BAB 4
IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

Implementasi sistem adalah tahap penerapan sistem yang telah dibuat pada tahap perancangan. Implementasi sistem bertujuan untuk menerjemahkan perancangan berdasarkan hasil analisis menjadi kode – kode bahasa pemrograman yang dapat dimengerti oleh mesin.

4.1.1 Perangkat Keras yang Digunakan

Spesifikasi perangkat keras yang digunakan untuk implementasi sistem monitoring ini dapat dilihat pada tabel 4.1 sebagai berikut.

Tabel 4. 1 Spesifikasi Perangkat Keras yang Digunakan

No.	Perangkat Keras	Spesifikasi
1.	Monitor	17”
2.	Processor	Processor dual core berkecepatan 2,7 GHz
3.	Memori / RAM	2 GB
4.	Harddisk	320 GB
5.	VGA Card	VGA On-Board
6.	Keyboard	Standar
7.	Mouse	Standar
8.	Printer	Standar

4.1.2 Perangkat Lunak yang Digunakan

Spesifikasi perangkat lunak yang digunakan untuk implementasi sistem monitoring ini dapat dilihat pada tabel 4.2 sebagai berikut.

Tabel 4. 2 Spesifikasi Perangkat Lunak yang Digunakan

No.	Perangkat Lunak	Spesifikasi
1.	Sistem Operasi	Minimal Windows XP
2.	*Browser*	Mozila Firefox, Crome, Opera

4.1.3 Implementasi *Database*

Implementasi *database* diambil berdasarkan perancangan basis data yang dibuat pada bab sebelumnya. Tabel – tabel yang diimplementasikan dapat dilihat pada tabel 4.4.

Tabel 4. 3 Implementasi Data

No.	Nama Tabel	File Yang Berhubungan
1.	yayasan	Yayasan.sql
2.	Anak	Anak.sql
3.	anak_tipe	Anak_tipe.sql
4.	proposal	Proposal.sql
5.	Biaya_anak	Biaya_anak.sql
6.	pencapaian	Pencapaian.sql
7.	anak_pencapaian	Anak_pencapaian.sql
8.	biaya_anak_detail	Biaya_anak_detail.sql
9.	proposal_detail	Proposal_detail.sql
10.	biaya_khusus	Biaya_khusus.sql

Database sistem yang dibangun pada aplikasi yang dibangun diimplementasikan pada *DBMS MySQL*. Implementasi tabel – tabel yang ada pada data sistem monitoring ini dapat dilihat pada tabel berikut.

1. Tabel Yayasan

   Tabel yayasan terdiri dari id_yayasan, username, password, nama_yayasan, alamat, email, status, cdate, mdate. Rincian yang terdapat pada tabel yayasan dapat dilihat pada tabel 4.4 sebagai berikut.

Tabel 4. 4 Tabel Yayasan

Tabel yayasan

```
CREATE TABLE IF NOT EXISTS `yayasan` (
  `id_yayasan` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(100) NOT NULL,
  `password` varchar(150) NOT NULL,
  `nama_yayasan` varchar(100) NOT NULL,
  `alamat` text NOT NULL,
  `email` varchar(100) NOT NULL,
  `status` char(1) NOT NULL,
  `cdate` int(11) NOT NULL,
  `mdate` int(11) NOT NULL,
  `id_petugas` int(11) NOT NULL,
  PRIMARY KEY (`id_yayasan`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=8 ;
```

2. Tabel Anak

   Tabel anak terdiri dari id_anak, id_tipe, id_yayasan, nama_anak, tempat_lahir, tgl_lahir, lokasi, status, idate, odate. Rincian yang terdapat pada tabel anak dapat dilihat pada tabel 4.5 sebagai berikut.

Tabel 4. 5 Tabel Anak

Tabel anak

```
CREATE TABLE IF NOT EXISTS `anak` (
  `id_anak` int(11) NOT NULL AUTO_INCREMENT,
  `id_tipe` int(11) NOT NULL,
  `id_yayasan` int(11) NOT NULL,
  `nama_anak` varchar(100) NOT NULL,
  `tempat_lahir` int(11) NOT NULL,
  `tgl_lahir` date NOT NULL,
  `nama_ortu` varchar(100) NOT NULL,
  `alamat_ortu` text NOT NULL,
  `lokasi` char(3) NOT NULL,
  `status` char(1) NOT NULL,
  `idate` date NOT NULL,
  `odate` date DEFAULT NULL,
  `cdate` int(11) NOT NULL,
  `mdate` int(11) NOT NULL,
  PRIMARY KEY (`id_anak`)
  FOREIGN        KEY        ('id_tipe')        REFERENCE
'dinsos_dimas','anak_tipe'  ('id_tipe')  ON  DELETE  RESTRICT
ON UPDATE RESTRICT ;
) ENGINE=InnoDB   DEFAULT  CHARSET=latin1  AUTO_INCREMENT=15
;
```

3. Tabel anak_tipe

   Tabel anak_tipe terdiri dari id_tipe dan nama_tipe. Rincian yang terdapat pada tabel anak_tipe dapat dilihat pada tabel 4.6 sebagai berikut.

