener<AuthResult>() {
                                @Override
                                public void
onComplete(@NonNull Task<AuthResult> task) {
                                    // If sign in fails,
display a message to the user. If sign in succeeds
                                    // the auth state listener
will be notified and logic to handle the
                                    // signed in user can be
handled in the listener.

progressBar.setVisibility(View.GONE);
                                    if (!task.isSuccessful())
{
                                        // there was an error
                                        if (password.length()
< 6) {

inputPassword.setError("Password too short, enter minimum
6 characters");
                                        } else {

Toast.makeText(LoginActivity.this, "Authentication failed,
check your email and password", Toast.LENGTH_LONG).show();
                                        }
                                    } else {
                                        Intent intent = new
Intent(LoginActivity.this, HomeActivity.class);
                                        startActivity(intent);
                                        finish();
                                    }
                                }
                            });
                        }
                    });
    }
}
```

2. Tabel *Source Code* Halaman Menu pada Aplikasi *Smart Parking*

```
package com.jtd.adit.tiketparkir;

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

```
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class HomeActivity extends AppCompatActivity {

    private Button parkir_mtr, parkir_mbl, history,
kehilangan, signOut;
    private FirebaseAuth.AuthStateListener authListener;
    private FirebaseAuth auth;

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

        //get firebase auth instance
        auth = FirebaseAuth.getInstance();

        //get current user
        final FirebaseUser user =
FirebaseAuth.getInstance().getCurrentUser();

        authListener = new
FirebaseAuth.AuthStateListener() {
            @Override
            public void onAuthStateChanged(@NonNull
FirebaseAuth firebaseAuth) {
                FirebaseUser user =
firebaseAuth.getCurrentUser();
                if (user == null) {
                    // user auth state is changed - user
is null
                    // launch login activity
                    startActivity(new
Intent(HomeActivity.this, LoginActivity.class));
                    finish();
                }
            }
        };

        parkir_mtr = findViewById(R.id.parkir_mtr);
        parkir_mbl = findViewById(R.id.parkir_mbl);
        history = findViewById(R.id.history);
        kehilangan = findViewById(R.id.hilang);
        signOut = findViewById(R.id.sign_out);


        parkir_mtr.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new
Intent(HomeActivity.this, MotorActivity.class));
```

```
            }
        });

        parkir_mbl.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new
Intent(HomeActivity.this, MobilActivity.class));
            }
        });

        history.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new
Intent(HomeActivity.this, HistoryActivity.class));
            }
        });

        kehilangan.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new
Intent(HomeActivity.this, KehilanganActivity.class));
            }
        });

        signOut.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                auth.signOut();
                finish();
                startActivity(new
Intent(HomeActivity.this, MainActivity.class));
            }
        });

    }
}
```

