

BAB IV

HASIL DAN PEMBAHASAN

1.1. Hasil Pengumpulan Data

Peneliti melakukan Observasi, Wawancara dan Studi Pustaka guna untuk mengumpulkan data. Dari hasil analisis kebutuhan yang peneliti lakukan dijelaskan sebagai berikut :

a. Observasi

Observasi dilakukan dengan melihat proses pengaduan mengenai pelayanan masyarakat tentang aspirasi pelayanan yang dilakukan saat ini di Kantor Desa Sumberanyar Kecamatan Paiton. Dari observasi yang peneliti lakukan, diketahui bahwa pengaduan atau penyampaian mengenai aspirasi yang ada di desa Sumberanyar masih belum terkomputerisasi serta dilakukan dengan cara manual yaitu dengan datang ke kantor desa kemudian menemui kepala desa dan aparatur desa demi mengajukan pengaduan apa saja yang ingin di adukan. Sehingga para warga desa sumberanyar masih kesulitan dengan adanya sistem yang saat ini dilakukan.

b. Wawancara

Wawancara adalah salah satu teknik untuk mengumpulkan data dengan cara bertanya kepada pihak yang berkaitan secara langsung. Wawancara dilaksanakan secara langsung dengan bapak Moh. Toha, S.Pd.I,Mm selaku kepala desa Sumberanyar Kecamatan Paiton. Wawancara dilakukan dengan tujuan agar peneliti mengetahui lebih lanjut tentang alur sistem pengaduan masyarakat yang sedang berjalan di kantor desa Sumberanyar. Dari wawancara yang dilakukan dapat disimpulkan bahwa sistem pengaduan yang dilakukan saat ini masih menggunakan sistem manual dan juga mengenai laporan yang disampaikan oleh setiap aparatur harus mengumpulkan warganya terlebih dahulu agar semua mengetahui tentang aspirasi apa saja yang masyarakat keluhkan. Dengan hasil laporan mengenai aspirasi tentang pelayanan masyarakat yang disampaikan dengan menggunakan surat.

c. Studi Pustaka

Hasil dari studi pustaka yaitu mengumpulkan data serta informasi-informasi tentang teori dan berbagai solusi dari permasalahan yang bersumber pada referensi serta topik yang saat ini diteliti. Data yang akan diperlukan untuk menyelesaikan masalah yang sedang terjadi.

1.2. Hasil Pengembangan Sistem

Setelah melakukan pengumpulan data, peneliti akan melakukan pengembangan terhadap sistem yang ada sebelumnya. Pada penelitian ini menggunakan metode *waterfall* dengan urutan langkah-langkah di mulai dari Analisis, Desain, Pengkodean, Uji Sistem dan *Maintenance*. Adapun hasil dari pengembangan sistem sebagai berikut :

1.2.1. Analisa Kebutuhan

Proses analisis kebutuhan untuk aplikasi yang akan dibuat dilakukan dengan melakukan observasi dan wawancara seperti yang telah dipaparkan sebelumnya. Adapun hasil dari analisis kebutuhan terbagi menjadi dua jenis analisis sistem, berupa analisis sistem lama dan analisis sistem baru

1. Analisis Sistem Lama

Dari hasil pengamatan yang dilakukan di Kantor Desa Sumberanyar Kecamatan Paiton, pada proses penyampaian informasi mengenai data pengaduan masyarakat masih dengan cara yang belum efektif yaitu dengan cara menyampaikan informasi menggunakan surat dan dengan datang langsung ke kantor desa. Berdasarkan hasil wawancara hal tersebut menimbulkan permasalahan seperti warga yang kurang nyaman dengan sistem saat ini yang sedang berjalan, sehingga jika ada proses pengaduan yang belum tersampaikan para masyarakat masih menunggu lama. Hal itu menyebabkan para warga kurang efektif dan efisien waktu dalam menanggapi aspirasinya.

2. Analisa Sistem Baru

Setelah mengetahui kelemahan dari sistem lama yang ada di Kantor Desa Sumberanyar Kecamatan Paiton, maka perlu di buatkan sistem baru yang dapat mempermudah dan mempercepat masyarakat dalam melakukan pengaduan. Dengan adanya sistem baru ini berfungsi untuk

membantu dalam proses penyampaian informasi agar lebih cepat dan pembuatan laporan pengaduan bisa lebih mudah.

Kebutuhan sistem informasi dapat Hasil dari studi pustaka yaitu mengumpulkan data serta informasi tentang landasan teori dan berbagai solusi permasalahan yang menjadikan referensi sebagai sumber serta dengan topik yang sedang diteliti. Data yang akan diperlukan untuk menyelesaikan masalah yang sedang terjadi.

Dapat juga diartikan sebagai kemampuan, syarat atau kriteria yang harus dimiliki oleh sistem informasi, sehingga apa yang diharapkan oleh pengguna dari sistem informasi dapat diwujudkan. Dalam analisis kebutuhan sistem dibagi menjadi 2, yaitu kebutuhan yang fungsional dan kebutuhan yang non fungsional:

a. Kebutuhan fungsional

- Masyarakat dapat melakukan pengaduan.
- Masyarakat dapat melihat riwayat pengaduan yang dilakukan.
- Masyarakat dapat mengetahui pengaduannya sudah ditanggapi atau belum.
- Masyarakat menerima notifikasi jika pengaduannya sudah ditanggapi.

b. Kebutuhan non fungsional

- Aplikasi ini dibuat hanya untuk masyarakat desa sumberanyar kecamatan paiton.
- Aplikasi ini hanya bisa diakses dengan android.

