BAB 4

IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dijelaskan secara rinci penerjemahan kebutuhan pembangunan aplikasi ke dalam representasi perangkat lunak sesuai dengan hasil analisis yang telah dilakukan. Setelah implementasi maka dilakukan pengujian sistem yang baru dimana akan dilihat kekurangan-kekurangan pada aplikasi yang baru untuk selanjutnya diadakan pengembangan sistem.

4.1. Implementasi

Tujuan implementasi sistem adalah untuk menjelaskan tentang manual modul kepada semua user yang akan menggunakan sistem. Sehingga user tersebut dapat merespon apa yang ditampilkan di sistem dan memberikan masukan kepada pembuat sistem untuk dilakukan perbaikan agar sistem lebih baik lagi.

4.1.1. Perangkat Lunak Pembangun

Perangkat lunak yang digunakan untuk membangun *"Aplikasi Pemandu Wisata Kebun Binatang Bandung Berbasis Android (BonBinTour)"* adalah sebagai berikut :

1. Sistem Operasi *Microsoft Windows XP Professional Service Pack 2.*
2. *WampServer 2.2* untuk *local internet* dan penyimpanan *database.*
3. *Adobe Dreamweaver CS 5* untuk penulisan kode program *PHP*.
4. *Eclipse Helios* untuk penulisan kode program *android.*
5. *Google Map API Android V 2.0.*
6. *www.bonbintour.kebunbinatangbandung.com* sebagai *web hosting* aplikasi *backend.*
7. *Junaio Browser* sebagai *augmented reality browser.*
8. *Mozilla Firefox* atau *Google Chrome* sebagai *web browser.*

4.1.2. Perangkat Keras Pembangun

Kebutuhan minimum perangkat keras *(hardware)* yang diperlukan untuk mengimplementasikan program aplikasi yang dibuat adalah perangkat keras komputer *PC* dan *handphone* yang *compatible* dengan spesifikasi yang disebutkan dibawah ini. Semakin tinggi spesifikasi komputer/*handphone* yang digunakan untuk menjalankan aplikasi, akan semakin baik. Kebutuhan minimumnya yaitu :

1. *Processor : 1.6 Ghz*.
2. *Memory/RAM : 1 GB.*
3. *Hardisk :* 150 GB.
4. *VGA : 224 MB.*
5. *Handphone* berbasis *Android API 11 (Honeycomb).*

4.1.3. Kebutuhan *Web Hosting*

Kebutuhan *web hosting* pada pada penelitian ini digunakan untuk keperluan aplikasi *backend* yaitu *web admin* dan *web service*, oleh karena itu agar sistem dapat berjalan maka perlu di-*upload* di *web hosting*. Adapun spesifikasi *web hosting* dan nama *domain* (situs) yang digunakan adalah http://bonbintour.kebunbinatangbandung.com.

Tabel 4. 1 Spesifikasi Web Hosting.

Disk Space	*200 MB.*
Bandwith	*Unmetered MB.*
Development Feature	1. *Web Control Panel (CPanel).* 2. *PHP 5.3.* 3. *MySQL 5.1.61.* 4. *CGI* dan *SGI.* 5. *View Error Log.* 6. *POP3 & SMTP Account.* 7. *SPAM Filter.* 8. *Domain Control Panel.*

4.1.4. Implementasi Basis Data *(Database)*

Implementasi basis data diambil berdasarkan perancangan basis data yang dibuat sebelumnya, Secara fisik, implementasi basis data diimplementasikan menggunakan perangkat lunak *MySQL Server 5.5.16*. Tabel-tabel berikut menggambarkan struktur tabel yang diimplementasikan pada basis data.

