BAB 4
IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini akan dilakukan implementasi dan pengujian terhadap aplikasi yang dibangun. Tahapan ini dilakukan setelah analisis dan perancangan selesai dilakukan dan selanjutnya akan diimplementasikan ke dalam bahasa pemrograman. Setelah implementasi maka dilakukan pengujian terhadap aplikasi. Aplikasi yang telah dibangun akan dilakukan pengujian untuk mengetahui apakah sistem tersebut dapat berjalan sesuai dengan tujuan atau tidak yang selanjutnya diadakan untuk pengembangan sistem.

4.1. Implementasi Sistem

Implementasi sistem merupakan tahap untuk menerapkan perancangan yang telah dilakukan terhadap sistem sehinggi siap untuk dioprasikan. Perangkat keras dan perangkat lunak yang digunakan untuk melakukan implementasi sistem dan pengujian aplikasi yang itu sebagai berikut.

4.1.1. Kebutuhan Perangkat Keras

Spesifikasi perangkat keras (*hardware*) yang digunakan dalam pneggunaan perangkat lunak berbasis *web* untuk aplikasi seleksi penerimaan beasiswa dapat dilihat pada tabel 4.1.

Tabel 4.1 Perangkat Keras yang digunakan pengguna

Nama Perangkat Keras	Spesifikasi
Processor	Dual Core 2.3 Ghz
Memori	1 GB
Hardisk	120 GB
Mouse	Standar
Keyboar	Standar
Model	Jaringan EVDO 3.5 Mbps sebgai koneksi internet

4.1.2. Kebutuhan Perangkat Lunak

Sedangkan spesifikasi perangkat lunak (*Software*) yang digunakan untuk mengembangkan pembangunan perangkat lunak berbasis *web* untuk aplikasi seleksi penerima beasiswa dapat dilihat pada tabel 4.2.

Tabel 4.2 Perangkat Lunak yang digunakan

Perangkat lunak	Spesifikasi
Sistem Operasi	Windows 7 Profesional
Perambah	Mozilla firefox

4.1.3. Implementasi Data

Pembuatan basis data dilakukan dengan menggunakan aplikasi DBMS MySQL. Implementasi basis data dalam bahasa SQL adalah sebagai berikut :

Tabel 4.3 Implementasi Data

No	Nama Tabel	Perintah SQL
1	musers	CREATE TABLE `musers` (`idu` int(11) NOT NULL AUTO_INCREMENT, `nama` char(50) NOT NULL, `username` char(50) NOT NULL, `password` char(100) NOT NULL, `email` char(50) DEFAULT NULL, `iduniv` int(11) NOT NULL, `idps` int(11) NOT NULL, `akses` char(6) NOT NULL, `status` int(11) NOT NULL, PRIMARY KEY (`idu`), CONSTRAINT `musers_ibfk_2` FOREIGN KEY (`idps`) REFERENCES `mperseleksi` (`idps`), CONSTRAINT `musers_ibfk_1` FOREIGN KEY (`iduniv`) REFERENCES `mdafuniv` (`iduniv`)) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;
2	mprofil	CREATE TABLE `mprofil` (`idprofil` int(11) NOT NULL AUTO_INCREMENT, `idu` int(11) NOT NULL, `tgllahir` date DEFAULT NULL, `alamatRmh` text, `alamatSkr` text, `nohp` char(12) DEFAULT NULL, `photo` char(100) DEFAULT NULL, PRIMARY KEY (`idprofil`), CONSTRAINT `mprofil_ibfk_1` FOREIGN KEY (`idu`) REFERENCES `musers` (`idu`)) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=latin1;

2	mdafuniv	CREATE TABLE `mdafuniv` (`iduniv` int(11) NOT NULL AUTO_INCREMENT, `namauniv` char(100) NOT NULL, PRIMARY KEY (`iduniv`)) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=latin1;
3	mperseleksi	CREATE TABLE `mperseleksi` (`idps` int(11) NOT NULL AUTO_INCREMENT, `thnperiode` int(11) DEFAULT NULL, `ket` text NOT NULL, `status` int(11) DEFAULT NULL, PRIMARY KEY (`idps`)) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;
4	mhslseleksi	CREATE TABLE `mhslseleksi` (`idhSeleksi` int(11) NOT NULL AUTO_INCREMENT, `idu` int(11) NOT NULL, `nilai` int(11) NOT NULL, ‘idps’ int(11) NOT NULL, PRIMARY KEY (`idhSeleksi`), CONSTRAINT `mhslseleksi_ibfk_1` FOREIGN KEY (`idu`) REFERENCES `musers` (`idu`), CONSTRAINT `mhslseleksi_ibfk_2` FOREIGN KEY (`idps`) REFERENCES `mperseleksi` (`idps`)) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;
5	mhisregis	CREATE TABLE `mhisregis` (`idhr` int(11) NOT NULL AUTO_INCREMENT, `idu` int(11) DEFAULT NULL, `idps` int(11) DEFAULT NULL, PRIMARY KEY (`idhr`), CONSTRAINT `mhisregis _ibfk_1` FOREIGN KEY (`idu`) REFERENCES `musers` (`idu`), CONSTRAINT `mhisregis _ibfk_2` FOREIGN KEY (`idps`) REFERENCES `mperseleksi` (`idps`)) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;
6	mhisperiode	CREATE TABLE `mhisperiode` (`idhp` int(11) NOT NULL AUTO_INCREMENT, `idu` int(11) DEFAULT NULL, `idps` int(11) DEFAULT NULL, PRIMARY KEY (`idhp`), CONSTRAINT `mhisperiode _ibfk_1` FOREIGN KEY (`idu`) REFERENCES `musers` (`idu`), CONSTRAINT `mhisperiode _ibfk_2` FOREIGN KEY (`idps`) REFERENCES `mhisperiode ` (`idps`)) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;
7	mhisfpass	CREATE TABLE `mhisfpass` (`idhfpass` int(11) NOT NULL AUTO_INCREMENT, `idu` int(11) NOT NULL, `tgl` date NOT NULL, PRIMARY KEY (`idhfpass`), CONSTRAINT `mhisfpass _ibfk_1` FOREIGN KEY (`idu`) REFERENCES `musers` (`idu`)

