BAB IV
IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini akan membahas implementasi dan pengujian terhadap sistem yang dibangun. Tahap ini dilakukan setelah perancangan selesai dilakukan dan selanjutnya akan diimplementasikan pada bahasa pemrograman yang digunakan. Setelah implementasi dilakukan, selanjutnya akan dilakukan pengujian sistem dimana akan dilihat kekurangan pada sistem yang dibangun untuk pengembangan sistem selanjutnya.Pengujian yang dilakukan adalah pengujian black box, dan pengujian beta.

IV.1 Implementasi Sistem

Tahap implementasi sistem ini bertujuan untuk menerapkan perancangan yang telah dibuat pada bab sebelumnya ke sistem yang sebenarnya agar bisa digunakan. Tahap implementasi sistem ini dibagi menjadi implementasi basis data dan implementasi antarmuka.

IV.1.1 Implementasi Basis Data

Implementasi ini merupakan tahap untuk pembuatan *database* yang sebelumnya telah dirancang dimana akan diimplementasikan secara langsung pada sistem yang dibangun. Implementasi *database* dalam SQL adalah sebagai berikut :

Tabel IV. 1 Implementasi Basis Data

No	Deskripsi	Syntax
1.	Membuat Basis Data	CREATE DATABASE `distanhut` ;
2.	Membuat Tabel Bantuan Pemerintah	CREATE TABLE `bantuan_pemerintah` (`id_bantuan` int(10) NOT NULL AUTO_INCREMENT, `nama` varchar(50) NOT NULL, `id_jenis` int(11) NOT NULL, `jumlah_tersedia` int(11) NOT NULL,

		`satuan` enum('KILOGRAM','EKOR','UNIT','BATANG','BUNGKUS','BOTOL') DEFAULT NULL, PRIMARY KEY (`id_bantuan`), KEY `bantuan_pemerintah_ibfk_1` (`id_jenis`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 ;
3.	Membuat Tabel Daemon	CREATE TABLE `daemons` (`Start` text NOT NULL, `Info` text NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8;
4.	Membuat Tabel Desa	CREATE TABLE `desa` (`id_desa` int(10) NOT NULL AUTO_INCREMENT, `nama_desa` varchar(20) DEFAULT NULL, `id_kecamatan` int(10) DEFAULT NULL, PRIMARY KEY (`id_desa`), KEY `id_kecamatan` (`id_kecamatan`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
5.	Membuat Tabel Gammu	CREATE TABLE IF NOT EXISTS `gammu` (`Version` int(11) NOT NULL DEFAULT '0') ENGINE=InnoDB DEFAULT CHARSET=utf8;
6.	Membuat Tabel hibah	CREATE TABLE `hibah` (`id_hibah` varchar(30) NOT NULL, `tanggal` date NOT NULL, `id_staf_bertugas` int(11) NOT NULL, `id_petani` int(11) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1;
7.	Membuat Tabel Hibah_detail	CREATE TABLE `hibah_detail` (`id_hibah` varchar(30) NOT NULL, `id_bantuan` int(11) NOT NULL, `jumlah_keluar` int(11) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1;
8.	Membuat Tabel Hibah_temp	CREATE TABLE `hibah_temp` (`id_bantuan` int(11) NOT NULL, `nama` varchar(30) NOT NULL, `jumlah_keluar` int(11) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1;
9.	Membuat Tabel inbox	CREATE TABLE `inbox` (`UpdatedInDB` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP, `ReceivingDateTime` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00', `Text` text NOT NULL, `SenderNumber` varchar(20) NOT NULL DEFAULT '',

		`Coding` enum('Default_No_Compression','Unicode_No_Compression','8bit','Default_Compression','Unicode_Compression') NOT NULL DEFAULT 'Default_No_Compression', `UDH` text NOT NULL, `SMSCNumber` varchar(20) NOT NULL DEFAULT '', `Class` int(11) NOT NULL DEFAULT '-1', `TextDecoded` text NOT NULL, `ID` int(10) unsigned NOT NULL AUTO_INCREMENT, `RecipientID` text NOT NULL, `Processed` enum('false','true') NOT NULL DEFAULT 'false', PRIMARY KEY (`ID`)) ENGINE=InnoDB DEFAULT CHARSET=utf8;
10.	Membuat Tabel Jenis	CREATE TABLE `jenis` (`id_jenis` int(11) NOT NULL AUTO_INCREMENT, `nama_jenis` varchar(30) NOT NULL, PRIMARY KEY (`id_jenis`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
11.	Membuat Tabel Kategori	CREATE TABLE `kategori` (`id_kategori` int(5) NOT NULL AUTO_INCREMENT, `nama_kategori` varchar(30) NOT NULL, PRIMARY KEY (`id_kategori`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
12.	Membuat Tabel Kecamatan	CREATE TABLE `kecamatan` (`id_kecamatan` int(10) NOT NULL AUTO_INCREMENT, `nama_kecamatan` varchar(20) DEFAULT NULL, PRIMARY KEY (`id_kecamatan`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
13.	Membuat Tabel Kelompok	CREATE TABLE `kelompok` (`id_kelompok` int(11) NOT NULL AUTO_INCREMENT, `nama_kelompok` varchar(30) DEFAULT NULL, PRIMARY KEY (`id_kelompok`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
14.	Membuat Tabel outbox	CREATE TABLE IF NOT EXISTS `outbox` (`UpdatedInDB` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP, `InsertIntoDB` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',

		`SendingDateTime` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00', `SendBefore` time NOT NULL DEFAULT '23:59:59', `SendAfter` time NOT NULL DEFAULT '00:00:00', `Text` text, `DestinationNumber` varchar(20) NOT NULL DEFAULT '', `Coding` enum('Default_No_Compression','Unicode_No_Compression','8bit','Default_Compression','Unicode_Compression') NOT NULL DEFAULT 'Default_No_Compression', `UDH` text, `Class` int(11) DEFAULT '-1', `TextDecoded` text NOT NULL, `ID` int(10) unsigned NOT NULL AUTO_INCREMENT, `MultiPart` enum('false','true') DEFAULT 'false', `RelativeValidity` int(11) DEFAULT '-1', `SenderID` varchar(255) DEFAULT NULL, `SendingTimeOut` timestamp NULL DEFAULT '0000-00-00 00:00:00', `DeliveryReport` enum('default','yes','no') DEFAULT 'default', `CreatorID` text NOT NULL, PRIMARY KEY (`ID`)) ENGINE=InnoDB DEFAULT CHARSET=utf8;
15	Membuat Tabel Outbox_multipart	CREATE TABLE IF NOT EXISTS `outbox_multipart` (`Text` text, `Coding` enum('Default_No_Compression','Unicode_No_Compression','8bit','Default_Compression','Unicode_Compression') NOT NULL DEFAULT 'Default_No_Compression', `UDH` text, `Class` int(11) DEFAULT '-1', `TextDecoded` text, `ID` int(10) unsigned NOT NULL DEFAULT '0', `SequencePosition` int(11) NOT NULL DEFAULT '1', PRIMARY KEY (`SequencePosition`)) ENGINE=InnoDB DEFAULT CHARSET=utf8;
16	Membuat Tabel Pengadaan	CREATE TABLE IF NOT EXISTS `pengadaan` (`id_pengadaan` varchar(30) NOT NULL,