1. Tabel *tbl_kingdom.*

```
CREATE TABLE IF NOT EXISTS `tbl_kingdom` (
  `id_kingdom` int(11) NOT NULL AUTO_INCREMENT,
  `nama_kingdom` varchar(50) NOT NULL,
  `jenis_kingdom` enum('hewan','tumbuhan') NOT NULL,
  PRIMARY KEY (`id_kingdom`),
  UNIQUE KEY `nama_kingdom` (`nama_kingdom`),
  KEY `jenis_kingdom` (`jenis_kingdom`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
AUTO_INCREMENT=1 ;
```

2. Tabel *tbl_filum.*

```
CREATE TABLE IF NOT EXISTS `tbl_filum` (
  `id_filum` int(11) NOT NULL AUTO_INCREMENT,
  `nama_filum` varchar(50) NOT NULL,
  `jenis_filum` enum('hewan','tumbuhan') NOT NULL,
  `fk_id_kingdom` int(11) DEFAULT NULL,
  PRIMARY KEY (`id_filum`),
  UNIQUE KEY `nama_filum` (`nama_filum`),
  KEY `jenis_filum` (`jenis_filum`),
  KEY `fk_id_kingdom` (`fk_id_kingdom`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1
AUTO_INCREMENT=1 ;

ALTER TABLE `tbl_filum`
  ADD CONSTRAINT `tbl_filum_ibfk_1` FOREIGN KEY
(`fk_id_kingdom`) REFERENCES `tbl_kingdom`
(`id_kingdom`) ON DELETE SET NULL ON UPDATE CASCADE;
```

3. Tabel *tbl_kelas.*

```
CREATE TABLE IF NOT EXISTS `tbl_kelas` (
  `id_kelas` int(11) NOT NULL AUTO_INCREMENT,
  `nama_kelas` varchar(50) NOT NULL,
  `jenis_kelas` enum('hewan','tumbuhan') NOT NULL,
  `fk_id_filum` int(11) DEFAULT NULL,
  PRIMARY KEY (`id_kelas`),
  UNIQUE KEY `nama_kelas` (`nama_kelas`),
  KEY `jenis_kelas` (`jenis_kelas`),
  KEY `fk_id_filum` (`fk_id_filum`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
AUTO_INCREMENT=1 ;

ALTER TABLE `tbl_kelas`
  ADD CONSTRAINT `tbl_kelas_ibfk_1` FOREIGN KEY
(`fk_id_filum`) REFERENCES `tbl_filum` (`id_filum`) ON
DELETE SET NULL;
```

4. Tabel *tbl_ordo.*

```
CREATE TABLE IF NOT EXISTS `tbl_ordo` (
  `id_ordo` int(11) NOT NULL AUTO_INCREMENT,
  `nama_ordo` varchar(50) NOT NULL,
  `jenis_ordo` enum('hewan','tumbuhan') NOT NULL,
  `fk_id_kelas` int(11) DEFAULT NULL,
  PRIMARY KEY (`id_ordo`),
  UNIQUE KEY `nama_ordo` (`nama_ordo`),
  KEY `fk_id_kelas` (`fk_id_kelas`),
  KEY `jenis_ordo` (`jenis_ordo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
AUTO_INCREMENT=1 ;

ALTER TABLE `tbl_ordo`
  ADD CONSTRAINT `tbl_ordo_ibfk_1` FOREIGN KEY
(`fk_id_kelas`) REFERENCES `tbl_kelas` (`id_kelas`) ON
DELETE SET NULL ON UPDATE CASCADE;
```

5. Tabel *tbl_famili.*

```
CREATE TABLE IF NOT EXISTS `tbl_famili` (
  `id_famili` int(11) NOT NULL AUTO_INCREMENT,
  `nama_famili` varchar(50) NOT NULL,
  `jenis_famili` enum('hewan','tumbuhan') NOT NULL,
  `fk_id_ordo` int(11) DEFAULT NULL,
  PRIMARY KEY (`id_famili`),
  UNIQUE KEY `nama_famili` (`nama_famili`),
  KEY `fk_id_ordo` (`fk_id_ordo`),
  KEY `jenis_famili` (`jenis_famili`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
AUTO_INCREMENT=1 ;

ALTER TABLE `tbl_famili`
  ADD CONSTRAINT `tbl_famili_ibfk_1` FOREIGN KEY
(`fk_id_ordo`) REFERENCES `tbl_ordo` (`id_ordo`) ON
DELETE SET NULL ON UPDATE CASCADE;
```