		) ENGINE=InnoDB DEFAULT CHARSET=latin1;
9	msyareko	CREATE TABLE `msyareko` (`kodeEko` char(5) NOT NULL, `ketEko` char(100) NOT NULL, `nilaiEko` int(11) NOT NULL, PRIMARY KEY (`kodeEko`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
10	msyaripk	CREATE TABLE `msyaripk` (`kodeIpk` char(5) NOT NULL, `ketIpk` char(100) NOT NULL, `nilaiIpk` int(11) NOT NULL, PRIMARY KEY (`kodeIpk`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
11	msyarorga	CREATE TABLE `msyarorga` (`kodeOrga` char(5) NOT NULL, `ketOrga` char(100) NOT NULL, `nilaiOrga` int(11) NOT NULL, PRIMARY KEY (`kodeOrga`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
12	msyarpiag	CREATE TABLE `msyarpiag` (`kodePiag` char(5) NOT NULL, `ketPiag` char(100) NOT NULL, `nilaiPiag` int(11) NOT NULL, PRIMARY KEY (`kodePiag`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
13	maddsyareko	CREATE TABLE `maddsyareko` (`idEko` int(11) NOT NULL AUTO_INCREMENT, `idu` int(11) NOT NULL, `kodeEko` char(5) NOT NULL, `idps` int(11) DEFAULT NULL, PRIMARY KEY (`idEko`), CONSTRAINT `maddsyareko _ibfk_1` FOREIGN KEY (`idu`) REFERENCES `musers` (`idu`), CONSTRAINT `maddsyareko_ibfk_2` FOREIGN KEY (`kodeEko`) REFERENCES `msyareko` (`kodeEko`), CONSTRAINT `maddsyareko_ibfk_3` FOREIGN KEY (`idps`) REFERENCES `mhisperiode ` (`idps`)) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;
14	maddsyaripk	CREATE TABLE `maddsyaripk` (`idIpk` int(11) NOT NULL AUTO_INCREMENT, `idu` int(11) NOT NULL, `kodeIpk` char(5) NOT NULL, `idps` int(11) DEFAULT NULL, PRIMARY KEY (`idIpk`), CONSTRAINT `maddsyaripk _ibfk_1` FOREIGN KEY (`idu`) REFERENCES `musers` (`idu`), CONSTRAINT `maddsyaripk_ibfk_2` FOREIGN KEY (`kodeIpk`) REFERENCES `msyaripk` (`kodeIpk`), CONSTRAINT `maddsyaripk_ibfk_3` FOREIGN KEY (`idps`) REFERENCES `mhisperiode ` (`idps`)) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;
15	maddsyarorga	CREATE TABLE `maddsyarorga` (`idOrga` int(11) NOT NULL AUTO_INCREMENT,

		`idu` int(11) NOT NULL, `kodeOrga` char(5) NOT NULL, `idps` int(11) DEFAULT NULL, PRIMARY KEY (`idOrga`), CONSTRAINT `maddsyarorga_ibfk_1` FOREIGN KEY (`idu`) REFERENCES `musers` (`idu`), CONSTRAINT `maddsyarorga_ibfk_2` FOREIGN KEY (`kodeOrga`) REFERENCES `msyarorga` (`kodeOrga`), CONSTRAINT `maddsyarorga_ibfk_3` FOREIGN KEY (`idps`) REFERENCES `mhisperiode ` (`idps`)) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;
16	maddsyarpiag	CREATE TABLE `maddsyarpiag` (`idPiag` int(11) NOT NULL AUTO_INCREMENT, `idu` int(11) NOT NULL, `kodePiag` char(5) NOT NULL, `idps` int(11) DEFAULT NULL, PRIMARY KEY (`idPiag`), CONSTRAINT `maddsyarpiag_ibfk_1` FOREIGN KEY (`idu`) REFERENCES `musers` (`idu`), CONSTRAINT `maddsyarpiag_ibfk_2` FOREIGN KEY (`kodePiag`) REFERENCES `msyarpiag` (`kodePiag`), CONSTRAINT `maddsyarpiag_ibfk_3` FOREIGN KEY (`idps`) REFERENCES `mhisperiode ` (`idps`)) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;

4.1.4. Implementasi Antarmuka

Implementasi antarmuka dilakukan terhadap setiap tampilan program yang dibagun dan pengkodeannya dalam bentuk *file* program. Berikut ini adalah implementasi antar muka untuk pengguna.

Tabel 4.4 Implementasi Antarmuka Administrator

Menu	Deskripsi	Nama File
Beranda	Merupakan halaman utama jika seorang administrator berhasil login.	cmainAdmin.php
Data Peserta	Merupakan halaman untuk mengelola akun peserta serta mendapatkan informasi dari profil peserta	cdataPeserta.php
Data Persyaratan Peserta	Memantau persyaratan peserta, dan mengelola periode seleksi.	cdataSyarat.php
Hasil Penilaian	Melihat hasil penilaian yang telah dilakukan oleh masing-masing peserta dan mecetak laporan untuk mendapatkan rekomendasi keputusan hasil seleksi	chasilPenilaian.php
Pengaturan	Halam ini untuk melakukan pengolahan data perguruan tinggi, nilai dari masing-masing kriteria dan profil administraotr.	cpengaturan.php

Daftar Perguruan Tinggi	Halaman ini berfungsi untuk menambah, mengubah dan mencari data perguruan tingi	caturUniv.php
Syarat Ekonomi	Halaman ini berfungsi untuk mengatur nilai atau keterangan pada Syarat Ekonomi	caturSyarEko.php
Syarat IPK	Halaman ini berfungsi untuk mengatur nilai atau keterangan pada Syarat IPK	caturSyarIpk.php
Syarat Organisasi	Halaman ini berfungsi untuk mengatur nilai atau keterangan pada Syarat Organisasi	caturSyarOrga.php
Syarat Piagam	Halaman ini berfungsi untuk mengatur nilai atau keterangan pada Syarat Piagam	caturSyarPiag.php
Profil	Halaman ini berfungsi untuk mengatur profil Administrator	cprofil.php

Tabel 4.5 Implementasi Antarmuka Peserta

Menu	Deskripsi	Nama File
Beranda	Merupakan halaman utama jika seorang peserta berhasil login.	cmainMember.php
Profil	Melihat dan mengelola profil untuk masing-masing peserta	cprofil.php
Data Persyaratan	Peserta dapat mengelola persyaratan yang harus di penuhi oleh masing-masing peserta	cpersyaratan.php

4.2. Pengujian Sistem

Tahap yang selanjutnya adalah thap pengujian sistem pada aplikasi seleksi penerima beasiswa. Tahap ini merupakaan hal terpenting yang bertujuan untuk menemukan kesalahan – kesalahan ataupun kekurangan – kekurangan pada palikasi yang dibangun. Pengujian ini bermaksud untuk mengetahui apakah aplikasi yang dibuat telah memenuhi kriteria yang sesuai dengan tujuan perancangan aplikasi atau belum. Pengujian aplikasi ini menggunakan pengujian *black box*, bertujuan pada persyaratan fungsional perangkat lunak.

4.2.1. Pengujian Alpha

Pengujian *alpha* dilakukan dengan menggunakan metode *black box* yang berfokus pada persyaratan fungsional perangkat lunak.

4.2.1.1. Metode Pengujian

Sebuah aplikasi yang akan diterapkan haruslah bebas dari kesalahan – kesalahan. Oleh karena itu, aplikasi harus diuji coba terlebih dahulu untuk menemukan kesalahan – kesalahan yang mungkin terjadi.