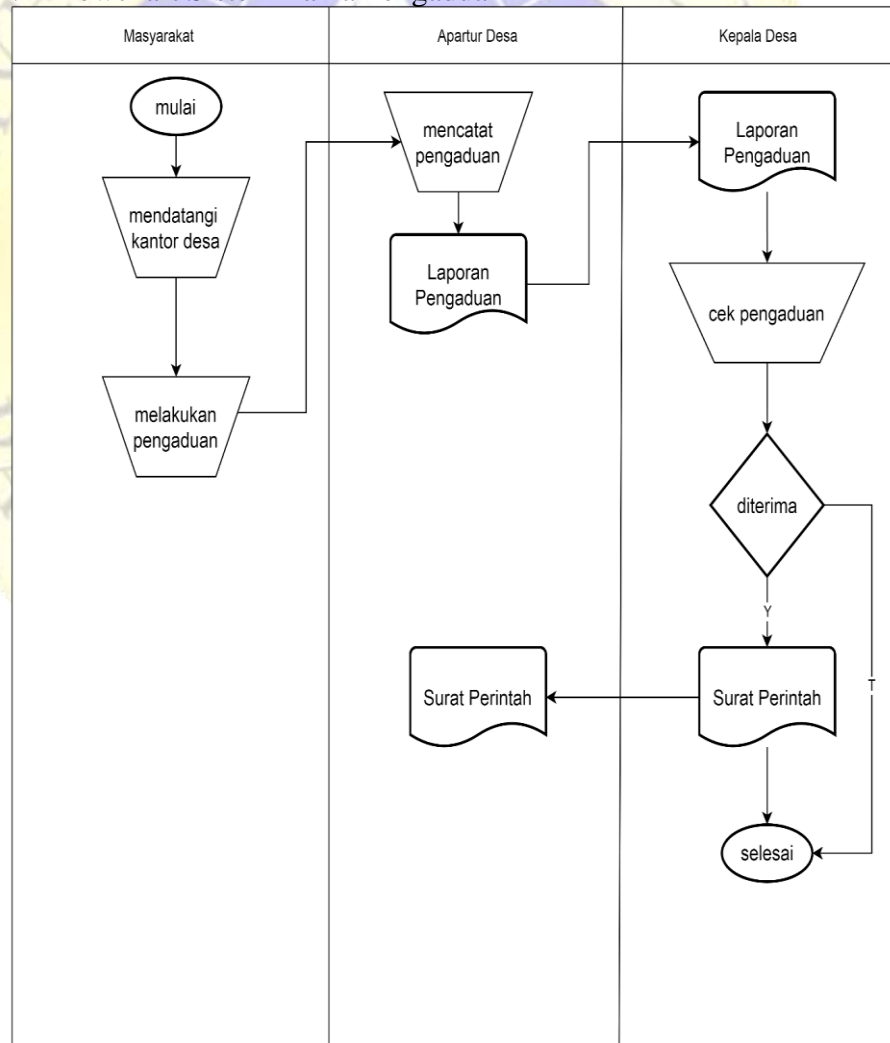
1.2.2. Desain Sistem

Pada tahap ini peneliti membuat rancangan sistem yang akan digunakan dalam menciptakan aplikasi. Yang dijadikan bahan pertimbangan dalam perancangan adalah hasil dari analisa kebutuhan sistem. Tahapan-tahapan desain ini meliputi alur sistem (*Flowchart*), alur data atau *Data Flow Diagram* (DFD), *Entity Relationship Diagram* (ERD), desain table dan desain tampilan aplikasi (*application interface*). Berikut penjelasan dari tahapan-tahapan yang telah disebutkan diatas.

a. Bagan Alur Sitem (Flowchat)

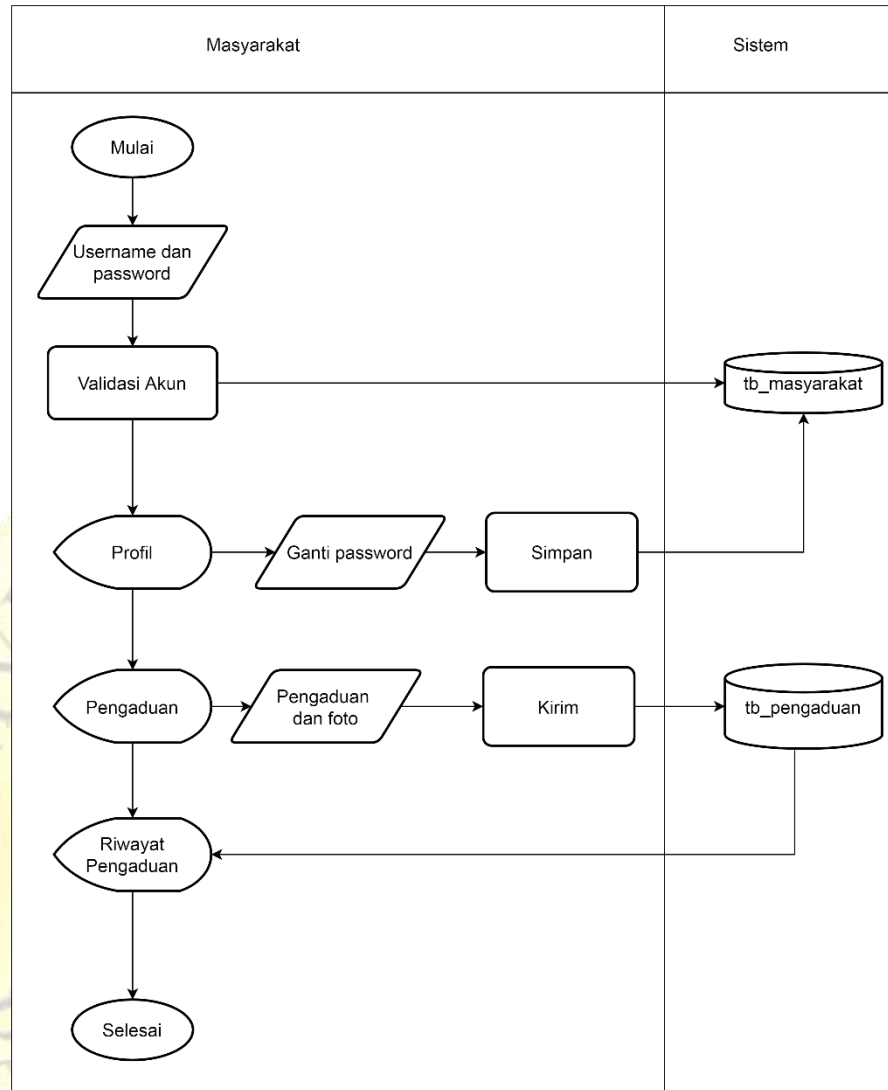
flowchart adalah suatu prosedur atau algoritma yang digunakan untuk menyelesaikan masalah dengan representasi secara simbolik, dalam analisa masalah akan membuat pengguna mudah dalam pengecekan bagian yang terlupakan dengan menggunakan *flowchart*, selain itu *flowchart* juga dijadikan sebagai fasilitas dalam berkomunikasi dalam tim suatu proyek dan antara pemrogram yang bekerja maka dibuatlah *flowchart* sistem yang saat ini digunakan serta *flowchart* sistem yang di usulkan agar perbedaan antara sistem yang sedang berjalan di Desa Sumberanyar Kecamatan Paiton dengan sistem baru yang di usulkan bisa lebih jelas dan lebih spesifik

1. Flowchart Sistem Lama Pengaduan



Gambar 4.1. Flowchart Sistem Lama

2. Flowchart Aplikasi Pengaduan



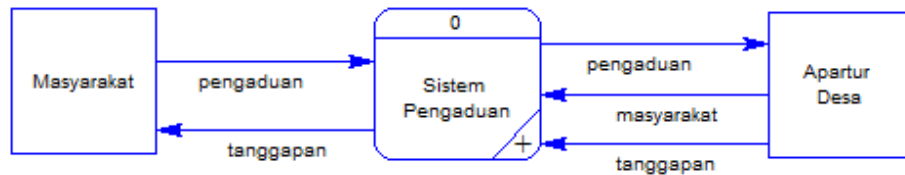
Gambar 4.2. Flowchart Aplikasi Pengaduan

b. Data Flow Diagram (DFD)

Data flow diagram (DFD) adalah suatu bagan alur yang menunjukkan bagian alur data pada sistem. Data flow diagram merupakan representasi dari keterkaitan entitas, jenis data, storage dan proses dalam bentuk sistem yang menggambarkan suatu sistem mulai dari gambaran sistem yang paling global sampai gambaran sistem yang terperinci. Dalam penguraian sistem tersebut dikenal dengan beberapa level diantaranya yaitu :