6. Tabel *tbl_genus.*

```
CREATE TABLE IF NOT EXISTS `tbl_genus` (
  `id_genus` int(11) NOT NULL AUTO_INCREMENT,
  `nama_genus` varchar(50) NOT NULL,
  `jenis_genus` enum('hewan','tumbuhan') NOT NULL,
  `fk_id_famili` int(11) DEFAULT NULL,
  PRIMARY KEY (`id_genus`),
  UNIQUE KEY `nama_genus` (`nama_genus`),
  KEY `jenis_genus` (`jenis_genus`),
  KEY `fk_id_famili` (`fk_id_famili`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
AUTO_INCREMENT=1 ;

ALTER TABLE `tbl_genus`
  ADD CONSTRAINT `tbl_genus_ibfk_1` FOREIGN KEY
(`fk_id_famili`) REFERENCES `tbl_famili` (`id_famili`)
ON DELETE SET NULL ON UPDATE CASCADE;
```

7. Tabel *tbl_spesies.*

```
CREATE TABLE IF NOT EXISTS `tbl_spesies` (
  `id_spesies` int(11) NOT NULL AUTO_INCREMENT,
  `nama_spesies` varchar(50) NOT NULL,
  `jenis_spesies` enum('hewan','tumbuhan') NOT NULL,
  `fk_id_genus` int(11) DEFAULT NULL,
  PRIMARY KEY (`id_spesies`),
  UNIQUE KEY `nama_spesies` (`nama_spesies`),
  KEY `jenis_spesies` (`jenis_spesies`),
  KEY `fk_id_genus` (`fk_id_genus`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
AUTO_INCREMENT=1 ;

ALTER TABLE `tbl_spesies`
  ADD CONSTRAINT `tbl_spesies_ibfk_1` FOREIGN KEY
(`fk_id_genus`) REFERENCES `tbl_genus` (`id_genus`) ON
DELETE SET NULL;
```

8. Tabel *tbl_hewan.*

```
CREATE TABLE IF NOT EXISTS `tbl_hewan` (
  `id_hewan` int(11) NOT NULL AUTO_INCREMENT,
  `nama_hewan` varchar(100) NOT NULL,
  `fk_id_spesies` int(11) DEFAULT NULL,
  `fk_id_genus` int(11) DEFAULT NULL,
  `fk_id_famili` int(11) DEFAULT NULL,
  `fk_id_ordo` int(11) DEFAULT NULL,
  `fk_id_kelas` int(11) DEFAULT NULL,
  `fk_id_filum` int(11) DEFAULT NULL,
  `fk_id_kingdom` int(11) DEFAULT NULL,
  `fisik` text,
  `makanan` text,
  `status` text,
  `populasi` text,
  `penyebaran` text,
  `habitat` text,
  `tingkah_laku` text,
  `jml_jantan` int(11) NOT NULL,
  `jml_betina` int(11) NOT NULL,
  PRIMARY KEY (`id_hewan`),
  UNIQUE KEY `nama_hewan` (`nama_hewan`),
  KEY `fk_id_famili` (`fk_id_famili`),
  KEY `fk_id_ordo` (`fk_id_ordo`),
  KEY `fk_id_kelas` (`fk_id_kelas`),
  KEY `fk_id_spesies` (`fk_id_spesies`),
  KEY `fk_id_genus` (`fk_id_genus`),
  KEY `fk_id_filum` (`fk_id_filum`),
  KEY `fk_id_kingdom` (`fk_id_kingdom`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1
AUTO_INCREMENT=1 ;
```

```
ALTER TABLE `tbl_hewan`
  ADD CONSTRAINT `tbl_hewan_ibfk_7` FOREIGN KEY
(`fk_id_kingdom`) REFERENCES `tbl_kingdom`
(`id_kingdom`) ON DELETE SET NULL ON UPDATE CASCADE,
  ADD CONSTRAINT `tbl_hewan_ibfk_1` FOREIGN KEY
(`fk_id_famili`) REFERENCES `tbl_famili` (`id_famili`)
ON DELETE SET NULL ON UPDATE CASCADE,
  ADD CONSTRAINT `tbl_hewan_ibfk_2` FOREIGN KEY
(`fk_id_ordo`) REFERENCES `tbl_ordo` (`id_ordo`) ON
DELETE SET NULL ON UPDATE CASCADE,
  ADD CONSTRAINT `tbl_hewan_ibfk_3` FOREIGN KEY
(`fk_id_kelas`) REFERENCES `tbl_kelas` (`id_kelas`) ON
DELETE SET NULL ON UPDATE CASCADE,
  ADD CONSTRAINT `tbl_hewan_ibfk_4` FOREIGN KEY
(`fk_id_spesies`) REFERENCES `tbl_spesies`
(`id_spesies`) ON DELETE SET NULL ON UPDATE CASCADE,
  ADD CONSTRAINT `tbl_hewan_ibfk_5` FOREIGN KEY
(`fk_id_genus`) REFERENCES `tbl_genus` (`id_genus`) ON
DELETE SET NULL ON UPDATE CASCADE,
  ADD CONSTRAINT `tbl_hewan_ibfk_6` FOREIGN KEY
(`fk_id_filum`) REFERENCES `tbl_filum` (`id_filum`) ON
DELETE SET NULL ON UPDATE CASCADE;
```