		`tanggal` date NOT NULL, `id_staf_bertugas` int(11) NOT NULL, `sumber_dana` enum('KABUPATEN','PROVINSI','PUSAT','') NOT NULL, PRIMARY KEY (`id_pengadaan`), KEY `id_staf_bertugas` (`id_staf_bertugas`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
17.	Membuat Tabel sentitems_detail	CREATE TABLE IF NOT EXISTS `sentitems_detail` (`id_sentitems` int(11) NOT NULL, `id_staf` int(11) NOT NULL, KEY `id_sentitems` (`id_sentitems`), KEY `id_staf` (`id_staf`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
17	Membuat Tabel Pengadaan_ detail	CREATE TABLE IF NOT EXISTS `pengadaan_detail` (`id_pengadaan` varchar(30) NOT NULL, `id_bantuan` int(11) NOT NULL, `jumlah_masuk` int(11) NOT NULL, KEY `id_pengadaan` (`id_pengadaan`), KEY `id_bantuan` (`id_bantuan`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
18.	Membuat Tabel Pengadaan_ temp	CREATE TABLE IF NOT EXISTS `pengadaan_temp` (`id_bantuan` int(11) NOT NULL, `nama` varchar(20) NOT NULL, `jumlah_masuk` int(11) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1;
19	Membuat Tabel Petani	CREATE TABLE IF NOT EXISTS `petani` (`id_petani` int(11) NOT NULL AUTO_INCREMENT, `id_kelompok` int(11) NOT NULL, `nik` varchar(20) NOT NULL, `nama_petani` varchar(40) NOT NULL, `no_hp_petani` varchar(15) NOT NULL, `id_desa` int(10) NOT NULL, `luas_lahan` varchar(50) NOT NULL, PRIMARY KEY (`id_petani`), KEY `id_kelompok` (`id_kelompok`), KEY `id_desa` (`id_desa`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=6 ;
20	Membuat Tabel Phones	CREATE TABLE IF NOT EXISTS `phones` (`ID` text NOT NULL,

		`UpdatedInDB` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP, `InsertIntoDB` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00', `TimeOut` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00', `Send` enum('yes','no') NOT NULL DEFAULT 'no', `Receive` enum('yes','no') NOT NULL DEFAULT 'no', `IMEI` varchar(35) NOT NULL, `Client` text NOT NULL, `Battery` int(11) NOT NULL DEFAULT '-1', `Signal` int(11) NOT NULL DEFAULT '-1', `Sent` int(11) NOT NULL DEFAULT '0', `Received` int(11) NOT NULL DEFAULT '0') ENGINE=InnoDB DEFAULT CHARSET=utf8;
21	Membuat Tabel Sentitems	CREATE TABLE IF NOT EXISTS `sentitems` (`UpdatedInDB` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP, `InsertIntoDB` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00', `SendingDateTime` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00', `DeliveryDateTime` timestamp NULL DEFAULT NULL, `Text` text NOT NULL, `DestinationNumber` varchar(20) NOT NULL DEFAULT '', `Coding` enum('Default_No_Compression','Unicode_No_Compression','8bit','Default_Compression','Unicode_Compression') NOT NULL DEFAULT 'Default_No_Compression', `UDH` text NOT NULL, `SMSCNumber` varchar(20) NOT NULL DEFAULT '', `Class` int(11) NOT NULL DEFAULT '-1', `TextDecoded` text NOT NULL, `ID` int(10) unsigned NOT NULL DEFAULT '0', `SenderID` varchar(255) NOT NULL, `SequencePosition` int(11) NOT NULL DEFAULT '1', `Status` enum('SendingOK','SendingOKNoReport','SendingErr

		or','DeliveryOK','DeliveryFailed','DeliveryPending','DeliveryUnknown','Error') NOT NULL DEFAULT 'SendingOK', `StatusError` int(11) NOT NULL DEFAULT '-1', `TPMR` int(11) NOT NULL DEFAULT '-1', `RelativeValidity` int(11) NOT NULL DEFAULT '-1', `CreatorID` text NOT NULL, PRIMARY KEY (`ID`)) ENGINE=InnoDB DEFAULT CHARSET=utf8;
22	Membuat Tabel Staf_ distanhut	CREATE TABLE IF NOT EXISTS `staf_distanhut` (`id_staf` int(10) NOT NULL AUTO_INCREMENT, `password` varchar(50) NOT NULL, `nama` varchar(50) NOT NULL, `nip` varchar(30) NOT NULL, `level` enum('ADMINISTRATOR','STAF') DEFAULT NULL, `jenis_kelamin` enum('LAKI-LAKI','PEREMPUAN') NOT NULL, `email` varchar(50) NOT NULL, `no_ponsel_staf` varchar(50) NOT NULL, `id_desa` int(11) DEFAULT NULL, PRIMARY KEY (`id_staf`), UNIQUE KEY `email` (`email`), KEY `id_desa` (`id_desa`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;

IV.1.2 Implementasi Antarmuka

Implementasi antarmuka bertujuan untuk mengetahui setiap tampilan yang telah dibangun dan pengkodeannya ke dalam bentuk file program, sehingga akan diketahui apakah sistem yang telah dibuat benar-benar sesuai dengan yang direncanakan. Berikut implementasi antarmuka beserta deskripsinya

Tabel IV. 2 Implementasi Antarmuka Administrator

No	Nama Antarmuka	Deskripsi	Nama File
1.	Login Admin	Antarmuka yang digunakan oleh Penguna untuk login ke sistem	formlogin.php
2.	Index administrator	Tampilan awal administrator jika berhasil login	Admin/index.php

3.	Kelola Staf	Antarmuka yang digunakan untuk melihat daftar staf dan mengelola (tambah, ubah, hapus) staf DISTANHUT	staf_kelola.php
4.	Tambah Staf	Antarmuka untuk menambah data staf DISTANHUT	staf_tambah.php
5.	Ubah Staf	Antarmuka untuk mengubah data staf DISTANHUT	staf_ubah.php
6.	Ubah data diri	Antarmuka untuk mengubah data diri sendiri (admin login)	diri_ubah.php
7.	*Backup* dan *Restore*	Antarmuka untuk backup dan restore basis data	Backuprestore.php