1. Context Diagram (DFD level 0)

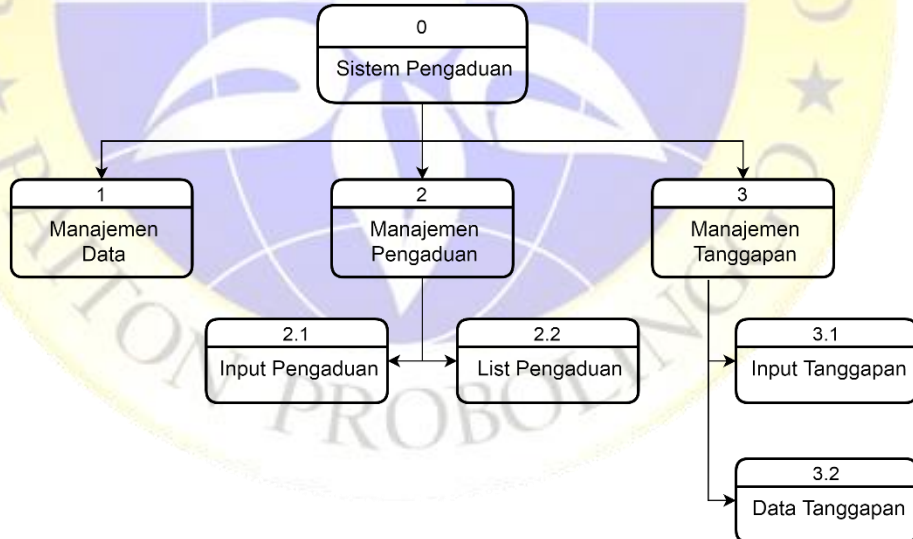
Diagram Context atau kata lain DFD level 0 menunjukkan relasi antara sistem informasi yang ada dengan entitas yang berkaitan secara global. Berikut *context Diagram* dari Aplikasi Monitoring Pelayanan Masyarakat Desa Sumberanyar Kecamatan Paiton.



Gambar 4.3. Diagram Flow Level 0

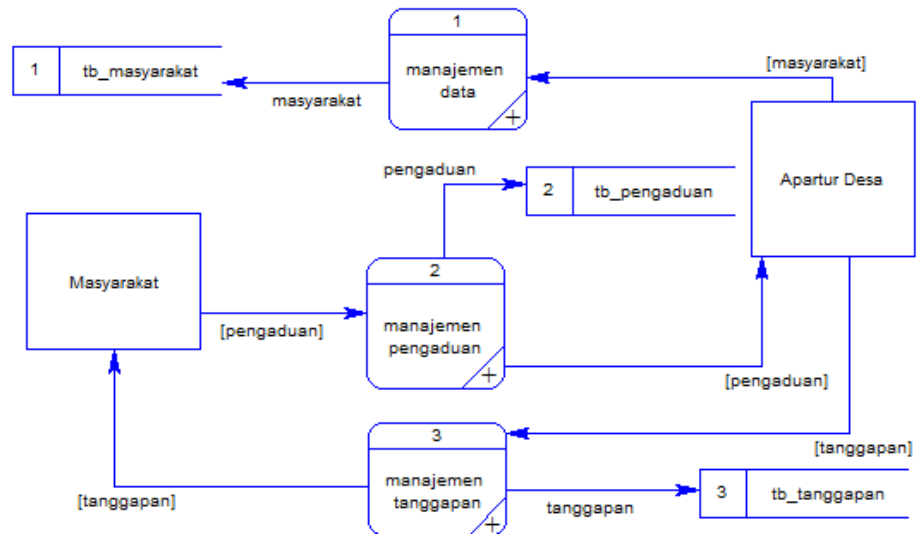
2. Bagan berjenjang

Bagan berjenjang adalah sebuah perancangan dari sistem yang menampilkan proses data yang ada pada suatu sistem aplikasi yang jelas dan terstruktur.