9. Tabel *tbl_tumbuhan.*

```
CREATE TABLE IF NOT EXISTS `tbl_tumbuhan` (
  `id_tumbuhan` int(11) NOT NULL AUTO_INCREMENT,
  `nama_tumbuhan` varchar(50) NOT NULL,
  `fk_id_spesies` int(11) DEFAULT NULL,
  `fk_id_genus` int(11) DEFAULT NULL,
  `fk_id_famili` int(11) DEFAULT NULL,
  `fk_id_ordo` int(11) DEFAULT NULL,
  `fk_id_kelas` int(11) DEFAULT NULL,
  `fk_id_filum` int(11) DEFAULT NULL,
  `fk_id_kingdom` int(11) DEFAULT NULL,
  `fisik` text,
  `perbanyakan` text,
  `status` text,
  `populasi` text,
  `penyebaran` text,
  `habitat` text,
  `manfaat` text,
  `jml` int(11) NOT NULL,
  PRIMARY KEY (`id_tumbuhan`),
  UNIQUE KEY `nama_tumbuhan` (`nama_tumbuhan`),
  KEY `fk_id_famili` (`fk_id_famili`),
  KEY `fk_id_ordo` (`fk_id_ordo`),
  KEY `fk_id_kelas` (`fk_id_kelas`),
  KEY `fk_id_spesies` (`fk_id_spesies`),
  KEY `fk_id_genus` (`fk_id_genus`),
  KEY `fk_id_filum` (`fk_id_filum`),
```

```
  KEY `fk_id_kingdom` (`fk_id_kingdom`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
AUTO_INCREMENT=1 ;

ALTER TABLE `tbl_tumbuhan`
  ADD CONSTRAINT `tbl_tumbuhan_ibfk_1` FOREIGN KEY
(`fk_id_famili`) REFERENCES `tbl_famili` (`id_famili`)
ON DELETE SET NULL ON UPDATE CASCADE,
  ADD CONSTRAINT `tbl_tumbuhan_ibfk_2` FOREIGN KEY
(`fk_id_ordo`) REFERENCES `tbl_ordo` (`id_ordo`) ON
DELETE SET NULL ON UPDATE CASCADE,
  ADD CONSTRAINT `tbl_tumbuhan_ibfk_3` FOREIGN KEY
(`fk_id_kelas`) REFERENCES `tbl_kelas` (`id_kelas`) ON
DELETE SET NULL ON UPDATE CASCADE,
  ADD CONSTRAINT `tbl_tumbuhan_ibfk_4` FOREIGN KEY
(`fk_id_spesies`) REFERENCES `tbl_spesies`
(`id_spesies`) ON DELETE SET NULL ON UPDATE CASCADE,
  ADD CONSTRAINT `tbl_tumbuhan_ibfk_5` FOREIGN KEY
(`fk_id_genus`) REFERENCES `tbl_genus` (`id_genus`) ON
DELETE SET NULL ON UPDATE CASCADE,
  ADD CONSTRAINT `tbl_tumbuhan_ibfk_6` FOREIGN KEY
(`fk_id_filum`) REFERENCES `tbl_filum` (`id_filum`) ON
DELETE SET NULL ON UPDATE CASCADE,
  ADD CONSTRAINT `tbl_tumbuhan_ibfk_7` FOREIGN KEY
(`fk_id_kingdom`) REFERENCES `tbl_kingdom`
(`id_kingdom`) ON DELETE SET NULL ON UPDATE CASCADE;
```