4.2.1.2. Rencana Pengujian

Pengujian program ini menggunakan metode *black box*. Pengujian *black box* merupakan pengujian program berdasarkan fungsional dari program. Tujuan dari metode *black box* ini adalah untuk menemukan kesalahan fungsi pada program. Pengujian dengan metode *black box* dilakukan dengan cara memberikan sejumlah data masukan pada aplikasi yang kemudian diproses sesuai dengan kebutuhan fungsionalnya untuk melihat apakah aplikasi yang kemudian diproses sesuai dengan kebutuhan fungsionalnya untuk melihat apakah aplikasi menghasilkan keluaran yang digunakan dan sesuai dengan fungsi dari program tersebut. apabila dari data masukan yang diberikan proses menghasilkan keluaran yang sesuai dengan kebutuhan fungsionalnya, maka aplikasi yang telah dibuah telah benar. Tetapi jika keluaran yang dihasilkan tidak sesuai dengan kebutuhan fungsionalnya, maka mash terdapat kesalahan pada program aplikasi.

Pengujian dilakukan dengan mencoba semua kemungkinan yang terjadi dan pengujian dilakukan berulang-ulang jika dalam pengujian ditamukan kesalahan maka akan dilakukan penelusuran atau perbaikan untuk memperbaiki kesalahan yang terjadi. Jika telah selesai melakukan perbaikan, makan akan dilakukan secara terus menerus sehingga diperoleh hasil yang terbaik. Rencana pengujian yang akan dilakukan pada perangkat lunak ini dapat dilihat pada tabel sebagai berikut :

5.

Tabel 4.6 Rancangan pengujian pada halaman login

Kelas Uji	Butir Uji	Jenis Pengujian
Login	Menampilkan form login	Black box
	Isi form login	
	Verifikasi username	
	Verifikasi password	
	Menampilkan halaman beranda	

Tabel 4.7 Rancangan pengujian pada tambah akun

Kelas Uji	Butir Uji	Jenis Pengujian
Tambah Akun	Menampilkan form tambah akun	Black box
	Isi form tambah akun	
	Verifikasi data masukan	
	Memasukan data ke dalam basis data	

Tabel 4.8 Rancangan pengujian pada ubah data akun

Kelas Uji	Butir Uji	Jenis Pengujian
Ubah data akun	Menampilkan form ubah data akun	Black box
	Mengubah data pada form ubah data akun	
	Verifikasi data masukan	
	Memasukan data ke dalam basis data	

Tabel 4.9 Rancangan pengujian hapus data akun

Kelas Uji	Butir Uji	Jenis Pengujian
Hapus data akun	Menampilkan pesan konfirmasi	Black box
	Menyetujui pesan konfirmasi	
	Menghapus data pada basis data	

Tabel 4.10 Rancangan pengujian pada tambah periode

Kelas Uji	Butir Uji	Jenis Pengujian
Tambah periode	Menampilkan form tambah periode	Black box
	Menambahkan periode baru	
	Verifikasi data masukan	
	Memasukan data ke dalam basis data	

Tabel 4.11 Rancangan pengujian pada ubah status periode

Kelas Uji	Butir Uji	Jenis Pengujian
Ubah status periode	Menampilkan form ubah status periode	Black box
	Verifikasi data masukan	
	Memasukan data ke dalam basis data	

Tabel 4. 12 Rancangan pengujian ubah deskripsi periode

Kelas Uji	Butir Uji	Jenis Pengujian
Ubah deskripsi periode	Manampilkan form ubah deskripsi periode	Black box
	Mengubah deskripsi periode	
	Verifikasi data masukan	
	Memasukan data kedalam basis data	

Tabel 4.13 Rancangan pengujian tambah data perguruan tinggi

Kelas Uji	Butir Uji	Jenis Pengujian
Tambah perguruan tinggi	Menampilkan form tambah perguruan tinggi	Black box
	Mengisi form tambah perguruan tinggi	
	Verifikasi data masukan	
	Memasukan data kedalam basis data	

Tabel 4.14 Rancangan pengujian ubah data perguruan tinggi

Kelas Uji	Butir Uji	Jenis Pengujian
Ubah data perguruan tinggi	Menampilkan form ubah perguruan tinggi	Black box
	Mengubah data perguruan tinggi	
	Verifikasi data masukan	
	Memasukan data kedalam basis data	

Tabel 4.15 Rancangan pengujian ubah data syarat ekonomi

Kelas Uji	Butir Uji	Jenis Pengujian
Ubah data syarat ekonomi	Menampilkan form ubah syarat ekonomi	Black box
	Mengubah data syarat ekonomi	
	Verifikasi data masukan	
	Memasukan data kedalam basis data	

Tabel 4.16 Rancangan pengujian ubah syarat IPK

Kelas Uji	Butir Uji	Jenis Pengujian
Ubah data syarat IPK	Menampilkan form ubah syarat IPK	Black box
	Mengubah data syarat IPK	
	Verifikasi data masukan	
	Memasukan data kedalam basis data	

Tabel 4.17 Rancangan pengujian ubah syarat organisasi

Kelas Uji	Butir Uji	Jenis Pengujian
Ubah data syarat organisasi	Menampilkan form ubah syarat organisasi	Black box
	Mengubah data syarat organisasi	
	Verfikasi data masukan	
	Memasukan data kedalam basis data	

Tabel 4.18 Rancangan pengujian ubah syarat piagam penghargaan

Kelas Uji	Butir Uji	Jenis Pengujian
Ubah data syarat piagam penghargaan	Menampilkan form ubah data syarat piagam penghargaan	Black box
	Mengubah data syarat piagam penghargaan	
	Verifikasi data masukan	
	Memasukan data kedalam basis data	

Tabel 4.19 Rancangan pengujian ubah data profil

Kelas Uji	Butir Uji	Jenis pengujian
Ubah data profil	Menampilkan form ubah data profil	Black box
	Mengubah data profil	
	Verfikasi data masukan	
	Memasukan data kedalam basis data	

Tabel 4.20 Rancangan pengujian isi persyratan wajib

Kelas Uji	Butir Uji	Jenis Pengujian
Isi persyaratan wajib	Menampilkan form ubah data akun	Black box
	Mengisi data persyaratan wajib	
	Verifikasi data masukan	

	Memasukan data ke dalam basis data	

Tabel 4.21 Rancangan pengujian ubah persyaratan wajib

Kelas Uji	Butir Uji	Jenis Pengujian
Ubah persyaratan wajib	Menampilkan form ubah persyaratan wajib	Black box
	Mengubah data pada form ubah persyaratan wajib	
	Verifikasi data masukan	
	Memasukan data ke dalam basis data	

Tabel 4.22 Rancangan pengujian tambah persyaratan tidak wajib

Kelas Uji	Butir Uji	Jenis Pengujian
Tambah persyaratan tidak wajib	Menampilkan form tambah persyaratan tidak wajib	Black box
	Mengisi persyaratan tidak wajib	
	Verifikasi data masukan	
	Memasukan data ke dalam basis data	

Tabel 4.23 Rancangan pengujian ubah persyaratan tidak wajib

Kelas Uji	Butir Uji	Jenis Pengujian
Ubah persyaratan tidak wajib	Menampilkan form ubah data persyaratan tidak wajib	Black box
	Mengubah data pada persyaratan tidak wajib	
	Verifikasi data masukan	
	Memasukan data ke dalam basis data	