Tabel 4. 6 Tabel anak_tipe

Tabel anak_tipe

CREATE TABLE IF NOT EXISTS `anak_tipe` (
  `id_tipe` int(11) NOT NULL AUTO_INCREMENT,
  `nama_tipe` varchar(100) NOT NULL,
  PRIMARY KEY (`id_tipe`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;

4. Tabel proposal

   Tabel proposal terdiri dari id_proposal, id_yayasan, tipe, tgl_pengajuan, total, dan sukses. Rincian yang terdapat pada tabel proposal dapat dilihat pada tabel 4.7 sebagai berikut.

Tabel 4. 7 Tabel Proposal

Tabel proposal

CREATE TABLE IF NOT EXISTS `proposal` (
  `id_proposal` int(11) NOT NULL AUTO_INCREMENT,
  `id_yayasan` int(11) NOT NULL,
  `tipe` varchar(25) NOT NULL COMMENT 'Reimbursement, Pengajuan',
  `tgl_pengajuan` date NOT NULL,
  `total` bigint(20) NOT NULL,
  `status` varchar(15) NOT NULL COMMENT 'Ajukan, Setujui, Sukses',
  `cdate` int(11) NOT NULL,
  `mdate` int(11) NOT NULL,
  `id_petugas` int(11) NOT NULL,
  PRIMARY KEY (`id_proposal`)
  FOREIGN KEY (‘id_yayasan’) REFERENCE ‘dinsos_dimas’,’yayasan’ (‘id_yayasan’) ON DELETE RESTRICT ON UPDATE RESTRICT
)
 	
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;

5. Tabel proposal detail

   Tabel proposal detail terdiri dari id_prop_detail, id_proposal, id_detail, id_khusus, nama_peruntukan, tempo, tgl_peruntukan. Rincian yang terdapat pada tabel proposal detail dapat dilihat pada tabel 4.8 sebagai berikut.

Tabel 4. 8 Tabel Proposal_detail

Tabel proposal_detail

```
CREATE TABLE IF NOT EXISTS `proposal_detail` (
  `id_prop_detail` int(11) NOT NULL AUTO_INCREMENT,
  `id_proposal` int(11) NOT NULL,
  `id_detail` int(11) NOT NULL,
  `id_khusus` int(11) NOT NULL,
  `nama_peruntukan` varchar(150) NOT NULL,
  `tempo` int(11) NOT NULL,
  `tgl_peruntukan` date NOT NULL,
  `deskripsi` text NOT NULL,
  `total` bigint(20) NOT NULL,
  `lpj` int(11) NOT NULL,
  `status` varchar(15) NOT NULL COMMENT 'Tunda, Setujui,
Sukses',
  `keterangan` varchar(15) DEFAULT NULL COMMENT 'Lebih,
Pas, Kurang',
  `penggunaan` bigint(20) NOT NULL DEFAULT '0',
  `cdate` int(11) NOT NULL,
  `mdate` int(11) NOT NULL,
  PRIMARY KEY (`id_prop_detail`)
  FOREIGN       KEY       ('id_proposal')       REFERENCE
'dinsos_dimas','proposal'    ('id_proposal')    ON    DELETE
RESTRICT ON UPDATE RESTRICT,
FOREIGN         KEY         ('id_detail')         REFERENCE
'dinsos_dimas','biaya_detail'   ('id_detail')   ON   DELETE
RESTRICT ON UPDATE RESTRICT,
FOREIGN         KEY         ('id_khusus')         REFERENCE
'dinsos_dimas','biaya_khusus'   ('id_khusus')   ON   DELETE
RESTRICT ON UPDATE RESTRICT
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=24
;
```

6. Tabel pencapaian

   Tabel pencapaian terdiri dari id_pencapaian dan nama_pencapaian. Rincian yang terdapat pada tabel pencapaian dapat dilihat pada tabel 4.9 sebagai berikut.

