

ABSTRAK

Prans Cafe merupakan sebuah cafe sekaligus rumah makan yang berlokasi di Jalan Mpu Sedah, Nglingsi, Klaten Selatan Provinsi Jawa Tengah. Cafe ini memiliki banyak karyawan untuk membantu mengelola bisnis agar berjalan dengan lancar. Dengan banyaknya karyawan yang bekerja, pemilik café membutuhkan sebuah sistem yang dapat membantu mengelola presensi untuk mengetahui kinerja karyawannya dalam bekerja. Presensi adalah kegiatan mengambil data untuk membahas jumlah yang diambil pada suatu acara atau kegiatan, dimana suatu kegiatan dapat diselesaikan dengan baik yang melibatkan keaktifan seseorang. Sama halnya dengan mengelola bisnis Prans Cafe yang akan membutuhkan banyak orang. Oleh karena itu perlu adanya pengawasan yang baik terhadap kehadiran karyawan. Perkembangan pengguna smartphone yang semakin meningkat dapat dimanfaatkan untuk mempermudah dan meningkatkan pengalaman proses bekerja. Penggunaan mobile yang menyediakan lebih banyak efisien dan efektif terkait dengan kemudahan mengakses dan mengambil informasi, namun menggunakan aplikasi QR Code atau barcode juga bisa digunakan sebagai presensi langsung pengganti sistem lama dimana proses presensi yang dilakukan masih manual yaitu dengan melakukan tanda tangan sebagai bukti kehadiran. Sistem presensi dengan QR Code juga mampu menghindari adanya kendala-kendala yang biasa terjadi, seperti laporan yang hilang atau rusak.

Kata Kunci: Presensi, QR Code, Mobile

Tanggal Pertama Publikasi : 18 Oktober 2023

Daftar Nama Pengusul

N a m a : 1. Khansa Khairun Nisa
Kewarganegaraan : Indonesia
Alamat : Metuk Kidul RT 013 RW 05 Tegalyoso Klaten Selatan Jawa Tengah
Kode Pos : 57424
No Hp : 081393357878
Alamat email : khansakhairunnisa431@gmail.com

N a m a : 2. Sutarman M.Kom., Ph.D.
Kewarganegaraan : Indonesia
Alamat : Jl Janti Karangjambe No 130-B Banguntapan Bantul Yogyakarta
Kode Pos : 55198
No Hp : 082134406672
Alamat email : sutarman@uty.ac.id

N a m a : 3. Murti Retnowo, S.Kom., M.Cs.
Kewarganegaraan : Indonesia
Alamat : Celeban UH III/505 RT 20 RW 05 Yogyakarta
Kode Pos : 55167
No Hp : 087812779345
Alamat email : murti.retnowo@uty.ac.id

N a m a : 4. Agus Suhendar,S.T.,M.Eng
Kewarganegaraan : Indonesia
Alamat : Degan II, 082/041, Banjararum, Kalibawang, Kulon Progo, Yogyakarta
Kode Pos : 55672
No Hp : 081578888331
Alamat email : agus.suhendar@staff.uty.ac.id

N a m a : 5. Ikrimach S.Kom., M.Cs.
Kewarganegaraan : Indonesia
Alamat : Karang Gondang RT 020 RW 000 Pendowo Harjo Sewon Bantul
D.I Yogyakarta
Kode Pos : -
No Hp : 081578888331
Alamat email : ikrimach@uty.ac.id

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta:

N a m a : Khansa Khairun Nisa
Kewarganegaraan : Indonesia
Alamat : Metuk Kidul RT 013 RW 05 Tegalyoso Klaten Selatan Jawa Tengah

N a m a : Sutarman M.Kom. Ph.D.
Kewarganegaraan : Indonesia
Alamat : Jl. Janti Karangjambe 130-B Banguntapan Bantul D.I.Yogyakarta

N a m a : Murti Retnowo S.Kom., M.Cs.
Kewarganegaraan : Indonesia
Alamat : Celeban UH III/505 RT 20 RW 05 Yogyakarta D.I.Yogyakarta

N a m a : Agus Suhendar S.T. M.Eng.
Kewarganegaraan : Indonesia
Alamat : Degan II, Banjararum, Kalibawang, Kulonprogo D.I.Yogyakarta

N a m a : Ikrimach S.Kom., M.Cs.
Kewarganegaraan : Indonesia
Alamat : Jln Raya Bantul No 211 RT 12 Karang Gondang Pendowo Harjo Sewon Bantul D.I.Yogyakarta

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:
 - Berupa : Source Code Program Berbasis Android
 - Berjudul : Implementasi Presensi Karyawan menggunakan QR Code Berbasis Android (Studi Kasus: Prans Café)
 - Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyek kekayaan intelektual lainnya sebagaimana dimaksud dalam Pasal 68 ayat (2);
 - Bukan merupakan Ekspresi Budaya Tradisional sebagaimana dimaksud dalam Pasal 38;
 - Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
 - Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;
 - Bukan merupakan Ciptaan seni lukis yang berupa logo atau tanda pembeda yang digunakan sebagai merek dalam perdagangan barang/jasa atau digunakan sebagai lambang organisasi, badan usaha, atau badan hukum sebagaimana dimaksud dalam Pasal 65 dan;
 - Bukan merupakan Ciptaan yang melanggar norma agama, norma susila, ketertiban umum, pertahanan dan keamanan negara atau melanggar peraturan perundang-undangan sebagaimana dimaksud dalam Pasal 74 ayat (1) huruf d Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.
2. Sebagai pemohon mempunyai kewajiban untuk menyimpan asli contoh ciptaan yang dimohonkan dan harus memberikan apabila dibutuhkan untuk kepentingan penyelesaian sengketa perdata maupun pidana sesuai dengan ketentuan perundang-undangan.
3. Karya Cipta yang saya mohonkan pada Angka 1 tersebut di atas tidak pernah dan tidak sedang dalam sengketa pidana dan/atau perdata di Pengadilan.

4. Dalam hal ketentuan sebagaimana dimaksud dalam Angka 1 dan Angka 3 tersebut di atas saya / kami langgar, maka saya / kami bersedia secara sukarela bahwa:
- permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau
 - Karya Cipta yang telah terdaftar dalam Daftar Umum Ciptaan Direktorat Hak Cipta, Direktorat Jenderal Hak Kekayaan Intelektual, Kementerian Hukum Dan Hak Asasi Manusia R.I dihapuskan sesuai dengan ketentuan perundang-undangan yang berlaku.
 - Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam berperkara dan/atau sedang dalam gugatan di Pengadilan maka status kepemilikan surat pencatatan elektronik tersebut ditangguhkan menunggu putusan Pengadilan yang berkekuatan hukum tetap.

Demikian Surat pernyataan ini saya/kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Yogyakarta, 19 Oktober 2023



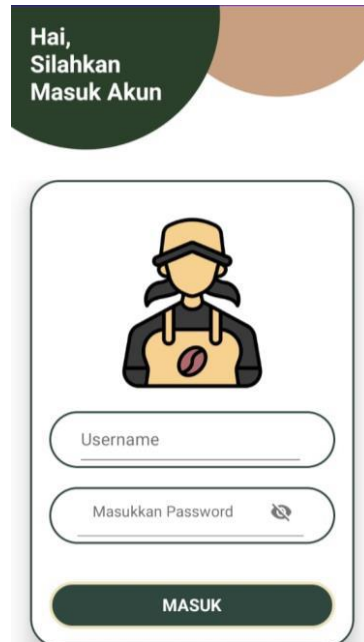
1. Khansa Khairun Nisa
2. Sutarman, M.Kom., Ph.D.
3. Murti Retnowo, S.Kom., M.Cs
4. Agus Suhendar S.T., M.Eng.
5. Ikrimach, S.Kom., M.Cs.

Pemegang Hak Cipta*

* Semua pemegang hak cipta agar menandatangani di atas materai.

MANUAL SISTEM

IMPLEMENTASI PRESENSI KARYAWAN MENGGUNAKAN QR CODE BERBASIS ANDROID (STUDI KASUS: PRANS CAFÉ)



OLEH

1. Khansa Khairun Nisa
2. Sutarman, M.Kom., Ph.D.
3. Murti Retnowo, S.Kom., M.Cs
4. Agus Suhendar S.T., M.Eng.
5. Ikrimach, S.Kom., M.Cs.

UNIVERSITAS TEKNOLOGI YOGYAKARTA

TAHUN 2023

DAFTAR ISI

JUDUL.....	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR.....	ii
1. TAMPILAN DAN KODE PROGRAM	3
1.1 Halaman Utama	3
1.1.1. Halaman Login.....	3
1.1.2. Potongan Source Code Login	3
1.2 Halaman Dashboard.....	5
1.2.1. Tampilan Dashboard Utama	5
1.2.2. Source Code Beranda.....	5
1.3 Halaman Scan QR Code	9
1.3.1. Tampilan Halaman Scan QR Code	9
1.3.2. Source Code Scan QR Code	9
1.4 Halaman Generate QR Code.....	12
1.4.1 Tampilan Halaman Generate QR Code	12
1.4.2 Source Code Halaman Generate QR Code	12
1.5 Halaman Riwayat Karyawan	15
1.5.1 Tampilan Halaman Riwayat Karyawan	15
1.5.2 Source Code Riwayat Karyawan	15
1.6 Halaman Riwayat Presensi	18
1.6.1 Tampilan Riwayat Presensi.....	18
1.6.2 Source Code Riwayat Presensi	18
1.7 Halaman Ijin dan Sakit Karyawan	21
1.7.1 Tampilan Halaman Ijin dan Sakit Karyawan.....	21
1.7.2 Source Code Ijin dan Sakit Karyawan	21
1.8 Halaman Cuti Karyawan.....	24
1.8.1 Tampilan Cuti Karyawan.....	24
1.8.2 Source Code Cuti Karyawan.....	24
1.9 Halaman Riwayat Admin.....	27
1.9.1 Tampilan Riwayat Admin	27
1.9.2 Source Code Riwayat Admin.....	27
1.10 Tambah Karyawan	31
1.10.1 Halaman Tambah Karyawan	31
1.10.2 Source Code Tambah Karyawan	31
2. PENGGUNAAN PROGRAM.....	36
2.1 Hasil Implementasi Antarmuka	36

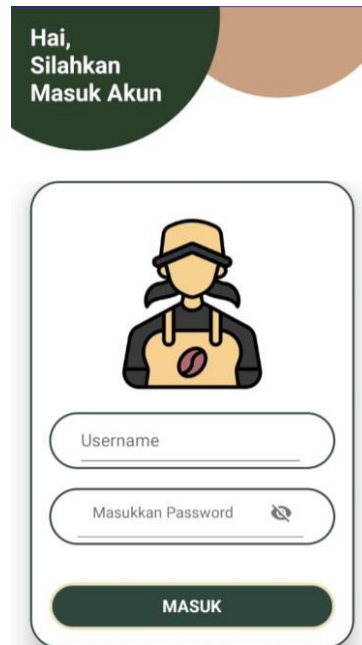
DAFTAR GAMBAR

Gambar 1 - Halaman <i>Login</i>	3
Gambar 2 – Halaman QR Code Scanner	5
Gambar 3 – Halaman Scan QR Code	9
Gambar 4 – Halaman Register QR Code.....	12
Gambar 5 – Halaman Riwayat Karyawan	15
Gambar 6 – Riwayat Presensi	18
Gambar 8 – Ijin dan Sakit Karyawan.....	21
Gambar 9 – Cuti Karyawan	24
Gambar 10 – Riwayat Admin	27
Gambar 11. Tambah Karyawan	31
Gambar 12 – Halaman Login.....	36
Gambar 13 – Scan QR Code.....	37
Gambar 14 - Generate QR Code	37
Gambar 15 – Riwayat Karyawan	38
Gambar 16 - Riwayat Presensi QR Code.....	38
Gambar 17 – Pengajuan Ijin Sakit	39
Gambar 18 – Halaman Cuti Karyawan	39
Gambar 19– Tambah Karyawan	40