Tabel IV. 3 Implementasi Antarmuka Staf DISTANHUT

No	Nama Antarmuka	Deskripsi	Nama File
1.	Login Staf	Antarmuka yang digunakan oleh Penguna untuk login ke sistem	formlogin.php
2.	Index Staf	Tampilan awal administrator jika berhasil login	index.php
3.	Kelola Bantuan	Antarmuka yang digunakan untuk melihat daftar bantuan pemerintah dan mengelola (tambah, ubah, hapus) bantuan	bantuan_kelola.php
4.	Tambah Bantuan	Antarmuka untuk menambah data bantuan pemerintah	bantuan_tambah.php
5.	Ubah Bantuan	Antarmuka untuk mengubah data bantuan pemerintah	bantuan_ubah.php
6	Kelola Petani	Antarmuka yang digunakan untuk melihat dan mengelola (tambah, ubah, hapus) daftar Petani	petani_kelola.php
7	Tambah Petani	Antarmuka untuk menambah data petani	petani_tambah.php
8	Ubah Petani	Antarmuka untuk mengubah data petani	petani_ubah.php
9	Kelola Pengadaan	Antarmuka yang digunakan untuk melihat	pengadaan_kelola.php

		dan mengelola (tambah, hapus) pengadaan bantuan	
10	Tambah Pengadaan	Antarmuka untuk menambah data pengadaan	pengadaan_tambah.php
11	Kelola Hibah	Antarmuka yang digunakan untuk melihat dan mengelola (tambah, hapus) hibah bantuan	hibah_kelola.php
12	Tambah Hibah	Antarmuka untuk menambah data hibah	hibah_tambah.php
13	Kelola jenis Bantuan	Antarmuka yang digunakan untuk melihat daftar jenis bantuan pemerintah dan mengelola (tambah, ubah, hapus) jenis bantuan	jenis_kelola.php
14	Tambah Bantuan	Antarmuka untuk menambah data jenis bantuan pemerintah	jenis_tambah.php
15	Ubah Bantuan	Antarmuka untuk mengubah data jenis bantuan pemerintah	jenis_ubah.php
16	Kelola Kelompok Petani	Antarmuka yang digunakan untuk melihat dan mengelola (tambah, ubah, hapus) data kelompok petani	kelompok_kelola.php
17	Tambah Kelompok Petani	Antarmuka untuk menambah data kelompok petani	kelompok_tambah.php
18	Ubah Kelompok Petani	Antarmuka untuk mengubah data kelompok petani	kelompok_ubah.php
19	Kelola Kecamatan	Antarmuka yang digunakan untuk melihat dan mengelola (tambah, ubah, hapus) data kecamatan	kecamatan_kelola.php
20	Tambah Kecamatan	Antarmuka untuk menambah data kecamatan	kecamatan_tambah.php
21	Ubah Kecamatan	Antarmuka untuk mengubah data kecamatan	kecamatan_ubah.php
22	Kelola Desa	Antarmuka yang digunakan untuk melihat dan mengelola (tambah, ubah, hapus) data desa	desa_kelola.php

23	Tambah Desa	Antarmuka untuk menambah data desa	desa _tambah.php
24	Ubah Desa	Antarmuka untuk mengubah data desa	desa _ubah.php
25	Kirim pesan	Antarmuka untuk mengirim pesan ke petani	kirim_sms.php
26	Kotak masuk	Antarmuka menampilkan daftar pesan masuk	terima_sms.php
27	Kotak Keluar	Antar muka menampilkan pesan keluar	terkirim_sms.php
28	Balas pesan	Antarmuka untuk membalas pesak masuk	balas_sms.php
29	Laporan Hibah	Antarmuka untuk menampilkan laporan hibah	laporan_hibah.php
30	Laporan pesan	Antarmuka yang menampilkan grafik pesan masuk	laporan_pesan.php

IV.2 Pengujian

Pengujian sistem dilakukan untuk menguji sistem dimana pada tahap ini dapat diketahui kesalahan - kesalahan sehingga dapat dilakukan perbaikan dan pengembangan sistem maka dapat diketahui kesesuaian sistem dengan kebutuhannya. Pengujian yang digunakan untuk menguji sistem yang baru adalah dengan menggunakan metode pengujian *black box*. Pada pengujian *black box* lebih fokus pada persyaratan fungsional perangkat lunak dan pengujian ini berusaha menemukan kesalahan dalam kategori sebagai berikut :

1. Kesalahan interface
2. Fungsi - fungsi yang tidak sesuai dengan prosedur
3. Kesalahan kinerja sistem

IV.2.1 Rencana Pengujian *Black Box*

Skenarion pengujian digunakan untuk menjelaskan pengujian terhadap suatu sistem. Skenario ini memaparkan urutan dan hal yang akan diuji yang dilakukan pada sistem. Adapun skenario pengujian yang dilakukan pada sistem ini adalah sebagai berikut.

Tabel IV. 4 Rencana Pengujian

Kasus Uji	**Detail Pengujian**	**Jenis Pengujian**
Login	Validasi Data login	*Black Box*
	Verifikasi Data login	*Black Box*
Kelola Data Staf	Tambah Staf	*Black Box*
	Ubah Staf	*Black Box*
	Hapus Staf	*Black Box*
Backup dan *Restore* basis data	Backup basis data	*Black Box*
	Restore basis data	*Black Box*
Data Bantuan Pemerintah	Tambah Bantuan Pemerintah	*Black Box*
	Ubah Bantuan Pemerintah	*Black Box*
	Hapus Bantuan Pemerintah	*Black Box*
Data Petani	Tambah Petani	*Black Box*
	Ubah Petani	*Black Box*
	Hapus Petani	*Black Box*
Data Pengadaan	Tambah Pengadaan	*Black Box*
	Hapus Pengadaan	*Black Box*
Data Hibah	Tambah Hibah	*Black Box*
	Hapus Hibah	*Black Box*
Laporan Hibah	Pratinjau Laporan Hibah	*Black Box*
	Cetak Laporan Hibah	*Black Box*
Laporan Pesan Masuk	Pratinjau Laporan Pesan Masuk	*Black Box*
	Cetak Laporan Pesan Masuk	*Black Box*
SMS *Gateway*	Kirim Pesan ke Petani	*Black Box*
	Terima Pesan dari Petani	*Black Box*
	Autoreply pesan dari Petani	*Black Box*
Data Jenis Bantuan	Tambah Jenis Bantuan	*Black Box*
	Ubah Jenis Bantuan	*Black Box*
	Hapus Jenis Bantuan	*Black Box*
Data Kelompok Tani	Tambah Kelompok Tani	*Black Box*

	Ubah Kelompok Tani	*Black Box*
	Hapus Kelompok Tani	*Black Box*
Data Desa	Tambah Desa	*Black Box*
	Ubah Desa	*Black Box*
	Hapus Desa	*Black Box*
Data Kecamatan	Tambah Kecamatan	*Black Box*
	Ubah Kecamatan	*Black Box*
	Hapus Kecamatan	*Black Box*

IV.2.2 Kasus dan Hasil Pengujian *Black Box*

Berdasarkan pada rencana pengujian blackbox maka dilakukan pengujian dan didapatkan hasil dari pengujian blackbox sebagai berikut.