Tabel 4. 9 Tabel Pencapaian

Tabel pencapaian

```
CREATE TABLE IF NOT EXISTS `pencapaian` (
  `id_pencapaian` int(11) NOT NULL AUTO_INCREMENT,
  `nama_pencapaian` varchar(100) NOT NULL,
  PRIMARY KEY (`id_pencapaian`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;
```

7. Tabel anak pencapaian

Tabel anak pencapaian terdiri dari id_anak, id_pencapaian, sdate, edate. Rincian yang terdapat pada tabel anak pencapaian dapat dilihat pada tabel 4.10 berikut.

Tabel 4. 10 Tabel Anak Pencapaian

Tabel pencapaian

```
CREATE TABLE IF NOT EXISTS `anak_pencapaian` (
  `id_anak` int(11) NOT NULL,
  `id_pencapaian` int(11) NOT NULL,
  `sdate` date NOT NULL,
  `edate` date NOT NULL
  FOREIGN KEY ('id_pencapaian') REFERENCE
'dinsos_dimas','pencapaian' ('id_pencapaian) ON DELETE
RESTRICT ON UPDATE RESTRICT) ENGINE=InnoDB DEFAULT
CHARSET=latin1;
```

8. Tabel biaya anak

Tabel biaya anak terdiri dari id_biaya_anak, id_yayasan, status, cdate, mdate. Rincian yang terdapat pada tabel biaya anak dapat dilihat pada tabel 4.11 berikut.

Tabel 4. 11 Tabel Biaya Anak

Tabel biaya anak

```
CREATE TABLE IF NOT EXISTS `biaya_anak` (
  `id_biaya_anak` int(11) NOT NULL AUTO_INCREMENT,
  `id_yayasan` int(11) NOT NULL,
  `status` varchar(15) NOT NULL COMMENT 'Tunda, Ajukan,
Batal, Setujui',
  `cdate` int(11) NOT NULL,
  `mdate` int(11) NOT NULL,
  PRIMARY KEY (`id_biaya_anak`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;
```

9. Tabel biaya anak detail

Tabel biaya anak detail terdiri dari id_detail, id_biaya_anak, id_anak, nama_peruntukan, tempo, tgl_peruntukan, deskripsi. Rincian yang terdapat pada tabel biaya anak detail dapat dilihat pada tabel 4.12 berikut.

Tabel 4. 12 Tabel Biaya Anak Detail

Tabel biaya_anak_detail

```
CREATE TABLE IF NOT EXISTS `biaya_anak_detail` (
  `id_detail` int(11) NOT NULL AUTO_INCREMENT,
  `id_biaya_anak` int(11) NOT NULL,
  `id_anak` int(11) NOT NULL,
  `id_yayasan` int(11) NOT NULL,
  `nama_peruntukan` varchar(150) NOT NULL,
  `tempo` int(11) NOT NULL,
  `tgl_peruntukan` date NOT NULL,
  `deskripsi` text NOT NULL,
  `total` bigint(20) NOT NULL,
  PRIMARY KEY (`id_detail`)
 FOREIGN       KEY       ('id_biaya_anak')       REFERENCE
'dinsos_dimas','biaya_anak'  ('id_biaya_anak')  ON  DELETE
RESTRICT ON UPDATE RESTRICT,
FOREIGN        KEY        ('id_yayasan')        REFERENCE
'dinsos_dimas','yayasan' ('id_yayasan') ON DELETE RESTRICT
ON UPDATE RESTRICT
FOREIGN  KEY  ('id_anak')  REFERENCE  'dinsos_dimas','anak'
('id_anak')  ON  DELETE  RESTRICT  ON  UPDATE  RESTRICT)
ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=20 ;
```

10. Tabel biaya khusus

Tabel biaya khusus terdiri dari id_khusus, id_yayasan, nama_peruntukan, tempo_tgl_peruntukan, deskripsi, total, status, cdate, mdate. Rincian yang terdapat pada tabel biaya khusus dapat dilihat pada tabel 4.13 berikut.