1. TAMPILAN DAN KODE PROGRAM

1.1 Halaman Utama

1.1.1. Halaman Login



Gambar 1 - Halaman *Login*

1.1.2. Potongan Source Code Login

```
1. package com.projek.tapengaduanmahasiswa;
2.
3. import androidx.annotation.NonNull;
4. import androidx.appcompat.app.AppCompatActivity;
5.
6. import android.app.ProgressDialog;
7. import android.content.Intent;
8. import android.os.Bundle;
9. import android.view.View;
10. import android.widget.Button;
11. import android.widget.TextView;
12. import android.widget.Toast;
13.
14. import com.google.firebase.auth.FirebaseAuth;
15. import com.google.firebase.database.DataSnapshot;
16. import com.google.firebase.database.DatabaseError;
17. import com.google.firebase.database.DatabaseReference;
18. import com.google.firebase.database.FirebaseDatabase;
19. import com.google.firebase.database.ValueEventListener;
20.
21. public class Login extends AppCompatActivity {
22.     TextView txtusr, txtpasswd;
23.     String struser, strpasswd;
24.     public String idus, strlevel, strlvl;
```

```

25.     Button btnlogin, btnbatal, btnregister;
26.     DatabaseReference ldb;
27.
28.     @Override
29.     protected void onCreate(Bundle savedInstanceState) {
30.         super.onCreate(savedInstanceState);
31.         setContentView(R.layout.activity_login);
32.         txtusr=findViewById(R.id.username);
33.         txtpasswd=findViewById(R.id.password);
34.         btnbatal=findViewById(R.id.btn_batal);
35.         btnlogin=findViewById(R.id.btnlogin);
36.         btnregister=findViewById(R.id.btn_register);
37.         ldb= FirebaseDatabase.getInstance().getReference();
38.
39.     }
40.
41.     public void cekpengguna(View e){
42.         struser=txtusr.getText().toString();
43.         strpasswd=txtpasswd.getText().toString();
44.         ldb.child("anggota")
45.             .orderByChild("nimpeng")
46.             .equalTo(struser)
47.             .addValueEventListener(new ValueEventListener() {
48.                 @Override
49.                 public void onDataChange(@NonNull DataSnapshot snapshot)
50.                 {
51.                     for(DataSnapshot data:snapshot.getChildren()){
52.                         String
passwd=data.child("ttlpeng").getValue().toString();
53.
54.                         if(passwd.equals(strpasswd)){
55.                             String
strgambar=data.child("imgpeng").getValue().toString();
56.                             idus
data.child("idpeng").getValue().toString();
57.                             if(strlevel.equals("0")) {
58.                                 //level Admin
59.                                 Intent macem = new Intent(Login.this,
Admin.class);
60.
61.                                 macem.putExtra("Username", struser);
62.                                 macem.putExtra("gambar",strgambar);
63.                                 macem.putExtra("peng","");
64.                                 macem.putExtra("lvl", strlevel);
65.                                 startActivity(macem);
66.                             }else{
67.                                 //Level Pengguna
68.                                 Intent nama = new Intent(Login.this,
MainActivity.class);
69.
70.                                 nama.putExtra("Username", struser);
71.                                 nama.putExtra("gambar",strgambar);
72.                                 nama.putExtra("peng",idus);
73.                                 nama.putExtra("lvl", strlevel);
74.                                 startActivity(nama);
75.                             }
76.                             //pengiriman Data / Variabel yang akan
digunakan untuk transaksi selanjutnya
77.
78.                             } else {
79.                                 Toast.makeText(Login.this, "Username Atau
Password Yang Anda Masukan Salah", Toast.LENGTH_SHORT).show();
80.                             }
81.                         }
82.                     }
83.                 }
84.             }
85.         }
86.     }
87.     @Override
88.     public void onCancelled(@NonNull DatabaseError error) {

```

```

84.
85.         }
86.     });
87. }
88. public void daftar(View e){
89.     Intent anggota= new Intent(Login.this, Register.class);
90.     startActivity(anggota);
91. }
92. public void keluar(View e){
93.     finish();
94. }
95.
96.
97. }

```

1.2 Halaman Dashboard

1.2.1. Tampilan Dashboard Utama



Gambar 2 – Halaman QR Code Scanner

1.2.2. Source Code Beranda

```

1. import com.bumptech.glide.Glide;
2. import androidx.appcompat.app.AlertDialog;
3. import androidx.appcompat.app.AppCompatActivity;
4. import androidx.core.app.ActivityCompat;
5. import androidx.core.content.ContextCompat;
6.
7. import android.Manifest;
8. import android.annotation.SuppressLint;
9. import android.content.DialogInterface;
10. import android.content.Intent;
11. import android.content.pm.PackageManager;
12. import android.os.Bundle;
13. import android.os.Handler;
14. import android.text.Layout;
15. import android.view.View;
16. import android.widget.ImageView;
17. import android.widget.TextView;
18. import android.widget.Toast;

```

```

19.
20. import com.google.firebase.database.DataSnapshot;
21. import com.google.firebase.database.DatabaseError;
22. import com.google.firebase.database.DatabaseReference;
23. import com.google.firebase.database.FirebaseDatabase;
24. import com.google.firebase.database.ValueEventListener;
25. import
    com.smarteist.autoimageslider.IndicatorView.animation.type.IndicatorAnimationType;
26. import com.smarteist.autoimageslider.SliderAnimations;
27. import com.smarteist.autoimageslider.SliderView;
28.
29. import org.checkerframework.checker.nullness.qual.NonNull;
30.
31. import java.text.DateFormat;
32. import java.text.SimpleDateFormat;
33. import java.util.ArrayList;
34. import java.util.Date;
35.
36. public class Dashboard extends AppCompatActivity {
37.
38.     String nik, nama;
39.     TextView btnscan, namatxt, niktxt, nohptxt, almttxt, textClock, textDate;
40.     ImageView foto;
41.     DatabaseReference ldb;
42.     SliderView sliderView;
43.     ArrayList<Integer> arrayList = new ArrayList<>();
44.     public String usr, level;
45.
46.     @Override
47.     protected void onCreate(Bundle savedInstanceState) {
48.         super.onCreate(savedInstanceState);
49.         setContentView(R.layout.activity_dashboard);
50.
51.         btnscan = findViewById(R.id.btnscan);
52.         namatxt = findViewById(R.id.namatxt);
53.         niktxt = findViewById(R.id.niktxt);
54.         nohptxt = findViewById(R.id.nohptxt);
55.         almttxt = findViewById(R.id.almttxt);
56.         ldb = FirebaseDatabase.getInstance().getReference();
57.         foto = (ImageView) findViewById(R.id.foto);
58.         textClock = findViewById(R.id.clock);
59.         textDate = findViewById(R.id.date);
60.         sliderView = findViewById(R.id.imageSlider);
61.         arrayList.add(R.drawable.pembuka);
62.         arrayList.add(R.drawable.k1);
63.         arrayList.add(R.drawable.k2);
64.         arrayList.add(R.drawable.k3);
65.         arrayList.add(R.drawable.k4);
66.         arrayList.add(R.drawable.penutup);
67.
68.         ImageSlider imageSlider = new ImageSlider(Dashboard.this, arrayList);
69.         sliderView.setIndicatorAnimation(IndicatorAnimationType.DROP);
70.
71.         sliderView.setSliderTransformAnimation(SliderAnimations.CUBEOUTDEPTHTRANSFORMATION);
72.         sliderView.setSliderAdapter(imageSlider);
73.         sliderView.startAutoCycle();
74.
75.         Handler handler = new Handler();
76.         handler.post(new Runnable() {
77.             @Override
78.             public void run() {
79.                 Date date = new Date();
80.
81.                 @SuppressWarnings("SimpleDateFormat")
82.                 DateFormat clockformat = new SimpleDateFormat("HH:mm:ss");

```

```

83.         @SuppressWarnings("SimpleDateFormat")
84.         DateFormat dateFormat = new SimpleDateFormat("EEEE, dd MMMM
yyyy");
85.
86.         textClock.setText(clockformat.format(new Date()));
87.         textDate.setText(dateFormat.format(new Date()));
88.
89.         handler.postDelayed(this, 1000);
90.
91.     }
92. });
93.
94.
95.     btnscan.setOnClickListener(s -> {
96.         if (ContextCompat.checkSelfPermission(Dashboard.this,
Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED) {
97.             if
(ActivityCompat.shouldShowRequestPermissionRationale(Dashboard.this,
Manifest.permission.CAMERA)) {
98.                 startScan();
99.             } else {
100.                ActivityCompat.requestPermissions(Dashboard.this, new
String[]{Manifest.permission.CAMERA}, 0);
101.            }
102.        } else {
103.            startScan();
104.        }
105.    });
106. }
107.
108. private void startScan() {
109.     Intent intent = new Intent(getApplicationContext(), Qrscanner.class);
110.     intent.putExtra("nik", nik);
111.     intent.putExtra("nama", nama);
112.     startActivityForResult(intent, 20);
113. }
114.
115. @Override
116. public void onBackPressed() {
117.     new AlertDialog.Builder(this)
118.         .setIcon(R.drawable.onboarding2)
119.         .setTitle(R.string.app_name)
120.         .setMessage("Apakah Anda yakin ingin keluar dari aplikasi
ini?")
121.         .setPositiveButton("Ya", new
DialogInterface.OnClickListener() {
122.             @Override
123.             public void onClick(DialogInterface dialog, int which) {
124.                 finish();
125.             }
126.         })
127.         .setNegativeButton("Tidak", new
DialogInterface.OnClickListener() {
128.             @Override
129.             public void onClick(DialogInterface dialog, int which) {
130.                 dialog.cancel();
131.             }
132.         })
133.         .show();
134. }
135.
136. @Override
137. public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
138.     super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
139.     if (requestCode == 0) {
140.         if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {

```

```

141.         startScan();
142.     } else {
143.         Toast.makeText(this, "Gagal membuka kamera!",
144.             Toast.LENGTH_LONG).show();
145.     }
146. }
147.
148. @Override
149. protected void onStart() {
150.     super.onStart();
151.     Intent masuk = getIntent();
152.     nik = masuk.getStringExtra("nik");
153.     nama = masuk.getStringExtra("nama");
154.     usr=masuk.getStringExtra("peng");
155.     level=masuk.getStringExtra("lvl");
156.     cekpengguna();
157. }
158.
159. public void cekpengguna() {
160.
161.     ldb.child("Data Register")
162.         .orderByChild("nik")
163.         .equalTo(nik)
164.         .addValueEventListener(new ValueEventListener() {
165.             @Override
166.             public void onDataChange(@NonNull DataSnapshot snapshot)
167.             {
168.                 for (DataSnapshot data : snapshot.getChildren()) {
169.                     // String
170.                     String tmpeng = data.child("tmpeng").getValue().toString();
171.                     String foto = data.child("foto").getValue().toString();
172.                     String nik = data.child("nik").getValue().toString();
173.                     String nama = data.child("nama").getValue().toString();
174.                     String nohp = data.child("nohp").getValue().toString();
175.                     String alamat = data.child("alamat").getValue().toString();
176.                     String url = "https://firebasestorage.googleapis.com/v0/b/taaplikasipresensi.appspot.com/o/gambar%2F" + namagambar + "?alt=media&token=2c2cdee3-13e6-48fd-be42-a47b71d81fal";
177.                     Glide.with(Dashboard.this).load(url).into(foto);
178.                     //gambar.setImageURI(url);
179.                     niktxt.setText(n);
180.                     namatxt.setText(nm);
181.                     nohptxt.setText(pr);
182.                     almttxt.setText(alm);
183.                 }
184.             }
185.         });
186.     }
187.
188. @Override
189. public void onCancelled(@NonNull DatabaseError error) {
190. }
191.
192. public void cuti(View e){
193.     Intent cuti=new Intent(Dashboard.this, Pcuti.class);
194.     cuti.putExtra("nik", nik);
195.     cuti.putExtra("nama", nama);
196.     startActivity(cuti);
197. }
198.
199. public void izinsakit(View e){

```

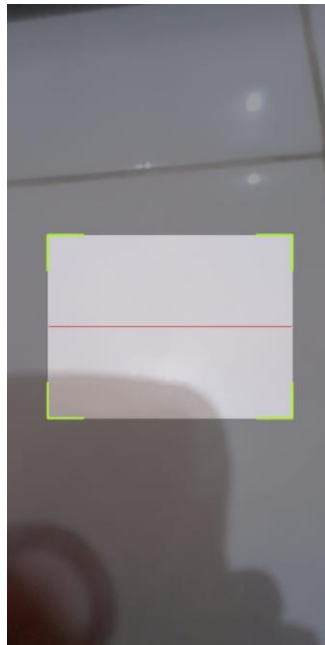
```

198.         Intent izinsakit=new Intent(Dashboard.this, Pizinsakit.class);
199.         izinsakit.putExtra("nik", nik);
200.         izinsakit.putExtra("nama", nama);
201.         startActivity(izinsakit);
202.     }
203.     public void keluar(View e){
204.         Intent keluar=new Intent(Dashboard.this, Keluar.class);
205.         keluar.putExtra("nik", nik);
206.         keluar.putExtra("nama", nama);
207.         keluar.putExtra("peng",nik);
208.         keluar.putExtra("lv", level);
209.         keluar.putExtra("lvl", level);
210.         startActivity(keluar);
211.     }
212.     public void panggilriwayat(View e) {
213.         Intent riwayat = new Intent(Dashboard.this, LayoutRiwayat.class);
214.         riwayat.putExtra("peng",nik);
215.         riwayat.putExtra("lv", level);
216.         riwayat.putExtra("nik", nik);
217.         riwayat.putExtra("lvl", level);
218.         startActivity(riwayat);
219.     }
220.
221. }

```

1.3 Halaman Scan QR Code

1.3.1. Tampilan Halaman Scan QR Code



Gambar 3 – Halaman Scan QR Code

1.3.2. Source Code Scan QR Code

```

1. import androidx.appcompat.app.AppCompatActivity;
2. import android.Manifest;
3. import android.content.Intent;

```



```

4. import android.database.Cursor;
5. import android.os.Bundle;
6. import android.widget.Toast;
7.
8. import com.google.android.gms.tasks.OnCompleteListener;
9. import com.google.android.gms.tasks.Task;
10. import com.google.firebase.database.DataSnapshot;
11. import com.google.firebase.database.DatabaseError;
12. import com.google.firebase.database.DatabaseReference;
13. import com.google.firebase.database.FirebaseDatabase;
14. import com.google.firebase.database.ValueEventListener;
15. import com.google.zxing.Result;
16.
17. import java.text.DateFormat;
18. import java.text.SimpleDateFormat;
19. import java.time.ZonedDateTime;
20. import java.time.format.DateTimeFormatter;
21. import java.util.Calendar;
22. import java.util.Date;
23. import java.util.Locale;
24.
25. import me.dm7.barcodescanner.zxing.ZXingScannerView;
26.
27. public class Qrscanner extends AppCompatActivity implements
    ZXingScannerView.ResultHandler {
28.     private ZXingScannerView mScannerView;
29.     DatabaseReference dbcek, refdbpeng;
30.     String nik, nama, snik, data, masuk, strstatus, iduser;
31.
32.     @Override
33.     protected void onCreate(Bundle savedInstanceState) {
34.         super.onCreate(savedInstanceState);
35.         mScannerView = new ZXingScannerView(this);
36.         mScannerView.setAspectTolerance(0.5f);
37.         setContentView(mScannerView);
38.
39.     }
40.
41.     @Override
42.     protected void onResume() {
43.         super.onResume();
44.         mScannerView.setResultHandler(this);
45.         mScannerView.startCamera();
46.     }
47.
48.     @Override
49.     protected void onStart() {
50.         super.onStart();
51.         Intent masuk=getIntent();
52.         nik=masuk.getStringExtra("nik");
53.         nama=masuk.getStringExtra("nama");
54.         snik="";
55.     }
56.
57.     @Override
58.     protected void onPause() {
59.         super.onPause();
60.         mScannerView.stopCamera();
61.     }
62.
63.     public void cekdata(String tanggal){
64.         FirebaseDatabase.getInstance().getReference("hasil presensi")
65.             .orderByChild("tanggal")
66.             //.startAt(Double.valueOf(nik))
67.             //.endAt(Double.valueOf(nik))
68.             .equalTo(tanggal)
69.             //.orderByChild("nik")
70.             //.equalTo(nik)

```

```

71.         .addListenerForSingleValueEvent(new ValueEventListener() {
72.             @Override
73.             public void onDataChange(DataSnapshot snapshot) {
74.                 String i=String.valueOf(snapshot.getChildrenCount());
75.                 // Toast.makeText(Qrscanner.this, i,
Toast.LENGTH_LONG).show();
76.                 if(Integer.parseInt(i)>0){
77.                     for(DataSnapshot snapshot1: snapshot.getChildren()){
78.                         /*if (snik == nik){
79.
80.                             } else {
81.                                 snik= (String)
snapshot.child("nik").getValue();
82.                             }*/
83.                             Toast.makeText(Qrscanner.this, "Anda sudah
presensi", Toast.LENGTH_LONG).show();
84.                         }
85.                         /*refdbpeng =
FirebaseDatabase.getInstance().getReference("hasil presensi");
86.                             presensi(data, masuk, snik, nama, strstatus);*/
87.                         }else{
88.                             Toast.makeText(Qrscanner.this, "Anda Berhasil
Presensi", Toast.LENGTH_LONG).show();
89.                             refdbpeng =
FirebaseDatabase.getInstance().getReference("hasil presensi");
90.                             presensi(data, masuk, nik, nama, strstatus);
91.                         }
92.                     }
93.
94.             @Override
95.             public void onCancelled(DatabaseError error) {
96.
97.             }
98.         });
99.     }
100.
101.     @Override
102.     public void handleResult(Result rawResult) {
103.         DateFormat currentTime = new SimpleDateFormat("HH:mm:ss");
104.         String tanggal =
currentTime.format(Calendar.getInstance().getTime());
105.         String dtStart = "2010-10-15T11:00:00Z";
106.         String converted = ZonedDateTime.parse(dtStart)
.format(DateTimeFormatter.ofPattern(
107.             "HHmmss",
108.             Locale.ENGLISH
109.         ));
110.
111.         String saiki = ZonedDateTime.now()
.format(DateTimeFormatter.ofPattern(
112.             "HHmmss",
113.             Locale.ENGLISH
114.         ));
115.
116.         masuk = ZonedDateTime.now()
.format(DateTimeFormatter.ofPattern(
117.             "HH:mm:ss",
118.             Locale.ENGLISH
119.         ));
120.
121.         data = rawResult.getText().toString();
122.         //String strstatus;
123.
124.         if (saiki.isEmpty() ) {
125.             Toast.makeText(Qrscanner.this, "Presensi
Gagal!",
126.                 Toast.LENGTH_SHORT).show();

```

```

131.         } else {
132.             // Toast.makeText(this, "Presensi Anda Hari Ini Berhasil!",
Toast.LENGTH_LONG).show();
133.             if (Integer.valueOf(saiki) > Integer.valueOf(converted)){
134.                 strstatus="Terlambat";
135.             } else {
136.                 strstatus="Hadir";
137.             }
138.
139.             cekdata(data);
140.         }
141.         finish();
142.     }
143.
144.     public void presensi(String data, String saiki, String nik, String nama,
String strstatus) {
145.         String key = refdbpeng.push().getKey();
146.         Qrscannerclass pemilih = new Qrscannerclass(key, data, saiki, nik ,
nama, strstatus);
147.         refdbpeng.child(key).setValue(pemilih);
148.     }
149. }

```

1.4 Halaman Generate QR Code

1.4.1 Tampilan Halaman Generate QR Code



Gambar 4 – Halaman Register QR Code

1.4.2 Source Code Halaman Generate QR Code

```

1. import android.app.DatePickerDialog;
2. import android.content.Intent;
3. import android.graphics.Bitmap;
4. import android.os.Bundle;
5. import android.provider.MediaStore;
6. import android.view.View;
7. import android.view.animation.Animation;
8. import android.view.animation.AnimationUtils;
9. import android.widget.Button;
10. import android.widget.DatePicker;
11. import android.widget.EditText;