IV.2.2.1 Pengujian Login

Berikut dapat dilihat pengujian dari login administrator dan staf DISTANHUT dengan kasus dan hasil uji kemudian diamati hasil dari pengujian tersebut.

Tabel IV. 5 Pengujian *Login*

Kasus dan Hasil Uji (Data Valid)			
Data masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
username : admin password : admin atau username : staf1 password : staf	e-mail dan *password* yang dimasukkan akan tampil pada masing-masing kolom isian. Pengguna akan dialihkan kehalaman utama sesuai hak akses	e-mail dan *password* tampil pada kolom isian Pengguna dialihkan kehalaman utama sesuai hak akses	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Invalid)			

username : admin *password* : 123	Gagal login, menampilkan pesan kesalahan dan kembali ke halaman login	Tampil pesan kesalahan "email atau password salah "dan kembali ke halaman login	[✓] Diterima [ ] Ditolak
e-mail : (kosong) *password* : (kosong)	Gagal login, Menampilkan pesan kolom isian wajib	Menampilkan pesan kolom isian wajib diisi	[✓] Diterima [ ] Ditolak

IV.2.2.2 Pengujian Kelola Data Staf

Berikut dapat dilihat pengujian kelola data staf oleh administrator dengan kasus dan hasil uji kemudian diamati hasil dari pengujian tersebut.

Tabel IV. 6 Pengujian Kelola Data Staf

Kasus dan Hasil Uji (Data Valid)			
Data masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih Menu Kelola Staf	Administrator dapat masuk ke halaman pengelolaan data staf	Dapat masuk ke halaman pengelolaan data staf	[✓] Diterima [ ] Ditolak
Pilih *Button* Tambah Staf Masukkan Data Nama Staf : Hasbullah Nip : 19600202 198603 1 003 Jenis Kelamin : Laki-laki Level : Administrator Nomor Ponsel : 083820120199	Menampilkan form Tambah staf Jika pengisian data benar (tidak ada duplikasi pada NIP, e-mail,no hp) maka akan meyimpan data dan menampilkan daftar staf	Menampilkan form tambah staf Dapat menyimpan data dan menampilkan daftar staf	[✓] Diterima [ ] Ditolak

username: staf1 Password : staf			
Pilih *Button* Ubah pada salah satu Staf (Contoh : Hasbullah) Ubah Data Nama Staf : Hasbullah Akbar Nip : 19600202 198603 1 003 Jenis Kelamin : Laki-laki Level : Administrator Nomor Ponsel : 083820120199 username: staf Password : staf12	Menampilkan form Ubah staf Jika pengisian data benar (tidak ada duplikasi pada NIP, e-mail,no hp) maka akan meyimpan data dan menampilkan daftar staf	Dapat menampilkan form ubah staf Dapat menyimpan data dan menmapilkan daftar staf	[✓] Diterima [ ] Ditolak
Pilih *Button* hapus pada salah satu staf	Menghapus data	Data terhapus	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Invalid)			
Tambah data dengan kolom isian kosong	Gagal menambahkan data, menampilkan pesan ada field yang wajib diisi	Tampil pesan kesalahan "field harus diisi"	[✓] Diterima [ ] Ditolak
Tambahkan data dengan duplikasi (NIP, nomor HP, e-mail)	Gagal menambahkan data, menampilkan pesan kesalahan	Menampilkan pesan kesalahan, tidak bisa menambahkan data	[✓] Diterima [ ] Ditolak

IV.2.2.3 Pengujian Backup dan Restore

Berikut dapat dilihat pengujian dari login administrator dan staf DISTANHUT dengan kasus dan hasil uji kemudian diamati hasil dari pengujian tersebut.

Tabel IV. 7 Pengujian *Backup* dan *Restore*

Kasus dan Hasil Uji (Data Valid)			
Data masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih menu *Backup & Restore Database* Pilih Backup Pilih *Button* Proses Backup	Menampilkan halaman backup & *Restore* Menampilkan menu backup membackup basis data	Dapat menampilkan halaman backup & *Restore* Dapat menampilkan menu backup Dapat membackup basis data	[✓] Diterima [ ] Ditolak
Pilih menu *Backup & Restore Database* Pilih Restore Pilih *File* Pilih *Button* Proses Restore	Menampilkan halaman backup & *Restore* Menampilkan menu restore Merestore basis data	Dapat menampilkan halaman backup & *Restore* Dapat Menampilkan menu restore Dapat merestore basis data	[✓] Diterima [ ] Ditolak

IV.2.2.4 Pengujian Kelola Data Bantuan Pemerintah

Berikut dapat dilihat pengujian kelola data bantuan pemerintah oleh staf DISTANHUT dengan kasus dan hasil uji kemudian diamati hasil dari pengujian tersebut.

Tabel IV. 8 Pengujian Kelola Data Bantuan Pemerintah

Kasus dan Hasil Uji (Data Valid)			
Data masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih Menu Kelola Bantuan Pemerintah	Staf dapat masuk ke halaman pengelolaan data bantuan pemerintah	Dapat masuk ke halaman pengelolaan data bantuan pemerintah	[✓] Diterima [ ] Ditolak
Pilih *Button* Tambah Bantuan Masukkan Data Nama : Itik Satuan : Ekor Jenis : Bibit Jumlah: 7010	Menampilkan form Tambah Bantuan Jika pengisian data benar (tidak ada duplikasi pada nama) maka akan meyimpan data dan menampilkan pesan "Berhasil Menambahkan Data"	Menampilkan form tambah bantuan Dapat menyimpan data dan menampilkan pesan "Berhasil Menambahkan Data"	[✓] Diterima [ ] Ditolak
Pilih *Button* ubah pada bantuan (misal "itik") Masukkan Data Nama : Itik Petelur Satuan : Ekor Jenis : Bibit Jumlah: 7010	Menampilkan form Ubah Bantuan Jika pengisian data benar (tidak ada duplikasi nama) pengubahan data akan dilakukan dan menampilkan pesan "Berhasil Mengubah Data!"	Dapat menampilkan form ubah staf Dapat mengubah data dan menampilkan pesan "Berhasil Mengubah Data!"	[✓] Diterima [ ] Ditolak
Pilih *Button* hapus pada salah satu bantuan pemerintah	Menghapus data	Data terhapus	[✓] Diterima [ ] Ditolak

Kasus dan Hasil Uji (Data Invalid)			
Tambah data dengan kolom isian kosong	Gagal menambahkan data, menampilkan pesan ada field yang wajib diisi	Tampil pesan kesalahan "field harus diisi"	[✓] Diterima [ ] Ditolak
Tambahkan data dengan duplikasi (Nama)	Gagal menambahkan data, menampilkan pesan kesalahan	Menampilkan pesan kesalahan, tidak bisa menambahkan data	[✓] Diterima [ ] Ditolak

IV.2.2.5 Pengujian Kelola Data Petani

Berikut dapat dilihat pengujian kelola data petani oleh staf DISTANHUT dengan kasus dan hasil uji kemudian diamati hasil dari pengujian tersebut.