10. Tabel *tbl_fasilitas_umum.*

```
CREATE TABLE IF NOT EXISTS `tbl_fasilitas_umum` (
  `id_fasilitas_umum` int(11) NOT NULL AUTO_INCREMENT,
  `nama_fasilitas_umum` varchar(50) NOT NULL,
  `ket` text,
  PRIMARY KEY (`id_fasilitas_umum`),
  UNIQUE KEY `nama_fasilitas_umum`
(`nama_fasilitas_umum`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
AUTO_INCREMENT=1 ;
```

11. Tabel *tbl_poi.*

```
CREATE TABLE IF NOT EXISTS `tbl_poi` (
  `id_poi` int(11) NOT NULL AUTO_INCREMENT,
  `nama_poi` varchar(50) NOT NULL,
  `latitude` double NOT NULL,
  `longitude` double NOT NULL,
  `altitude` double NOT NULL,
  `fk_id_jenis_poi` int(11) DEFAULT NULL,
  `fk_id_hewan` int(11) DEFAULT NULL,
  `fk_id_tumbuhan` int(11) DEFAULT NULL,
  `fk_id_fasilitas_umum` int(11) DEFAULT NULL,
  PRIMARY KEY (`id_poi`),
  UNIQUE KEY `nama_poi` (`nama_poi`),
  KEY `fk_id_jenis_poi` (`fk_id_jenis_poi`),
  KEY `fk_id_hewan` (`fk_id_hewan`),
  KEY `fk_id_tumbuhan` (`fk_id_tumbuhan`),
  KEY `fk_id_fasilitas_umum` (`fk_id_fasilitas_umum`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1
AUTO_INCREMENT=1 ;
```

4.1.5. Implementasi *Class*

Implementasi *class* pada aplikasi yang dibangun merupakan implementasi dari perancangan *class diagram* yang terbentuk dari *use case diagram* kedalam bentuk fisik file-file dengan ekstensi *.java* maupun ekstensi *.php* yang digunakan dalam pembuatan dan jalannya proses dalam sistem. Berikut beberapa implementasi *class* yang telah terdefinisi, sedangkan untuk *source code*-nya dilampirkan.

Tabel 4. 2 Implementasi Class.

No	Nama *Class*	Nama File Fisik
1.	*Login*	*login.php*
2.	*v_login*	*v_login.php*
3.	*Kingdom*	*kingdom.php*
4.	*Mdl_Kingdom*	*mdl_kingdom.php*
5.	*list_kingdom*	*list_kingdom.php*
6.	*detail_kingdom*	*detail_kingdom.php*
7.	*ubah_kingdom*	*ubah_kingdom.php*