```

```

12. import android.widget.ImageButton;
13. import android.widget.ImageView;
14. import android.widget.TextView;
15. import android.widget.Toast;
16.
17. import androidx.appcompat.app.AppCompatActivity;
18.
19. import com.google.zxing.BarcodeFormat;
20. import com.google.zxing.MultiFormatWriter;
21. import com.google.zxing.WriterException;
22. import com.google.zxing.common.BitMatrix;
23.
24. import java.text.SimpleDateFormat;
25. import java.util.Calendar;
26. import java.util.Date;
27. import java.util.Locale;
28.
29. public class GenerateCode extends AppCompatActivity {
30.     public final static int QRCodeWidth = 500;
31.     private static final long START_TIME_IN_MILLIS = 30000;
32.     Bitmap bitmap;
33.     TextView text;
34.     private TextView download;
35.     private TextView generate;
36.     private ImageView iv;
37.     private EditText etTanggal;
38.     Animation sideAnim;
39.
40.     @Override
41.     protected void onCreate(Bundle savedInstanceState) {
42.         super.onCreate(savedInstanceState);
43.         setContentView(R.layout.activity_generate_code);
44.         text = findViewById(R.id.text);
45.         download = findViewById(R.id.download);
46.         download.setVisibility(View.INVISIBLE);
47.         generate = findViewById(R.id.generate);
48.         iv = findViewById(R.id.image);
49.         etTanggal = findViewById(R.id.etTanggal);
50.         sideAnim = AnimationUtils.loadAnimation(this, R.anim.side_anim);
51.
52.         text.setAnimation(sideAnim);
53.
54.         etTanggal.setOnClickListener(view -> {
55.             Calendar tanggalJemput = Calendar.getInstance();
56.             DatePickerDialog.OnDateSetListener date = (view1, year,
monthOfYear, dayOfMonth) -> {
57.                 tanggalJemput.set(Calendar.YEAR, year);
58.                 tanggalJemput.set(Calendar.MONTH, monthOfYear);
59.                 tanggalJemput.set(Calendar.DAY_OF_MONTH, dayOfMonth);
60.
61.                 String strFormatDefault = "d MMMM yyyy";
62.                 SimpleDateFormat simpleDateFormat = new
SimpleDateFormat(strFormatDefault, Locale.getDefault());
63.
64.                 etTanggal.setText(simpleDateFormat.format(tanggalJemput.getTime()));
65.             });
66.             new DatePickerDialog(GenerateCode.this, date,
67.                 tanggalJemput.get(Calendar.YEAR),
68.                 tanggalJemput.get(Calendar.MONTH),
69.                 tanggalJemput.get(Calendar.DAY_OF_MONTH)).show();
70.         });
71.
72.         generate.setOnClickListener(new View.OnClickListener() {
73.             @Override
74.             public void onClick(View v) {
75.                 if (etTanggal.getText().toString().trim().length() == 0){
76.

```

```

77.         Toast.makeText(GenerateCode.this, "Enter Text",
Toast.LENGTH_SHORT).show();
78.     }
79.     else{
80.         try{
81.             bitmap =
textToImageEncode(etTanggal.getText().toString());
82.             iv.setImageBitmap(bitmap);
83.             download.setVisibility(View.VISIBLE);
84.             download.setOnClickListener(new View.OnClickListener()
{
85.                 @Override
86.                 public void onClick(View v) {
87.                     MediaStore.Images.Media.insertImage(getContentResolver(), bitmap,
"code_scanner"
88.                     , null);
89.                     Toast.makeText(GenerateCode.this, "QRCode
Tersimpan Ke Galeri", Toast.LENGTH_SHORT)
90.                         .show();
91.                 }
92.             });
93.         }catch (WriterException e){
94.             e.printStackTrace();
95.         }
96.     }
97. }
98. }
99. });
100. }
101.
102. private Bitmap textToImageEncode(String value) throws WriterException {
103.     BitMatrix bitMatrix;
104.     try {
105.         bitMatrix = new MultiFormatWriter().encode(value,
106.             BarcodeFormat.DATA_MATRIX.QR_CODE, QRCodeWidth,
QRCodeWidth, null);
107.     } catch (IllegalArgumentException e) {
108.         return null;
109.     }
110.
111.     int bitMatrixWidth = bitMatrix.getWidth();
112.     int bitMatrixHeight = bitMatrix.getHeight();
113.     int[] pixels = new int[bitMatrixWidth * bitMatrixHeight];
114.
115.     for (int y = 0; y < bitMatrixHeight; y++) {
116.         int offSet = y * bitMatrixWidth;
117.         for (int x = 0; x < bitMatrixWidth; x++) {
118.             pixels[offSet + x] = bitMatrix.get(x, y) ?
119.                 getResources().getColor(R.color.black) :
getResources().getColor(R.color.white);
120.         }
121.     }
122.     Bitmap bitmap = Bitmap.createBitmap(bitMatrixWidth, bitMatrixHeight,
Bitmap.Config.ARGB_4444);
123.     bitmap.setPixels(pixels, 0, 500, 0, 0, bitMatrixWidth,
bitMatrixHeight);
124.     return bitmap;
125. }
126. }

```

1.5 Halaman Riwayat Karyawan

1.5.1 Tampilan Halaman Riwayat Karyawan



Gambar 5 – Halaman Riwayat Karyawan

1.5.2 Source Code Riwayat Karyawan

```
1. import android.app.Activity;
2. import android.content.Context;
3. import android.content.Intent;
4. import android.os.Bundle;
5. import android.os.Parcelable;
6. import android.view.LayoutInflater;
7. import android.view.View;
8. import android.view.ViewGroup;
9. import android.widget.AdapterView;
10. import android.widget.Button;
11. import android.widget.ImageView;
12. import android.widget.ListView;
13. import android.widget.SimpleAdapter;
14. import android.widget.TextView;
15. import android.widget.Toast;
16.
17. import com.bumptech.glide.Glide;
18. import com.google.firebase.database.DataSnapshot;
19. import com.google.firebase.database.DatabaseError;
20. import com.google.firebase.database.DatabaseReference;
21. import com.google.firebase.database.FirebaseDatabase;
22. import com.google.firebase.database.ValueEventListener;
23.
24. import java.util.ArrayList;
25. import java.util.HashMap;
26. import java.util.List;
27. import java.util.Map;
28.
29. public class Riwayat extends Activity implements AdapterView.OnItemClickListener
    {
30.     String nama, nik, tanggal, jam, status, usr, levelpeng;
31.     Activity activity;
32.     ListView listView;
33.     Riwayat.ListAdapter adapter;
34.     Button btnBack;
35.     DatabaseReference dbm, refdbpeng;
```

```

36.     private ArrayList<HashMap<String, String>> list = new ArrayList<>();
37.
38.     @Override
39.     protected void onCreate(Bundle savedInstanceState) {
40.         super.onCreate(savedInstanceState);
41.         setContentView(R.layout.coba_riwayat);
42.
43.         activity = this;
44.         btnBack = (Button) findViewById(R.id.btnBack);
45.
46.         listView = findViewById(R.id.listview);
47.         listView.setOnItemClickListener(this);
48.         listView.setOverScrollMode(View.OVER_SCROLL_NEVER);
49.
50.         dbm= FirebaseDatabase.getInstance().getReference();
51.         // refdbpeng=dbm.child("laporan");
52.         refdbpeng = FirebaseDatabase.getInstance().getReference();
53.         //getcekaduan();
54.         // aduan();
55.         btnBack.setOnClickListener(new View.OnClickListener() {
56.             @Override
57.             public void onClick(View view) {
58.                 onBackPressed();
59.             }
60.         });
61.     }
62. }
63.
64. public void presensi(){
65.     if(levelpeng.equals("1")){
66.         getcekpresensi();
67.     }
68. }
69.
70. public void getcekpresensi(){
71.     list = new ArrayList<HashMap<String, String>>();
72.     //Toast.makeText(calonrt.this, nort, Toast.LENGTH_SHORT).show();
73.     refdbpeng.child("hasil presensi")
74.         .orderByChild("nik")
75.         .equalTo(usr)
76.         .addValueEventListener(new ValueEventListener() {
77.             @Override
78.             public void onDataChange(DataSnapshot dataSnapshot) {
79.                 int i = 1;
80.                 for (DataSnapshot snapshot : dataSnapshot.getChildren())
81.                 {
82.                     String id = (String) snapshot.getKey();
83.                     String idpeng = (String)
84. snapshot.child("idpeng").getValue();
85.                     //String nik = (String)
86. snapshot.child("nik").getValue();
87.                     //String nama = (String)
88. snapshot.child("nama").getValue();
89.                     String tanggal = (String)
90. snapshot.child("tanggal").getValue();
91.                     String jam = (String)
92. snapshot.child("jam").getValue();
93.                     String status = (String)
94. snapshot.child("status").getValue();
95.
96.                     HashMap<String, String> data = new HashMap<>();
97.                     data.put("key", id);
98.                     data.put("content", ""+i++);
99.
100.                    data.put("idpeng", idpeng);
101.                    //data.put("nik", nik);
102.                    //data.put("nama", nama);
103.                    data.put("tanggal", tanggal);

```

```

97.         data.put("jam", jam);
98.         data.put("status", status);
99.
100.            list.add(data);
101.        }
102.
103.            adapter = new Riwayat.ListAdapter(activity, list,
104.                R.layout.awnitem_dish_riwayat,        new
String[]{"content", "tanggal", "jam", "status"},
105.                new int[]{        R.id.tanggal,        R.id.jam,
R.id.status});
106.
107.            Parcelable state = listView.onSaveInstanceState();
108.            listView.setAdapter(adapter);
109.            listView.onRestoreInstanceState(state);
110.            adapter.notifyDataSetChanged();
111.
112.        }
113.        @Override
114.        public void onCancelled(DatabaseError databaseError) {
115.        }
116.    });
117.    }
118.
119.    public class ListAdapter extends SimpleAdapter {
120.        private Context mContext;
121.        public LayoutInflater inflater = null;
122.
123.        public ListAdapter(Context context, List<? extends Map<String, ?>>
data, int resource, String[] from, int[] to) {
124.            super(context, data, resource, from, to);
125.            mContext = context;
126.            inflater = (LayoutInflater)
mContext.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
127.        }
128.
129.        @Override
130.        public View getView(int position, View convertView, ViewGroup parent)
{
131.            View vi = convertView;
132.            if (convertView == null)
133.                vi = inflater.inflate(R.layout.awnitem_dish_riwayat, null);
134.
135.            HashMap<String, Object> data = (HashMap<String, Object>)
getItem(position);
136.            final TextView id = vi.findViewById(R.id.id);
137.            //final TextView nik = vi.findViewById(R.id.nik);
138.            //final TextView nama = vi.findViewById(R.id.nama);
139.            final TextView tanggal = vi.findViewById(R.id.tanggal);
140.            final TextView jam = vi.findViewById(R.id.jam);
141.            final TextView status = vi.findViewById(R.id.status);
142.
143.
144.            final String strid=(String) data.get("id");
145.            //final String strnik = (String) data.get("nik");
146.            //final String strnama= (String) data.get("nama");
147.            final String strtanggal = (String) data.get("tanggal");
148.            final String strjam=(String) data.get("jam");
149.            final String strstatus=(String) data.get("status");
150.
151.            id.setText(strid);
152.            //nik.setText("nik : " + strnik);
153.            //nama.setText("nama : " + strnama);
154.            tanggal.setText("Tanggal Presensi : " + strtanggal);
155.            jam.setText("Waktu Presensi : " + strjam);
156.            status.setText(strstatus);
157.
158.

```



```

159.         return vi;
160.     }
161. }
162.
163. @Override
164. protected void onStart() {
165.     super.onStart();
166.     Intent masuk=getIntent();
167.     usr=masuk.getStringExtra("peng");
168.     levelpeng=masuk.getStringExtra("lv");
169.     Toast.makeText(this, levelpeng, Toast.LENGTH_LONG).show();
170.     presensi();
171. }
172.
173. @Override
174. public void onItemClick(AdapterView<?> adapterView, View view, int i, long
175.     1) {
176. }
177. }
178.
179.