3. Tabel *Source Code* Halaman Parkir Motor pada Aplikasi *Smart Parking*

```
package com.jtd.adit.tiketparkir;

import android.annotation.SuppressLint;
import android.app.Activity;
import android.app.AlertDialog;
import android.bluetooth.BluetoothAdapter;
```

```
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.ClipData;
import android.content.ClipboardManager;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.drawable.BitmapDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.os.Handler;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
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.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.EventListener;
import com.google.firebase.firestore.FirebaseFirestore;
import
com.google.firebase.firestore.FirebaseFirestoreException;
import
com.google.firebase.firestore.QueryDocumentSnapshot;
import com.google.firebase.firestore.QuerySnapshot;
import com.google.zxing.BarcodeFormat;
import com.google.zxing.MultiFormatWriter;
import com.google.zxing.WriterException;
import com.google.zxing.common.BitMatrix;
import
com.google.zxing.integration.android.IntentIntegrator;
import com.google.zxing.integration.android.IntentResult;
import com.journeyapps.barcodescanner.BarcodeEncoder;
import com.jtd.adit.tiketparkir.camera.CameraActivity;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.HashMap;
```

```
import java.util.Map;
import java.util.Set;
import java.util.Timer;
import java.util.TimerTask;
import java.util.UUID;


public class MotorActivity extends AppCompatActivity {

    private static final String TAG = "MainActivity";
    private static final String KEY_DAY = "day";
    private static final String KEY_TIME = "timeIn";
    private static final String KEY_PLAT = "noPlat";
    private static final String KEY_JENIS =
"jenisKendaraan";
    private static final String KEY_TIMEOUT = "timeOut";
    private static final String KEY_TARIF = "tarif";
    private static final String KEY_STATUS = "status";
    public String statusp;

    private FirebaseFirestore db =
FirebaseFirestore.getInstance();

    private EditText plat, plat2;
    private TextView timein;
    private TextView TVStatus;
    private TextView noplat;
    private TextView label1;
    private TextView label2;
    private TextView label3;
    private TextView label4;
    private TextView label5;
    private TextView label6;
    private TextView hasil, hasil2, hasil3, hasil4;
    private ImageView qrcode;
    private TextView hari, nama;
    private Button generate;
    private Button proses;
    private Button scan;
    private Button hapus;
    private Button btnTakePicture;

    private FirebaseAuth.AuthStateListener authListener;
    private FirebaseAuth auth;


    //printer
    BluetoothAdapter bluetoothAdapter;
    BluetoothSocket bluetoothSocket;
    BluetoothDevice bluetoothDevice;

    OutputStream outputStream;
    InputStream inputStream;
```

```
    Thread thread;

    byte[] readBuffer;
    int readBufferPosition;
    volatile boolean stopWorker;

    TextView lblPrinterName;

    DatabaseHelper mDatabaseHelper;
    private int taskId;

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

        getSupportActionBar().setTitle("Parkir Motor");

        //Get Firebase auth instance
        auth = FirebaseAuth.getInstance();

        //get current user
        final FirebaseUser user =
FirebaseAuth.getInstance().getCurrentUser();

        String email = user.getEmail().toString();

        nama = (TextView) findViewById(R.id.nama);

        nama.setText(email);


        TVStatus = (TextView) findViewById(R.id.TVStatus);
        plat = (EditText) findViewById(R.id.plat);
        plat2 = (EditText) findViewById(R.id.plat2);
        timein = (TextView) findViewById(R.id.txt4);
        noplat = (TextView) findViewById(R.id.txt5);
        label1 = (TextView) findViewById(R.id.txt1);
        label2 = (TextView) findViewById(R.id.txt2);
        qrcode = (ImageView) findViewById(R.id.qrcode);
        generate = (Button) findViewById(R.id.generate);
        proses = (Button) findViewById(R.id.proses);
        label3 = (TextView) findViewById(R.id.note);
        label4 = (TextView) findViewById(R.id.txt6);
        label5 = (TextView) findViewById(R.id.note2);
        label6 = (TextView) findViewById(R.id.note3);
        scan = (Button) findViewById(R.id.btnScan);
        hapus = (Button) findViewById(R.id.hapus);
        btnTakePicture = (Button)
findViewById(R.id.btnTakePicture);
        hasil = (TextView) findViewById(R.id.hasil);
        hasil2 = (TextView) findViewById(R.id.hasil2);
        hasil3 = (TextView) findViewById(R.id.hasil3);
```

```
        hasil4 = (TextView) findViewById(R.id.hasil4);

        mDatabaseHelper = new DatabaseHelper(this);

        final Calendar calendar = Calendar.getInstance();
        SimpleDateFormat DateFormat = new
SimpleDateFormat("yyyy-MM-dd");
        final String currentDate =
DateFormat.format(calendar.getTime());

        hari = findViewById(R.id.txt3);
        TextView textViewDate = findViewById(R.id.time);
        textViewDate.setText(currentDate);


        generate.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String text = plat.getText().toString();


db.collection("Berlangganan").addSnapshotListener(new
EventListener<QuerySnapshot>() {
                    @Override
                    public void onEvent(QuerySnapshot
queryDocumentSnapshots, FirebaseFirestoreException e) {
                        if (e != null) {
                            Log.d(TAG, "Error" +
e.getMessage());
                        }
                        for (DocumentSnapshot doc :
queryDocumentSnapshots) {
                            String DBNoPlat =
doc.getString("noPlat");
                            if
(plat.getText().toString().equals(DBNoPlat)) {
                                statusp = "Berlangganan";

TVStatus.setVisibility(View.VISIBLE);
                            } else {

TVStatus.setVisibility(View.GONE);
                                statusp = "Tidak
Berlangganan";
                            }
                        }
                    }
                });
                if (text != null && !text.isEmpty()) {
                    try {
                        MultiFormatWriter
multiFormatWriter = new MultiFormatWriter();
```

```
                        BitMatrix bitMatrix =
multiFormatWriter.encode(text, BarcodeFormat.QR_CODE, 225,
225);
                        BarcodeEncoder barcodeEncoder =
new BarcodeEncoder();
                        Bitmap bitmap =
barcodeEncoder.createBitmap(bitMatrix);
                        qrcode.setImageBitmap(bitmap);
                    } catch (WriterException e) {
                        e.printStackTrace();
                    }
                }
            }
        });

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

                Calendar calendar1 =
Calendar.getInstance();
                SimpleDateFormat format = new
SimpleDateFormat("HH.mm.ss");
                String time_in =
format.format(calendar1.getTime());
                String nomorplat =
plat.getText().toString().trim();

                hari.setText(currentDate);
                timein.setText(time_in);
                noplat.setText(nomorplat);

            }
        });

        scan.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View v) {
                IntentIntegrator intentIntegrator = new
IntentIntegrator(MotorActivity.this);

intentIntegrator.setDesiredBarcodeFormats(IntentIntegrator
.QR_CODE);
                intentIntegrator.setCameraId(0);

intentIntegrator.setOrientationLocked(false);
                intentIntegrator.setPrompt("Scanning");
                intentIntegrator.setBeepEnabled(true);

intentIntegrator.setBarcodeImageEnabled(true);
                intentIntegrator.initiateScan();
```

```
            }
        });

        hapus.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                plat.setText("");
                hari.setText("");
                timein.setText("");
                noplat.setText("");
            }
        });

        // Create object of controls
