

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 sehingga siap untuk dioperasikan. Implementasi sistem meliputi implementasi perangkat keras, implementasi perangkat lunak, implementasi data, implementasi antar muka.

1. Implementasi Perangkat Keras

Perangkat keras (*hardware*) yang digunakan untuk membangun sistem di platform mobile dan website untuk aplikasi Giving everyday dapat dilihat pada tabel 4.1 dan tabel 4.2 :

Tabel 4. 1 Implementasi Perangkat Keras di Platform Mobile

Nama Perangkat Keras	Spesifikasi
Processor	Quad-core 1.2 GHz
Memory Internal	8 GB
RAM	1 GB
Layar	4.7 inches
Sistem Operasi	Android 4.4.2 Kitkat

2. Kebutuhan Perangkat Lunak

Perangkat lunak (*software*) untuk proses implementasi sistem meliputi sistem operasi, IDE, perambah dan text editor untuk menulis kode. Tabel 4.3 merupakan implementasi perangkat lunak yang digunakan untuk membangun aplikasi Giving everyday di platform mobile dan website.

Tabel 4. 3 Implementasi Perangkat Lunak

Nama Perangkat Lunak	Spesifikasi
Sistem Operasi	Windows 8.1 profesional 64 bit

IDE	Android Studio 1.2
Perambah	Google Chrome
Text Editor	Sublime Text 3

3. Implementasi Data

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

Tabel 4. 4 Implementasi Data

No	Nama Tabel	Perintah SQL
1	user	<pre>CREATE TABLE `user` (`id_user` int(11) NOT NULL AUTO_INCREMENT, `nama` varchar(50) NOT NULL, `email` varchar(100) NOT NULL, `foto` varchar(100) NOT NULL, `passwordenkrip` varchar(100) NOT NULL, `salt` varchar(10) NOT NULL, `tanggal_buat` date NOT NULL DEFAULT '0000-00-00', `tanggal_update` date NOT NULL DEFAULT '0000-00-00', `user_level` int(1) NOT NULL DEFAULT '2', `konfirmasi` int(1) NOT NULL DEFAULT '0', `kota` varchar(100) NOT NULL, `deskripsi` text NOT NULL, `mata_uang` varchar(4) NOT NULL DEFAULT 'IDR', `nilai_tukar` double NOT NULL DEFAULT '0', `bahasa` varchar(15) NOT NULL DEFAULT 'English', `confirm_code` varchar(100) NOT NULL, PRIMARY KEY (`id_user`), UNIQUE KEY `email` (`email`)) ENGINE=InnoDB AUTO_INCREMENT=27 DEFAULT CHARSET=latin1</pre>
2	t_berita	<pre>CREATE TABLE `t_perencanaan` (`id_perencanaan` int(11) NOT NULL AUTO_INCREMENT, `judul` varchar(50) NOT NULL DEFAULT 'judul', `budget` double NOT NULL DEFAULT '0', `total_pengeluaran` double NOT NULL DEFAULT '0', `lama_hari` int(2) NOT NULL DEFAULT '0', `tanggal_berangkat` date NOT NULL DEFAULT '0000-00-00', `tanggal_kembali` date NOT NULL DEFAULT '0000-00-00', `status` varchar(20) NOT NULL DEFAULT 'Plan', `id_user` int(11) NOT NULL DEFAULT '0', `id_jenis` int(1) NOT NULL DEFAULT '0', `id_kab_asal` int(4) NOT NULL DEFAULT '0', `id_kab_tujuan` int(4) NOT NULL DEFAULT '0', PRIMARY KEY (`id_perencanaan`), KEY `fk_user` (`id_user`), KEY `fk_jenis` (`id_jenis`), KEY `fk_asal` (`id_kab_asal`), KEY `fk_tujuan` (`id_kab_tujuan`), CONSTRAINT `fk_asal` FOREIGN KEY (`id_kab_asal`) REFERENCES `t_kabupaten_asal` (`id_kab_asal`) ON DELETE CASCADE ON UPDATE CASCADE,</pre>