```

1.6 Halaman Riwayat Presensi

1.6.1 Tampilan Riwayat Presensi



Gambar 6 – Riwayat Presensi

1.6.2 Source Code Riwayat Presensi

```

1. package com.example.taaplikasipresensi;
2.
3. import androidx.appcompat.app.AlertDialog;
4. import androidx.appcompat.app.AppCompatActivity;
5. import androidx.appcompat.widget.SearchView;
6. import androidx.recyclerview.widget.GridLayoutManager;
7. import androidx.recyclerview.widget.RecyclerView;
8.
9. import android.content.Intent;
10. import android.os.Bundle;
11. import android.os.Parcelable;

```

```

12. import android.view.View;
13. import android.view.animation.Animation;
14. import android.view.animation.AnimationUtils;
15. import android.widget.AdapterView;
16. import android.widget.TextView;
17. import android.widget.Toast;
18.
19. import com.google.android.material.floatingactionbutton.FloatingActionButton;
20. import com.google.firebase.database.DataSnapshot;
21. import com.google.firebase.database.DatabaseError;
22. import com.google.firebase.database.DatabaseReference;
23. import com.google.firebase.database.FirebaseDatabase;
24. import com.google.firebase.database.ValueEventListener;
25.
26. import java.util.ArrayList;
27. import java.util.List;
28.
29. public class RiwayatPresensi extends AppCompatActivity {
30.     String nik, nama, usr, levelpeng;
31.     TextView text;
32.     Animation bottomanim;
33.     DatabaseReference databaseReference;
34.     ValueEventListener eventListener;
35.     RecyclerView recyclerView;
36.     List<Qrscannerclass> dataList;
37.     MyAdapterPresensi adapter;
38.     SearchView searchView;
39.
40.     @Override
41.     protected void onCreate(Bundle savedInstanceState) {
42.         super.onCreate(savedInstanceState);
43.         setContentView(R.layout.activity_riwayat_presensi);
44.
45.         recyclerView = findViewById(R.id.recyclerView);
46.         text = findViewById(R.id.text);
47.         bottomanim = AnimationUtils.loadAnimation(this,R.anim.bottom_anim);
48.         searchView = findViewById(R.id.search);
49.         searchView.clearFocus();
50.
51.         text.setAnimation(bottomanim);
52.
53.         GridLayoutManager          gridLayoutManager          =          new
GridLayoutManager(RiwayatPresensi.this, 1);
54.         recyclerView.setLayoutManager(gridLayoutManager);
55.
56.         //AlertDialog.Builder          builder          =          new
AlertDialog.Builder(RiwayatPcutiAdmin.this);
57.         //builder.setCancelable(false);
58.         // builder.setView(R.layout.progress_layout);
59.         //AlertDialog dialog = builder.create();
60.         // dialog.show();
61.
62.         searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
63.             @Override
64.             public boolean onQueryTextSubmit(String query) {
65.                 return false;
66.             }
67.
68.             @Override
69.             public boolean onQueryTextChange(String newText) {
70.                 searchList(newText);
71.                 return true;
72.             }
73.         });
74.
75.     }
76.     @Override
77.     protected void onStart() {

```

```

78.         super.onStart();
79.         Intent masuk = getIntent();
80.         nik = masuk.getStringExtra("nik");
81.         nama = masuk.getStringExtra("nama");
82.         usr = masuk.getStringExtra("peng");
83.         levelpeng = masuk.getStringExtra("lv");
84.         cuti();
85.     }
86.
87.     private void cuti() {
88.         if(levelpeng.equals("1")){
89.             getcekcuti();
90.         }
91.     }
92.
93.     private void getcekcuti() {
94.         dataList = new ArrayList<>();
95.         adapter = new MyAdapterPresensi(RiwayatPresensi.this, dataList);
96.         recyclerView.setAdapter(adapter);
97.         databaseReference = FirebaseDatabase.getInstance().getReference("hasil
presensi");
98.         //dialog.show();
99.         eventListener = databaseReference
100.             .orderByChild("nik")
101.             .equalTo(usr)
102.             .addValueEventListener(new ValueEventListener() {
103.                 @Override
104.                 public void onDataChange(DataSnapshot snapshot) {
105.                     dataList.clear();
106.                     //int i = 1;
107.                     for (DataSnapshot itemSnapshot :
snapshot.getChildren()) {
108.                         Qrscannerclass qrscannerclass =
itemSnapshot.getValue(Qrscannerclass.class);
109.
110.                         Qrscannerclass.setIdpeng(itemSnapshot.getKey());
111.
112.                         //pizinsakitclass.getKey();
113.
114.                         dataList.add(Qrscannerclass);
115.                     }
116.                     adapter.notifyDataSetChanged();
117.                     //dialog.dismiss();
118.                 }
119.
120.
121.                 @Override
122.                 public void onCancelled(DatabaseError databaseError) {
123.                     //dialog.dismiss();
124.                 }
125.             });
126.     }
127.
128.     public void searchList(String text) {
129.         ArrayList<Qrscannerclass> searchList = new ArrayList<>();
130.         for (Qrscannerclass Qrscannerclass : dataList) {
131.             if
(Qrscannerclass.getNama().toLowerCase().contains(text.toLowerCase())) {
132.                 searchList.add(Qrscannerclass);
133.             } else {
134.                 if
(Qrscannerclass.getNik().toLowerCase().contains(text.toLowerCase())) {
135.                     searchList.add(Qrscannerclass);
136.                 } else {
137.                     if
(Qrscannerclass.getTanggal().toLowerCase().contains(text.toLowerCase())) {
138.                         searchList.add(Qrscannerclass);
139.                     } else {

```

```

140.         if
141.         (Qrscannerclass.getJam().toLowerCase().contains(text.toLowerCase())) {
142.             searchList.add(Qrscannerclass);
143.         } else {
144.             if
145.             (Qrscannerclass.getStatus().toLowerCase().contains(text.toLowerCase())) {
146.                 searchList.add(Qrscannerclass);
147.             }
148.         }
149.     }
150. }
151.     adapter.searchDataList(searchList);
152. }
153.
154. }

```

1.7 Halaman Ijin dan Sakit Karyawan

1.7.1 Tampilan Halaman Ijin dan Sakit Karyawan



Gambar 8 – Ijin dan Sakit Karyawan

1.7.2 Source Code Ijin dan Sakit Karyawan

```

1. import androidx.appcompat.app.AlertDialog;
2. import androidx.appcompat.app.AppCompatActivity;
3. import androidx.appcompat.widget.SearchView;
4. import androidx.recyclerview.widget.GridLayoutManager;
5. import androidx.recyclerview.widget.RecyclerView;
6.
7. import android.content.Intent;
8. import android.os.Bundle;
9. import android.view.View;
10. import android.view.animation.Animation;
11. import android.view.animation.AnimationUtils;
12. import android.widget.TextView;
13. import android.widget.Toast;
14.

```

```

15. import com.google.android.material.floatingactionbutton.FloatingActionButton;
16. import com.google.firebase.database.DataSnapshot;
17. import com.google.firebase.database.DatabaseError;
18. import com.google.firebase.database.DatabaseReference;
19. import com.google.firebase.database.FirebaseDatabase;
20. import com.google.firebase.database.ValueEventListener;
21.
22. import java.util.ArrayList;
23. import java.util.List;
24.
25. public class RiwayatPizinsakitAdmin extends AppCompatActivity {
26.     String nik, nama, usr, levelpeng;
27.     TextView text;
28.     Animation bottomAnim;
29.     FloatingActionButton fab;
30.     DatabaseReference databaseReference;
31.     ValueEventListener eventListener;
32.     RecyclerView recyclerView;
33.     List<pizinsakitclass> dataList;
34.     MyAdapterISA adapter;
35.     SearchView searchView;
36.
37.     @Override
38.     protected void onCreate(Bundle savedInstanceState) {
39.         super.onCreate(savedInstanceState);
40.         setContentView(R.layout.activity_riwayat_pizinsakit_admin);
41.
42.         recyclerView = findViewById(R.id.recyclerView);
43.         fab = findViewById(R.id.fab);
44.         searchView = findViewById(R.id.search);
45.         searchView.clearFocus();
46.         text = findViewById(R.id.text);
47.         bottomAnim = AnimationUtils.loadAnimation(this, R.anim.bottom_anim);
48.
49.         text.setAnimation(bottomAnim);
50.
51.         GridLayoutManager          gridLayoutManager          =          new
GridLayoutManager(RiwayatPizinsakitAdmin.this, 1);
52.         recyclerView.setLayoutManager(gridLayoutManager);
53.
54.         AlertDialog.Builder          builder          =          new
AlertDialog.Builder(RiwayatPizinsakitAdmin.this);
55.         builder.setCancelable(false);
56.         builder.setView(R.layout.progress_layout);
57.         AlertDialog dialog = builder.create();
58.         dialog.show();
59.
60.         dataList = new ArrayList<>();
61.
62.         adapter = new MyAdapterISA(RiwayatPizinsakitAdmin.this, dataList);
63.         recyclerView.setAdapter(adapter);
64.
65.         databaseReference          =
FirebaseDatabase.getInstance().getReference("Pengajuan Izin dan Sakit");
66.         dialog.show();
67.
68.         eventListener          =          databaseReference.addValueEventListener(new
ValueEventListener() {
69.             @Override
70.             public void onDataChange(DataSnapshot snapshot) {
71.                 dataList.clear();
72.                 for (DataSnapshot itemSnapshot : snapshot.getChildren()) {
73.                     pizinsakitclass          pizinsakitclass          =
itemSnapshot.getValue(pizinsakitclass.class);
74.
75.                     pizinsakitclass.setKey(itemSnapshot.getKey());
76.
77.                     //pizinsakitclass.getKey();

```

```

78.
79.         dataList.add(pizinsakitclass);
80.     }
81.     adapter.notifyDataSetChanged();
82.     dialog.dismiss();
83. }
84.
85.     @Override
86.     public void onCancelled(DatabaseError error) {
87.         dialog.dismiss();
88.     }
89. });
90.
91.     searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
92.         @Override
93.         public boolean onQueryTextSubmit(String query) {
94.             return false;
95.         }
96.
97.         @Override
98.         public boolean onQueryTextChange(String newText) {
99.             searchList(newText);
100.            return true;
101.        }
102.    });
103.
104.     fab.setOnClickListener(new View.OnClickListener() {
105.         @Override
106.         public void onClick(View view) {
107.             Intent intent = new Intent(RiwayatPizinsakitAdmin.this,
108.                 PizinsakitAdmin.class);
109.             startActivity(intent);
110.         }
111.     });
112. }
113.
114.
115.     @Override
116.     protected void onStart() {
117.         super.onStart();
118.         Intent masuk = getIntent();
119.         nik = masuk.getStringExtra("nik");
120.         nama = masuk.getStringExtra("nama");
121.         usr = masuk.getStringExtra("peng");
122.         levelpeng = masuk.getStringExtra("lv");
123.     }
124.
125.     public void searchList(String text) {
126.         ArrayList<pizinsakitclass> searchList = new ArrayList<>();
127.         for (pizinsakitclass pizinsakitclass : dataList) {
128.             if
129.             (pizinsakitclass.getDataKet().toLowerCase().contains(text.toLowerCase())) {
130.                 searchList.add(pizinsakitclass);
131.             } else {
132.                 if
133.                 (pizinsakitclass.getDataTawal().toLowerCase().contains(text.toLowerCase())) {
134.                     searchList.add(pizinsakitclass);
135.                 } else {
136.                     if
137.                     (pizinsakitclass.getDataTakhir().toLowerCase().contains(text.toLowerCase())) {
138.                         searchList.add(pizinsakitclass);
139.                     } else {