        Button btnConnect = (Button)
findViewById(R.id.btnConnect);
        Button btnDisconnect = (Button)
findViewById(R.id.btnDisconnect);
        Button btnPrint = (Button)
findViewById(R.id.btnPrint);


        lblPrinterName = (TextView)
findViewById(R.id.lblPrinterName);

        btnConnect.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                try {
                    FindBluetoothDevice();
                    openBluetoothPrinter();

                } catch (Exception ex) {
                    ex.printStackTrace();
                }
            }
        });
        btnDisconnect.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                try {
                    disconnectBT();
                } catch (Exception ex) {
                    ex.printStackTrace();
                }
            }
        });
        btnPrint.setOnClickListener(view -> {
            try {
                printData();
```

```
            } catch (Exception ex) {
                ex.printStackTrace();
            }
        });

        btnTakePicture.setOnClickListener(view -> {
            try {
                Intent intent = new
Intent(MotorActivity.this, CameraActivity.class);
                startActivityForResult(intent, 200);
            } catch (Exception ex) {
                ex.printStackTrace();
            }
        });
    }

    void FindBluetoothDevice() {

        try {

            bluetoothAdapter =
BluetoothAdapter.getDefaultAdapter();
            if (bluetoothAdapter == null) {
                lblPrinterName.setText("No Bluetooth
Adapter found");
            }
            if (!bluetoothAdapter.isEnabled()) {
                Intent enableBT = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
                startActivityForResult(enableBT, 0);
            }

            Set<BluetoothDevice> pairedDevice =
bluetoothAdapter.getBondedDevices();

            if (pairedDevice.size() > 0) {
                for (BluetoothDevice pairedDev :
pairedDevice) {

                    // My Bluetoth printer name is
BlueTooth Printer
                    if
(pairedDev.getName().equals("RPP02N")) {
                        bluetoothDevice = pairedDev;
                        lblPrinterName.setText("Bluetooth
Printer Attached: " + pairedDev.getName());
                        break;
                    }
                }
            }

            lblPrinterName.setText("Bluetooth Printer
Attached");
```

```
        } catch (Exception ex) {
            ex.printStackTrace();
        }

    }

    // Open Bluetooth Printer

    void openBluetoothPrinter() throws IOException {
        try {

            //Standard uuid from string //
            UUID uuidSting = UUID.fromString("00001101-
0000-1000-8000-00805f9b34fb");
            bluetoothSocket =
bluetoothDevice.createRfcommSocketToServiceRecord(uuidStin
g);
            bluetoothSocket.connect();
            outputStream =
bluetoothSocket.getOutputStream();
            inputStream =
bluetoothSocket.getInputStream();

            beginListenData();

        } catch (Exception ex) {

        }
    }

    void beginListenData() {
        try {

            final Handler handler = new Handler();
            final byte delimiter = 10;
            stopWorker = false;
            readBufferPosition = 0;
            readBuffer = new byte[1024];

            thread = new Thread(new Runnable() {
                @Override
                public void run() {

                    while
(!Thread.currentThread().isInterrupted() && !stopWorker) {
                        try {
                            int byteAvailable =
inputStream.available();
                            if (byteAvailable > 0) {
                                byte[] packetByte = new
byte[byteAvailable];

inputStream.read(packetByte);
```

```
                                    for (int i = 0; i <
byteAvailable; i++) {
                                        byte b =
packetByte[i];
                                        if (b == delimiter) {
                                            byte[] encodedByte
= new byte[readBufferPosition];
                                            System.arraycopy(

readBuffer, 0,

encodedByte, 0,

encodedByte.length
                                            );
                                            final String data
= new String(encodedByte, "US-ASCII");
                                            readBufferPosition
= 0;
                                            handler.post(new
Runnable() {
                                                @Override
                                                public void
run() {

lblPrinterName.setText(data);
                                                }
                                            });
                                        } else {

readBuffer[readBufferPosition++] = b;
                                        }
                                    }
                                }
                            } catch (Exception ex) {
                                stopWorker = true;
                            }
                        }

                    }
                });

                thread.start();
            } catch (Exception ex) {
                ex.printStackTrace();
            }
        }

        // Printing Text & Bitmap to Bluetooth Printer //
        void printData() throws IOException {
            try {
```

```
            Bitmap bmp = ((BitmapDrawable)
qrcode.getDrawable()).getBitmap();
            String baris1 = "\n" +
label1.getText().toString();
            String baris2 = "\n" +
label2.getText().toString();
            String tanggal = "\n\n Day:" +
hari.getText().toString();
            String waktu = "\n Time In : " +
timein.getText().toString();
            String msg = "\n No. Plat: " +
noplat.getText().toString();
            String last = "\n\n" +
label3.getText().toString();
            String last2 = "\n Jenis Kendaraan : " +
label4.getText().toString();
            String baris3 = "\n" +
label5.getText().toString();
            String baris4 = "\n" +
label6.getText().toString();

            msg += "\n";
            outputStream.write(baris1.getBytes());
            outputStream.write(baris2.getBytes());
            outputStream.write(last.getBytes());
            outputStream.write(baris3.getBytes());
            outputStream.write(baris4.getBytes());
            outputStream.write(tanggal.getBytes());
            outputStream.write(waktu.getBytes());
            outputStream.write(last2.getBytes());
            outputStream.write(msg.getBytes());


            if (bmp != null) {
                byte[] command = Utils.decodeBitmap(bmp);

outputStream.write(PrinterCommands.ESC_ALIGN_LEFT);
                printText(command);

            } else {
                Log.e("Print Error", "file no");
            }

            lblPrinterName.setText("Printing Text...");
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }

    // Disconnect Printer //
    void disconnectBT() throws IOException {
        try {
            stopWorker = true;
```

```
            outputStream.close();
            inputStream.close();
            bluetoothSocket.close();
            lblPrinterName.setText("Printer
Disconnected.");
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }

    private void printText(byte[] msg) {
        try {
            // Print normal text
            outputStream.write(msg);


        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    //cloud firestore//
    public void saveNote(View v) {

        String nm = nama.getText().toString();
        String day = hari.getText().toString();
        String time = timein.getText().toString();
        String platnomor = noplat.getText().toString();
        String jenis = label4.getText().toString();

        Map<String, Object> note = new HashMap<>();
        note.put(KEY_DAY, day);
        note.put(KEY_TIME, time);
        note.put(KEY_PLAT, platnomor);
        note.put(KEY_JENIS, jenis);
        note.put(KEY_TIMEOUT, " ");
        note.put(KEY_TARIF, "Rp 0");
        note.put(KEY_STATUS, statusp);