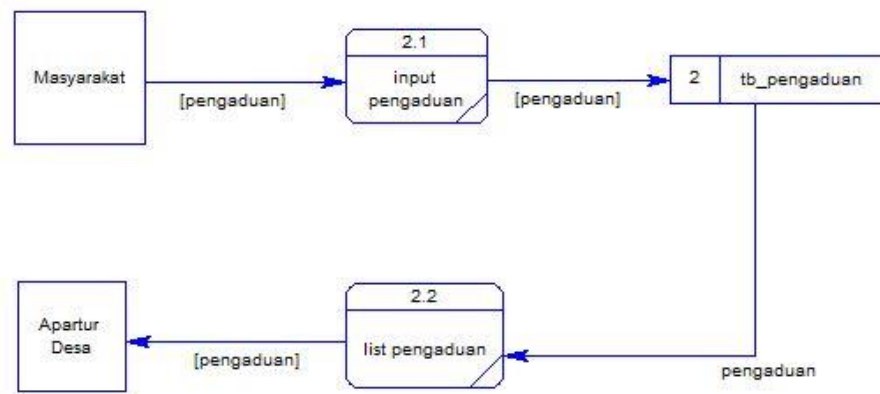
Gambar 4.4. Bagan Berjenjang

3. Data Flow Diagram Level 1



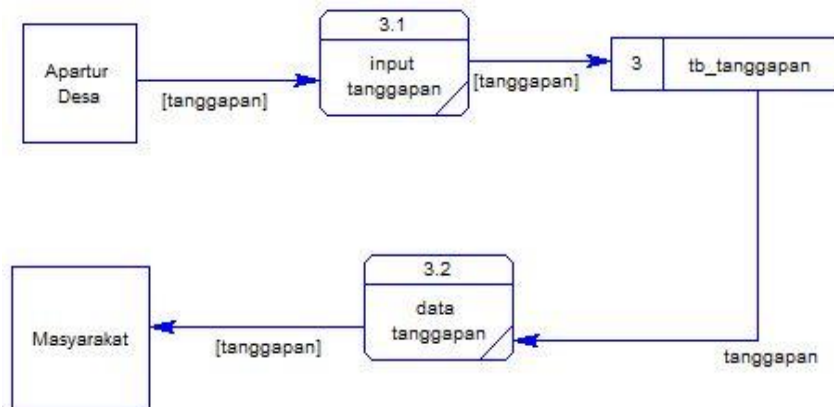
Gambar 4.5. Diagram Flow Level 1

4. Data Flow Diagram Level 2 Proses 2



Gambar 4.6. Diagram Flow Level 2 Proses 2

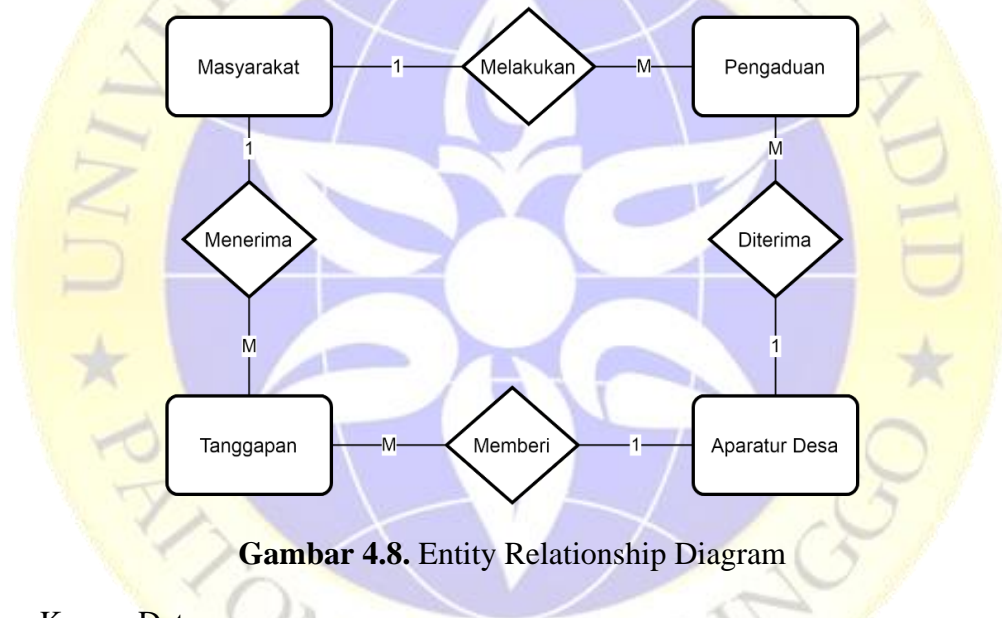
5. Data Flow Diagram Level 2 Proses 3



Gambar 4.7. Diagram Flow Level 2 Proses 3

c. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan gambaran awal dari *database* yang sering digunakan. ERD dikembangkan dari matematika dalam teori himpunan. ERD digunakan dalam pemodelan database relasional. Model ERD berisi komponen entitas dan sekumpulan relasi, setiap komponen terdapat atribut yang mewakili suatu entitas, sehingga bisa diketahui relasi antara entitas dengan atributnya. Selain itu ERD juga mempunyai relasi antar entitas seperti *one to one*, *one to many* dan *many to many*. Serta dilengkapi kamus data, sehingga atribut-atribut yang terlibat dapat diketahui secara langsung dimana atribut yang digaris bawah menjadi kunci atau key).



Gambar 4.8. Entity Relationship Diagram

Kamus Data:

Masyarakat : {nik*, nama, password, tempat_lahir, tanggal_lahir, jenis_kelamin, agama, alamat, pekerjaan, status_perkawinan, token}

Aparatur desa : {nik*, nama_petugas, tempat_lahir, tanggal_lahir, jenis_kelamin, agama, status_perkawinan, pekerjaan, username, password, telp, alamat }

Pengaduan : {id*, id_tanggapan, tgl, nik, nama, isi_laporan, foto, bidang, status}

Tanggapan : {id_tanggapan*, id_pengaduan, isi_tanggapan}

d. Struktur Database

Tahapan desain database merupakan tahap mendesain database hasil perancangan dari entity relationship diagram yang kemudian ditentukan struktur dari tabel berupa kolom, tipe data, kunci utama dan lain-lain. Berikut adalah struktur tabel database yang akan digunakan.

Tabel 4.1. Database Masyarakat

No.	Nama	Tipe Data	Ukuran	Status
1.	Nik	Bigint	16	Primary Key
2.	Nama	Varchar	255	
3.	Password	Varchar	255	
4.	Tempat_lahir	Varchar	50	
5.	Tanggal_lahir	Varchar	20	
6.	Jenis_kelamin	Enum	('Laki-laki','Perempuan')	
7.	Alamat	Text	-	
8.	Agama	Enum	'Islam','Kristen Protestan','Kristen Katolik','Buddha','Hindu','Konghucu'	
9.	Status_Perkawinan	Enum	'Belum Kawin','Kawin','Cerai Hidup','Cerai Mati'	
10.	Pekerjaan	Varchar	50	
11.	Token	Varchar	200	

Tabel 4.2. Database Aparatur Desa

No.	Field Name	Field Type	Field Size	Key
1.	Nik	Bigint	16	Primary Key
2.	Nama_petugas	Varchar	100	
3.	Tempat_lahir	Varchar	50	
4.	Tanggal_lahir	Varchar	20	
5.	Jenis_kelamin	Enum	('Laki-laki','Perempuan')	
6.	Agama	Enum	'Islam','Kristen Protestan','Kristen Katolik','Buddha','Hindu','Konghucu'	
7.	Status_Perkawinan	Enum	'Belum Kawin','Kawin','Cerai Hidup','Cerai Mati'	
8.	Pekerjaan	Varchar	100	
9.	Username	Varchar	20	
10.	Password	Varchar	20	
11.	Telp	Varchar	100	
12.	Alamat	Varchar	100	

Tabel 4.3. Database Pengaduan

No.	Field Name	Field Type	Field Size	Key
1.	Id	Bigint	16	Primary Key
2.	Tanggal	Varchar	50	
3.	Nik	int	20	Foreign Key
4.	Nama	Varchar	50	
5.	Isi laporan	Varchar	500	
6.	Foto	Varchar	255	
7.	Bidang	Varchar	20	
8.	Status	Enum	'Ditanggapi','Proses','Selesai'	
9.	Tanggapan	Varchar	100	

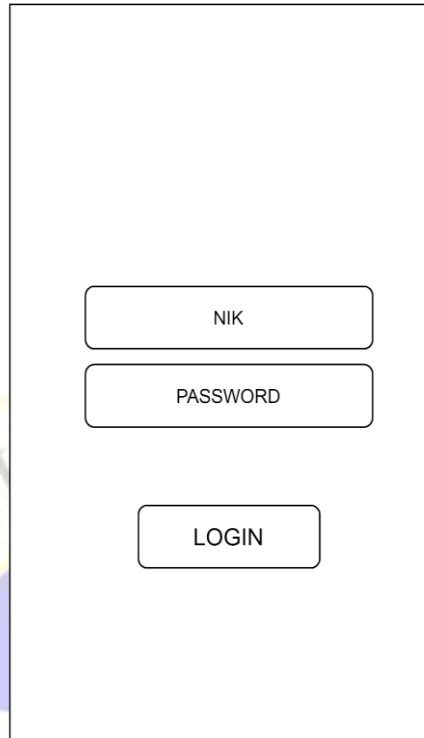
Tabel 4.4. Database Tanggapan

No.	Field Name	Field Type	Field Size	Key
1.	Id_tanggapan	Int	5	Primary Key
2.	Id_pengaduan	int	20	Foreign Key
3.	Isi_anggapan	Varchar	500	

e. Desain Interface

Desain Interface adalah desain tampilan antar muka program aplikasi. Karena masih berupa desain, maka bisa jadi dalam program sebenarnya terdapat penambahan dan pengurangan, akan tetapi penambahan dan pengurangan tersebut tidak akan terlepas dari fungsi pokok masing-masing prosedur. Maka desain interface diklasifikasikan sebagai berikut:

1) Desain Login



A vertical rectangular form with a white background and a black border. It contains three rounded rectangular input fields stacked vertically. The top field is labeled 'NIK', the middle field is labeled 'PASSWORD', and the bottom field is labeled 'LOGIN'.

Gambar 4.9. Desain login

2) Desain Dashboard



A vertical rectangular form with a white background and a black border. At the top right is a rounded rectangular field labeled 'Tanggal'. Below it, centered, are the labels 'Nama Masyarakat' and 'NIK'. At the bottom, there are four rounded rectangular buttons: 'Profil', 'Pengaduan', 'Riwayat Pengaduan', and 'Keluar'.

Gambar 4.10. Desain Dashboard

3) Desain Form Pengaduan

A vertical wireframe for a complaint form. It features a large rounded rectangular input field at the top. Below it is another rounded rectangular input field containing the labels 'Nik', 'Nama', 'Keterangan', and 'Bidang'. At the bottom of the form is a rectangular button labeled 'Kirim'.