```

```

140.         if
141.         (pizinsakitclass.getNama().toLowerCase().contains(text.toLowerCase())) {
142.             searchList.add(pizinsakitclass);
143.         }
144.     }
145. }
146. }
147. }
148.     adapter.searchDataList(searchList);
149. }
150. }

```

1.8 Halaman Cuti Karyawan

1.8.1 Tampilan Cuti Karyawan



Gambar 9 – Cuti Karyawan

1.8.2 Source Code Cuti Karyawan

```

1. import androidx.appcompat.app.AlertDialog;
2. import androidx.appcompat.app.AppCompatActivity;
3. import androidx.appcompat.widget.SearchView;
4. import androidx.recyclerview.widget.GridLayoutManager;
5. import androidx.recyclerview.widget.RecyclerView;
6.
7. import android.content.Intent;
8. import android.os.Bundle;
9. import android.os.Parcelable;
10. import android.view.View;
11. import android.view.animation.Animation;
12. import android.view.animation.AnimationUtils;
13. import android.widget.AdapterView;
14. import android.widget.TextView;
15. import android.widget.Toast;
16.
17. import com.google.android.material.floatingactionbutton.FloatingActionButton;
18. import com.google.firebase.database.DataSnapshot;
19. import com.google.firebase.database.DatabaseError;
20. import com.google.firebase.database.DatabaseReference;
21. import com.google.firebase.database.FirebaseDatabase;

```

```

22. import com.google.firebase.database.ValueEventListener;
23.
24. import java.util.ArrayList;
25. import java.util.List;
26.
27. public class RiwayatCuti extends AppCompatActivity {
28.     String nik, nama, usr, levelpeng;
29.     FloatingActionButton fab;
30.     Animation sideAnim;
31.     TextView text;
32.     DatabaseReference databaseReference;
33.     ValueEventListener eventListener;
34.     RecyclerView recyclerView;
35.     List<pcuticlass> dataList;
36.     MyAdapterCuti adapter;
37.     SearchView searchView;
38.
39.     @Override
40.     protected void onCreate(Bundle savedInstanceState) {
41.         super.onCreate(savedInstanceState);
42.         setContentView(R.layout.activity_riwayat_cuti);
43.
44.         recyclerView = findViewById(R.id.recyclerView);
45.         fab = findViewById(R.id.fab);
46.         searchView = findViewById(R.id.search);
47.         searchView.clearFocus();
48.         sideAnim = AnimationUtils.loadAnimation(this, R.anim.side_anim);
49.         text = findViewById(R.id.text);
50.
51.         text.setAnimation(sideAnim);
52.
53.         GridLayoutManager          gridLayoutManager          =          new
GridLayoutManager(RiwayatCuti.this, 1);
54.         recyclerView.setLayoutManager(gridLayoutManager);
55.
56.         //AlertDialog.Builder          builder          =          new
AlertDialog.Builder(RiwayatPcutiAdmin.this);
57.         //builder.setCancelable(false);
58.         // builder.setView(R.layout.progress_layout);
59.         //AlertDialog dialog = builder.create();
60.         // dialog.show();
61.
62.         searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
63.             @Override
64.             public boolean onQueryTextSubmit(String query) {
65.                 return false;
66.             }
67.
68.             @Override
69.             public boolean onQueryTextChange(String newText) {
70.                 searchList(newText);
71.                 return true;
72.             }
73.         });
74.
75.     }
76.     @Override
77.     protected void onStart() {
78.         super.onStart();
79.         Intent masuk = getIntent();
80.         nik = masuk.getStringExtra("nik");
81.         nama = masuk.getStringExtra("nama");
82.         usr = masuk.getStringExtra("peng");
83.         levelpeng = masuk.getStringExtra("lv");
84.         cuti();
85.     }
86.
87.     private void cuti() {

```



```

88.         if(levelpeng.equals("1")){
89.             getcekcuti();
90.         }
91.     }
92.
93.     private void getcekcuti() {
94.         dataList = new ArrayList<>();
95.         adapter = new MyAdapterCuti(RiwayatCuti.this, dataList);
96.         recyclerView.setAdapter(adapter);
97.         databaseReference =
98.         FirebaseDatabase.getInstance().getReference("Pengajuan Cuti");
99.         //dialog.show();
100.        eventListener = databaseReference
101.            .orderByChild("nik")
102.            .equalTo(usr)
103.            .addValueEventListener(new ValueEventListener() {
104.                @Override
105.                public void onDataChange(DataSnapshot snapshot) {
106.                    dataList.clear();
107.                    //int i = 1;
108.                    for (DataSnapshot itemSnapshot :
109.                        snapshot.getChildren()) {
110.                            pcuticlass = pcuticlass
111.                            itemSnapshot.getValue(pcuticlass.class);
112.                            pcuticlass.setKey(itemSnapshot.getKey());
113.                            //pizinsakitclass.getKey();
114.                            dataList.add(pcuticlass);
115.                        }
116.                        adapter.notifyDataSetChanged();
117.                        //dialog.dismiss();
118.                    }
119.
120.
121.                @Override
122.                public void onCancelled(DatabaseError databaseError) {
123.                    //dialog.dismiss();
124.                }
125.            });
126.    }
127.
128.    public void searchList(String text) {
129.        ArrayList<pcuticlass> searchList = new ArrayList<>();
130.        for (pcuticlass pcuticlass : dataList) {
131.            if
132.            (pcuticlass.getDataAl().toLowerCase().contains(text.toLowerCase())) {
133.                searchList.add(pcuticlass);
134.            } else {
135.                if
136.                (pcuticlass.getDataTawal().toLowerCase().contains(text.toLowerCase())) {
137.                    searchList.add(pcuticlass);
138.                } else {
139.                    if
140.                    (pcuticlass.getDataTakhir().toLowerCase().contains(text.toLowerCase())) {
141.                        searchList.add(pcuticlass);
142.                    } else {
143.                        if
144.                        (pcuticlass.getNama().toLowerCase().contains(text.toLowerCase())) {
145.                            searchList.add(pcuticlass);
146.                        } else {
147.                            if
148.                            (pcuticlass.getStatus().toLowerCase().contains(text.toLowerCase())) {

```

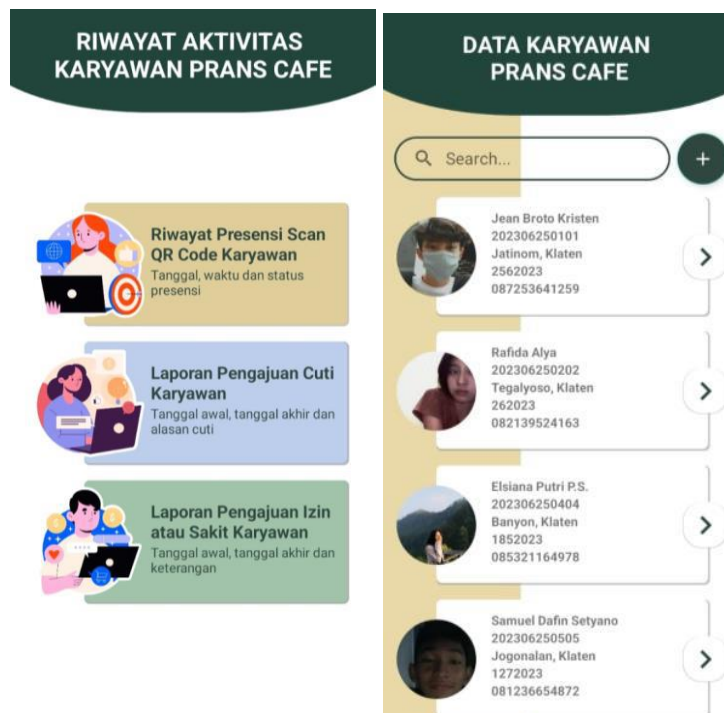
```

147.                                     searchList.add(pcuticlass);
148.                                     }
149.                                 }
150.                             }
151.                         }
152.                     }
153.                 }
154.             }
155.         adapter.searchDataList(searchList);
156.     }
157. }

```

1.9 Halaman Riwayat Admin

1.9.1 Tampilan Riwayat Admin



Gambar 10 – Riwayat Admin

1.9.2 Source Code Riwayat Admin

```

1. import androidx.appcompat.app.AppCompatActivity;
2.
3. import android.app.Activity;
4. import android.content.Context;
5. import android.content.Intent;
6. import android.os.Bundle;
7. import android.os.Parcelable;
8. import android.view.LayoutInflater;
9. import android.view.View;
10. import android.view.ViewGroup;
11. import android.widget.AdapterView;
12. import android.widget.AdapterView;
13. import android.widget.ArrayAdapter;
14. import android.widget.Button;
15. import android.widget.ListView;
16. import android.widget.SearchView;
17. import android.widget.SimpleAdapter;
18. import android.widget.TextView;
19. import android.widget.Toast;

```

```

19.
20. import com.google.android.gms.tasks.OnFailureListener;
21. import com.google.android.gms.tasks.OnSuccessListener;
22. import com.google.android.gms.tasks.Task;
23. import com.google.firebase.database.DataSnapshot;
24. import com.google.firebase.database.DatabaseError;
25. import com.google.firebase.database.DatabaseReference;
26. import com.google.firebase.database.FirebaseDatabase;
27. import com.google.firebase.database.ValueEventListener;
28.
29. import org.checkerframework.checker.units.qual.A;
30.
31. import java.util.ArrayList;
32. import java.util.HashMap;
33. import java.util.List;
34. import java.util.Map;
35.
36. public class RiwayatAdmin extends AppCompatActivity implements
    AdapterView.OnItemClickListener {
37.     String usr, levelpeng;
38.     private String nik;
39.     Activity activity;
40.     ListView listView;
41.     RiwayatAdmin.ListAdapter adapter;
42.     SearchView searchView;
43.     Button btnBack;
44.     DatabaseReference dbm, refdbpeng;
45.     private ArrayList<HashMap<String, String>> list = new ArrayList<>();
46.
47.     @Override
48.     protected void onCreate(Bundle savedInstanceState) {
49.         super.onCreate(savedInstanceState);
50.         setContentView(R.layout.coba_riwayat_admin);
51.
52.         activity = this;
53.         btnBack = (Button) findViewById(R.id.btnBack);
54.
55.         listView = findViewById(R.id.listview);
56.         listView.setOnItemClickListener(this);
57.         listView.setOverScrollMode(View.OVER_SCROLL_NEVER);
58.
59.         dbm = FirebaseDatabase.getInstance().getReference();
60.         refdbpeng = FirebaseDatabase.getInstance().getReference();
61.
62.         btnBack.setOnClickListener(new View.OnClickListener() {
63.             @Override
64.             public void onClick(View view) {
65.                 onBackPressed();
66.             }
67.         });
68.
69.     }
70.
71.
72.     public void presensi() {
73.         if (levelpeng.equals("0")) {
74.             getcekpresensiadmin();
75.             /*}else{
76.                 getcekpresensi();
77.             }*/
78.         }
79.     }
80.
81.     public void getcekpresensiadmin() {
82.         list = new ArrayList<HashMap<String, String>>();
83.         //Toast.makeText(calonrt.this, nort, Toast.LENGTH_SHORT).show();
84.         refdbpeng.child("hasil presensi")
85.             .orderByChild("idpeng")