Tabel IV. 9 Pengujian Kelola Data Petani

Kasus dan Hasil Uji (Data Valid)			
Data masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih Menu Kelola Petani	Staf dapat masuk ke halaman pengelolaan data petani	Dapat masuk ke halaman pengelolaan data petani	[✓] Diterima [ ] Ditolak
Pilih *Button* Tambah Petani Masukkan Data Nama Kelompok : Setia Kawan NIK : 1801132412740004 Nama : Bastiar No Ponsel: 087829445566 Domisili: Dendang	Menampilkan form Tambah Petani Jika pengisian data benar (tidak ada duplikasi pada NIK dan No Ponsel) maka akan meyimpan data dan menampilkan pesan "Berhasil Menambahkan Data"	Menampilkan form tambah petani Dapat menyimpan data dan menampilkan pesan "Berhasil Menambahkan Data"	[✓] Diterima [ ] Ditolak

Luas Lahan (Ha): 5			
Pilih *Button* ubah pada bantuan (misal "BASTIAR") Masukkan Data Nama Kelompok : Setia Kawan NIK : 1801132412740004 Nama : Bastiar Putra No Ponsel: 087829445566 Domisili: Dendang Luas Lahan (Ha): 15	Menampilkan form Ubah Bantuan Jika pengisian data benar (tidak ada duplikasi NIK dan nomor ponsel) pengubahan data akan dilakukan dan menampilkan pesan "Berhasil Mengubah Data!"	Dapat menampilkan form ubah staf Dapat mengubah data dan menampilkan pesan "Berhasil Mengubah Data!"	[✓] Diterima [ ] Ditolak
Pilih *Button* hapus pada salah satu petani	Menghapus data	Data terhapus	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Invalid)			
Tambah data dengan kolom isian kosong	Gagal menambahkan data, menampilkan pesan ada field yang wajib diisi	Tampil pesan kesalahan "field harus diisi"	[✓] Diterima [ ] Ditolak
Tambahkan data dengan duplikasi (NIK, No Ponsel)	Gagal menambahkan data, menampilkan pesan kesalahan	Menampilkan pesan kesalahan, tidak bisa menambahkan data	[✓] Diterima [ ] Ditolak

IV.2.2.6 Pengujian Kelola Data Pengadaan

Berikut dapat dilihat pengujian kelola data pengadaan oleh staf DISTANHUT dengan kasus dan hasil uji kemudian diamati hasil dari pengujian tersebut.

Tabel IV. 10 Pengujian Kelola Data Pengadaan

Kasus dan Hasil Uji (Data Valid)			
Data masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih Menu Kelola Pengadaan	Staf dapat masuk ke halaman pengelolaan pengadaan	Dapat masuk ke halaman pengelolaan pengadaan	[✓] Diterima [ ] Ditolak
Pilih *Button* Tambah Pengadaan Masukkan Data Nama Bantuan : ANVIL Jumlah Masuk: 10 Sumber Dana : Kabupaten Pilih *Button* Proses	Menampilkan form Tambah Pengadaan Jika pengisian data benar maka akan meyimpan data dan menampilkan pesan "Transaksi Berhasil"	Menampilkan form tambah pengadaan Dapat menyimpan data dan menampilkan pesan "Transaksi Berhasil"	[✓] Diterima [ ] Ditolak
Pilih *Button* hapus pada salah satu transaksi pengadaan	Menghapus data Menampilkan Pesan "Data Telah Dihapus"	Data terhapus Menampilkan Pesan "Data Telah Dihapus"	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Invalid)			
Tambah data dengan sumber dana kosong	Gagal menambahkan data, menampilkan pesan kesalahan	Tampil pesan kesalahan "Sumber dana harus diisi"	[✓] Diterima [ ] Ditolak

IV.2.2.7 Pengujian Kelola Data Hibah

Berikut dapat dilihat pengujian kelola data hibah oleh staf DISTANHUT dengan kasus dan hasil uji kemudian diamati hasil dari pengujian tersebut.

Tabel IV. 11 Pengujian Kelola Data Hibah

Kasus dan Hasil Uji (Data Valid)			
Data masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih Menu Kelola Hibah	Staf dapat masuk ke halaman pengelolaan pengadaan	Dapat masuk ke halaman pengelolaan pengadaan	[✓] Diterima [ ] Ditolak
Pilih *Button* Tambah Hibah Masukkan Data Nama Bantuan : ANTRACOL Jumlah Keluar: 10 Nama Bantuan : Sapi Jumlah Keluar: 5 Penerima : Bastiar Pilih *Button* Proses	Menampilkan form Tambah Pengadaan Jika pengisian data benar maka akan meyimpan data dan menampilkan pesan "Transaksi Berhasil"	Menampilkan form tambah pengadaan Dapat menyimpan data dan menampilkan pesan "Transaksi Berhasil"	[✓] Diterima [ ] Ditolak
Pilih *Button* hapus pada salah satu transaksi pengadaan	Menghapus data Menampilkan Pesan "Data Telah Dihapus"	Data terhapus Menampilkan Pesan "Data Telah Dihapus"	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Invalid)			

Tambah data dengan kolom isian kosong	Gagal menambahkan data, menampilkan pesan kesalahan	Tampil pesan kesalahan "penerima harus diisi"	[✓] Diterima [ ] Ditolak

IV.2.2.8 Pengujian Kelola Data Jenis Bantuan

Berikut dapat dilihat pengujian kelola data jenis bantuan oleh staf DISTANHUT dengan kasus dan hasil uji kemudian diamati hasil dari pengujian tersebut.