Tabel 4.24 Rancangan pengujian hapus persyaratan tidak wajib

Kelas Uji	Butir Uji	Jenis Pengujian
Hapus persyaratan tidak wajib	Menampilkan pesan konfirmasi persetujuan	Black box
	Menyetujui pesan konfirmasi	
	Menghapus data pada basis data	

Tabel 4.25 Rancangan pengujian lupa password

Kelas Uji	Butir Uji	Jenis pengujian
Lupa password	Menampilkan form lupa password	Black box
	Memasukan email	

	Verfikasi data masukan	
	Memasukan data kedalam basis data dan mengirim data ke email	

4.2.1.3. Kasus dan Hasil Pengujian

Pengujian black box bertujuan untuk mengukur kinerja dari perangkat lunak yang dibangun, apakah fungsinya dapat berjalan dengan baik atau tidak. Adapun kasus dan hasil pengujian yang dilakukan adalah sebagai berikut :

1. Pengujian login

   Berikut ini dapat dilihat pengujian dari login yang dapat dilihat pada tabel 4.26.

Tabel 4.26 Pengujian login

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data login, contoh masukan : Username : asdfg Password : asdfg	Mengisi data login dan *field* yang tersedia dan berhasil masuk kedalam halaman beranda	Dapat mengisi data login dan dapat masuk ke halaman beranda	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Username : (kosong) Password : (kosong)	Tampil pesan kesalahan "Login gagal ! username dan password salah"	Dapat menampilkan pesan kesalahan "Login gagal ! username dan password salah"	[√] Diterima [ ] Ditolak

2. Pengujian tambah akun peserta

   Berikut ini dapat dilihat pengujian dari tambah akun peserta yang dapat dilihat pada tabel 4.27.

Tabel 4.27 Pengujian tambah akun peserta

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data logn, contoh masukan :	Mengisi data tambah akun berdasarkan *field* yang tersedia, berhasil	Dapat mengisi form tambah akun lalu memasukan	[√] Diterima [ ] Ditolak

Nama : dadang s Email : dd@domain.com	memasukan ke dalam basis data dan penampilkan pesan "Akun baru atas nama dadang s berhasil di tambahkan"	data kedalam basis data dan menampilkan pesan "Akun baru atas nama dadang s berhasil di tambahkan"	
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Nama : (kosong) Email : (kosong)	Menampilkan pesan "**Data masukan salah !** masukan nama dan email sesuai dengan ketentuan"	Dapat menampilkan pesan kesalahan "**Data masukan salah !** masukan nama dan email sesuai dengan ketentuan"	[√] Diterima [ ] Ditolak

3. Pengujian ubah data akun peserta

   Berikut ini dapat dilihat pengujian dari ubah data akun peserta yang dapat dilihat pada tabel 4.28.

Tabel 4.28 Pengujian ubah data akun peserta

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data logn, contoh masukan : Nama : dadang s Email : dd@domain.com	Mengubah data akun berdasarkan *field* yang tersedia, berhasil memasukan ke dalam basis data dan penampilkan pesan "**Berhasil !** Akun akun atas nama Nurul Hadi berhasil diubah"	Dapat mengisi form ubah akun lalu dapat mengubah data dalam basis data dan menampilkan pesan "**Berhasil !** Akun akun atas nama Nurul Hadi berhasil diubah"	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Nama : (kosong) Email : (kosong)	Menampilkan pesan "**Data masukan salah !** masukan nama dan email sesuai dengan ketentuan"	Dapat menampilkan pesan kesalahan "**Data masukan salah !** masukan nama dan email sesuai dengan ketentuan"	[√] Diterima [ ] Ditolak

4. Pengujian hapus akun peserta

Berikut ini dapat dilihat pengujian dari hapus data akun peserta yang dapat dilihat pada tabel 4.29.

Tabel 4.29 Pengujian hapus akun peserta

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memilih data yang akan dihapus	Menampilkan pesan konfirmasi hapus, menghapus data pada basis data dan penampilkan pesan "**Berhasil !** Akun peserta berhasil dihapus"	Dapat menampilkan konfirmasi hapus dan menampilkan pesan "**Berhasil !** Akun peserta berhasil dihapus"	[√] Diterima [ ] Ditolak

5. Pengujian tambah periode baru

   Berikut ini dapat dilihat pengujian dari tambah periode baru yang dapat dilihat pada tabel 4.30.

Tabel 4.30 Pengujian tambah periode baru

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data tambah periode, contoh masukan : Deskripsi : pada periode ini akan	Menambah data periode baru berdasarkan *field* yang tersedia, berhasil memasukan ke dalam basis data dan penampilkan pesan "**Berhasil !** periode baru berhasil ditambahkan"	Dapat mengisi form tambah akun lalu dapat menambah data dalam basis data dan menampilkan pesan "**Berhasil !** periode baru berhasil ditambahkan"	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Deskripsi: (kosong)	Menampilkan pesan "**Data masukan salah !** isi deskripsi sesuai dengan ketentuan"	Dapat menampilkan pesan kesalahan "**Data masukan salah !** isi deskripsi sesuai dengan ketentuan"	[√] Diterima [ ] Ditolak

6. Pengujian ubah deskripsi periode

   Berikut ini dapat dilihat pengujian dari ubah deskripsi periode yang dapat dilihat pada tabel 4.31.

Tabel 4.31 Pengujian ubah deskripsi periode

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data, contoh masukan : Deskripsi : pada periode ini akan	Mengubah deskripsi berdasarkan *field* yang tersedia, berhasil mengubah data dalam basis data dan penampilkan pesan "**Berhasil ! deskripsi** berhasil diubah"	Dapat mengisi form ubah deskripsi lalu dapat mengubah data dalam basis data dan menampilkan pesan "**Berhasil !** deskripsi berhasil diubah"	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Deskripsi: (kosong)	Menampilkan pesan "**Data masukan salah !** isi deskripsi sesuai dengan ketentuan"	Dapat menampilkan pesan kesalahan "**Data masukan salah !** isi deskripsi sesuai dengan ketentuan"	[√] Diterima [ ] Ditolak

7. Pengujian ubah status periode

   Berikut ini dapat dilihat pengujian dari ubah status periode yang dapat dilihat pada tabel 4.32.

Tabel 4.32 Pengujian ubah status periode

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Menset non aktif pada periode	Mengubah status periode dari aktif menjadi tidak aktif dan penampilkan pesan "**Berhasil ! periode seleksi telah di non aktifkan**"	Dapat mengubah status dari aktif menjadi tidak aktif dan menampilkan pesan "**Berhasil !** periode seleksi telah di non aktifkan"	[√] Diterima [ ] Ditolak

8. Pengujian tambah perguruan tinggi

   Berikut ini dapat dilihat pengujian dari tambah perguruan tinggi yang dapat lihat pada tabel 4.33.