Tabel 4. 13 Tabel Biaya Khusus

Tabel biaya_khusus

```
CREATE TABLE IF NOT EXISTS `biaya_khusus` (
  `id_khusus` int(11) NOT NULL AUTO_INCREMENT,
  `id_yayasan` int(11) NOT NULL,
  `nama_peruntukan` varchar(150) NOT NULL,
  `tempo` int(11) NOT NULL,
  `tgl_peruntukan` date NOT NULL,
  `deskripsi` text NOT NULL,
  `total` bigint(20) NOT NULL,
  `status` varchar(15) NOT NULL COMMENT 'Tunda, Ajukan,
Batal',
  `cdate` int(11) NOT NULL,
  `mdate` int(11) NOT NULL,
  PRIMARY KEY (`id_khusus`)
  FOREIGN        KEY        ('id_yayasan')        REFERENCE
'dinsos_dimas','yayasan' ('id_yayasan') ON DELETE RESTRICT
ON UPDATE RESTRICT
)ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;
```

4.1.4 Implementasi Antarmuka

Implementasi antarmuka dilakukan untuk setiap tampilan program yang dibangun beserta pengkodeannya dalam bentuk file program. Implementasi antarmuka yang dibangun terdiri dari implementasi antarmuka Yayasan atau Lembaga dan implementasi antarmuka petugas. Implementasi antarmuka untuk Yayasan atau Lembaga dapat dilihat pada tabel 4.14.

Tabel 4. 14 Implementasi Antarmuka Yayasan / Lembaga

Menu	Deskripsi	Nama File
Login	Halaman Yayasan untuk mengisi data *login* agar dapat masuk kedalam sistem	index.php
Lupa *Password*	Halaman Yayasan yang digunakan jika lupa *password*	lupa_*password* .php
Beranda	Halaman awal saat berhasil *login*	awal.php
Pengolahan data anak	Halaman yang digunakan untuk mengelola data anak.	anak.php
Pengolahan data biaya anak	Halaman yang digunakan untuk mengolah biaya anak	Biaya_anak.php
Pengolahan data biaya khusus	Halaman yang digunakan untuk mengolah biaya khusus	Biaya_khusus.php
Pengolahan data proposal	Halaman yang digunakan untuk menambahkan proposal	Proposal.php

Implementasi antarmuka kedua adalah implementasi antarmuka untuk Petugas. Implementasi antarmuka untuk Petugas dapat dilihat pada tabel 4.15.

Tabel 4. 15 Implementasi Antarmuka Petugas

Menu	Deskripsi	Nama File
Login	Halaman untuk mengisi data *login* agar dapat masuk kedalam sistem	index.php
Lupa *Password*	Halaman yang digunakan jika lupa *password*	lupa_*password* .php
Beranda	Halaman awal saat berhasil *login*	awal.php
Yayasan	Halaman yang digunakann jika ingin mengolah yayasan	yayasan.php
Proposal	Halaman yang digunakan untuk mengolah proposal	proposal.php proposal_detail.php
Monitoring	Halaman yang digunakan jika ingin melihat data hasil monitoring	monitoring.php
Setting	Halaman yang digunakan jika ingin mengolah setting	setting.php

4.2 Pengujian Sistem

Pengujian sistem dilakukan untuk menguji apakah sistem yang dibangun sudah layak untuk digunakan. Pengujian yang digunakan untuk menguji sistem monitoring ini adalah menggunakan pengujian *black box* dan pegujian *beta.* Pengujian *black box* ini berfokus pada persyaratan fungsional dari perangkat lunak yang dibangun.

4.2.1 Rencana Pengujian

Rencana pengujian terdiri dari rencana pengujian untuk yayasan atau lembaga dan pengujian untuk petugas.

4.2.1.1 Rencana Pengujian Halaman Yayasan

Rencana pengujian untuk halaman Admin dapat dilihat pada tabel 4.16 sebagai berikut.

Tabel 4. 16 Rencana Pengujian Halaman Yayasan

Kelas Uji	Butir Uji	Jenis Pengujian
Login	Input data *login*	Blackbox
	Verifikasi	Blackbox
	Lupa *password*	Blackbox
Anak	Tambah anak	Blackbox
	Edit anak	Blackbox
Proposal	Tambah proposal	Blackbox
		Blackbox
		Blackbox

4.2.1.2 Rencana Pengujian Halaman Petugas

Rencana pengujian untuk halaman Petugas dapat dilihat pada tabel 4.17 sebagai berikut.