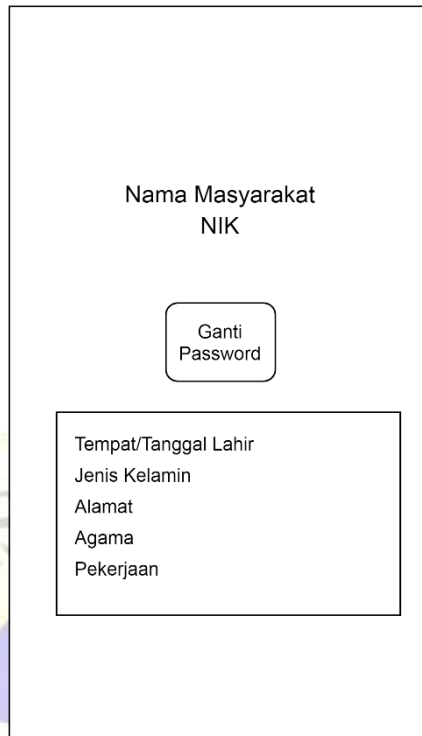
Gambar 4.11. Desain form pengaduan

4) Desain Form Data Masyarakat

A vertical wireframe for a community data form. It is divided into three main sections. The top section is a rounded rectangular box labeled 'Riwayat Pengaduan'. The middle section is a table with three columns: 'Diproses', 'Diterima', and 'Selesai'. The bottom section is a large rounded rectangular box labeled 'Hasil Pengaduan'.

Gambar 4.12. Desain riwayat pengaduan

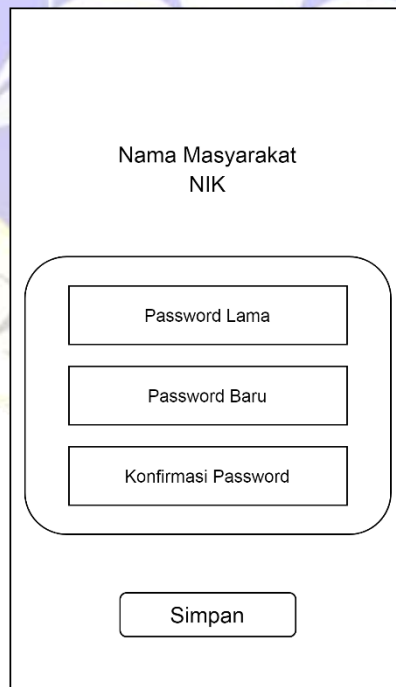
5) Desain Profil



The diagram shows a vertical rectangular profile form. At the top, it contains the text "Nama Masyarakat" and "NIK". Below this is a rounded rectangular button labeled "Ganti Password". Underneath the button is a larger rectangular box containing a list of fields: "Tempat/Tanggal Lahir", "Jenis Kelamin", "Alamat", "Agama", and "Pekerjaan".

Gambar 4.13. Desain profil

6) Desain Form Ganti Password



The diagram shows a vertical rectangular form for changing a password. At the top, it contains the text "Nama Masyarakat" and "NIK". Below this is a large rounded rectangular container that holds three stacked rectangular input fields labeled "Password Lama", "Password Baru", and "Konfirmasi Password". Below this container is a separate rounded rectangular button labeled "Simpan".

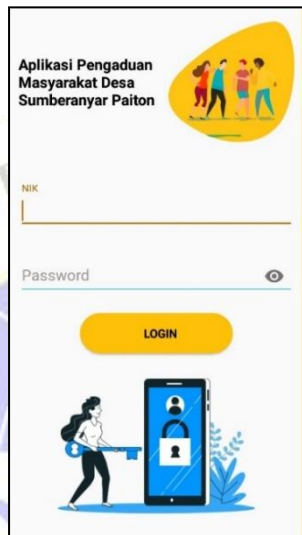
Gambar 4.14. Desain Form Ganti Password

1.2.3. Implementasi

Hasil dari pengkodean dihasilkan aplikasi yang bisa digunakan oleh masyarakat sebagai pengguna, berikut hasil dari pengkodean aplikasi.

1. Form login

Masyarakat yang akan menggunakan aplikasi ini dan harus login terlebih dulu sebelum menggunakan sistem ini.



Gambar 4.15. Screenshot Form login

Segmen Program 4.1. Form login

```
78. StringRequest simpan = new
79. StringRequest(Request.Method.POST, koneksi.login,
80. new Response.Listener<String>() {
81. @Override
82. public void onResponse(String response) { try {
83. if (response.contains("1")) {
84. JSONObject jsonObject =new JSONObject(response);
85. JSONArray hasil=jsonObject.getJSONArray("login");
86. for (int i = 0; i < hasil.length(); i++) {
87. JSONObject c = hasil.getJSONObject(i);
88. String nm = c.getString(koneksi.nama);
89. String id = c.getString(koneksi.nik);
90. String alm = c.getString(koneksi.alamat);
91. startActivity(new Intent(getApplicationContext(),
92. DashboardActivity.class));
93. SharedPreferences pref =
94. getSharedPreferences("akun", MODE_PRIVATE);
95. SharedPreferences.Editor editor = pref.edit();
96. editor.putString(koneksi.nik, id.toString());
97. editor.putString(koneksi.nama, nm.toString());
98. editor.putString(koneksi.alamat, alm.toString());
99. editor.commit();
100. finish();
101. }
102. }elseif (response.contains("0")) {
103. new SweetAlertDialog(LoginActivity.this,
104. SweetAlertDialog.ERROR_TYPE)
```

Segmen Program 4.2. Lanjutan form login

```
101. .setTitleText("Oops...")
102. .setContentText("NIK atau Password anda salah!")
103. .show();
104. }
105. }catch (JSONException e) {
106. e.printStackTrace();
107. }
108. }, new Response.ErrorListener() {
109. @Override
110. public void onErrorResponse(VolleyError error) {
111. Log.e("Pengaduan", "onErrorResponse: " +
    error.getMessage());
112. }
113. }) {
114. @Override
115. protected Map<String, String> getParams() throws
    AuthFailureError {
116. Map<String, String> param = new HashMap<>();
117. param.put(koneksi.nik, ur);
118. param.put(koneksi.password, pw);
119. param.put("token", token);
120. return param;
121. }
122. };
123. RequestQueue requestQueue =
    Volley.newRequestQueue(getApplicationContext());
124. requestQueue.add(simpan);
```

2. Dashboard

Dashboard adalah adalah form yang menampilkan fitur-fitur yang ada dalam sistem ini.



Gambar 4.16. Screenshot Dashboard

Segmen Program 4.3. Dashboard

```
1. case R.id.cv_profil:
2.     Intent pro=new Intent (DashboardActivity.this,
3.     ProfilActivity.class);
4.     startActivity (pro);
5.     break;
6. case R.id.cv_pengaduan:
7.     Intent pen=new Intent (DashboardActivity.this,
8.     PengaduanActivity.class);
9.     pen.putExtra (koneksi.tgl, tgl1);
10.    startActivity (pen);
11.    break;
12. case R.id.cv_history:
13.    Intent his=new Intent (DashboardActivity.this,
14.    HistoryActivity.class);
15.    startActivity (his);
16.    break;
17. case R.id.cv_keluar:
18.    Intent keluar = new Intent (DashboardActivity.this,
19.    LoginActivity.class);
20.    startActivity (keluar);
21.    SharedPreferences pref =
22.    getSharedPreferences ("akun",MODE_PRIVATE);
23.    SharedPreferences.Editor editor = pref.edit ();
24.    editor.putString (koneksi.nik, "0");
25.    editor.commit ();
26.    finish ();
27.    break;
```

3. Profil

Profil adalah layout yang akan menampilkan data dari masyarakat sesuai login dari masyarakat. Data yang telah tersimpan di database akan di tampilkan pada menu profil.