Tabel 4.33 Pengujian tambah perguruan tinggi

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data, contoh masukan : Nama perguruan tinggi : Universitas Komputer Indonesia	Menambah perguruan tinggi berdasarkan *field* yang tersedia, berhasil menambah data kedalam basis data dan penampilkan pesan "**Berhasil ! perguruan tinggi baru berhasil ditambahkan**"	Dapat mengisi form tambah perguruan tinggi lalu dapat menambah data dalam basis data dan menampilkan pesan "**Berhasil ! perguruan tinggi baru berhasil ditambahkan**"	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Nama perguruan tinggi: (kosong)	Menampilkan pesan "**Data masukan salah !** isi nama perguruan tinggi sesuai dengan ketentuan"	Dapat menampilkan pesan kesalahan "**Data masukan salah !** isi nama perguruan tinggi sesuai dengan ketentuan"	[√] Diterima [ ] Ditolak

9. Pengujian ubah data perguruan tinggi

   Berikut ini dapat dilihat pengujian dari ubah data perguruan tinggi yang dapat dilihat pada tabel 4.34.

Tabel 4.34 Pengujian ubah data perguruan tinggi

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data, contoh masukan : Nama perguruan tinggi : Universitas Komputer Indonesia	Mengubah data perguruan tinggi berdasarkan *field* yang tersedia, berhasil mengubah data dalam basis data dan penampilkan pesan "**Berhasil ! perguruan tinggi baru berhasil diubah**"	Dapat mengisi form ubah perguruan tinggi lalu dapat mengubah data dalam basis data dan menampilkan pesan "**Berhasil ! perguruan tinggi baru berhasil diubah**"	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Nama perguruan tinggi: (kosong)	Menampilkan pesan "**Data masukan salah !** isi nama perguruan	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak

	tinggi sesuai dengan ketentuan"	"**Data masukan salah !** isi nama perguruan tinggi sesuai dengan ketentuan"	

10. Pengujian ubah data syarat ekonomi

Berikut ini dapat dilihat pengujian dari ubah data syarat ekonomi yang dapat dilihat pada tabel 4.35.

Tabel 4.35 Pengujian ubah data syarat ekonomi

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data, contoh masukan : Ketarangan : memliki penghasilan Rp. 2.000.000 per bulan Nilai : 304	Mengubah data syarat ekonomi berdasarkan *field* yang tersedia, berhasil mengubah data didalam basis data dan penampilkan pesan "**Berhasil ! data berhasil diubah**"	Dapat mengisi form ubah syarat ekonomi lalu dapat mengubah data dalam basis data dan menampilkan pesan "**Berhasil ! data berhasil diubah**"	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Keterangani : (kosong) Nilai : (kosong)	Menampilkan pesan "**Data masukan salah !** isi keterangan dan nilai sesuai dengan ketentuan"	Dapat menampilkan pesan kesalahan "**Data masukan salah !** isi keterangan dan nilai sesuai dengan ketentuan"	[√] Diterima [ ] Ditolak

11. Pengujian ubah data syarat IPK

Berikut ini dapat dilihat pengujian dari ubah data syarat IPK yang dapat dilihat pada tabel 4.36.

Tabel 4.36 Pengujian ubah data syarat IPK

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data, contoh masukan : Ketarangan : memliki IPK 3.2	Mengubah data syarat ekonomi berdasarkan *field* yang tersedia, berhasil mengubah data didalam basis data dan	Dapat mengisi form ubah syarat ekonomi lalu dapat mengubah data dalam basis data	[√] Diterima [ ] Ditolak

Nilai : 304	penampilkan pesan "**Berhasil ! data berhasil diubah**"	dan menampilkan pesan "**Berhasil ! data berhasil diubah**"	
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Keterangani : (kosong) Nilai : (kosong)	Menampilkan pesan "**Data masukan salah !** isi keterangan dan nilai sesuai dengan ketentuan"	Dapat menampilkan pesan kesalahan "**Data masukan salah !** isi keterangan dan nilai sesuai dengan ketentuan"	[√] Diterima [ ] Ditolak

12. Pengujian ubah data syarat organisasi

    Berikut ini dapat dilihat pengujian dari ubah data syarat organisasi yang dapat dilihat pada tabel 4.37.

Tabel 4.37 Pengujian ubah data syarat organisasi

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data, contoh masukan : Ketarangan : menjabat sebagai ketua Nilai : 304	Mengubah data syarat ekonomi berdasarkan *field* yang tersedia, berhasil mengubah data didalam basis data dan penampilkan pesan "**Berhasil ! data berhasil diubah**"	Dapat mengisi form ubah syarat ekonomi lalu dapat mengubah data dalam basis data dan menampilkan pesan "**Berhasil ! data berhasil diubah**"	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Keterangani : (kosong) Nilai : (kosong)	Menampilkan pesan "**Data masukan salah !** isi keterangan dan nilai sesuai dengan ketentuan"	Dapat menampilkan pesan kesalahan "**Data masukan salah !** isi keterangan dan nilai sesuai dengan ketentuan"	[√] Diterima [ ] Ditolak

13. Pengujian ubah data syarat piagam penghargaan

    Berikut ini dapat dilihat pengujian dari ubah data piagam penghargaan yang dapat dilihat pada tabel 4.38.

Tabel 4.38 Pengujian ubah data syarat piagam penghargaan

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data, contoh masukan : Ketarangan : menjuarai cerdas cermat tingkat kabupaten Nilai : 304	Mengubah data syarat ekonomi berdasarkan *field* yang tersedia, berhasil mengubah data didalam basis data dan penampilkan pesan "**Berhasil ! data berhasil diubah**"	Dapat mengisi form ubah syarat ekonomi lalu dapat mengubah data dalam basis data dan menampilkan pesan "**Berhasil ! data berhasil diubah**"	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Keterangani : (kosong) Nilai : (kosong)	Menampilkan pesan "**Data masukan salah !** isi keterangan dan nilai sesuai dengan ketentuan"	Dapat menampilkan pesan kesalahan "**Data masukan salah !** isi keterangan dan nilai sesuai dengan ketentuan"	[√] Diterima [ ] Ditolak

14. Pengujian ubah data profil

Berikut ini dapat dilihat pengujian dari ubah data profil yang dapat dilihat pada tabel 4.39.

Tabel 4.39 Pengujian ubah data profil

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data, contoh masukan : Nama : raden gusti Email : raden@kumandang.or.id No hp : 0984744744	Mengubah data profil berdasarkan *field* yang tersedia, berhasil mengubah data didalam basis data dan penampilkan pesan "**Berhasil ! data berhasil diubah**"	Dapat mengisi form ubah data profil lalu dapat mengubah data dalam basis data dan menampilkan pesan "**Berhasil ! data berhasil diubah**"	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Nama : (kosong) Email : (kosong) No hp : (kosong)	Menampilkan pesan "**Data masukan salah !** masukan	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak

	nama, tanggal lahir, no hp, dan email sesuai dengan ketentuan"	"**Data masukan salah !** masukan nama, tanggal lahir, no hp, dan email sesuai dengan ketentuan"	

15. Pengujian isi persyaratan wajib

Berikut ini dapat dilihat pengujian dari isi persyaratan wajib yang dapat dilihat pada tabel 4.40.