Tabel 4. 17 Rencana Pengujian Halaman Petugas

Kelas Uji	Butir Uji	Jenis Pengujian
Login	Input data *login*	Blackbox
	Verifikasi	Blackbox
	Lupa *password*	Blackbox
Lihat detail yayasan	Tambah yayasan	Blackbox
	Hapus yayasan	Blackbox
Monitoring	Monitoring yang sesuai dengan data yang dipilih	Blackbox
Proposal	Lihat detail proposal	Blackbox
	Setujui proposal	Blackbox
Setting	Ubah setting	Blackbox
	Tambah setting	Blackbox
	Hapus setting	Blackbox

4.2.2 Pengujian Black Box

Pengujian *black box* adalah pengujian yang berfokus kepada keluaran yang dihasilkan dari data perangkat lunak yang dibangun. Teknik yang digunakan dalam pengujian sistem monitoring ini adalah teknik *equivalence partitioning* yaitu masing – masing kelas *equivalensi partition* diproses dimana program yang akan memproses anggota kelas – kelas tersebut secara *equivalen*.

4.2.2.1 Pengujian Metode *Equivalence partitioning* Halaman Yayasan

Pengujian metode *equivalence partitioning* pada halaman Yayasan terdiri dari pengujian *login*, anak dan proposal.

1. Pengujian *Login*

Pengujian dari *login* dengan data *valid* kemudian diamati dan menghasilkan sebuah kesimpulan. Pengujian *login* dapat dilihat pada tabel 4.18.

Tabel 4. 18 Pengujian *Login* dengan Data *Valid*

Kasus dan hasil uji (data *valid*)			
Data masukan	***Valid* class**	**Pengamatan**	**Kesimpulan**
*User*name Contoh masukan : Yayasan	[A-Z | a-z | 0-9]	Data *user*name dapat diterima jika input data : 1. Kurang dari 10 karakter 2. Boleh menggunakan angka, huruf atau kombinasi dari keduanya	[√] diterima [ ] ditolak
Password Contoh masukan : 123456	[A-Z | a-z | 0-9]	Data *password* dapat diterima jika input data : 1. Kurang dari 150 karakter 2. Boleh menggunakan angka, huruf atau kombinasi dari keduanya	[√] diterima [ ] ditolak

Pengujian dari *login* dengan data *invalid* kemudian diamati dan menghasilkan sebuah kesimpulan. Pengujian *login* dapat dilihat pada tabel 4.19.

Tabel 4. 19 Pengujian *Login* dengan Data *Invalid*

Kasus dan hasil uji (data *invalid*)			
Data masukan	***Valid* class**	**Pengamatan**	**Kesimpulan**
*User*name Contoh masukan : Yayasan	Data kosong	Dapat menampilkan pesan kesalahan bahwa *user*name harus diisi (tidak boleh kosong)	[ ] diterima [√] ditolak
Password Contoh masukan : 123456	Data kosong	Dapat menampilkan pesan kesalahan bahwa *password* harus diisi (tidak boleh kosong)	[ ] diterima [√] ditolak

2. Pengujian Anak

Pengujian dari anak dengan data *valid* kemudian diamati dan menghasilkan sebuah kesimpulan. Pengujian anak dapat dilihat pada tabel 4.20.

Tabel 4. 20 Pengujian Anak dengan Data *Valid*

Kasus dan hasil uji (data *valid*)			
Data masukan	***Valid* class**	**Pengamatan**	**Kesimpulan**
Nama anak yang dimasukan Contoh masukan : Fitria	[A-Z | a-z]	Data nama akun dapat diterima jika input data : 1. Kurang dari 50 karakter 2. Boleh menggunakan huruf kapital maupun huruf kecil atau kombinasi dari keduanya	[√] diterima [ ] ditolak
tipe anak yang dimasukkan Contoh masukan : sekolah	[A-Z | a-z]	Data tipe anak dapat diterima jika input data : 1. Kurang dari 50 karakter 2. Boleh menggunakan huruf kapital maupun huruf kecil atau kombinasi dari	[√] diterima [ ] ditolak

		keduanya	
Tambah tanggal lahir Contoh masukan : 10/8/2000	[date]	Data tanggal lahir dapat diterima jika input data berupa date	[√] diterima [ ] ditolak

Pengujian dari anak dengan data *invalid* kemudian diamati dan menghasilkan sebuah kesimpulan. Pengujian anak dapat dilihat pada tabel 4.21

.