Tabel IV. 12 Pengujian Kelola Data Jenis Bantuan

Kasus dan Hasil Uji (Data Valid)			
Data masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih Menu Kelola Jenis Bantuan	Staf dapat masuk ke halaman pengelolaan data jenis bantuan	Dapat masuk ke halaman pengelolaan data jenis bantuan	[✓] Diterima [ ] Ditolak
Klik *Button* Tambah Jenis Masukkan Data Nama Jenis : Mesin	Menampilkan form Tambah Jenis Bantuan menampilkan pesan "Berhasil Menambahkan Data"	Menampilkan form tambah jenis Dapat menyimpan data dan menampilkan pesan "Berhasil Menambahkan Data"	[✓] Diterima [ ] Ditolak
Pilih *Button* ubah pada jenis bantuan (misal "Mesin") Masukkan Data Nama : Mesin Potong	Menampilkan form Ubah Bantuan menampilkan pesan "Berhasil Mengubah Data!"	Dapat menampilkan form ubah jenis Dapat mengubah data dan menampilkan pesan "Berhasil Mengubah Data!"	[✓] Diterima [ ] Ditolak
Pilih *Button* hapus pada salah satu jenis bantuan	Menghapus data	Data terhapus	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Invalid)			
Tambah data dengan kolom isian kosong	Gagal menambahkan data, menampilkan	Tampil pesan kesalahan "field harus diisi"	[✓] Diterima [ ] Ditolak

	pesan ada field yang wajib diisi		
Tambahkan data dengan duplikasi (Nama)	Gagal menambahkan data, menampilkan pesan kesalahan	Menampilkan pesan kesalahan, tidak bisa menambahkan data	[✓] Diterima [ ] Ditolak

IV.2.2.9 Pengujian Kelola Kelompok Tani

Berikut dapat dilihat pengujian kelola data petani oleh staf DISTANHUT dengan kasus dan hasil uji kemudian diamati hasil dari pengujian tersebut.

Tabel IV. 13 Pengujian Kelola Data Kelompok Tani

Kasus dan Hasil Uji (Data Valid)			
Data masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih Menu Kelola Petani	Staf dapat masuk ke halaman pengelolaan data kelompok tani	Dapat masuk ke halaman pengelolaan data kelompok tani	[✓] Diterima [ ] Ditolak
Pilih *Button* Tambah Kelompok Tani Masukkan Data Nama Kelompok : Setia Kawan	Menampilkan form Tambah kelompok tani Jika pengisian data benar (tidak ada duplikasi) maka akan meyimpan data dan menampilkan pesan "Berhasil Menambahkan Data"	Menampilkan form tambah kelompok tani Dapat menyimpan data dan menampilkan pesan "Berhasil Menambahkan Data"	[✓] Diterima [ ] Ditolak
Pilih *Button* ubah pada bantuan (misal "Setia Kawan") Ubah Nama Menjadi : Setia Kawan Lama	Menampilkan form ubah kelompok tani Jika pengisian data benar (tidak ada duplikasi Nama) pengubahan data akan dilakukan dan menampilkan pesan "Berhasil Mengubah Data!"	Dapat menampilkan form ubah kelompok tani Dapat mengubah data dan menampilkan pesan "Berhasil Mengubah Data!"	[✓] Diterima [ ] Ditolak
Pilih *Button* hapus pada salah satu petani	Menghapus data	Data terhapus	[✓] Diterima

			[ ] Ditolak
Kasus dan Hasil Uji (Data Invalid)			
Tambah data dengan kolom isian kosong	Gagal menambahkan data, menampilkan pesan ada field yang wajib diisi	Tampil pesan kesalahan "field harus diisi"	[✓] Diterima [ ] Ditolak
Tambahkan data dengan duplikasi (Nama)	Gagal menambahkan data, menampilkan pesan kesalahan	Menampilkan pesan kesalahan, tidak bisa	[✓] Diterima [ ] Ditolak

IV.2.2.10 Pengujian Kelola Desa

Berikut dapat dilihat pengujian kelola data desa oleh staf DISTANHUT dengan kasus dan hasil uji kemudian diamati hasil dari pengujian tersebut.

Tabel IV. 14 Pengujian Kelola Desa

Kasus dan Hasil Uji (Data Valid)			
Data masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih Menu Kelola Desa	Staf dapat masuk ke halaman pengelolaan desa	Dapat masuk ke halaman pengelolaan desa	[✓] Diterima [ ] Ditolak
Klik *Button* Tambah desa Masukkan Data Nama Desa: Dendang	Menampilkan form tambah desa menampilkan pesan "Berhasil Menambahkan Data"	Menampilkan form tambah desa Dapat menyimpan data dan menampilkan pesan "Berhasil Menambahkan Data"	[✓] Diterima [ ] Ditolak
Pilih *Button* ubah pada desa (misal "Dendang") Masukkan Data Nama : Dendang Baru	Menampilkan form ubah desa menampilkan pesan "Berhasil Mengubah Data!"	Dapat menampilkan form ubah desa Dapat mengubah data dan menampilkan pesan "Berhasil Mengubah Data!"	[✓] Diterima [ ] Ditolak
Pilih *Button* hapus pada salah satu desa	Menghapus data	Data terhapus	[✓] Diterima

			[ ] Ditolak
Kasus dan Hasil Uji (Data Invalid)			
Tambah data dengan kolom isian kosong	Gagal menambahkan data, menampilkan pesan ada field yang wajib diisi	Tampil pesan kesalahan "field harus diisi"	[✓] Diterima [ ] Ditolak

IV.2.2.11 Pengujian Kelola Kecamatan

Berikut dapat dilihat pengujian kelola data kecamatan oleh staf DISTANHUT dengan kasus dan hasil uji kemudian diamati hasil dari pengujian tersebut.

Tabel IV. 15 Pengujian Kelola Kecamatan

Kasus dan Hasil Uji (Data Valid)			
Data masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih Menu Kelola Kecamatan	Staf dapat masuk ke halaman pengelolaan kecamatan	Dapat masuk ke halaman pengelolaan kecamatan	[✓] Diterima [ ] Ditolak
Klik *Button* Tambah kecamatan Masukkan Data Nama Kecamatan: Manggar	Menampilkan form tambah kecamatan menampilkan pesan "Berhasil Menambahkan Data"	Menampilkan form tambah kecamatan Dapat menyimpan data dan menampilkan pesan "Berhasil Menambahkan Data"	[✓] Diterima [ ] Ditolak
Pilih *Button* ubah pada kecamatan (misal "Manggar") Masukkan Data Nama : Manggar Baru	Menampilkan form ubah kecamatan menampilkan pesan "Berhasil Mengubah Data!"	Dapat menampilkan form ubah kecamatan Dapat mengubah data dan menampilkan pesan "Berhasil Mengubah Data!"	[✓] Diterima [ ] Ditolak

Pilih *Button* hapus pada salah satu kecamatan	Menghapus data	Data terhapus	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Invalid)			
Tambah data dengan kolom isian kosong	Gagal menambahkan data, menampilkan pesan ada field yang wajib diisi	Tampil pesan kesalahan "field harus diisi"	[✓] Diterima [ ] Ditolak

IV.2.2.12 Pengujian SMS

Berikut dapat dilihat pengujian sms (autoreply, pemesanan) oleh staf DISTANHUT dengan kasus dan hasil uji kemudian diamati hasil dari pengujian tersebut.