Tabel 4.40 Pengujian isi persyaratan wajib

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memilih data masukan, contoh masukan : Taraf ekonomi, Nilai IPK, jabatan organisasi, piagam penghargaan di isi melalui *drop-down list*	Mengisi data persyaratan wajib berdasarkan *field* yang tersedia, berhasil mengisi data kedalam basis data dan penampilkan pesan "**Berhasil !** Persyaratan untuk kriteria Ekonomi, IPK, dan Organisasi berhasil diisi."	Dapat mengisi form isi persyaratan wajib lalu dapat mengisi data dalam basis data dan menampilkan pesan "**Berhasil !** Persyaratan untuk kriteria Ekonomi, IPK, dan Organisasi berhasil diisi."	[√] Diterima [ ] Ditolak

16. Pengujian ubah persyaratan wajib

Berikut ini dapat dilihat pengujian dari ubah persyaratan wajib yang dapat dilihat pada tabel 4.41.

Tabel 4.41 Pengujian ubah persyaratan wajib

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengubah data, contoh masukan : Taraf ekonomi, Nilai IPK, jabatan organisasi, piagam penghargaan di isi melalui *drop-down list*	Mengubah data persyaratan wajib berdasarkan *field* yang tersedia, berhasil mengubah data didalam basis data dan penampilkan pesan "**Berhasil !** Persyaratan untuk kriteria Ekonomi,	Dapat mengisi form ubah persyaratan wajib lalu dapat mengubah data dalam basis data dan menampilkan pesan "**Berhasil !** Persyaratan untuk	[√] Diterima [ ] Ditolak

	IPK, dan Organisasi berhasil diubah."	kriteria Ekonomi, IPK, dan Organisasi berhasil diubah."	

17. Pengujian isi persyaratan tidak wajib

Berikut ini dapat dilihat pengujian dari isi persyaratan tidak wajib yang dapat dilihat pada tabel 4.42.

Tabel 4.42 Pengujian isi persyaratan tidak wajib

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memilih data masukan, contoh masukan : Jenis piagam penghargaan di isi melalui *drop-down list*	Mengisi data persyaratan tidak wajib berdasarkan *field* yang tersedia, berhasil mengisi data kedalam basis data dan penampilkan pesan "**Berhasil !** Persyaratan untuk kriteria piagam berhasil ditambahkan."	Dapat mengisi form isi persyaratan tidak wajib lalu dapat mengisi data dalam basis data dan menampilkan pesan "**Berhasil !** Persyaratan untuk kriteria piagam berhasil ditambahkan."	[√] Diterima [ ] Ditolak

18. Pengujian ubah persyaratan tidak wajib

Berikut ini dapat dilihat pengujian dari ubah persyaratan tidak wajib yang dapat dilihat pada tabel 4.43.

Tabel 4.43 Pengujian isi persyaratan tidak wajib

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memilih data masukan, contoh masukan : Jenis piagam penghargaan di isi melalui *drop-down list*	Mengubah data persyaratan tidak wajib berdasarkan *field* yang tersedia, berhasil mengubah data kedalam basis data dan penampilkan pesan "**Berhasil !** Persyaratan untuk kriteria piagam berhasil diubah."	Dapat mengisi form ubah persyaratan tidak wajib lalu dapat mengubah data dalam basis data dan menampilkan pesan "**Berhasil !** Persyaratan untuk kriteria piagam berhasil diubah."	[√] Diterima [ ] Ditolak

19. Pengujian hapus persyaratan tidak wajib

Berikut ini dapat dilihat pengujian dari hapus persyaratan tidak wajib yang dapat dilihat pada tabel 4.44.

Tabel 4.44 Pengujian hapus persyaratan tidak wajib

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memilih data yang akan dihapus	Menampilkan pesan konfirmasi hapus, menghapus data pada basis data dan penampilkan pesan "**Berhasil !** Persyaratan untuk kriteria Piagam berhasil dihapus."	Dapat menampilkan konfirmasi hapus dan menampilkan pesan "**Berhasil !** Persyaratan untuk kriteria Piagam berhasil dihapus"	[√] Diterima [ ] Ditolak

4.2.1.4. Kesimpulan Pengujian Alpha

Berdasarkan pengujian yang telah dilakukan, maka kesimpulan yang didapatkan bahwa poses-proses pada pembangunan aplikasi seleksi penerima beasiswa di organisasi KUMANDANG Bandung telah melalui tahap perbaikan dan sudah dimaksimalkan bergitupun secara fungsional sudah menghasilkan output yang diharapkan sehingga sistem sudah layak untuk digunakan.

4.2.2. Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif yang dilakukan secara berlangsung oleh pengguna yang nantinya akan menggunakan aplikasi seleksi penerima beasiswa ini dinataranya adalah Divisi Litabang di KUMANDANG Bandung dan peserta seleksis yang berasal dari kabupaten Pandeglang dengan status sebagai mahasiswa aktif yang melanjutkan studinya di Bandung.

Penelitian ini dilakukan di lingkungan organisasi KUMANDANG Bandung dengan menggunakan teknik wawancara untuk divisi litbang dan kuisioner untuk calon calon peserta. Hal yang dilakukan agar dapat mengetahui

sejauh mana aplikasi yang dibangun ini dapat membantu dan dapat menyelesaikan masalah yang suda dijelaskan pada perumusan masalah.

4.2.2.1. Skenario Pengujian Beta

Kuisioner Pengujian merupakan teknik pengolahan data yang merupakan langkah dalam mengolah data yang telah didapat untuk dijadikan hasil penelitian sehinggan dapat ditarik kesimpulan. Berikut ini adalah tahapan dalam Kuisioner Pengujian.

1. Wawancara

Wawancara dilakukan pada calon pengguna aplikasi seleksi penerima beasiswa di organisasi Keluarga Mahasiswa Pandeglang (KUMANDANG) Bandung pada bagian admin. Adapaun pertanyaan yang diajukan pada saat wawancara adalah sebagai berikut :

1) Apakah aplikasi seleksi penerima beasiswa sudah dapat menyajikan informasi seputar seleksi penerima beasiswa ?
2) Apakah informasi nilai peserta yang disajikan sesuai dengan data yang ada ?
3) Apakah aplikasi sudah dapat menyajikan data peserta dengan baik ?
4) Apakah aplikasi sudah dapat menyajikan data seleksi penerima beasiswa dengan baik ?
5) Apakah aplikasi dapat digunakan untuk proses seleksi diperiode yang akan datang ?