Tabel 4. 21 Pengujian Anak dengan Data *Invalid*

Kasus dan hasil uji (data *valid*)			
Data masukan	***Valid* class**	**Pengamatan**	**Kesimpulan**
Nama anak yang dimasukan Contoh masukan : Fitria	Data kosong	Data nama akun dapat diterima jika input data : 3. Kurang dari 50 karakter 4. Boleh menggunakan huruf kapital maupun huruf kecil atau kombinasi dari keduanya	[√] diterima [ ] ditolak
tipe anak yang dimasukkan Contoh masukan : sekolah	Data kosong	Data tipe anak dapat diterima jika input data : 3. Kurang dari 50 karakter 4. Boleh menggunakan huruf kapital maupun huruf kecil atau kombinasi dari keduanya	[√] diterima [ ] ditolak
Tambah tanggal lahir Contoh masukan : 10/8/2000	Data kosong	Data tanggal lahir dapat diterima jika input data berupa date	[√] diterima [ ] ditolak

3. Pengujian Pengolahan Data Proposal

Pengujian dari pengolahan data proposal dengan data *valid* kemudian diamati dan menghasilkan sebuah kesimpulan. Pengujian pengolahan data proposal dapat dilihat pada tabel 4.22.

Tabel 4. 22 Pengolahan Data Proposal dengan Data *Valid*

Kasus dan hasil uji (data *valid*)			
Data masukan	***Valid* class**	**Pengamatan**	**Kesimpulan**
Tanggal peruntukan Contoh masukan : 10-10-2013	[date]	Data nama pengguna dapat diterima jika input data berupa tanggal	[√] diterima [ ] ditolak
Total Contoh masukan : 2000000	[0-9]	Data email dapat diterima jika input data berupa angka	[√] diterima [ ] ditolak

Pengujian dari pengolahan data proposal dengan data *invalid* kemudian diamati dan menghasilkan sebuah kesimpulan. Pengujian pengolahan data proposal dapat dilihat pada tabel 4.23.

Tabel 4. 23 Pengolahan Data Proposal dengan Data *Invalid*

Kasus dan hasil uji (data *valid*)			
Data masukan	***Valid* class**	**Pengamatan**	**Kesimpulan**
Tanggal peruntukan Contoh masukan : 10-10-2013	Data kosong	Data nama pengguna dapat diterima jika input data berupa tanggal	[√] diterima [ ] ditolak
Total Contoh masukan : 2000000	Data kosong	Data email dapat diterima jika input data berupa angka	[√] diterima [ ] ditolak

4.2.2.2 Pengujian Metode Equivalence Pertitioning Halaman Petugas

Pengujian metode *equivalence partitioning* pada halaman Petugas terdiri dari pengujian *login*, lihat detail yayasan, monitoring, proposal, setting.

1. Pengujian *Login* Petugas

Pengujian dari *login* Petugas dengan data *valid* kemudian diamati dan menghasilkan sebuah kesimpulan. Pengujian *login* Petugas dapat dilihat pada tabel 4.24.

Tabel 4. 24 Pengujian *Login* Petugas dengan Data *Valid*

Kasus dan hasil uji (data *valid*)			
Data masukan	***Valid* class**	**Pengamatan**	**Kesimpulan**
*User*name Contoh masukan : petugas	[A-Z | a-z | 0-9]	Data *user*name dapat diterima jika input data : 1. Kurang dari 10 karakter 2. Boleh menggunakan angka, huruf atau kombinasi dari keduanya	[√] diterima [ ] ditolak
Password Contoh masukan : 123456	[A-Z | a-z | 0-9]	Data *password* dapat diterima jika input data : 1. Kurang dari 150 karakter 2. Boleh menggunakan angka, huruf atau kombinasi dari keduanya	[√] diterima [ ] ditolak

Pengujian dari *login* Petugas dengan data *invalid* kemudian diamati dan menghasilkan sebuah kesimpulan. Pengujian *login* Petugas dapat dilihat pada tabel 4.25.

Tabel 4. 25 Pengujian *Login* Kepala Bidang dengan Data *Invalid*

Kasus dan hasil uji (data *invalid*)			
Data masukan	***Valid* class**	**Pengamatan**	**Kesimpulan**
*User*name Contoh masukan : petugas	Data kosong	Dapat menampilkan pesan kesalahan bahwa *user*name harus diisi (tidak	[√] diterima [ ] ditolak

		boleh kosong)	
Password Contoh masukan : 123456	Data kosong	Dapat menampilkan pesan kesalahan bahwa *password* harus diisi (tidak boleh kosong)	[√] diterima [ ] ditolak

2. Pengujian Lihat Detail Yayasan

Pengujian dari detail yayasan dengan data *valid* kemudian diamati dan menghasilkan sebuah kesimpulan. Pengujian detail yayasan dapat dilihat pada tabel 4.26.