db.collection("data").document(nm).collection("kendaraan")
.document().set(note)
                .addOnSuccessListener(new
OnSuccessListener<Void>() {
                    @Override
                    public void onSuccess(Void aVoid) {
                        Toast.makeText(MotorActivity.this,
"Saved to server", Toast.LENGTH_SHORT).show();
                    }
                })
                .addOnFailureListener(new
OnFailureListener() {
                    @Override
```

```
                    public void onFailure(@NonNull
Exception e) {
                        Toast.makeText(MotorActivity.this,
"Failed!", Toast.LENGTH_SHORT).show();
                        Log.d(TAG, e.toString());
                    }
                });

        /*
db.collection("data").document(nm).collection("kendaraan")
.whereEqualTo("noPlat",noplat).get()
                .addOnCompleteListener(new
OnCompleteListener<QuerySnapshot>() {
                    @Override
                    public void onComplete(@NonNull
Task<QuerySnapshot> task) {
                        if (task.isSuccessful())
                            for (QueryDocumentSnapshot
document : task.getResult()) {
                            }
                        else {

                        }
                    }
                }); */


        String newEntry = hari.getText().toString();
        String newEntry2 = timein.getText().toString();
        String newEntry3 = noplat.getText().toString();
        String newEntry4 = label4.getText().toString();

        if (noplat.length() != 0) {
            AddData(newEntry +
                    "\nTime In : " + newEntry2 +
                    "\nNo.Plat : " + newEntry3 +
                    "\nJenis Kendaraan : " + newEntry4);
        } else {
            Toast.makeText(MotorActivity.this, "You must
put something in the text field!",
Toast.LENGTH_SHORT).show();
        }
    }

    public void AddData(String newEntry) {
        boolean insertData =
mDatabaseHelper.addData(newEntry);

        if (insertData) {
            Toast.makeText(MotorActivity.this, "Saved to
History", Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(MotorActivity.this, "Something
```

```
went wrong", Toast.LENGTH_SHORT).show();
        }
    }

    //result scanning//
    @Override
    protected void onActivityResult(int requestCode, int
resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode,
data);
        final IntentResult result =
IntentIntegrator.parseActivityResult(requestCode,
resultCode, data);
        final String nm = nama.getText().toString();
        if (result != null && result.getContents() !=
null) {

db.collection("data").document(nm).collection("kendaraan")
.whereEqualTo("noPlat", result.getContents()).get()
                  //document(result.getContents()).get()
                  .addOnCompleteListener(new
OnCompleteListener<QuerySnapshot>() {
                      @Override
                      public void onComplete(@NonNull
Task<QuerySnapshot> task) {
                          if (task.isSuccessful()) {
                              for (final
QueryDocumentSnapshot document : task.getResult()) {
                                  Calendar calendar =
Calendar.getInstance();
                                  SimpleDateFormat
format = new SimpleDateFormat("HH.mm.ss");
                                  final String masuk =
document.getString(KEY_TIME);
                                  final String keluar =
format.format(calendar.getTime());
                                  final String
statusparkir = document.getString(KEY_STATUS);


                                  Timer updateTimer =
new Timer();

updateTimer.schedule(new TimerTask() {

@SuppressLint("SetTextI18n")
                                      public void run()
{
                                          try {


SimpleDateFormat format = new
SimpleDateFormat("HH.mm.ss");
```

```
                                                    Date date1
= format.parse(keluar);
                                                    Date date2
= format.parse(masuk);
                                                    long mills
= date1.getTime() - date2.getTime();

Log.v("Data1", "" + date1.getTime());

Log.v("Data2", "" + date2.getTime());
                                                    int hours
= (int) (mills / (1000 * 60 * 60));
                                                    int mins =
(int) (mills / (1000 * 60)) % 60;
                                                    int harga
= 0;

                                                    if
(plat2.getText().toString().equals(statusparkir)) {
                                                        harga
= 0;
                                                    } else {
                                                        if
(hours < 1) {

harga = 2000;
                                                        } else
if (hours >= 1 && hours < 9) {

harga = 2000 + (hours * 1000);
                                                        } else
{

harga = 10000;
                                                        }
                                                    }

                                                    String
diff = hours + ":" + mins; // updated value every second

hasil.setText(keluar);

hasil2.setText(masuk);

hasil3.setText(diff + " status : " + statusparkir);

hasil4.setText("Rp " + harga);
                                                } catch
(Exception e) {

e.printStackTrace();
                                                }
```

```
                                                String nm =
nama.getText().toString();
                                                String out =
hasil.getText().toString();
                                                String trf =
hasil4.getText().toString();


db.collection("data").document(nm).collection("kendaraan")
.document(document.getId()).update(KEY_TIMEOUT, out);

db.collection("data").document(nm).collection("kendaraan")
.document(document.getId()).update(KEY_TARIF, trf);
                                            }

                                        }, 0, 1000);

                                        if (document.exists())
{
                                            Log.d(TAG,
"DocumentSnapshot data: " + document.getData());
                                        } else {
                                            Log.d(TAG, "No
such document");
                                        }
                                    }
                                }

                                new
AlertDialog.Builder(MotorActivity.this)
                                        .setTitle("Scan
Result")

.setMessage(result.getContents())

.setPositiveButton("Copy", new
DialogInterface.OnClickListener() {
                                            @Override
                                            public void
onClick(DialogInterface dialog, int which) {

ClipboardManager manager = (ClipboardManager)
getSystemService(CLIPBOARD_SERVICE);
                                                ClipData data
= ClipData.newPlainText("Result", result.getContents());

manager.setPrimaryClip(data);
                                            }

}).setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
                                            @Override
                                            public void
```

```
onClick(DialogInterface dialog, int which) {

dialog.dismiss();
                                            }
                                        }).create().show();

                            }
                        });
        }

        // From camera activity
        if (requestCode == 200) {
            if (resultCode == Activity.RESULT_OK) {
                Uri fotoPlat = data.getData();
                String platNomor =
data.getStringExtra(CameraActivity.EXTRA_PLAT_NOMOR);
                plat.setText(platNomor);
            } else if (resultCode ==
Activity.RESULT_CANCELED) {
                Toast.makeText(this, "device tidak
memiliki kamera", Toast.LENGTH_LONG).show();
            }
        }
    }
}
```