Tabel berikut ini adalah jawaban dari saudara Ikhwan sebagai koordinator divisi litbang yang nantinya akan bertindak sebagai admin pada aplikasi seleksi penerima beasiswa terhadap kelima pertanyaan diatas :

Tabel 4.45 Hasil wawancara pengguna sebagai admin

No	Pertanyaan	Jawaban
1	Apakah aplikasi seleksi penerima beasiswa sudah dapat menyajikan informasi seputar seleksi penerima beasiswa ?	Aplikasi ini sudah dapat menampilkan informasi seputar seleksi

2	Apakah informasi nilai peserta yang disajikan sesuai dengan data yang ada ?	Informasi nilai masing-masing peserta sesuai dengan data pada halaman admin
3	Apakah aplikasi sudah dapat menyajikan data peserta dengan baik ?	Data peserta yang ditampilkan sudah cukup baik
4	Apakah aplikasi sudah dapat menyajikan data seleksi penerima beasiswa dengan baik ?	Aplikasi sudah dapat menyajikan data seleksi dari masing-masing peserta
5	Apakah aplikasi dapat digunakan untuk proses seleksi diperiode yang akan datang ?	Aplikasi ini sudah dapat memenuhi kebutuhan dalam proses seleksi dan sudah dapat digunkana di periode seleksi slanjutnya.

Berdasarkan hasil wawancara, dapat disimpulkan bahwa aplikasi yang dibangun sudah cukup memenuhi tujuan untuk memermudah divisi litbang dalam pengolahan data peserta dan data seleksi penerima beasiswa.

4.2.2.2. Penghitungan Hasil Kuisioner

Penghitungan kuisioner menggunakan skala *Likert*. Menurut Sugiyono, skala *Likert* digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau kelompok orang tentang fenomena sosial. Fenomena sosial disini adalah tanggapan responden mengenai aplikasi seleksi penerima beasiswa yang dibuat untuk membantu KUMANDANG Bandung dalam melakukan proses seleksi pemberian beasiswa terhadapat mahasiswa asal Pandeglang yang melanjutkan studinya di Bandung. Berikut adalah langkah – langkah penelitian untuk menghitung Hasil Kuisioner.

a. Pada kuisioner ini, memberikan lima kategori nilai pilihan jawaban beserta skor dari masing – masing jawaban yang dapat dilihat pada tabel 4.46.

Tabel 4.46 Kategori Nilai

Kategori	Nilai
Sangat setuju	5
Setuju	4
Cukup setuju	3
Kurang setuju	2
Tidak setuju	1

b. Untuk menghitung jumlah maksimum skor kriterium jumlah pertanyaan = 5, dan jumlah responden = 30 yaitu dengan rumus :

$$\text{Kriterium} = \text{Jumlah soal} \times \text{Responden} \quad (4.1)$$

Maka nilai dari kuisioner yang dibuat adalah 5 x 30 = 150

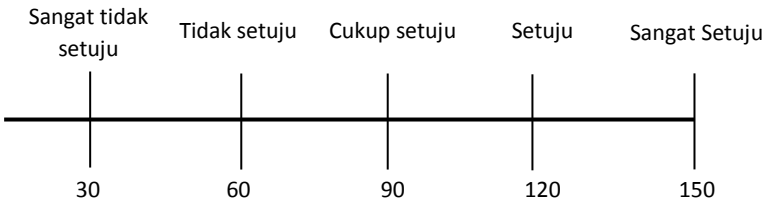
c. Sedangkan untuk mengetahui jumlah jawaban dari responden dalam bentuk presentase, digunakan rumus sebagai berikut.

$$P = \frac{total\ nilai}{skor\ ideal}\ x\ 100\ \% \quad (4.2)$$

Keterangan :

P	: Nilai presentase yang dicari
Total nilai	: Jumlah frekuensi dikalikan nilai yang ditetapkan jawaban
Skor ideal	: Nilai tertinggi dilakukan dengan jumlah sampel
100 %	: Bilangan tetap

d. Skor yang telah didapat selanjutnya dimasukan kedalam bentuk *interval ratting scale* sebagai berikut :



Gambar 4.1 Interval ratting scale

2. Kuesioner

Berikut ini adalah pertanyaan kuisioner yang akan diajukan kepada anggota KUMANDANG Bandung yang nantinya akan menjadi peserta seleksi penerima beasiswa. Pertanyaannya adalah sebagai berikut.

1) Apakah proses penilaian yang dilakukan oleh masing-masing peserta dapat dilakukan dengan mudah ?
2) Apakah hasil penilaian pada aplikasi sesuai dengan kriteria yang peserta miliki ?
3) Apakah informasi hasil penilaian pada aplikasi sesuai dengan proses penialai yang sudah dilakukan ?
4) Apakah hasil penilaian pada aplikasi dapat diterima dengan baik ?
5) Apakah aplikasi dapat digunakan untuk proses seleksi diperiode yang akan datang ?

Jawaban untuk kuisioner :

a. Sangat setuju
b. Setuju
c. Cukup setuju
d. Kurang setuju
e. Tidak setuju

Cara Perhitungan :

Berikut ini adalah hasil peresentase masing – masing jawaban yang sudah dihitung nilainya dengan menggunakan rumus diatas. Kuesioner ini telah diujikan kepada 30 orang calon pengguna aplikasi seleksi penerima beasiswa ini. Hasil perhitungan presentase masing – masing jawaban dapat dilihat pada tabel berikut :

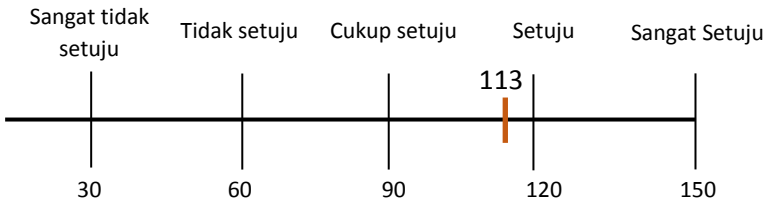
1) Apakah proses penilaian yang dilakukan oleh masing-masing peserta dapat dilakukan dengan mudah ?

Tabel 4.47 Kuesioner pertanyaan 1

Kategori Jawaban	Skor	Frekuensi Jawaban	Jumlah Skor	Nilai Presentase (%)
Sangat setuju	5	2	10	(113/(30x5))*100) =75,3%
Setuju	4	19	76	

Cukup setuju	3	9	27	
Tidak setuju	2	0	0	
Sangat tidak setuju	1	0	0	
Jumlah		30	113	

Berdasarkan perhitungan diatas, jumlah skor yang didapat adalah sebanyak 113 skor dengan skala kategori jawaban diantara cukup setuju dan setuju. Sedangkan hasil dari nilai presentasi responden adalah 75,3% dari nilai yang diaharapkan sebesar 100%. Maka dapat disimpulkan bahwa aplikasi ini dapat melakukan proses penilaian dengan mudah. Secara kontinium dapat digambarkan seperti pada gambar 4.2.