Tabel 4. 26 Pengujian Akun Kepala Bidang dengan Data *Valid*

Kasus dan hasil uji (data *valid*)			
Data masukan	***Valid* class**	**Pengamatan**	**Kesimpulan**
Nama yayasan yang ditambah Contoh masukan : yayasan	[A-Z | a-z]	Data nama akun dapat diterima jika input data : 1. Kurang dari 50 karakter 2. Boleh menggunakan huruf kapital maupun huruf kecil atau kombinasi dari keduanya	[√] diterima [ ] ditolak
Email yayasan Contoh masukan : yayasan@yahoo.com	[A-Z | a-z | 0-9 | simbol]	Data email dapat diterima jika input data : 1. Kurang dari 50 karakter 2. Boleh menggunakan angka, huruf, simbol atau kombinasi dari ketiganya	[√] diterima [ ] ditolak
Data *password* Contoh masukan : *Password* lama : 123456 *Password* baru :	[A-Z | a-z | 0-9 | simbol]	Data *password* dapat diterima jika input data : 1. Kurang dari 150 karakter	[√] diterima [ ] ditolak

Kasus dan hasil uji (data *valid*)			
Data masukan	***Valid* class**	**Pengamatan**	**Kesimpulan**
aha123 *Password* baru : aha123		2. Boleh menggunakan angka, huruf, simbol atau kombinasi dari ketiganya	

Pengujian dari detail yayasan dengan data *invalid* kemudian diamati dan menghasilkan sebuah kesimpulan. Pengujian detail yayasan dapat dilihat pada tabel 4.27.

Tabel 4. 27 Pengujian Akun Kepala Bidang dengan Data *Invalid*

Kasus dan hasil uji (data *invalid*)			
Data masukan	***Valid* class**	**Pengamatan**	**Kesimpulan**
Nama yayasan yang ditambah Contoh masukan : yayasan	Data kosong	Dapat menampilkan pesan kesalahan bahwa nama akun harus diisi (tidak boleh kosong)	[√] diterima [ ] ditolak
Email yayasan Contoh masukan : yayasan@yahoo.com	Data kosong	Dapat menampilkan pesan kesalahan bahwa email harus diisi dengan format xxx@xxx.xx (tidak boleh kosong)	[√] diterima [ ] ditolak
Data *password* Contoh masukan : *Password* lama : 123456 *Password* baru : aha123 *Password* baru : aha123	Data kosong	Dapat menampilkan pesan kesalahan bahwa : 1. *Password* lama harus diisi (tidak boleh kosong) 2. *Password* baru minimal 6 karakter 3. *Password* baru tidak sama	[√] diterima [ ] ditolak

3. Pengujian Proposal

Pengujian dari proposal dengan data *valid* kemudian diamati dan menghasilkan sebuah kesimpulan. Pengujian proposal dapat dilihat pada tabel 4.28.

Tabel 4. 28 Proposal dengan Data *Valid*

Kasus dan hasil uji (data *valid*)			
Data masukan	***Valid* class**	**Pengamatan**	**Kesimpulan**
Nama Proposal yang dipilih Contoh masukan : Ajukan	Pilih [nama yayasan…]	Memilih data proposal yang sebelumnya telah diinputkan oleh yayasan	[√] diterima [ ] ditolak

Pengujian dari proposal dengan data *invalid* kemudian diamati dan menghasilkan sebuah kesimpulan. Pengujian proposal dapat dilihat pada tabel 4.29.

Tabel 4. 29 Proposal dengan Data *Invalid*

Kasus dan hasil uji (data *invalid*)			
Data masukan	***Valid* class**	**Pengamatan**	**Kesimpulan**
Nama Proposal yang dipilih Contoh masukan : Ajukan	Tidak dipilih	Tidak menampilkan menu halaman berikutnya, karena *field* tersebut belum dipilih	[√] diterima [ ] ditolak

4. Pengujian Pemilihan Monitoring

Pengujian dari pemilihan monitoring dengan data *valid* kemudian diamati dan menghasilkan sebuah kesimpulan. Pengujian pemilihan monitoring dapat dilihat pada tabel 4.30

.

Tabel 4. 30 Pengujian Pemilihan Monitoring dengan Data *Valid*

Kasus dan hasil uji (data *valid*)			
Data masukan	***Valid* class**	**Pengamatan**	**Kesimpulan**
Monitoring yang dipilih Contoh masukan : Monitoring pencapaian anak	Pilih [monitoring …]	Memilih monitoring	[√] diterima [ ] ditolak

Pengujian dari pemilihan monitoring dengan data *invalid* kemudian diamati dan menghasilkan sebuah kesimpulan. Pengujian pemilihan monitoring dapat dilihat pada tabel 4.31.