4. Tabel *Source Code* Halaman Parkir Mobil pada Aplikasi *Smart Parking*

```
package com.jtd.adit.tiketparkir;

import android.annotation.SuppressLint;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.ClipData;
import android.content.ClipboardManager;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.drawable.BitmapDrawable;

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

import android.net.Uri;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
```

```
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.firestore.FirebaseFirestore;
import
com.google.firebase.firestore.QueryDocumentSnapshot;
import com.google.firebase.firestore.QuerySnapshot;
import com.google.zxing.BarcodeFormat;
import com.google.zxing.MultiFormatWriter;
import com.google.zxing.WriterException;
import com.google.zxing.common.BitMatrix;
import
com.google.zxing.integration.android.IntentIntegrator;
import com.google.zxing.integration.android.IntentResult;
import com.journeyapps.barcodescanner.BarcodeEncoder;
import com.jtd.adit.tiketparkir.camera.CameraActivity;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.HashMap;
import java.util.Map;

// Import neccessor namespace
import android.bluetooth.BluetoothAdapter;
import  android.bluetooth.BluetoothDevice;
import  android.bluetooth.BluetoothSocket;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

import java.util.Set;
import java.util.Timer;
import java.util.TimerTask;
import java.util.UUID;
import android.os.Handler;

public class MobilActivity extends AppCompatActivity {

    private static final String TAG = "MainActivity";
    private static final String KEY_DAY = "day";
    private static final String KEY_TIME = "timeIn";
    private static final String KEY_PLAT = "noPlat";
    private static final String KEY_JENIS =
"jenisKendaraan";
    private static final String KEY_TIMEOUT = "timeOut";
    private static final String KEY_TARIF = "tarif";
```

```
    private FirebaseFirestore db =
FirebaseFirestore.getInstance();

    private EditText plat;
    private TextView timein;
    private TextView noplat;
    private TextView label1;
    private TextView label2;
    private TextView label3;
    private TextView label4;
    private TextView label5;
    private TextView label6;
    private TextView hasil, hasil2, hasil3, hasil4;
    private ImageView qrcode;
    private TextView hari, nama;
    private Button generate;
    private Button proses;
    private Button scan;
    private Button hapus;
    private Button btnTakePicture;

    private FirebaseAuth.AuthStateListener authListener;
    private FirebaseAuth auth;

    //printer
    BluetoothAdapter bluetoothAdapter;
    BluetoothSocket bluetoothSocket;
    BluetoothDevice bluetoothDevice;

    OutputStream outputStream;
    InputStream inputStream;
    Thread thread;

    byte[] readBuffer;
    int readBufferPosition;
    volatile boolean stopWorker;

    TextView lblPrinterName;

    DatabaseHelper mDatabaseHelper;

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

        getSupportActionBar().setTitle("Parkir Mobil");

        //Get Firebase auth instance
        auth = FirebaseAuth.getInstance();

        //get current user
        final FirebaseUser user =
```

```
FirebaseAuth.getInstance().getCurrentUser();

        String email = user.getEmail().toString();

        nama = (TextView)findViewById(R.id.nama);

        nama.setText(email);

        plat = (EditText) findViewById(R.id.plat);
        timein = (TextView) findViewById(R.id.txt4);
        noplat = (TextView) findViewById(R.id.txt5);
        label1 = (TextView) findViewById(R.id.txt1);
        label2 = (TextView) findViewById(R.id.txt2);
        qrcode = (ImageView) findViewById(R.id.qrcode);
        generate = (Button) findViewById(R.id.generate);
        proses = (Button) findViewById(R.id.proses);
        label3 = (TextView) findViewById(R.id.note);
        label4 = (TextView) findViewById(R.id.txt6);
        label5 = (TextView) findViewById(R.id.note2);
        label6 = (TextView) findViewById(R.id.note3);
        scan = (Button) findViewById(R.id.btnScan);
        hapus = (Button) findViewById(R.id.hapus);
        btnTakePicture = (Button)
findViewById(R.id.btnTakePicture);
        hasil = (TextView) findViewById(R.id.hasil);
        hasil2 = (TextView) findViewById(R.id.hasil2);
        hasil3 = (TextView) findViewById(R.id.hasil3);
        hasil4 = (TextView) findViewById(R.id.hasil4);

        mDatabaseHelper = new DatabaseHelper(this);

        final Calendar calendar = Calendar.getInstance();
        SimpleDateFormat DateFormat = new
SimpleDateFormat("yyyy-MM-dd");
        final String currentDate =
DateFormat.format(calendar.getTime());
        //final String currentDate =
DateFormat.getDateInstance(DateFormat.FULL).format(calenda
r.getTime());

        hari = findViewById(R.id.txt3);
        TextView textViewDate = findViewById(R.id.time);
        textViewDate.setText(currentDate);


        generate.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String text = plat.getText().toString();
                if (text!= null && !text.isEmpty()){
                    try{
                        MultiFormatWriter
```

```
multiFormatWriter = new MultiFormatWriter();
                        BitMatrix bitMatrix =
multiFormatWriter.encode(text, BarcodeFormat.QR_CODE, 225,
225);
                        BarcodeEncoder barcodeEncoder =
new BarcodeEncoder();
                        Bitmap bitmap =
barcodeEncoder.createBitmap(bitMatrix);
                        qrcode.setImageBitmap(bitmap);
                    } catch (WriterException e){
                        e.printStackTrace();
                    }
                }
            }
        });

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

                Calendar calendar1 =
Calendar.getInstance();
                SimpleDateFormat format = new
SimpleDateFormat("HH.mm.ss");
                String time_in =
format.format(calendar1.getTime());
                String nomorplat =
plat.getText().toString().trim();

                hari.setText(currentDate);
                timein.setText(time_in);
                noplat.setText(nomorplat);

            }
        });

        scan.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View v) {
                IntentIntegrator intentIntegrator = new
IntentIntegrator(MobilActivity.this);

intentIntegrator.setDesiredBarcodeFormats(IntentIntegrator
.QR_CODE);
                intentIntegrator.setCameraId(0);

intentIntegrator.setOrientationLocked(false);
                intentIntegrator.setPrompt("Scanning");
                intentIntegrator.setBeepEnabled(true);

intentIntegrator.setBarcodeImageEnabled(true);
```

```
                intentIntegrator.initiateScan();
            }
        });

        hapus.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                plat.setText("");
                hari.setText("");
                timein.setText("");
                noplat.setText("");
            }
        });

        // Create object of controls
        Button btnConnect = (Button)
findViewById(R.id.btnConnect);
        Button btnDisconnect = (Button)
findViewById(R.id.btnDisconnect);
        Button btnPrint = (Button)
findViewById(R.id.btnPrint);


        lblPrinterName = (TextView)
findViewById(R.id.lblPrinterName);

        btnConnect.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                try{
                    FindBluetoothDevice();
                    openBluetoothPrinter();

                }catch (Exception ex){
                    ex.printStackTrace();
                }
            }
        });
        btnDisconnect.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                try{
                    disconnectBT();
                }catch (Exception ex){
                    ex.printStackTrace();
                }
            }
        });
        btnPrint.setOnClickListener(new
View.OnClickListener() {
```

```
            @Override
            public void onClick(View view) {
                try{
                    printData();
                }catch (Exception ex){
                    ex.printStackTrace();
                }
            }
        });
        btnTakePicture.setOnClickListener(view -> {
            try {
                Intent intent = new
Intent(MobilActivity.this, CameraActivity.class);
                startActivityForResult(intent, 200);
            } catch (Exception ex) {
                ex.printStackTrace();
            }
        });
    }

    void FindBluetoothDevice(){

        try{

            bluetoothAdapter =
BluetoothAdapter.getDefaultAdapter();
            if(bluetoothAdapter==null){
                lblPrinterName.setText("No Bluetooth
Adapter found");
            }
            if(!bluetoothAdapter.isEnabled()){
                Intent enableBT = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
                startActivityForResult(enableBT,0);
            }

            Set<BluetoothDevice> pairedDevice =
bluetoothAdapter.getBondedDevices();

            if(pairedDevice.size()>0){
                for(BluetoothDevice
pairedDev:pairedDevice){

                    // My Bluetoth printer name is
BlueTooth Printer

if(pairedDev.getName().equals("RPP02N")){
                        bluetoothDevice=pairedDev;
                        lblPrinterName.setText("Bluetooth
Printer Attached: "+pairedDev.getName());
                        break;
                    }
                }
```

```
            }

            lblPrinterName.setText("Bluetooth Printer
Attached");
        }catch(Exception ex){
            ex.printStackTrace();
        }

    }

    // Open Bluetooth Printer

    void openBluetoothPrinter() throws IOException{
        try{

            //Standard uuid from string //
            UUID uuidSting = UUID.fromString("00001101-
0000-1000-8000-00805f9b34fb");

bluetoothSocket=bluetoothDevice.createRfcommSocketToServic
eRecord(uuidSting);
            bluetoothSocket.connect();

outputStream=bluetoothSocket.getOutputStream();
            inputStream=bluetoothSocket.getInputStream();

            beginListenData();

        }catch (Exception ex){

        }
    }

    void beginListenData(){
        try{

            final Handler handler =new Handler();
            final byte delimiter=10;
            stopWorker =false;
            readBufferPosition=0;
            readBuffer = new byte[1024];

            thread=new Thread(new Runnable() {
                @Override
                public void run() {

                    while
(!Thread.currentThread().isInterrupted() && !stopWorker){
                        try{
                            int byteAvailable =
inputStream.available();
                            if(byteAvailable>0){
                                byte[] packetByte = new
```

```
byte[byteAvailable];

inputStream.read(packetByte);

                                    for(int i=0;
i<byteAvailable; i++){
                                        byte b =
packetByte[i];
                                        if(b==delimiter){
                                            byte[] encodedByte
= new byte[readBufferPosition];
                                            System.arraycopy(

readBuffer,0,