Gambar 4.2 Interval ratting scale pertanyaan 1

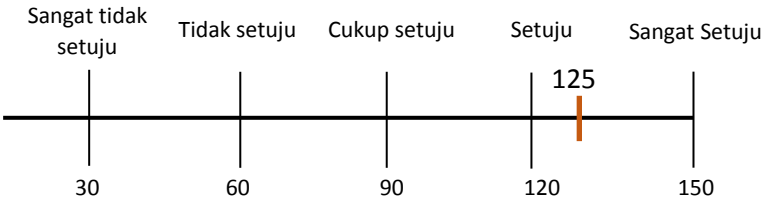
2) Apakah hasil penilaian pada aplikasi sesuai dengan kriteria yang peserta miliki ?

Tabel 4.48 Kuesioner pertanyaan 2

Kategori Jawaban	Skor	Frekuensi Jawaban	Jumlah Skor	Nilai Presentase (%)
Sangat setuju	5	13	65	(125/(30x5))*100) =83,3%
Setuju	4	9	36	
Cukup setuju	3	8	24	
Tidak setuju	2	0	0	
Sangat tidak setuju	1	0	0	
Jumlah		30	125	

Berdasarkan perhitungan diatas, jumlah skor yang didapat adalah sebanyak 125 skor dengan skala kategori jawaban diantara setuju dan sangat setuju. Sedangkan hasil dari nilai presentasi responden adalah 83,3% dari nilai yang diaharapkan sebesar 100%. Maka dapat disimpulkan bahwa aplikasi ini dapat melakukan penilaian berdsarkan kriteria yang ada. Secara kontinium dapat

digambarkan seperti pada gambar 4.3.



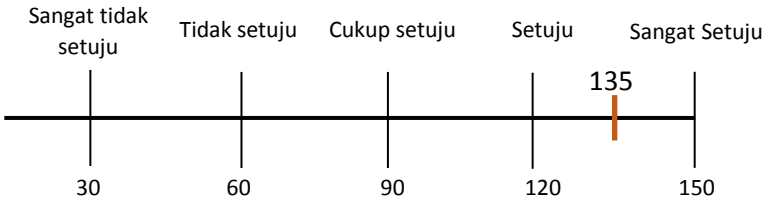
Gambar 4.3 Interval ratting scale pertanyaan 2

3) Apakah informasi hasil penilaian pada aplikasi sesuai dengan proses penialai yang sudah dilakukan ?

Tabel 4.49 Kuesioner pertanyaan 3

Kategori Jawaban	Skor	Frekuensi Jawaban	Jumlah Skor	Nilai Presentase (%)
Sangat setuju	5	16	80	(135/(30x5))*100) =90%
Setuju	4	13	52	
Cukup setuju	3	1	3	
Tidak setuju	2	0	0	
Sangat tidak setuju	1	0	0	
Jumlah		30	135	

Berdasarkan perhitungan diatas, jumlah skor yang didapat adalah sebanyak 135 skor dengan skala kategori jawaban diantara setuju dan sangat setuju. Sedangkan hasil dari nilai presentasi responden adalah 90% dari nilai yang diaharapkan sebesar 100%. Maka dapat disimpulkan bahwa aplikasi ini dapat memberikan informasi sesuai dengan data yang ada. Secara kontinium dapat digambarkan seperti pada gambar 4.4.



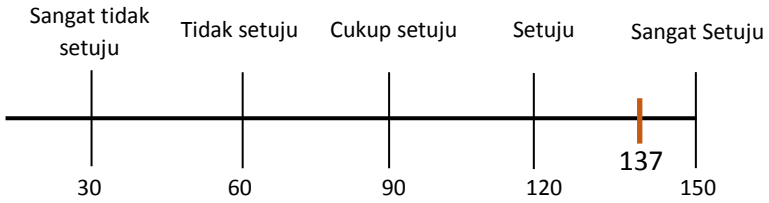
Gambar 4.4 Interval ratting scale pertanyaan 3

4) Apakah hasil penilaian pada aplikasi dapat diterima dengan baik ?

Tabel 4.50 Kuesioner pertanyaan 4

Kategori Jawaban	Skor	Frekuensi Jawaban	Jumlah Skor	Nilai Presentase (%)
Sangat setuju	5	17	85	(137/(30x5))*100) =91,3%
Setuju	4	13	52	
Cukup setuju	3	0	0	
Tidak setuju	2	0	0	
Sangat tidak setuju	1	0	0	
Jumlah		30	137	

Berdasarkan perhitungan diatas, jumlah skor yang didapat adalah sebanyak 137 skor dengan skala kategori jawaban diantara setuju dan sangat setuju. Sedangkan hasil dari nilai presentasi responden adalah 91,3% dari nilai yang diaharapkan sebesar 100%. Maka dapat disimpulkan bahwa hasil penilaian pada aplikasi ini dapat diterima dengan baik oleh peserta. Secara kontinium dapat digambarkan seperti pada gambar 4.5.



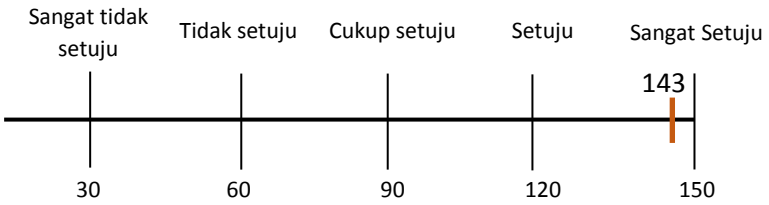
Gambar 4.5 Interval ratting scale pertanyaan 4

5) Apakah aplikasi dapat digunakan untuk proses seleksi diperiode yang akan datang ?

Tabel 4.51 Kuesioner pertanyaan 5

Kategori Jawaban	Skor	Frekuensi Jawaban	Jumlah Skor	Nilai Presentase (%)
Sangat setuju	5	23	115	(143/(30x5))*100) =95,3%
Setuju	4	7	28	
Cukup setuju	3	0	0	
Tidak setuju	2	0	0	
Sangat tidak setuju	1	0	0	
Jumlah		30	143	

Berdasarkan perhitungan diatas, jumlah skor yang didapat adalah sebanyak 143 skor dengan skala kategori jawaban diantara setuju dan sangat setuju. Sedangkan hasil dari nilai presentasi responden adalah 95,3% dari nilai yang diaharapkan sebesar 100%. Maka dapat disimpulkan bahwa peserta berharap aplikasi ini dapat digunakan di periode seleksi selanjutnya. Secara kontinium dapat digambarkan seperti pada gambar 4.6.



Gambar 4.6 Interval ratting scale pertanyaan 5

4.2.2.3. Kesimpulan Hasil Pengujian Beta

Berdasarkan hasil pengujian beta, dapat diambil kesimpulan bahwa :

1. Proses penilaian pada aplikasi dapat dilakukan dengan mudah oleh masing-masing peserta
2. Aplikasi dapat menampilkan data persyaratan sesuai dengan data dari masing-masing peserta
3. Informasi hasil penilaian yang di tampilkan pada aplikasi sudah sesuai dengan data yang ada.
4. Hasi penilaian pada aplikasi dapat diterim dengan baik oleh masing-masing peserta.
5. Aplikasi yang dibangun dapat digunakan pada periode seleksi penerima beasiswa selanjutnya.