Tabel 4. 31 Pengujian Pemilihan Monitoring dengan Data *Invalid*

Kasus dan hasil uji (data *invalid*)			
Data masukan	***Valid* class**	**Pengamatan**	**Kesimpulan**
Monitoring yang dipilih Contoh masukan : Monitoring pencapaian anak	Tidak dipilih	Tidak menampilkan menu halaman berikutnya, karena *field* tersebut belum dipilih	[√] diterima [ ] ditolak

5. Pengujian Setting

Pengujian dari setting dengan data *valid* kemudian diamati dan menghasilkan sebuah kesimpulan. Pengujian setting dapat dilihat pada tabel 4.32.

Tabel 4. 32 Pengujian Setting

Kasus dan hasil uji (data *valid*)			
Data masukan	***Valid* class**	**Pengamatan**	**Kesimpulan**
Setting tipe anak yang dipilih Contoh masukan : SMP	Pilih [tipe anak…]	Menginputkan data tipe anak	[√] diterima [ ] ditolak

Pengujian dari setting dengan data *invalid* kemudian diamati dan menghasilkan sebuah kesimpulan. Pengujian setting dapat dilihat pada tabel 4.33.

Tabel 4. 33 Pengujian Setting dengan Data *Invalid*

Kasus dan hasil uji (data *invalid*)			
Data masukan	***Valid* class**	**Pengamatan**	**Kesimpulan**
Setting tipe anak yang dipilih Contoh masukan : SMP	Tidak dipilih	Tidak menampilkan menu halaman berikutnya, karena *field* tersebut belum dipilih	[√] diterima [ ] ditolak

4.2.3 Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian sistem yag dilakukan secara menyeluruh telah memberikan suatu kesimpulan bahwa proses pada sistem monitoring ini telah melalui tahap perbaikan dan sudah dioptimalkan terhadap proses – proses tersebut dan secara fungsional sistem monitoring ini sudah dapat digunakan dan menghasilkan respon sistem yang diharapkan.

4.2.4 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif dimana sistem yang dibangun dapat diuji secara langsung ke pengguna yaitu Yayasan dan Petugas Dinsos dengan melakukan wawancara.

4.2.4.1 Wawancara

Wawancara dilakukan kepada petugas dinsos dan yayasan pada sistem informasi monitoring ini.

a. Pengujian Beta Wawancara dengan Petugas

Berikut adalah pertanyaan yang diajukan kepada Petugas dan jawaban yang telah diberikan.

1. Apakah website Sistem Informasi Monitoring Pemberian Bantuan Kepada Anak Terlantar ini dapat mengintegrasikan data pada tiap-tiap yayasan yang ada di Kota Bandung?

Jawaban :

Ya, sistem ini dapat mengurangi data yang sama/distorsi data. Dengan adanya sistem ini maka kemungkinan data anak yang sama tidak akan terjadi.

2. Apakah pengolahan data pengguna dalam website Sistem Informasi Monitoring Pemberian Bantuan Kepada Anak Terlantar ini dapat membantu Dinas Sosial dalam hal pengumpulan informasi pertanggunawaban penggunaan dana pada tiap-tiap yayasan?

   Jawaban :

   Sistem ini membantu dinas sosial khususnya bidang PRS anak dan remaja untuk membantu mengawasi bantuan apakah tepat sasaran atau tidak. Sistem ini juga membantu karena dapat memasukan foto agar dapat memantau seperti apa di lapangan.

3. Apakah website Sistem Informasi Monitoring Pemberian Bantuan Kepada Anak Terlantar ini dapat membantu Dinas Sosial dalam memonitoring pencapaian anak yang ada pada tiap-tiap yayasan yang ada di Kota Bandung?

   Jawaban :

   Sistem ini memberikan kemudahan kepada Dinas Sosial khususnya bidang PRS anak dan remaja dalam hal pencapaian anak yang menerima bantuan dalam hal akademik, yang juga mempermudah dalam pembuatan laporan..

Dari hasil wawancara yang dilakukan dapat ditarik kesimpulan bahwa website Sistem Informasi Monitoring Pemberian Bantuan Kepada Anak Terlantar ini dapat membantu untuk melakukan pembuatan laporan, pertanggungjawaban penggunaan dana, monitoring anak dan membantu dalam pengolahan yayasan yang mudah digunakan.