Gambar 4.17. Screenshot Profil

Segmen Program 4.4. Profil

```
125. final StringRequest list = new
    StringRequest(Request.Method.POST, koneksi.profil,
126. new Response.Listener<String>() {
127. @Override
128. public void onResponse(String response) {
129. Log.e("complaint", "onResponse: "+response );
130. try {
131. JSONObject jsonObject =new JSONObject(response);
132. JSONArray hasil=jsonObject.getJSONArray("Hasil");
133. for (int i = 0; i <hasil.length() ; i++) {
134.     JSONObject c=hasil.getJSONObject(i);
135.     String tgl=c.getString(koneksi.tanggal);
136.     String tempat=c.getString(koneksi.tempat);
137.     String jk=c.getString(koneksi.jenis_kelamin);
138.     String agama=c.getString(koneksi.agama);
139.     String pekerjaan=c.getString(koneksi.pekerjaan);
140.     String alamat=c.getString(koneksi.alamat);
141.     if (jk.equals("L")){
142.         tjk.setText("Laki-Laki");
143.     }else if(jk.equals("P")){
144.         tjk.setText("Perempuan");
145.     }
146.     String ttl=tempat+", "+tgl;
147.     ttgl.setText(ttl);
148.     tagama.setText(agama);
149.     tpekerjaan.setText(pekerjaan);
150.     talamat.setText(alamat);
151.     }
152. }catch (JSONException e) {
153.     e.printStackTrace();
154. }
155. }
156. }, new Response.ErrorListener() {
157. @Override
158. public void onErrorResponse(VolleyError error) {
159. Log.e("complain", "onErrorResponse:
    "+error.getMessage() );
160. }
161. }){
162. @Override
163. protected Map<String, String> getParams() throws
    AuthFailureError {
164. Map<String, String> param=new HashMap<>();
165. param.put("nik",nik_);
166. return param;
167. }
168. };
169. RequestQueue requestQueue=
    Volley.newRequestQueue(getApplicationContext());
170. requestQueue.add(list);
```

4. Form Pengaduan

Form pengaduan digunakan masyarakat untuk melakukan pengaduan atau aspirasi ke desa sumberanyar kecamatan paiton.



Gambar 4.18. Screenshot form Pengaduan

Segmen Program 4.5. Form pengaduan

```

142. final ProgressDialog progressDialog = new
    ProgressDialog(this);
143. progressDialog.setMessage("Mengirim data. . .");
144. progressDialog.show();
145. String bid=bidang.getSelectedItem().toString();
146. StringRequest simpan = new
    StringRequest(Request.Method.POST,
    koneksi.pengaduan,
147. new Response.Listener<String>() {
148. @Override
149. public void onResponse(String response) {
150. progressDialog.dismiss();
151. Log.e("pengaduan", "onResponse: " + response);
152. new SweetAlertDialog(PengaduanActivity.this,
    SweetAlertDialog.SUCCESS_TYPE)
153. .setTitleText("Data Terkirim")
154. .setContentText("Pengaduan Anda Telah Terkirim!")
155. .setConfirmText("Ok")
156. .setConfirmClickListener(new
    SweetAlertDialog.OnSweetClickListener() {
157. @Override
158. public void onClick(SweetAlertDialog sDialog) {
159. Intent hc=new Intent(PengaduanActivity.this,
    DashboardActivity.class);
160. startActivity(hc);
161. finish();
162. }
163. })
164. .show();
165. }
166. }, new Response.ErrorListener() {
167. @Override
168. public void onErrorResponse(VolleyError error) {

```

Segmen Program 4.6. Lanjutan Form pengaduan

```
169. Log.e("Pengaduan", "onErrorResponse: " +
    error.getMessage());
170. }
171. }) {
172. @Override
173. protected Map<String, String> getParams() throws
    AuthFailureError {
174.     Map<String, String> param = new HashMap<>();
175.     param.put(koneksi.nik, nik_);
176.     param.put(koneksi.nama, nama_);
177.     param.put(koneksi.tgl, date);
178.     param.put(koneksi.alamat, alamat_);
179.     param.put(koneksi.isi,
        keterangan.getText().toString());
180.     param.put(koneksi.bidang, bid);
181.     param.put(koneksi.gambar, imageToString(bitmap));
182.     return param;
183. }
184. };
185. RequestQueue requestQueue =
    Volley.newRequestQueue(getApplicationContext());
186. requestQueue.add(simpan);
```

5. Riwayat Pengaduan

Riwayat pengaduan digunakan oleh masyarakat untuk melihat hasil dari pengaduan yang dilakukan.



Gambar 4.19. Screenshot Riwayat Pengaduan

Segmen Program 4.7. Riwayat Pengaduan

```
67. final StringRequest list = new
    StringRequest(Request.Method.POST,
    koneksi.list_proses,
68. new Response.Listener<String>() {
69. @Override
```

Segmen Program 4.8. Lanjutan Riwayat Pengaduan

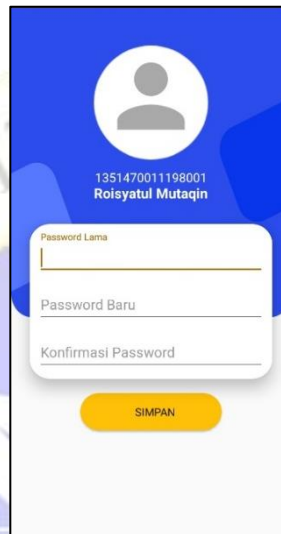
```
70. public void onResponse(String response) {
71.     Log.e("complaint", "onResponse: " + response);
72.     try {
73.         JSONObject jsonObject = new JSONObject(response);
74.         JSONArray hasil = jsonObject.getJSONArray("Hasil");
75.         tampil.clear();
76.         for (int i = 0; i < hasil.length(); i++) {
77.             JSONObject c = hasil.getJSONObject(i);
78.             String tgl = c.getString(koneksi.tgl);
79.             String ket = c.getString(koneksi.isi);
80.             String kategori = c.getString(koneksi.bidang);
81.             String status = c.getString(koneksi.status);
82.             String gambar = c.getString(koneksi.gambar);
83.             String tanggapan = c.getString(koneksi.tanggapan);
84.             if
85.                 (c.getString(koneksi.status).equals("Ditanggapi")) {
86.                 warna = "#B3AF1A1A";
87.             } else if
88.                 (c.getString(koneksi.status).equals("Proses")) {
89.                 warna = "#B303A9F4";
90.             } else if
91.                 (c.getString(koneksi.status).equals("Selesai")) {
92.                 warna = "#FF9800";
93.             }
94.             HashMap<String, String> map = new HashMap<String,
95.                 String>();
96.             map.put(koneksi.tanggal, tgl);
97.             map.put(koneksi.isi, ket);
98.             map.put(koneksi.bidang, kategori);
99.             map.put(koneksi.status, status);
100.            map.put(koneksi.gambar, gambar);
101.            map.put(koneksi.tanggapan, tanggapan);
102.            map.put("warna", warna);
103.            tampil.add(map);
104.        }
105.        adapter = new adapter(getActivity(), tampil);
106.        rv.setAdapter(adapter);
107.    } catch (JSONException e) {
108.        e.printStackTrace();
109.    }
110. }, new Response.ErrorListener() {
111.     @Override
112.     public void onErrorResponse(VolleyError error) {
113.         Log.e("complain", "onErrorResponse: " +
114.             error.getMessage());
115.     }
116. }) {
117.     @Override
118.     protected Map<String, String> getParams() throws
119.         AuthFailureError {
120.         Map<String, String> param = new HashMap<>();
121.         param.put(koneksi.nik, nik_.toString());
122.         return param;
123.     }
124. };
```