8.	*Filum*	*filum.php*
9.	*Mdl_Filum*	*mdl_filum.php*
10.	*list_filum*	*list_filum.php*
11.	*detail_filum*	*detail_filum.php*
12.	*ubah_filum*	*ubah_filum.php*
13.	*Kelas*	*kelas.php*
14.	*Mdl_Kelas*	*mdl_kelas.php*
15.	*list_kelas*	*list_kelas.php*
16.	*detail_kelas*	*detail_kelas.php*
17.	*ubah_kelas*	*ubah_kelas.php*
18.	*Ordo*	*ordo.php*
19.	*Mdl_Ordo*	*mdl_ordo.php*
20.	*list_ordo*	*list_ordo.php*
21.	*detail_ordo*	*detail_ordo.php*
22.	*ubah_ordo*	*ubah_ordo.php*
23.	*Famili*	*famili.php*
24.	*Mdl_Famili*	*mdl_famili.php*
25.	*list_famili*	*list_famili.php*
26.	*detail_famili*	*detail_famili.php*
27.	*ubah_famili*	*ubah_famili.php*
28.	*Genus*	*genus.php*
29.	*Mdl_Genus*	*mdl_genus.php*
30.	*list_genus*	*list_genus.php*
31.	*detail_genus*	*detail_genus.php*
32.	*ubah_genus*	*ubah_genus.php*
33.	*Spesies*	*spesies.php*

34.	*Mdl_Spesies*	*mdl_spesies.php*
35.	*list_spesies*	*list_spesies.php*
36.	*detail_spesies*	*detail_spesies.php*
37.	*ubah_spesies*	*ubah_spesies.php*
38.	*Hewan*	*hewan.php*
39.	*Mdl_Hewan*	*mdl_hewan.php*
40.	*list_hewan*	*list_hewan.php*
41.	*detail_hewan*	*detail_hewan.php*
42.	*ubah_hewan*	*ubah_hewan.php*
43.	*Tumbuhan*	*tumbuhan.php*
44.	*Mdl_Tumbuhan*	*mdl_tumbuhan.php*
45.	*list_tumbuhan*	*list_tumbuhan.php*
46.	*detail_tumbuhan*	*detail_tumbuhan.php*
47.	*ubah_tumbuhan*	*ubah_tumbuhan.php*
48.	*Fasilitas_Umum*	*fasilitas_umum.php*
49.	*Mdl_Fasilitas_Umum*	*mdl_fasilitas_umum.php*
50.	*list_fasilitas_umum*	*list_failitas_umum.php*
51.	*detail_fasilitas_umum*	*detail_fasilitas_umum.php*
52.	*ubah_fasilitas_umum*	*ubah_fasilitas_umum.php*
53.	*POI*	*poi.php*
54.	*Mdl_POI*	*mdl_poi.php*
55.	*list_poi*	*list_poi.php*
56.	*detail_poi*	*detail_poi.php*
57.	*ubah_poi*	*ubah_poi.php*
58.	*Suara*	*suara.php*
59.	*v_suara*	*v_suara.php*

60.	*Video*	*video.php*
61.	*v_video*	*v_video.php*
62.	*Gambar*	*gambar.php*
63.	*v_gambar*	*v_gambar.php*
64.	*Model*	*model.php*
65.	*v_model*	*v_model.php*
66.	*ARELXMLHelper*	*ARELXMLHelper.php*
67.	*ActivityARHewan*	*ActivityARHewan.java,*
68.	*ActivityARTumbuhan*	*ActivityARTumbuhan.java,*
69.	*ActivityARFasilitasUmum*	*ActivityARFasilitasUmum.java*
70.	*ActivityARView*	*ActivityARView.java*
71.	*ActivityScanMarker*	*ActivityScanMarker.java*
72.	*ActivityCurrentLocationMapV2*	*ActivityCurrentLocationMapV2.java*
73.	*ActivityBonBinTour*	*ActivityBonBinTour.java*
74.	*index.php*	*index.php*
75.	*index.xml*	*index.xml*
76.	*SherlockFragmentActivity*	*com.actionbarsherlock.app.SherlockFragmentActivity*
77.	*MetaioCloudPlugin*	*com.metaio.cloud.plugin.MetaioCloudPlugin*
78.	*OnMyLocationChangeListener*	*com.google.android.gms.maps.GoogleMap.OnMyLocationChangeListener*
79.	*ActivityPOIHewan*	*ActivityPOIHewan.java*
80.	*ServerRequestPOIHewan*	*ServerRequestPOIHewan.java*
81.	*POIHewan*	*POIHewan.java*
82.	*ListAdapterPOIHewan*	*ListAdapterPOIHewan.java*
83.	*ActivityPOITumbuhan*	*ActivityPOITumbuhan.java*
84.	*ServerRequestPOITumbuhan*	*ServerRequestPOITumbuhan.java*
85.	*POITumbuhan*	*POITumbuhan.java*