		<p>CONSTRAINT `fk_jenis` FOREIGN KEY (`id_jenis`) REFERENCES `t_jenis` (`id_jenis`) ON DELETE CASCADE ON UPDATE CASCADE, CONSTRAINT `fk_tujuan` FOREIGN KEY (`id_kab_tujuan`) REFERENCES `t_kabupaten_tujuan` (`id_kab_tujuan`) ON DELETE CASCADE ON UPDATE CASCADE, CONSTRAINT `fk_user` FOREIGN KEY (`id_user`) REFERENCES `user` (`id_user`) ON DELETE CASCADE ON UPDATE CASCADE) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1</p>
3	t_detail_berita	<p>CREATE TABLE `t_detail_perencanaan` (`id_detail_berita` int(11) NOT NULL AUTO_INCREMENT, `judul` varchar(100) NOT NULL, `judul` varchar(4) NOT NULL, `tempat` double NOT NULL DEFAULT '0', `waktu` double NOT NULL DEFAULT '0', `keterangan` text NOT NULL, `foto` varchar(100) NOT NULL, `lng` decimal(10,6) NOT NULL DEFAULT '0.000000', `lat` decimal(10,6) NOT NULL DEFAULT '0.000000', `tanggal_mulai` date NOT NULL DEFAULT '0000-00-00', `tanggal_selesai` date NOT NULL DEFAULT '0000-00-00', `status` varchar(20) NOT NULL, `id_perencanaan` int(11) NOT NULL, `id_kategori` int(1) NOT NULL, PRIMARY KEY (`id_detail_perencanaan`), KEY `fk_kategori` (`kategori`), KEY `fk_perencanaan` (`id_perencanaan`), KEY `fk_kategori` (`id_kategori`), CONSTRAINT `fk_kategori` FOREIGN KEY (`id_kategori`) REFERENCES `t_kategori` (`id_kategori`) ON DELETE CASCADE ON UPDATE CASCADE, CONSTRAINT `fk_perencanaan` FOREIGN KEY (`id_perencanaan`) REFERENCES `t_perencanaan` (`id_perencanaan`) ON DELETE CASCADE ON UPDATE CASCADE) ENGINE=InnoDB AUTO_INCREMENT=52 DEFAULT CHARSET=latin1</p>
4	t_jenis	<p>CREATE TABLE `t_jenis` (`id_jenis` int(1) NOT NULL AUTO_INCREMENT, `nama_jenis` varchar(30) NOT NULL, PRIMARY KEY (`id_jenis`)) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1</p>
5	t_kategori	<p>CREATE TABLE `t_kategori` (`id_kategori` int(1) NOT NULL AUTO_INCREMENT, `nama_kategori` varchar(30) DEFAULT NULL, PRIMARY KEY (`id_kategori`)) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=latin1</p>
6	t_deteksi	<p>CREATE TABLE `t_kabupaten_asal` (`id_kab_asal` int(4) NOT NULL AUTO_INCREMENT, `nama_kab` varchar(45) CHARACTER SET utf8 NOT NULL, PRIMARY KEY (`id_kab_asal`)) ENGINE=InnoDB AUTO_INCREMENT=498 DEFAULT CHARSET=latin1</p>
7	t_rekomendasi	<p>CREATE TABLE `t_kabupaten_tujuan` (`id_kab_tujuan` int(4) NOT NULL AUTO_INCREMENT, `nama_kab` varchar(45) CHARACTER SET utf8 NOT NULL, PRIMARY KEY (`id_kab_tujuan`)) ENGINE=InnoDB AUTO_INCREMENT=498 DEFAULT CHARSET=latin1</p>