Tabel IV. 16 Pengujian autoreply SMS

Kasus dan Hasil Uji (Data Valid)			
Data masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Klik Button Kirim SMS	Mengirim SMS	Dapat Mengirim SMS	[✓] Diterima [ ] Ditolak
Menu Kotak Masuk	Menampilkan pesan masuk	Dapat menampilkan pesan masuk	[✓] Diterima [ ] Ditolak
Menu Kotak Keluar	Menampilkan pesan keluar	Dapat menampilkan pesan keluar	[✓] Diterima [ ] Ditolak
Membalas otomatis (*autoreply*) "INFO#ANTRACOL"	Dapat Membalas otomatis ke nomor pengirim	Dapat Membalas otomatis	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Invalid)			
Petani mengirim pesan diluar kata kunci "TOLONG#OBAT#ANTRACOL"	Dapat membalas ke Petani berisi pesan kesalahan	Dapat membalas ke Petani berisi pesan kesalahan	[✓] Diterima [ ] Ditolak
Nomor tidak terdaftar mengirim pesan	Membalas berisi pesan kesalahan nomor tidak terdaftar	Membalas berisi pesan kesalahan nomor tidak terdaftar	[✓] Diterima [ ] Ditolak

IV.2.2.13 Pengujian laporan

Berikut dapat dilihat pengujian kelola laporan oleh staf DISTANHUT dengan kasus dan hasil uji kemudian diamati hasil dari pengujian tersebut.

Tabel IV. 17 Pengujian Laporam

Kasus dan Hasil Uji (Data Valid)			
Data masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Klik Menu Cetak Laporan Pilih tanggal periode laporan	Mencetak Laporan	Dapat mencetak Laporan	[✓] Diterima [ ] Ditolak

Laporan yang dimaksud adalah laporan kotak masuk, laporan hibah dan laporan pengadaan bantuan pemerintah seperti bibit, pupuk, mesin, obat dan benih.

IV.2.2.14 Pengujian Kelola Akun

Berikut dapat dilihat pengujian kelola akun oleh pengguna *login* dengan kasus dan hasil uji kemudian diamati hasil dari pengujian tersebut.

Tabel IV. 18 Pengujian Kelola Akun

Kasus dan Hasil Uji (Data Valid)			
Data masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Klik Pengaturan akun	Menampilkan menu pengaturan akun	Dapat menampilkan pengaturan akun	[✓] Diterima [ ] Ditolak
Ubah nomor ponsel Nomor lama : 083820120199 Nomor baru:	Mengubah data akun login	Dapat mengubah akun login	[✓] Diterima [ ] Ditolak

089656868836			
Kasus dan Hasil Uji (Data Invalid)			
Ubah username username lama : staf1 username baru: staf2	Menampilkan pesan kesalahan data duplikat	Menampilkan pesan kesalahan	[✓] Diterima [ ] Ditolak

IV.2.2.15 Pengujian *Logout*

Berikut dapat dilihat pengujian kelola data petani oleh staf DISTANHUT dengan kasus dan hasil uji kemudian diamati hasil dari pengujian tersebut.

Tabel IV. 19 Pengujian Logout

Kasus dan Hasil Uji (Data Valid)			
Data masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Klik Tombol Log Out	Keluar dan kembali ke halaman login	Dapat Kembali ke halaman Login	[✓] Diterima [ ] Ditolak

IV.2.3 Kesimpulan Pengujian *Black Box*

Berdasarkan hasil pengujian dengan kasus *sample* uji yang telah dilakukan memberikan kesimpulan bahwa pada proses sudah benar. Secara fungsional sistem sudah dapat menghasilkan *output* yang diharapkan.

IV.2.4 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif dimana diuji secara langsung ke lapangan, dengan melakukan wawancara atau kuesioner mengenai kepuasan pengguna terhadap sistem yang telah dibangun dan untuk mengetahui apakah sistem sesuai dengan apa yang diharapkan. Adapun skenario pengujian beta adalah sebagai berikut :

IV.2.4.1 Skenario Pengujian Beta Administrator

Pengujian beta yang dilakukan terhadap administrator adalah teknik wawancara, berikut daftar pertanyaan dan hasil wawancara:

1. Apakah sistem informasi ini meningkatkan efisiensi kerja ?
   Jawab : Ya, Sistem ini meningkatkan efisiensi kerja, kesalahan-kesalahan bisa diminimalisir sehingga informasi lebih cepat dan akurat.
2. Apakah sistem informasi ini mudah dipahami dan dipelajari?
   Jawab : Ya, Sistem ini mudah dipahami dan dipelajari.
3. Apakah ada kendala dalam mengoperasikan sistem ini?
   Jawab : Tidak ada kendala dalam mengoperasikan sistem ini, satu atau dua kali pelatihan pengguna akan terbiasa.
4. Apakah sistem ini memudahkan pekerjaan DISTANHUT dalam mengelola data?
   Jawab : Ya, Sistem ini memudahkan dalam mengelola data.
5. Fitur apa yang dirasakan paling bermanfaat?
   Jawab : Yang dirasakan paling bermanfaat adalah backup data dan manajemen pengguna, administrator bisa dengan mudah mengatur hak akses pengguna.
6. Apakah sistem memenuhi kebutuhan administrator?
   Jawab : Ya, sistem yang di buat sudah mampu memenuhi kebutuhan administrator.

IV.2.4.2 Skenario Pengujian Beta Operator

Pengujian beta yang dilakukan terhadap operator (staf DISTANHUT) adalah teknik wawancara, berikut daftar pertanyaan dan hasil wawancara:

1. Apakah sistem informasi ini meningkatkan efisiensi kerja ?
   Jawab : Ya, Sistem ini meningkatkan efisiensi kerja, input data lebih cepat, dan kesalahan yang biasanya dilakukan bisa diminimalisir.
2. Apakah sistem informasi ini mudah dipahami dan dipelajari?
   Jawab : Ya, Sistem ini mudah dipahami dan dipelajari.
3. Apakah ada kendala dalam mengoperasikan sistem ini?
   Jawab : Tidak ada kendala dalam mengoperasikan sistem ini, orang yang terbiasa dengan komputer tidak akan menglami kesulitan dalam mengoperasikan program ini.
4. Apakah sistem ini memudahkan pekerjaan DISTANHUT dalam mengelola data?

Jawab : Ya, Sistem ini memudahkan dalam mengelola data, selain itu data lebih akurat dan sikron sehingga tidak perlu bekerja lebih dari sekali untuk memeriksa keakuratan data.

5. Fitur apa yang dirasakan paling bermanfaat?
   Jawab : Yang dirasakan paling bermanfaat bagi operator adalah fitur SMS *gateway*, petani dan DISTANHUT bisa berkomunikasi lebih cepat dan dokumentasi yang lebih baik, selain itu fitur hibah juga sangat bermanfaat dimana sebelumnya daftar transaksi yang dibuat secara manual rentan akan kesalahan operator.
6. Apakah sistem memenuhi kebutuhan operator?
   Jawab : Ya, sistem yang di buat sudah mampu memenuhi kebutuhan operator, mulai dari hubungan dengan petani, data di DISTANHUT ataupun pembuatan laporan untuk diberikan kepada kepala dinas.