86.	*ListAdapterPOITumbuhan*	*ListAdapterPOITumbuhan.java*
87.	*ActivityPOIFasilitasUmum*	*ActivityPOIFasilitasUmum.java*
88.	*ServerRequestPOIFasilitasUmum*	*ServerRequestPOIFasilitasUmum.java*
89.	*POIFasilitasUmum*	*POIFasilitasUmum.java*
90.	*ListAdapterPOIFasilitasUmum*	*ListAdapterPOIFasilitasUmum.java*
91.	*ActivityVideoPlayer*	*ActivityVideoPlayer.java*
92.	*ActivityVideoPlayerView*	*ActivityVideoPlayerView.java*

4.1.6. Implementasi Antarmuka

Implementasi antarmuka dilakukan dengan setiap halaman yang dibuat baik aplikasi *backend* yang dibangun menggunakan *web* maupun aplikasi *frontend* yang dibangun di perangkat *mobile*. Berikut ini beberapa tampilan antarmuka yang telah di implementasikan.

4.1.6.1. Antarmuka Aplikasi *Backend*

Berikut ini implementasi antarmuka dari aplikasi *backend BonBinTour* beserta *file* program sebagai tampilan dari masing-masing antarmuka yang terlihat pada tabel 4.3. Adapun tampilan implementasi antarmuka aplikasi *backend BonBinTour* tertera pada lampiran B implementasi antarmuka.

Tabel 4. 3 Implementasi Antarmuka Aplikasi Backend.

No.	Halaman	Deskripsi	Nama File
1.	*Login*	Digunakan sebagai halaman *login administrator.*	*login.php*
2.	Utama *(backend)*	Digunakan sebagai halaman untuk menampilkan keseluruhan fungsional aplikasi pada sisi *backend.*	*utama.php*
3.	*Taksonomi*	Digunakan sebagai halaman untuk menampilkan daftar taksonomi yang akan diolah.	*taksonomi.php*

4.	Kategori	Digunakan sebagai halaman untuk menampilkan kategori objek wisata, yang meliputi hewan, tumbuhan dan fasilitas umum.	*kategori.php*
5.	Koordinat	Digunakan sebagai halaman untuk menampilkan gambaran peta posisi *POI* dan pengolahan koordinat *POI.*	*koordinat.php*
6.	*Kingdom*	Digunakan sebagai halaman untuk mengolah data *kingdom.*	*kingdom.php*
7.	*Filum*	Digunakan sebagai halaman untuk mengolah data *filum.*	*filum.php*
8.	*Kelas*	Digunakan sebagai halaman untuk mengolah data *kelas.*	*kelas.php*
9.	*Ordo*	Digunakan sebagai halaman untuk mengolah data *ordo.*	*ordo.php*
10.	*Famili*	Digunakan sebagai halaman untuk mengolah data *famili.*	*famili.php*
11.	*Genus*	Digunakan sebagai halaman untuk mengolah data *genus.*	*genus.php*
12.	*Spesies*	Digunakan sebagai halaman untuk mengolah data *spesies.*	*spesies.php*
13.	Hewan	Digunakan sebagai halaman untuk mengolah data hewan.	*hewan.php*
14.	Tumbuhan	Digunakan sebagai halaman untuk mengolah data tumbuhan.	*tumbuhan.php*
15.	Fasilitas Umum	Digunakan sebagai halaman untuk mengolah data fasilitas umum.	*fasilitas_umum.php*
16.	Peta	Digunakan sebagai halaman untuk menampilkan posisi *POI* dalam bentuk peta.	*peta.php*
17.	*POI*	Digunakan sebagai halaman untuk mengolah data *POI.*	*poi.php*