```

```

86.         .addValueEventListener(new ValueEventListener() {
87.             @Override
88.             public void onDataChange(DataSnapshot dataSnapshot) {
89.                 int i = 1;
90.                 for (DataSnapshot snapshot : dataSnapshot.getChildren())
91.                 {
92.                     String id = (String) snapshot.getKey();
93.                     String idpeng = (String)
94. snapshot.child("idpeng").getValue();
95.                     String nik = (String)
96. snapshot.child("nik").getValue();
97.                     String nama = (String)
98. snapshot.child("nama").getValue();
99.                     String tanggal = (String)
100. snapshot.child("tanggal").getValue();
101.                     String jam = (String)
102. snapshot.child("jam").getValue();
103.                     String status = (String)
104. snapshot.child("status").getValue();
105.
106.                     HashMap<String, String> data = new HashMap<>();
107.                     data.put("key", id);
108.                     data.put("content", "" + i++);
109.
110.                     data.put("idpeng", idpeng);
111.                     data.put("nik", nik);
112.                     data.put("nama", nama);
113.                     data.put("tanggal", tanggal);
114.                     data.put("jam", jam);
115.                     data.put("status", status);
116.
117.                     list.add(data);
118.                 }
119.
120.                 adapter = new RiwayatAdmin.ListAdapter(activity,
121. list,
122. R.layout.awnitem_dish_riwayat_admin, new
123. String[]{"content", "nik", "nama", "tanggal", "jam", "status"},
124. new int[]{R.id.nik, R.id.nama, R.id.tanggal,
125. R.id.jam, R.id.status});
126.
127.                 Parcelable state = listView.onSaveInstanceState();
128.                 listView.setAdapter(adapter);
129.                 listView.onRestoreInstanceState(state);
130.                 adapter.notifyDataSetChanged();
131.             }
132.
133.             @Override
134.             public void onCancelled(DatabaseError databaseError) {
135.             }
136.         });
137.     }
138.
139.     public class ListAdapter extends SimpleAdapter {
140.         private Context mContext;
141.         public LayoutInflater inflater = null;
142.
143.         public ListAdapter(Context context, List<? extends Map<String, ?>>
144. data, int resource, String[] from, int[] to) {
145.             super(context, data, resource, from, to);
146.             mContext = context;
147.             inflater = (LayoutInflater)
148. mContext.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
149.         }
150.
151.         @Override

```

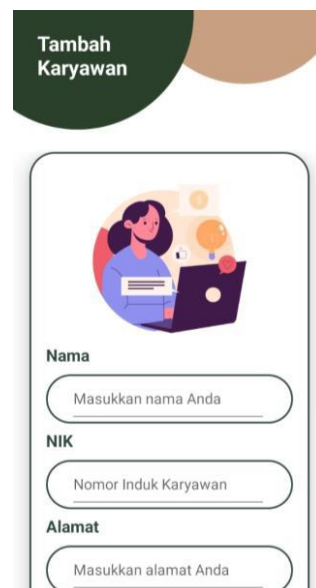
```

142.         public View getView(int position, View convertView, ViewGroup parent)
143.         {
144.             View vi = convertView;
145.             if (convertView == null)
146.                 vi = inflater.inflate(R.layout.awnitem_dish_riwayat_admin,
147.                                     null);
148.             HashMap<String, Object> data = (HashMap<String, Object>)
149.             getItem(position);
150.             final TextView id = vi.findViewById(R.id.id);
151.             final TextView nik = vi.findViewById(R.id.nik);
152.             final TextView nama = vi.findViewById(R.id.nama);
153.             final TextView tanggal = vi.findViewById(R.id.tanggal);
154.             final TextView jam = vi.findViewById(R.id.jam);
155.             final TextView status = vi.findViewById(R.id.status);
156.
157.             final String strid=(String) data.get("id");
158.             final String strnik = (String) data.get("nik");
159.             final String strnama= (String) data.get("nama");
160.             final String strtanggal = (String) data.get("tanggal");
161.             final String strjam=(String) data.get("jam");
162.             final String strstatus=(String) data.get("status");
163.
164.             id.setText(strid);
165.             nik.setText("NIK                      : " + strnik);
166.             nama.setText("Nama                      : " + strnama);
167.             tanggal.setText("Tanggal Presensi : " + strtanggal);
168.             jam.setText("Waktu Presensi      : " + strjam);
169.             status.setText( strstatus);
170.
171.             return vi;
172.         }
173.     }
174.
175.
176.     @Override
177.     protected void onStart() {
178.         super.onStart();
179.         Intent masuk=getIntent();
180.         usr=masuk.getStringExtra("peng");
181.         levelpeng=masuk.getStringExtra("lv");
182.         Toast.makeText(this, levelpeng, Toast.LENGTH_LONG).show();
183.         presensi();
184.     }
185.
186.     @Override
187.     public void onItemClick(AdapterView<?> adapterView, View view, int i, long
188.     l) {
189.     }
190.
191. }

```

1.10 Tambah Karyawan

1.10.1 Halaman Tambah Karyawan



Gambar 11. Tambah Karyawan

1.10.2 Source Code Tambah Karyawan

```
1. import androidx.appcompat.app.AlertDialog;
2. import androidx.appcompat.app.AppCompatActivity;
3. import android.app.DatePickerDialog;
4. import android.content.DialogInterface;
5. import android.content.Intent;
6. import android.graphics.Bitmap;
7. import android.graphics.drawable.BitmapDrawable;
8. import android.net.Uri;
9. import android.os.Bundle;
10. import android.provider.MediaStore;
11. import android.view.View;
12. import android.view.animation.Animation;
13. import android.view.animation.AnimationUtils;
14. import android.widget.AdapterView;
15. import android.widget.Button;
16. import android.widget.DatePicker;
17. import android.widget.EditText;
18. import android.widget.ImageView;
19. import android.widget.Spinner;
20. import android.widget.TextView;
21. import android.widget.Toast;
22. import com.google.android.gms.tasks.OnFailureListener;
23. import com.google.android.gms.tasks.OnSuccessListener;
24. import com.google.firebase.database.DatabaseReference;
25. import com.google.firebase.database.FirebaseDatabase;
26. import com.google.firebase.storage.FirebaseStorage;
27. import com.google.firebase.storage.OnProgressListener;
28. import com.google.firebase.storage.StorageReference;
29. import com.google.firebase.storage.UploadTask;
30. import java.io.ByteArrayOutputStream;
31. import java.util.ArrayList;
32. import java.util.Calendar;
33. import java.util.UUID;
34.
35. public class TambahKaryawan extends AppCompatActivity {
```

```

36.
37.     private DatabaseReference databaseReference;
38.
39.     TextView btncari, btnregister, btnlogin, text;
40.     EditText editregister1, editregister2, editregister3, editregister4,
editregister5, editregister6;
41.     String strnama, strnik, stralamat, strnohp, strtatl, strlv;
42.     DatabaseReference dbm, refdbpeng, dbrefusr;
43.     private ImageView imageContainer;
44.     private StorageReference reference;
45.     Animation sideAnim;
46.     private int tahun, bulan, tanggal;
47.     private static int SPLASH_TIMER = 1500;
48.     private ArrayList<String> arrayList = new ArrayList<>();
49.
50.     private static final int REQUEST_CODE_CAMERA = 1;
51.     private static final int REQUEST_CODE_GALLERY = 2;
52.
53.     @Override
54.     protected void onCreate(Bundle savedInstanceState) {
55.         super.onCreate(savedInstanceState);
56.         setContentView(R.layout.activity_tambah_karyawan);
57.
58.         editregister1 = findViewById(R.id.editregister1);
59.         editregister2 = findViewById(R.id.editregister2);
60.         editregister3 = findViewById(R.id.editregister3);
61.         editregister4 = findViewById(R.id.editregister4);
62.         editregister5 = findViewById(R.id.editregister5);
63.         editregister6 = findViewById(R.id.editregister6);
64.         btncari = findViewById(R.id.btn_cari);
65.         btnregister = findViewById(R.id.btn_register);
66.         btnlogin = findViewById(R.id.btn_login);
67.         imageContainer = findViewById(R.id.imageContainer);
68.         text = findViewById(R.id.text);
69.         sideAnim = AnimationUtils.loadAnimation(this, R.anim.side_anim);
70.         dbm = FirebaseDatabase.getInstance().getReference();
71.         refdbpeng = dbm.child("Karyawan");
72.         dbrefusr = dbm.child("akun");
73.         reference = FirebaseStorage.getInstance().getReference();
74.         text.setAnimation(sideAnim);
75.
76.         editregister5.setOnClickListener(new View.OnClickListener() {
77.             @Override
78.             public void onClick(View v) {
79.                 Calendar calendar = Calendar.getInstance();
80.                 tahun = calendar.get(Calendar.YEAR);
81.                 bulan = calendar.get(Calendar.MONTH);
82.                 tanggal = calendar.get(Calendar.DAY_OF_MONTH);
83.
84.                 DatePickerDialog dialog;
85.                 dialog = new DatePickerDialog(TambahKaryawan.this, new
DatePickerDialog.OnDateSetListener() {
86.                     @Override
87.                     public void onDateSet(DatePicker view, int year, int month,
int dayOfMonth) {
88.                         tahun = year;
89.                         bulan = month+1;
90.                         tanggal = dayOfMonth;
91.
92.                         editregister5.setText(tanggal + "" + bulan + "" + tahun);
93.                     }
94.                 }, tahun, bulan, tanggal);
95.                 dialog.show();
96.             }
97.         });
98.     }
99.
100.     public void uploadImage(View e) {