IV.2.4.3 Skenario Pengujian Beta Petani

Pengujian beta yang dilakukan terhadap Petani menggunakan kuesioner, kuesioner disebarkan menggunakan teknik sampling yaitu *Sample Random Sampling* yang disebarkan kepada 20 orang pengguna. Dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian penerapan sistem yang baru. Kuesioner ini terdiri dari 5 pertanyaan untuk pengguna (contoh kuesioner dapat dilihat pada lampiran). Skala pengukuran yang digunakan adalah skala likert dengan skala 1 sampai 4, dengan ketentuan skala untuk tiap pertanyaan berikut :

Tabel IV. 20 Skala Pengukuran Kuesioner

No	Keterangan
1	Sangat Setuju
2	Setuju
3	Kurang Setuju
4	Tidak Setuju

Berikut kuesioner yang diajukan ke Petani:

1. Sistem Informasi yang dibangun memudahkan petani untuk berkomunikasi dengan pihak DISTANHUT
   A. Sangat Setuju
   B. Setuju
   C. Kurang Setuju
   D. Tidak Setuju
2. Sistem Informasi yang dibangun bisa meningkatkan kinerja Petani dan DISTANHUT
   A. Sangat Setuju
   B. Setuju
   C. Kurang Setuju
   D. Tidak Setuju
3. Dengan aplikasi SMS *gateway* semua petani mendapat hak yang sama dalam hibah bantuan karena data bersifat transparan antara petani dan DISTANHUT
   A. Sangat Setuju
   B. Setuju
   C. Kurang Setuju
   D. Tidak Setuju
4. Instruksi yang digunakan untuk menggunakan fitur SMS *gateway* mudah dipahami
   A. Sangat Setuju
   B. Setuju
   C. Kurang Setuju
   D. Tidak Setuju
5. Fitur SMS Gateway memudahkan Petani dalam bertanya maupun memberi masukan ke pihak DISTANHUT
   A. Sangat Setuju
   B. Setuju
   C. Kurang Setuju
   D. Tidak Setuju

Berdasarkan data hasil kuisoner tersebut, dapat ditentukan persentase masing-masing jawaban dengan menggunakan rumus : Y = P/Q * 100%

Keterangan :

P = Banyaknya jawaban responden tiap soal.

Q = Jumlah responden

Y = Nilai persentase

Berikut ini adalah persentase masing-masing jawaban. Kuisoner ini diujikan kepada 20 orang petani.

1. Sistem Informasi yang dibangun memudahkan petani untuk berkomunikasi dengan pihak DISTANHUT.

Tabel IV. 21 Hasil Pengujian Kuesioner Pertanyaan No 1

No	Pilihan	Keterangan	Jumlah Responden	Persentase (%)
1	A	Sangat Setuju	6	30
	B	Setuju	14	70
	C	Kurang Setuju	0	0
	D	Tidak Setuju	0	0
			Total	100%

Berdasarkan hasil kuesioner diatas maka dapat disimpulkan bahwa sebanyak 6 orang atau 30% menyatakan sangat setuju dan 14 orang atau 70% mengatakan setuju bahwa sistem informasi yang dibangun memudahkan petani untuk berkomunikasi dengan pihak DISTANHUT.

2. Sistem Informasi yang dibangun bisa meningkatkan kinerja Petani dan DISTANHUT.

Tabel IV. 22 Hasil Pengujian Kuesioner Pertanyaan No 2

No	Pilihan	Keterangan	Jumlah Responden	Persentase (%)
2	A	Sangat Setuju	4	20
	B	Setuju	16	80
	C	Kurang Setuju	0	0
	D	Tidak Setuju	0	0
			Total	100%

Berdasarkan hasil kuesioner diatas maka dapat disimpulkan bahwa sebanyak 4 orang atau 20% menyatakan sangat setuju dan 16 orang atau 80% mengatakan setuju bahwa sistem informasi yang dibangun meningkatkan kinerja Petani dan DISTANHUT

3. Dengan aplikasi SMS *gateway* semua petani mendapat hak yang sama dalam hibah bantuan karena data bersifat transparan antara petani dan DISTANHUT.

Tabel IV. 23 Hasil Pengujian Kuesioner Pertanyaan No 3

No	Pilihan	Keterangan	Jumlah Responden	Persentase (%)
3	A	Sangat Setuju	2	10
	B	Setuju	18	90
	C	Kurang Setuju	0	0
	D	Tidak Setuju	0	0
			Total	100%

Berdasarkan hasil kuesioner diatas maka dapat disimpulkan bahwa sebanyak 2 orang atau 10% menyatakan sangat setuju dan 18 orang atau 90% mengatakan setuju bahwa dengan aplikasi SMS gateway semua petani mendapat hak yang sama dalam hibah bantuan.

4. Instruksi yang digunakan untuk menggunakan fitur SMS *gateway* mudah dipahami.

Tabel IV. 24 Hasil Pengujian Kuesioner Pertanyaan No 4

No	Pilihan	Keterangan	Jumlah Responden	Persentase (%)
4	A	Sangat Setuju	4	20
	B	Setuju	16	80
	C	Kurang Setuju	0	0
	D	Tidak Setuju	0	0
			Total	100%

Berdasarkan hasil kuesioner diatas maka dapat disimpulkan bahwa sebanyak 4 orang atau 20% menyatakan sangat setuju dan 16 orang atau 80%

mengatakan setuju instruksi yang digunakan untuk menggunakan fitur SMS gateway mudah dipahami

5. Fitur SMS Gateway memudahkan Petani dalam bertanya maupun memberi masukan ke pihak DISTANHUT.

Tabel IV. 25 Hasil Pengujian Kuesioner Pertanyaan No 5

No	Pilihan	Keterangan	Jumlah Responden	Persentase (%)
5	A	Sangat Setuju	6	30
	B	Setuju	14	70
	C	Kurang Setuju	0	0
	D	Tidak Setuju	0	0
			Total	100%

Berdasarkan hasil kuesioner diatas maka dapat disimpulkan bahwa sebanyak 6 orang atau 30% menyatakan sangat setuju dan 14 orang atau 70% mengatakan setuju fitur SMS Gateway memudahkan Petani dalam bertanya maupun memberi masukan ke pihak DISTANHUT.

IV.2.5 Kesimpulan Pengujian Beta

Berdasarkan hasil pengujian beta berupa wawancara dan kuesioner kepada staf DISTANHUT dan petani, pembangunan sistem ini sudah sesuai dengan tujuan yaitu memberikan kemudahan dalam proses pengelolaan data, media komunikasi petani dan DISTANHUT, serta pembuatan laporan.