encodedByte,0,

encodedByte.length
                                            );
                                            final String data
= new String(encodedByte,"US-ASCII");

readBufferPosition=0;
                                            handler.post(new
Runnable() {
                                                @Override
                                                public void
run() {

lblPrinterName.setText(data);
                                                }
                                            });
                                        }else{

readBuffer[readBufferPosition++]=b;
                                        }
                                    }
                                }
                            }catch(Exception ex){
                                stopWorker=true;
                            }
                        }

                    }
                });

                thread.start();
            }catch (Exception ex){
                ex.printStackTrace();
            }
        }

        // Printing Text & Bitmap to Bluetooth Printer //
```

```
    void printData() throws  IOException {
        try {

            Bitmap bmp =
((BitmapDrawable)qrcode.getDrawable()).getBitmap();
            String baris1 = "\n" +
label1.getText().toString();
            String baris2 = "\n"+
label2.getText().toString();
            String tanggal = "\n\n Day:" +
hari.getText().toString();
            String waktu = "\n Time In : " +
timein.getText().toString();
            String msg = "\n No. Plat: " +
noplat.getText().toString();
            String last = "\n\n" +
label3.getText().toString();
            String last2 = "\n Jenis Kendaraan : " +
label4.getText().toString();
            String baris3 = "\n"+
label5.getText().toString();
            String baris4 = "\n"+
label6.getText().toString();

            msg += "\n";
            outputStream.write(baris1.getBytes());
            outputStream.write(baris2.getBytes());
            outputStream.write(last.getBytes());
            outputStream.write(baris3.getBytes());
            outputStream.write(baris4.getBytes());
            outputStream.write(tanggal.getBytes());
            outputStream.write(waktu.getBytes());
            outputStream.write(last2.getBytes());
            outputStream.write(msg.getBytes());


            if(bmp!=null){
                byte[] command = Utils.decodeBitmap(bmp);

outputStream.write(PrinterCommands.ESC_ALIGN_LEFT);
                printText(command);

            }else {
                Log.e("Print Error","file no");
            }

            lblPrinterName.setText("Printing Text...");
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }

    // Disconnect Printer //
```

```
    void disconnectBT() throws IOException{
        try {
            stopWorker=true;
            outputStream.close();
            inputStream.close();
            bluetoothSocket.close();
            lblPrinterName.setText("Printer
Disconnected.");
        }catch (Exception ex){
            ex.printStackTrace();
        }
    }

    private void printText(byte[] msg) {
        try {
            // Print normal text
            outputStream.write(msg);


        } catch (IOException e) {
            e.printStackTrace();
        }
    }


    //cloud firestore//
    public void saveNote(View v){
        String nm = nama.getText().toString();
        String day = hari.getText().toString();
        String time = timein.getText().toString();
        String platnomor = noplat.getText().toString();
        String jenis = label4.getText().toString();

        Map<String,Object> note = new HashMap<>();
        note.put(KEY_DAY, day);
        note.put(KEY_TIME, time);
        note.put(KEY_PLAT, platnomor);
        note.put(KEY_JENIS, jenis);
        note.put(KEY_TIMEOUT, " ");
        note.put(KEY_TARIF, "Rp 0 ");


db.collection("data").document(nm).collection("kendaraan")
.document().set(note)
                .addOnSuccessListener(new
OnSuccessListener<Void>() {
                    @Override
                    public void onSuccess(Void aVoid) {
                        Toast.makeText(MobilActivity.this,
"Saved to server", Toast.LENGTH_SHORT).show();
                    }
                })
                .addOnFailureListener(new
```

```
OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull
Exception e) {
                        Toast.makeText(MobilActivity.this,
"Failed!", Toast.LENGTH_SHORT).show();
                        Log.d(TAG, e.toString());
                    }
                });

        String newEntry = hari.getText().toString();
        String newEntry2 = timein.getText().toString();
        String newEntry3 = noplat.getText().toString();
        String newEntry4 = label4.getText().toString();

        if (noplat.length() != 0) {
            AddData(newEntry +
                    "\nTime In : "+ newEntry2 +
                    "\nNo.Plat : "+ newEntry3 +
                    "\nJenis Kendaraan : "+ newEntry4);
        } else {
            Toast.makeText(MobilActivity.this,"You must
put something in the text field!",
Toast.LENGTH_SHORT).show();
        }
    }

    public void AddData(String newEntry) {
        boolean insertData =
mDatabaseHelper.addData(newEntry);

        if (insertData) {
            Toast.makeText(MobilActivity.this,"Saved to
History", Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(MobilActivity.this,"Something
went wrong", Toast.LENGTH_SHORT).show();
        }
    }

    //result scanning//
    @Override
    protected void onActivityResult(int requestCode, int
resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode,
data);
        final IntentResult result =
IntentIntegrator.parseActivityResult(requestCode,
resultCode, data);
        if (result!=null && result.getContents()!=null){
            String nm = nama.getText().toString();

db.collection("data").document(nm).collection("kendaraan")
```

```
.whereEqualTo("noPlat",result.getContents()).get()
                    //document(result.getContents()).get()
                    .addOnCompleteListener(new
OnCompleteListener<QuerySnapshot>() {
                        @Override
                        public void onComplete(@NonNull
Task<QuerySnapshot> task) {
                            if (task.isSuccessful()) {
                                for (final
QueryDocumentSnapshot document : task.getResult()) {
                                    Calendar calendar =
Calendar.getInstance();
                                    SimpleDateFormat
format = new SimpleDateFormat("HH.mm.ss");
                                    final String masuk =
document.getString(KEY_TIME);
                                    final String keluar =
format.format(calendar.getTime());

                                    Timer updateTimer =
new Timer();

updateTimer.schedule(new TimerTask() {

@SuppressLint("SetTextI18n")
                                        public void run()
{
                                            try {


SimpleDateFormat format = new
SimpleDateFormat("HH.mm.ss");
                                            Date date1
= format.parse(keluar);
                                            Date date2
= format.parse(masuk);
                                            long mills
= date1.getTime() - date2.getTime();

Log.v("Data1", "" + date1.getTime());

Log.v("Data2", "" + date2.getTime());
                                            int hours
= (int) (mills / (1000 * 60 * 60));
                                            int mins =
(int) (mills / (1000 * 60)) % 60;
                                            int harga
= 0;

                                            if (hours
< 1) {
                                                harga
= 3000;
```

```
                                                    }
                                                    else if
(hours > 1 && hours < 9) {
                                                        harga
= 3000 + (hours * 1000);
                                                    }
                                                    else {
                                                        harga
= 12000;
                                                    }
                                                    String
diff = hours + ":" + mins; // updated value every second

hasil.setText(keluar);

hasil2.setText(masuk);

hasil3.setText(diff);

hasil4.setText("Rp " + harga);
                                                } catch
(Exception e) {

e.printStackTrace();
                                                }


                                                String nm =
nama.getText().toString();
                                                String out =
hasil.getText().toString();
                                                String trf =
hasil4.getText().toString();


db.collection("data").document(nm).collection("kendaraan")
.document(document.getId()).update(KEY_TIMEOUT, out);

db.collection("data").document(nm).collection("kendaraan")
.document(document.getId()).update(KEY_TARIF, trf);
                                                    }

                                                }, 0, 1000);

                                                if (document.exists())
{
                                                    Log.d(TAG,
"DocumentSnapshot data: " + document.getData());
                                                } else {
                                                    Log.d(TAG, "No
such document");
                                                }
                                            }
                                        }
```

```
                                new
AlertDialog.Builder(MobilActivity.this)
                                        .setTitle("Scan
Result")

.setMessage(result.getContents())

.setPositiveButton("Copy", new
DialogInterface.OnClickListener() {
                                            @Override
                                            public void
onClick(DialogInterface dialog, int which) {

ClipboardManager manager = (ClipboardManager)
getSystemService(CLIPBOARD_SERVICE);
                                                ClipData data
= ClipData.newPlainText("Result", result.getContents());

manager.setPrimaryClip(data);
                                            }

}).setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
                                @Override
                                public void
onClick(DialogInterface dialog, int which) {
                                    dialog.dismiss();
                                }
                            }).create().show();

                        }
                    });

        }
        // From camera activity
        if (requestCode == 200) {
            if (resultCode == Activity.RESULT_OK) {
                Uri fotoPlat = data.getData();
                String platNomor =
data.getStringExtra(CameraActivity.EXTRA_PLAT_NOMOR);
                plat.setText(platNomor);
            } else if (resultCode ==
Activity.RESULT_CANCELED) {
                Toast.makeText(this, "device tidak
memiliki kamera", Toast.LENGTH_LONG).show();
            }
        }
    }
}
```