```

```

101.         strnama = editregister1.getText().toString();
102.         strnik = editregister2.getText().toString();
103.         // strprodi = edprodi.getText().toString();
104.         stralamat = editregister3.getText().toString();
105.         strnohp = editregister4.getText().toString();
106.         strtttl = editregister5.getText().toString();
107.         strlvl = editregister6.getText().toString();
108.
109.
110.         if (strnama.isEmpty() || strnik.isEmpty()) {
111.             Toast.makeText(TambahKaryawan.this, "Semua Data Tidak Boleh
Kosong!", Toast.LENGTH_SHORT).show();
112.
113.         } else {
114.             imageContainer.setDrawingCacheEnabled(true);
115.             imageContainer.buildDrawingCache();
116.             Bitmap bitmap = ((BitmapDrawable)
imageContainer.getDrawable()).getBitmap();
117.             ByteArrayOutputStream stream = new ByteArrayOutputStream();
118.
119.             bitmap.compress(Bitmap.CompressFormat.JPEG, 100, stream);
120.             byte[] bytes = stream.toByteArray();
121.
122.             String namaFile = UUID.randomUUID() + ".jpg";
123.             String pathImage = "gambar/" + namaFile;
124.
125.             UploadTask uploadTask =
reference.child(pathImage).putBytes(bytes);
126.             uploadTask.addOnSuccessListener(new
OnSuccessListener<UploadTask.TaskSnapshot>() {
127.                 @Override
128.                 public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
129.                     Toast.makeText(TambahKaryawan.this, "Uploading Berhasil",
Toast.LENGTH_SHORT).show();
130.                 }
131.             })
132.                 .addOnFailureListener(new OnFailureListener() {
133.                     @Override
134.                     public void onFailure(Exception e) {
135.                         Toast.makeText(TambahKaryawan.this, "Uploading
Gagal", Toast.LENGTH_SHORT).show();
136.                     }
137.                 })
138.                 .addOnProgressListener(new
OnProgressListener<UploadTask.TaskSnapshot>() {
139.                     @Override
140.                     public void onProgress(UploadTask.TaskSnapshot
taskSnapshot) {
141.                         double progress = (100.0 *
taskSnapshot.getBytesTransferred()) / taskSnapshot.getTotalByteCount();
142.                     }
143.                 });
144.             dbrefusr = FirebaseDatabase.getInstance().getReference("Data
Login");
145.             createuser(strnik, strtttl, strlvl);
146.
147.             refdbpeng = FirebaseDatabase.getInstance().getReference("Data
Register");
148.             tambahanggota(strnama, strnik, stralamat, strnohp, strtttl, strlvl,
namaFile);
149.             // Toast.makeText(Register.this, "Username : " + strnim + "dan
Password : " + strtttl, Toast.LENGTH_LONG).show();
150.             bersih();
151.         }
152.     }
153.
154.     public void createuser(String nik, String ttl, String level) {
155.         String key = dbrefusr.push().getKey();

```



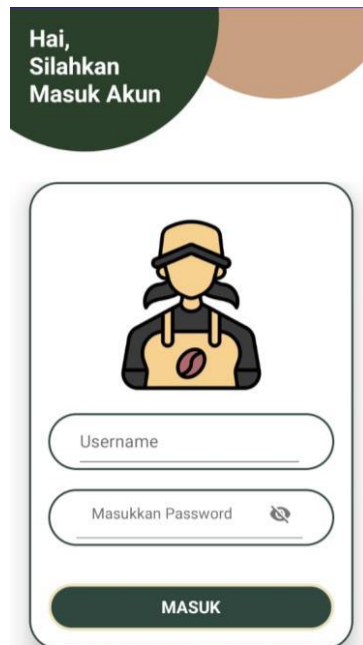
```
216.
217.     public void bersih() {
218.         editregister1.setText("");
219.         editregister2.setText("");
220.         editregister3.setText("");
221.         editregister4.setText("");
222.         editregister5.setText("");
223.         editregister6.setText("");
224.     }
225.
226.     public void tutup(View e) {
227.         onBackPressed();
228.     }
229.
230. }
```

2. PENGGUNAAN PROGRAM

2.1 Hasil Implementasi Antarmuka

2.1.1 Halaman Login

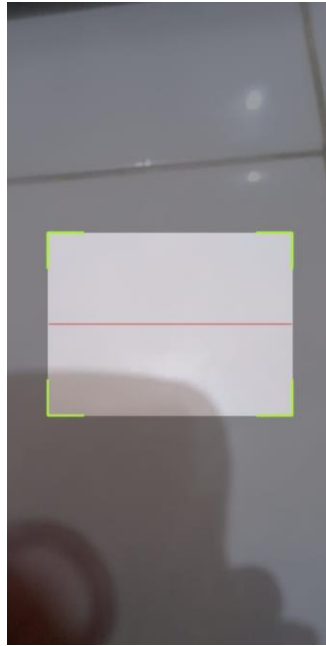
Halaman Masuk akun pada sistem informasi presensi qrcode berbasis android berfungsi sebagai tempat untuk melakukan validasi user sesuai hak aksesnya. Sebelum mengoperasikan aplikasi, halaman masuk akun terdapat pada aplikasi untuk karyawan dan admin. Halaman ini berisi form untuk pengguna melakukan login. Terdapat masukan username dan password yang harus pengguna isi untuk bisa melakukan login. Apabila username dan password yang dimasukan salah, maka user tidak dapat masuk ke dalam sistem dan tidak dapat mengoperasikan aplikasi. Halaman Masuk Akun ditunjukkan pada Gambar 12



Gambar 12 – Halaman Login

2.1.2 Halaman Scan QR Code

Halaman Scan QR Code adalah halaman yang ditampilkan setelah karyawan memilih menu scan pesensi. Halaman ini berfungsi untuk melakukan scan kode presensi yang dibuat oleh admin. Halaman Scan ditunjukkan pada Gambar 13



Gambar 13 – Scan QR Code

2.1.3 Generate QR Code

Halaman Generate QR Code pada sistem informasi presensi qrcode berbasis android berfungsi untuk membuat QR Code presensi dengan memilih tanggal hari ini. Dimana, admin dapat menginputkan tanggal terlebih dahulu dan sistem akan membuat qr code presensi. Halaman Generate QR Code Ditunjukkan pada gambar 14



Gambar 14 - Generate QR Code

2.1.4 Halaman Riwayat Karyawan

Halaman Riwayat Karyawan berisi aktivitas karyawan sesuai akun yang terdiri dari riwayat presensi QR Code, Pengajuan Izin Sakit dan Pengajuan cuti. Halaman Riwayat Karyawan Ditunjukkan pada gambar 15



Gambar 15 – Riwayat Karyawan

2.1.5 Halaman Riwayat Presensi QR Code

Halaman riwayat presensi ini dapat diakses oleh karyawan. Sistem akan menampilkan riwayat presensi sesuai akun. Halaman riwayat presensi ditunjukkan pada Gambar 16.



Gambar 16 - Riwayat Presensi QR Code

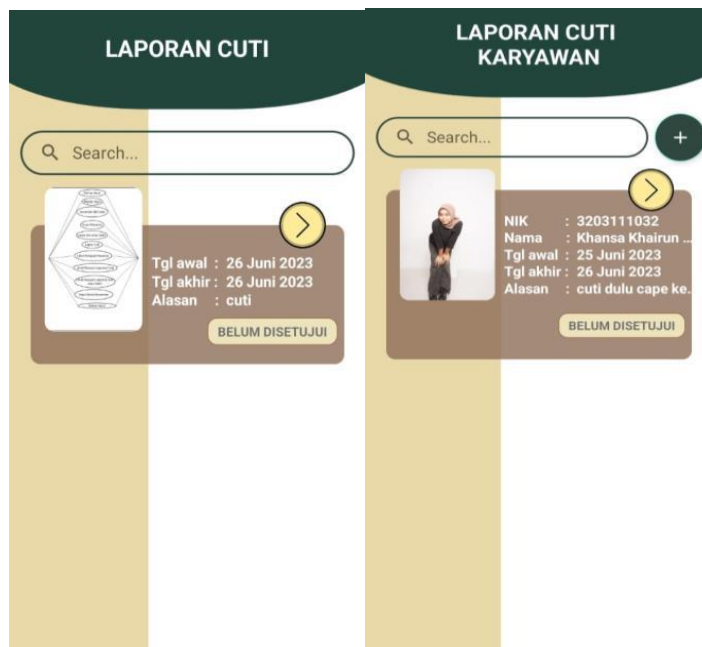
2.1.6 Halaman Pengajuan Ijin Sakit

Halaman koleksi buku tampil ketika *user* memilih menu koleksi buku. Tampilan ini berisi *form* pencarian, daftar kategori, dan *list* koleksi buku semua tipe. Tampilan halaman koleksi buku dapat dilihat pada Gambar 17



Gambar 17 – Pengajuan Ijin Sakit

2.1.7 Halaman Cuti Karyawan



Gambar 18 – Halaman Cuti Karyawan

Halaman riwayat cuti dapat diakses oleh user yaitu karyawan dan admin. Sistem akan menampilkan halaman dashboard, lalu admin dan karyawan bisa melihat riwayat. Didalam

menu Riwayat user dapat memilih menu Riwayat cuti. Untuk karyawan hanya dapat melihat riwayat pengajuan cuti sesuai akun dan untuk Admin dapat melihat riwayat pengajuan cuti dari semua akun karyawan yang telah melakukan presensi. Halaman riwayat pengajuan cuti ditunjukkan pada Gambar 18

2.1.8 Halaman Tambah Karyawan

Halaman Tambah Karyawan Hanya dilakukan oleh admin. Terdapat form yang berisi masukan nomor induk karyawan, nama karyawan, nomor handphone, alamat, tanggal lahir, level, serta harus upload foto profil. Halaman daftar akun ditunjukkan pada Gambar 19



Tambah Karyawan

Nama
Masukkan nama Anda

NIK
Nomor Induk Karyawan

Alamat
Masukkan alamat Anda

Gambar 19– Tambah Karyawan

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini :

N a m a : Khansa Khairun Nisa
Alamat : Metuk Kidul RT 013 RW 05 Tegalyoso Klaten Selatan Jawa Tengah

N a m a : Sutarman M.Kom. Ph.D.
Alamat : Jl. Janti Karangjambe 130-B Banguntapan Bantul D.I.Yogyakarta

N a m a : Murti Retnowo S.Kom., M.Cs.
Alamat : Celeban UH III/505 RT 20 RW 05 Yogyakarta D.I.Yogyakarta

N a m a : Agus Suhendar S.T. M.Eng.
Alamat : Degan II, Banjararum, Kalibawang, Kulonprogo D.I.Yogyakarta

N a m a : Ikrimach S.Kom., M.Cs.
Alamat : Jln Raya Bantul No 211 RT 12 Karang Gondang Pendowo Harjo Sewon Bantul D.I.Yogyakarta

Adalah **Pihak I** selaku pencipta, dengan ini menyerahkan karya ciptaan saya kepada :

N a m a : Universitas Teknologi Yogyakarta
Alamat : Jl. Siliwangi (Ringroad Utara), Jombor, Sleman, D.I.Yogyakarta, 55285

Adalah **Pihak II** selaku Pemegang Hak Cipta berupa Program Komputer yang berjudul Implementasi Presensi Karyawan menggunakan QR Code Berbasis Android (Studi Kasus: Prans Café) untuk didaftarkan di Direktorat Hak Cipta dan Desain Industri, Direktorat Jenderal Kekayaan Intelektual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.

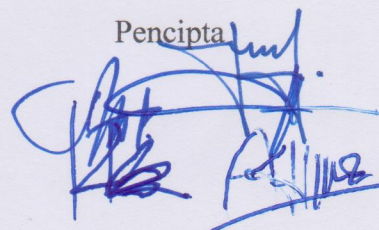
Demikianlah surat pengalihan hak ini kami buat, agar dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 18 Oktober 2023

Pemegang Hak Cipta
Rektor UTY

Dr. Bambang Moertono S, MM., Akt, CA.

Pencipta



1. Khansa Khairun Nisa
2. Sutarman, M.Kom., Ph.D.
3. Murti Retnowo, S.Kom., M.Cs
4. Agus Suhendar S.T., M.Eng.
5. Ikrimach, S.Kom., M.Cs.