Segmen Program 4.9. Lanjutan 2 Riwayat Pengaduan

```
120. RequestQueue requestQueue =  
    Volley.newRequestQueue (getActivity());  
121. requestQueue.add(list);
```

6. Form Password

Form gani password bisa masyarakat gunakan untuk mengganti password akun, agar tidak sembarangan orang bisa masuk menggunakan akun orang lain.



Gambar 4.20. Screenshot Form Ganti Password

Segmen Program 4.10. Form ganti password

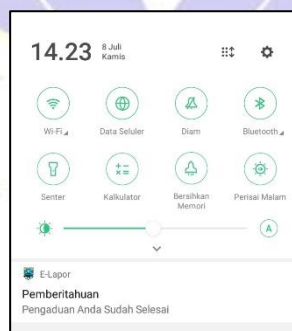
```
71. String lama=plama.getText().toString();  
72. String baru=pbaru.getText().toString();  
73. String konfirm=pkonf.getText().toString();  
74. if (lama.equals("")) {  
75.     plama.setError("Belum diisi");  
76.     plama.requestFocus();  
77. } else if (baru.equals("")) {  
78.     pbaru.setError("Belum diisi");  
79.     pbaru.requestFocus();  
80. } else if (konfirm.equals("")) {  
81.     pkonf.setError("Belum diisi");  
82.     pkonf.requestFocus();  
83. }else if (baru.equals(konfirm)){  
84.     StringRequest simpan = new  
StringRequest(Request.Method.POST, koneksi.gpass,  
85.     new Response.Listener<String>() {  
86.     @Override  
87.     public void onResponse(String response) {  
88.     Log.e("pengaduan", "onResponse: " + response);  
89.     new SweetAlertDialog(GantipassActivity.this,  
SweetAlertDialog.SUCCESS_TYPE)  
90.     .setTitleText("Berhasil")  
91.     .setContentText("Password Telah diganti!")  
92.     .setConfirmText("Ok")
```

Segmen Program 4.11. Lanjutan Form ganti password

```
93.     .setConfirmClickListener(new
SweetAlertDialog.OnSweetClickListener() {
94.     @Override
95.     public void onClick(SweetAlertDialog sDialog) {
96.     Intent hc = new Intent(GantipassActivity.this,
DashboardActivity.class);
97.     startActivity(hc);
98.     }
99.     })
100. .show();
101. }
102. }, new Response.ErrorListener() {
103. @Override
104. public void onResponse(VolleyError error) {
105. Log.e("Pengaduan", "onErrorResponse: " +
error.getMessage());
106. }
107. }) {
108. @Override
109. protected Map<String, String> getParams() throws
AuthFailureError {
110.     Map<String, String> param = new HashMap<>();
111.     param.put(koneksi.nik, nik_);
112.     param.put(koneksi.plama, lama);
113.     param.put(koneksi.pbaru, baru);
114.     return param;
115. }
116. };
117. RequestQueue requestQueue =
Volley.newRequestQueue(getApplicationContext());
118. requestQueue.add(simpan);
```

7. Notifikasi

Notifikasi akan muncul ketika aparture desa menanggapi atau menyelesaikan pengaduan yang dilakukan masyarakat.



Gambar 4.21. Screenshot notifikasi

Segmen Program 4.12. PHP notifikasi

```
66. $notif = $this->M_all->tampilnotif($id);
67. foreach ($notif->result_array() as $pp) {
68.     $tokens[]=$pp['token'];
69. }
70. $url = 'https://fcm.googleapis.com/fcm/send';
```

Segmen Program 4.13. Lanjutan PHP notifikasi

```
71. $msg = array
72. (
73. 'body' => "Pengaduan Anda Sudah Ditanggapi",
74. 'title' => "Pemberitahuan",
75. 'sound' => 'default'
76. );
77. $fields = array
78. (
79. 'registration_ids' => $tokens,
80. 'notification' => $msg
81. );
82. $headers = array(
83. 'Authorization:key = AAAAuG7-
HiI:APA91bG8CL8OtAIn1SWuokKM5NrC38BhPqq1p3KYm1ksoVZS
6ZKFUBRhBpLilidXeDq8pShf039skiqaB2DFsyN4dDrpgKYhHANK
eNPctNKZv2uwXlkmoXefjIzOU7SIIIt4XHD_QbAGb',
84. 'Content-Type: application/json'
85. );
86. $ch = curl_init();
87. curl_setopt($ch, CURLOPT_URL, $url);
88. curl_setopt($ch, CURLOPT_POST, true);
89. curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
90. curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
91. curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
92. curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
93. curl_setopt($ch, CURLOPT_POSTFIELDS,
json_encode($fields));
94. $result = curl_exec($ch);
95. if ($result === FALSE) {
96. die('Curl failed: ' . curl_error($ch));
97. }
98. curl_close($ch);
99. echo "<script>";
100. echo "alert('Pengaduan Berhasil Ditanggapi')";
101. echo "</script>";
102. redirect('kelola','refresh');
103. return $result;
```

1.2.4. Hasil Uji

Pengujian aplikasi yang dilakukan oleh peneliti ada 2, yaitu pengujian blackbox atau internal dan pengujian eksternal yang menggunakan kuesioner sebagai media untuk pengujian. Berikut hasil pengujian internal dan eksternal.

1. Pengujian Black Box

Tabel 4.5 merupakan hasil pengujian internal *Black Box* yang diuji oleh langsung oleh dosen Universitas Nurul Jadid, bapak Fathur Rizal, M.Kom selaku dosen mata kuliah android.