5. Tabel *Source Code* Halaman *History* pada Aplikasi *Smart Parking*

```
package com.jtd.adit.tiketparkir;

import android.database.Cursor;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.widget.ArrayAdapter;
import android.widget.ListAdapter;
import android.widget.ListView;

import java.util.ArrayList;


public class HistoryActivity extends AppCompatActivity {

    private static final String TAG = "HistoryActivity";

    DatabaseHelper mDatabaseHelper;

    private ListView mListView;

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

        getSupportActionBar().setTitle("History");

        mListView = (ListView)
findViewById(R.id.listView);
        mDatabaseHelper = new DatabaseHelper(this);

        populateListView();
    }

    private void populateListView() {
        Log.d(TAG, "populateListView: Displaying data in
the ListView.");

        //get the data and append to a list
        Cursor data = mDatabaseHelper.getData();
        ArrayList<String> listData = new ArrayList<>();
        while (data.moveToNext()) {
            //get the value from the database in column 1
            //then add it to the ArrayList
            listData.add(data.getString(1));
        }
        //create the list adapter and set the adapter
        ListAdapter adapter = new ArrayAdapter<>(this,
android.R.layout.simple_list_item_1, listData);
        mListView.setAdapter(adapter);

    }
```

```
}
```

6. Tabel *Source Code* Halaman Kehilangan Tiket pada Aplikasi *Smart Parking*

```
package com.jtd.adit.tiketparkir;

import android.app.ProgressDialog;
import android.content.Intent;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;

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

import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.material.snackbar.Snackbar;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.OnProgressListener;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;

import java.io.ByteArrayOutputStream;
import java.util.HashMap;
import java.util.Map;
import java.util.UUID;

public class KehilanganActivity extends AppCompatActivity
{

    ImageView imageViewAdd;
    private Button btnupload;
    private Uri filepath;
    public EditText ETnamapemarkir, ETnoplat, ETnoSTNK;
    private final int PICK_IMAGE_REQUEST = 71;
    private FirebaseFirestore DBFirestore;
    String generatedFilePath;
    FirebaseStorage storage;
    StorageReference storageReference;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_kehilangan);
        getSupportActionBar().setTitle("Kehilangan
Tiket");
        imageViewAdd = (ImageView)
findViewById(R.id.imageViewAdd);
        btnupload = (Button) findViewById(R.id.btnupload);
        storageReference =
FirebaseStorage.getInstance().getReference();
        DBFirestore = FirebaseFirestore.getInstance();
        ETnamapemarkir = (EditText)
findViewById(R.id.ETnamapemarkir);
        ETnoplat = (EditText) findViewById(R.id.ETnoplat);
        ETnoSTNK = (EditText) findViewById(R.id.ETnoSTNK);
    }

    private void uploadImage() {

        if (filepath != null) {
            /** menampilkan process window **/
            final ProgressDialog progressDialog = new
ProgressDialog(this);
            progressDialog.setTitle("Uploading...");
            progressDialog.show();

            /** mengakses storage foto **/
            StorageReference ref =
storageReference.child("images/" +
UUID.randomUUID().toString());
            ref.putFile(filepath)
                    .addOnSuccessListener(new
OnSuccessListener<UploadTask.TaskSnapshot>() {
                        @Override
                        public void
onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                            progressDialog.dismiss();

ref.getDownloadUrl().addOnSuccessListener(uri ->
generatedFilePath = uri.toString());

Toast.makeText(KehilanganActivity.this, "Uploaded",
Toast.LENGTH_SHORT).show();
                        }
                    })
                            .

                    addOnFailureListener(new
OnFailureListener() {
                        @Override
                        public void onFailure(@NonNull
Exception e) {
                            progressDialog.dismiss();

Toast.makeText(KehilanganActivity.this, "Failed " +
```

```
e.getMessage(), Toast.LENGTH_SHORT).show();
                        }
                    })
                            .

                    addOnProgressListener(new
OnProgressListener<UploadTask.TaskSnapshot>() {
                        @Override
                        public void
onProgress(UploadTask.TaskSnapshot taskSnapshot) {
                            double progress = (100.0 *
taskSnapshot.getBytesTransferred() / taskSnapshot
                                    .getTotalByteCount());

progressDialog.setMessage("Uploaded " + (int) progress +
"%");
                        }
                    });
        }
    }

    private void uploadImageCamera() {

        if (imageViewAdd != null) {

            imageViewAdd.setDrawingCacheEnabled(true);
            imageViewAdd.buildDrawingCache();
            Bitmap bitmapAdd =
imageViewAdd.getDrawingCache();
            ByteArrayOutputStream baos = new
ByteArrayOutputStream();
            bitmapAdd.compress(Bitmap.CompressFormat.JPEG,
100, baos);
            byte[] dataAdd = baos.toByteArray();

            final ProgressDialog progressDialog = new
ProgressDialog(this);
            progressDialog.setTitle("Uploading...");
            progressDialog.show();

            StorageReference ref =
storageReference.child("images/" +
UUID.randomUUID().toString());
            ref.putBytes(dataAdd)
                    .addOnSuccessListener(new
OnSuccessListener<UploadTask.TaskSnapshot>() {
                        @Override
                        public void
onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                            progressDialog.dismiss();

ref.getDownloadUrl().addOnSuccessListener(uri ->
generatedFilePath = uri.toString());
```

```
Toast.makeText(KehilanganActivity.this, "Uploaded",
Toast.LENGTH_SHORT).show();
                        }
                    })
                    .addOnFailureListener(new
OnFailureListener() {
                        @Override
                        public void onFailure(@NonNull
Exception e) {
                            progressDialog.dismiss();

Toast.makeText(KehilanganActivity.this, "Failed " +
e.getMessage(), Toast.LENGTH_SHORT).show();
                        }
                    })
                    .addOnProgressListener(new
OnProgressListener<UploadTask.TaskSnapshot>() {
                        @Override
                        public void
onProgress(UploadTask.TaskSnapshot taskSnapshot) {
                            double progress = (100.0 *
taskSnapshot.getBytesTransferred() / taskSnapshot
                                    .getTotalByteCount());

progressDialog.setMessage("Uploaded " + (int) progress +
"%");
                        }
                    });
        }
    }

    public void Camera(View view) {
        Intent takePicture = new
Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        startActivityForResult(takePicture, 0);//zero can
be replaced with any action code
    }


    protected void onActivityResult(int requestCode, int
resultCode, Intent
            imageReturnedIntent) {
        if (resultCode != RESULT_CANCELED) {
            super.onActivityResult(requestCode,
resultCode, imageReturnedIntent);

            switch (requestCode) {
                case 0:
                    if (resultCode == RESULT_OK) {


                        Bitmap photo = (Bitmap)
```

```
imageReturnedIntent.getExtras().get("data");

                        filepath =
imageReturnedIntent.getData();

imageViewAdd.setImageBitmap(photo);


                        uploadImageCamera();


                    }

                    break;
                case 1:
                    if (resultCode == RESULT_OK) {
                        Uri selectedImage =
imageReturnedIntent.getData();

imageViewAdd.setImageURI(selectedImage);
                        filepath =
imageReturnedIntent.getData();
                        uploadImage();
                    }
                    break;
            }
        }
    }


    public void btnupload(View view) {
        Snackbar.make(view, "Menghubungkan Ke Server",
Snackbar.LENGTH_LONG)
                .setAction("Action", null).show();

        String namapemarkir =
ETnamapemarkir.getText().toString();
        String noplat = ETnoplat.getText().toString();
        String noSTNK = ETnoSTNK.getText().toString();

        Map<String, String> userMap = new HashMap<>();
        userMap.put("namaPemarkir", namapemarkir);
        userMap.put("noPlat", noplat);
        userMap.put("noSTNK", noSTNK);
        userMap.put("url", generatedFilePath);


DBFirestore.collection("kehilangan").add(userMap).addOnSuc
cessListener(new OnSuccessListener<DocumentReference>() {
            @Override
            public void onSuccess(DocumentReference
documentReference) {
                Toast.makeText(KehilanganActivity.this,
```

```
"Berhasil Upload", Toast.LENGTH_SHORT).show();

            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                String error = e.getMessage();
                Toast.makeText(KehilanganActivity.this,
"Error : " + error, Toast.LENGTH_SHORT).show();

            }
        });

    }
}
```