4.1.6.2. Antarmuka Aplikasi *Frontend*

Berikut ini implementasi antarmuka dari aplikasi *frontend BonBinTour* beserta *file* program sebagai tampilan dari masing-masing antarmuka yang terlihat pada tabel 4.4. Adapun tampilan implementasi antarmuka aplikasi *frontend BonBinTour* tertera pada lampiran A tampilan antarmuka.

Tabel 4. 4 Implementasi Antarmuka Aplikasi Frontend.

No.	Halaman	Deskripsi	Nama File
1.	Utama *(frontend)*	Digunakan sebagai halaman untuk menampilkan antarmuka utama aplikasi pada sisi *frontend.*	*ActivityUtama.java*
2.	*AR* Hewan Sekitar	Digunakan sebagai halaman untuk menampilkan *augmented reality* hewan di sekitar pengguna.	*ActivityARHewan.java*
3.	*AR* Tumbuhan Sekitar	Digunakan sebagai halaman untuk menampilkan *augmented reality* tumbuhan di sekitar pengguna.	*ActivityARTumbuhan.java*
4.	*AR* Fasilitas Umum Sekitar	Digunakan sebagai halaman untuk menampilkan *augmented reality* fasilitas umum di sekitar pengguna.	*ActivityARFasilitasUmum.java*
5.	*AR Scan Marker*	Digunakan sebagai halaman untuk menampilkan *augmented reality* yang terdapat pada *marker*.	*ActivityARScanMarker.java*
6.	Identifikasi Koordinat	Digunakan sebagai halaman untuk mengidentifikasi informasi posisi pengguna *(latitude, longitude).*	*ActivityCurrentLocationMapV2.java*

4.2. Pengujian

Dalam penelitian ini pengujian yang dilakukan terhadap sistem yaitu berupa rencana pengujian *betha*, dikarenakan aplikasi masih berupa *prototype* dan *betha* merupakan rencana pengujian yang tepat untuk aplikasi yang masih berupa *prototype*.

4.2.1. Rencana Pengujian *Betha*

Pengujian *betha* merupakan pengujian yang dilakukan dengan tujuan untuk mengetahui sejauh mana kualitas sistem, apakah sudah memenuhi harapan atau belum. Untuk itu dalam pengujian *betha* dilakukan wawancara/*interview* kepada petugas kebun binatang Bandung, untuk menguji fungsionalitas dari aplikasi yang dibuat.

Interview yang dilakukan kepada petugas kebun binatang Bandung terdiri dari 8 pertanyaan dengan 1 responden yang nantinya akan diposisikan sebagai *admin* aplikasi. Untuk laporan hasil *interview* dengan pihak terkait di Kebun Binatang Bandung dapat dilihat pada lampiran D hasil *interview*.

4.3. Kesimpulan dan Hasil Pengujian

Dalam pengujian *betha* yang berupa hasil *interview* dengan pihak Kebun Binatang Bandung, maka dapat ditarik kesimpulan bahwa sistem yang dibangun memiliki segi penampilan antarmuka yang cukup menarik, sudah membantu mengetahui informasi keberadaan hewan, tumbuhan dan fasilitas umum di sekitar pengguna, informasi yang disajikan sudah informatif namun perlu adanya soialisasi pengenalan aplikasi kepada pengguna, konten informasi multimedia yang menarik yang dapat memberikan sisi edukasinya juga, penerapan teknologi *augmented reality*-nya sudah diterapkan dengan baik, aplikasi yang dibangun mudah digunakan dan aplikasi ini sudah sangat membantu dan bermanfaat bagi pengunjung dalam melakukan kunjungan wisata di Kebun Binatang Bandung.