Tabel 4.5. Pengujian Internal Blackbox

No.	Unit Uji	Masukan/ Kondisi	Hasil yang diharapkan	Kesesuaian	
				Ya	Tidak
1	Login	Akses untuk menuju ke dashboard	- Jika nik dan password sesuai maka program akan melanjutkan ke menu utama.	✓	
			- Jika tidak sesuai maka program tidak akan melanjutkan ke menu utama.	✓	
2	Dashbo ard	Profil	- Menampilkan data masyarakat yang berhasil login	✓	
		Pengaduan	- Menampilkan form pengaduan yang akan dikirim ke aparature desa.	✓	
		Hisroty	- Menampilkan hasil pengaduan dari masyarakat	✓	
3	Profil	Ganti Password	- Menampilkan Form untuk mengganti password	✓	
4	Log Out	Akses untuk menutup program	- Dapat menutup program (kembali ke menu login)	✓	

2. Hasil Kuesioner

Pada pengujian eksternal yang dilakukan dengan cara membagikan kuesioner yang di ujikan langsung kepada masyarakat desa sumberanyar kecamatan paiton. Peneliti perhitungan dengan skala likert yang digunakan untuk mengukur persepsi, sikap atau pendapat masyarakat melalui angket dan sering digunakan dalam riset yang berupa survei. Pada pengujian Aplikasi Sistem Informasi Pengaduan Masyarakat Desa Sumberanyar Kecamatan Paiton dilakukan kepada 15 masyarakat desa sumberanyar. Tabel 4.6 merupakan hasil kuesioner yang telah peneliti lakukan :

Tabel 4.6. Kuesioner

No.	Pertanyaan	Hasil Uji			
		SB	B	TB	STB
1	Apakah aplikasi ini dibutuhkan ?	9	4	2	-
2	Apakah aplikasi pengaduan ini bisa mempermudah dalam melakukan proses pengaduan ke desa?	12	3	-	-
3	Apakah aplikasi ini memberikan informasi kepada masyarakat tentang pengaduan yang dilakukan secara efektif dan efisien?	8	4	3	-
4	Apakah aplikasi pengaduan ini membantu masyarakat desa sumberanyar ?	8	5	2	-
5	Apakah aplikasi pengaduan ini sudah layak untuk digunakan?	11	3	1	-
6	Apakah desain tampilan aplikasi dapat digunakan dengan mudah (user friendly) atau tidak?	12	-	3	-
7	Apakah fungsi-fungsi yang digunakan sudah sesuai dengan kebutuhan atau tidak?	12	2	1	-

Seperti yang sudah dijelaskan pada BAB III pada poin pengujian eksternal yang menggunakan skala likert, berikut adalah hasil perhitungan hasilnya.

1. Bobot nilai setiap skala

Tabel 4. 7. keterangan dan bobot nilai

No	Keterangan	Bobot Nilai
1	SB = Sangat Baik	4
2	B = Baik	3
3	TB = Tidak Baik	2
4	STB = Sangat Tidak Baik	1

2. Nilai I (Rentang jarak) dan interpretasi persen

$$I = 100 / \text{jumlah likert}$$

$$I = 100 / 4 = 25$$

Jadi, nilai interval jarak yaitu 0% (terendah) hingga 100% (tertinggi) adalah sebanyak 25.

Tabel 4.8. Persentase skor interval

No	Keterangan	Persentase penilaian
1	STB = Sangat Tidak Baik	0% - 25%
2	TB = Tidak Baik	26% - 50%
3	B = Baik	51% - 75%
4	SB = Sangat Baik	76% - 100%

3. Penghitungan hasil jawaban responden

Y (skor tertinggi) dan X (skor terendah), untuk item penilaian sebagai berikut:

$$\begin{aligned} Y &= \text{skor tertinggi likert} \times \text{jumlah responden} \\ &= 4 \times 15 = 60 \end{aligned}$$

$$\begin{aligned} X &= \text{skor terendah likert} \times \text{jumlah responden} \\ &= 1 \times 15 = 15 \end{aligned}$$

• **Pertanyaan 1**

1. Menjawab SB = $9 \times 4 = 36$

2. Menjawab B = $4 \times 3 = 12$

3. Menjawab TB = $2 \times 2 = 4$

4. Menjawab STB = $0 \times 1 = 0$

$$\text{Total} = 36 + 12 + 4 + 0 + 0 = 52$$

• **Pertanyaan 2**

1. Menjawab SB = $12 \times 4 = 48$

2. Menjawab B = $3 \times 3 = 9$

3. Menjawab TB = $0 \times 2 = 0$

4. Menjawab STB = $0 \times 1 = 0$

$$\text{Total} = 48 + 9 + 0 + 0 + 0 = 57$$

- Pertanyaan 3

1. Menjawab SB = $8 \times 4 = 32$

2. Menjawab B = $4 \times 3 = 12$

3. Menjawab TB = $3 \times 2 = 6$

4. Menjawab STB = $0 \times 1 = 0$

$$\text{Total} = 32 + 12 + 6 + 0 + 0 = 50$$

- Pertanyaan 4

1. Menjawab SB = $8 \times 4 = 32$

2. Menjawab B = $5 \times 3 = 15$

3. Menjawab TB = $2 \times 2 = 4$

4. Menjawab STB = $0 \times 1 = 0$

$$\text{Total} = 32 + 15 + 4 + 0 + 0 = 51$$

- Pertanyaan 5

1. Menjawab SB = $11 \times 4 = 44$

2. Menjawab B = $3 \times 3 = 9$

3. Menjawab TB = $1 \times 2 = 2$

4. Menjawab STB = $0 \times 1 = 0$

$$\text{Total} = 44 + 9 + 2 + 0 + 0 = 55$$

- Pertanyaan 6

1. Menjawab SB = $12 \times 4 = 48$

2. Menjawab B = $0 \times 3 = 0$

3. Menjawab TB = $3 \times 2 = 6$

4. Menjawab STB = $0 \times 1 = 0$

$$\text{Total} = 48 + 0 + 6 + 0 + 0 = 54$$

- Pertanyaan 7

1. Menjawab SB = $12 \times 4 = 48$

2. Menjawab B = $2 \times 3 = 6$

3. Menjawab TB = $1 \times 2 = 2$

4. Menjawab STB = $0 \times 1 = 0$

$$\text{Total} = 48 + 6 + 2 + 0 + 0 = 56$$

4. Presentase jawaban responden

- Pertanyaan 1

$$\text{Total skor} / Y \times 100 = 52 / 60 \times 100 = 86,67\%$$

Dari perhitungan diatas, dapat disimpulkan sebanyak 86,67% dari 15 responden menyatakan sangat baik.

- Pertanyaan 2

$$\text{Total skor} / Y \times 100 = 57 / 60 \times 100 = 95\%$$

Dari perhitungan diatas, dapat disimpulkan sebanyak 95% dari 15 responden menyatakan sangat baik.

- Pertanyaan 3

$$\text{Total skor} / Y \times 100 = 50 / 60 \times 100 = 83,33\%$$

Dari perhitungan diatas, dapat disimpulkan sebanyak 83,33% dari 15 responden menyatakan sangat baik.

- Pertanyaan 4

$$\text{Total skor} / Y \times 100 = 51 / 60 \times 100 = 85\%$$

Dari perhitungan diatas, dapat disimpulkan sebanyak 85% dari 15 responden menyatakan sangat baik.

- Pertanyaan 5

$$\text{Total skor} / Y \times 100 = 55 / 60 \times 100 = 91,67\%$$

Dari perhitungan diatas, dapat disimpulkan sebanyak 91,67% dari 15 responden menyatakan sangat baik.

- Pertanyaan 6

$$\text{Total skor} / Y \times 100 = 54 / 60 \times 100 = 90\%$$

Dari perhitungan diatas, dapat disimpulkan sebanyak 90% dari 15 responden menyatakan sangat baik.

- Pertanyaan 7

$$\text{Total skor} / Y \times 100 = 56 / 60 \times 100 = 93,33\%$$

Dari perhitungan diatas, dapat disimpulkan sebanyak 93,33% dari 15 responden menyatakan sangat baik.

Dari pengujian yang dilakukan dapat disimpulkan bahwa persentase tertinggi diperoleh dari pertanyaan ke-2 sebesar 95% dari 15 penguji menyatakan aplikasi Sangat Baik untuk digunakan.