Lampiran 4. Transkrip Hasil Wawancara Pihak Juru Parkir Dengan Bapak I Gede Adi Indrawan Di Rumah Sakit Umum Daerah (RSUD) Buleleng

1. Pertanyaan : Sistem parkir digital yang ada saat ini dikelola oleh siapa?

   Jawaban : Sistem parkir disini dikelola oleh pihak rumah sakit umum sendiri.

2. Pertanyaan : Bagaimana cara kerja sistem parkir digital saat ini?

   Jawaban : Pertama-tama gerbang masuk atau *Barrier gate* nantinya akan mendeteksi kendaraan secara otomatis. Baik itu bagi pengendara sepeda motor maupun mobil, semuanya akan terdeteksi secara langsung. Selanjutnya pengendara menekan tombol hijau yang berguna untuk mencetak karcis transaksi setelah itu nantinya tiket tersebut akan keluar secara otomatis apabila ditekan. Pada saat keluar, tempelkan kertas karcis yang menjadi bukti transaksi masuk pengendara di area parkir. Selain itu, nantinya akan dilakukan proses verifikasi barcode melalui ticket dispenser. Apabila datanya valid maka secara otomatis tarif parkir akan muncul di layar. Kemudian siapkan kartu elektronik dengan saldo yang cukup untuk pembayaran dan tempelkan atau tap di payment reader atau bisa dengan pembayaran *cash*. Jika transaksi berhasil, maka bukti pembayaran akan tercetak secara otomatis. Setelah itu, palang pintu alias *barrier gate* tersebut akan terbuka dengan sendirinya. Jika sudah melewati palang pintu tersebut, maka nantinya *Barrier gate* akan tertutup secara otomatis.

3. Pertanyaan : Untuk juru parkir yang bertugas menjaga sistem ada berapa per harinya?

   Jawaban : Disini terdapat 2 orang untuk menjaga sistem parkir,

menggunakan 2 *shift* kerja per harinya meliputi pagi dan sore hari

4. Pertanyaan : Berapa besaran biaya parkir baik itu motor atau mobil?

   Jawaban : Untuk mobil Rp 2.000 dan mobil Rp 3.000

5. Pertanyaan : Untuk biaya parkir motor atau mobil disini perhari atau perjam?

   Jawaban : Untuk biaya parkir disini perhari, terdapat juga sistem langganan jadi orang yang kerja di rumah sakit ini akan berlangganan biaya parkirnya per bulan.

6. Pertanyaan : Apakah parkir disini terdapat jenis parkir langganan?

   Jawaban : Iya, disini bisa juga membuat member untuk parkir berlangganan. Jadi orang-orang yang bekerja disini pasti memiliki member untuk parkir berlangganan, bebas keluar masuk kapan saja tanpa perlu membayar per hari.

7. Pertanyaan : Apakah sistem parkir disini menerima pembayaran secara non tunai atau hanya tunai saja?

   Jawaban : Untuk pembayaran parkir disini bisa *cash* dan bisa menggunakan *e-money*.

8. Pertanyaan : Apakah sistem parkir digital disini memiliki sistem pengamanan cctv?

   Jawaban : Iya, terdapat di depan sistem parkir untuk mengawasi parkir digital disini.

9. Pertanyaan : Apa kelebihan dan kekurangan dengan adanya sistem parkir digital saat ini?

   Jawaban : Kelebihannya sistem parkir tidak banyak antrian, bukti

transaksi dapat tersimpan lebih rapi dan bisa bekerja sama dengan bank, kekurangannya orang-orang yang tidak parkir berlangganan jika keluar masuk disini harus membayar berkali-kali.

Lampiran 5. Gambar Wawancara Pihak Juru Parkir Di Rumah Sakit Umum Daerah (RSUD) Buleleng

RIWAYAT HIDUP

Nyoman Aditya Wiradarma lahir di Waingapu pada tanggal 3 Maret 2001. Penulis lahir dari pasangan suami istri Bapak I Wayan Darmayasa dan Ibu Nyoman Sukeni. Penulis berkebangsaan Indonesia dan beragama Hindu. Kini penulis beralamat di Jalan Sri Rama Barat Dusun Tista, Baktiseraga, Kecamatan Buleleng Kabupaten Buleleng, Provinsi Bali.

Penulis menyelesaikan pendidikan dasar di SD Negeri 125 Banyuasri dan lulus pada tahun 2013. Kemudian penulis melanjutkan di SMP Negeri 2 Singaraja dan lulus pada tahun 2016. Pada tahun 2019, penulis lulus dari SMA Negeri 2 Singaraja jurusan MIPA. Penulis masih terdaftar sebagai mahasiswa Program Studi S1 Ilmu Komputer di Universitas Pendidikan Ganesha.