4. Implementasi Antar Muka

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

Tabel 4. 5 Implementasi Antar Muka di platform mobile

Menu	Deskripsi	Nama File
Halaman Awal	Merupakan halaman ketika pengguna pertama kali membuka aplikasi Giving everyday	awal.xml
Login	Merupakan halaman login pengguna yang sudah terdaftar	login.xml
Register	Merupakan halaman daftar pengguna baru	register.xml
Forgot Password	Merupakan halaman untuk mendapatkan password baru	forgot_password.xml
Profile	Merupakan halaman profil pengguna	fragment_profile.xml
My Plan	Merupakan halaman untuk menampilkan perencanaan jamaah	fragment_myplan.xml
My Expense	Merupakan halaman untuk menampilkan pengeluaran pengguna	fragment_my_expense.xml
History	Merupakan halaman untuk menampilkan riwayat perjalanan jamaah pengguna	fragment_history.xml
Create Plan	Merupakan halaman untuk membuat perencanaan jamaah	create_my_plan.xml
Get Recommendation	Merupakan halaman untuk mendapatkan rekomendasi jamaah	rekomendasi.xml
Add Plan	Merupakan halaman untuk menambahkan perencanaan kegiatan jamaah	create_add_plan.xml
Edit Plan	Merupakan halaman untuk mengubah perencanaan kegiatan jamaah	edit_detail_plan.xml
Add Expenses	Merupakan halaman untuk menambah pengeluaran kegiatan jamaah	add_expense.xml
Edit Expenses	Merupakan halaman untuk mengubah pengeluaran kegiatan jamaah	edit_expense.xml
Search History	Merupakan halaman untuk mencari riwayat jamaah	search_history.xml
Setting	Merupakan halaman untuk mengatur pengaturan aplikasi giving everyday	fragment_setting.xml

Tabel 4. 6 Implementasi Antar Muka di platform website

Menu	Deskripsi	Nama File
Login	Merupakan halaman untuk login admin	login_form.php

Lupa password	Merupakan halaman untuk mendapatkan password baru admin	Forgot.php
Pengguna	Merupakan halaman untuk melihat data pengguna	v_user.php
dashboard	Merupakan halaman utama ketika admin sudah login	main_page.php
Konfirmasi Pengguna	Merupakan halaman untuk mengelola pengguna yang belum konfirmasi pendaftaran	v_user_un.php
Riwayat	Merupakan halaman untuk menampilkan riwayat jamaah pengguna	v_myplan_user.php
Detail Riwayat	Merupakan halaman untuk menampilkan detail riwayat jamaah pengguna	v_detailplan.php

4.2 Pengujian Sistem

Tahap yang selanjutnya adalah tahap pengujian sistem pada perangkat lunak *giving everyday*. Tahap ini merupakan hal terpenting yang bertujuan untuk menemukan kesalahan – kesalahan ataupun kekurangan – kekurangan pada aplikasi yang dibangun. Pengujian ini bermaksud untuk mengetahui apakah aplikasi yang dibuat telah memenuhi kriteria yang sesuai dengan tujuan perancangan aplikasi atau belum. Pengujian terhadap sistem perangkat lunak *giving everyday* akan menggunakan strategi pengujian *white-box*, pengujian alpha (*black-box*) dan pengujian beta.

1. Pengujian *White-Box*

Pengujian *white-box* atau *white-box testing* merupakan metode perancangan test case yang menggunakan struktur kontrol dari perancangan prosedural dalam mendapatkan test case. Adapun metode yang digunakan dalam pengujian *white-box* ini adalah metode Basis Path. Metode Basis Path memungkinkan pendesain kasus uji untuk membuat perkiraan logik yang kompleks dari desain prosedural dan menggunakan perkiraan ini untuk mendefinisikan aliran eksekusi.

1. Pengujian *White-Box* Algoritma Similarity Nearest Neighbor

a. Pengujian *white-box* rekomendasi jamaah menggunakan algoritma similarity nearest neighbor.

Berikut pseudocode algoritma similarity nearest neighbor

```

1 $q = mysql_query("SELECT * FROM `v_plan` WHERE `status` = 'History' AND `id_user` != '$id' ");
2 $response["data"] = array();
3 while ($a = mysql_fetch_array($q)){
4     $hasil = ($s1*$w1 + $s2*$w2 + $s3*$w3 + $s4*$w4)/($w1+$w2+$w3+$w4);
5     if ($hasil >= 0.52) {
6         $output = array();
7         $output["id_plan"] = $a["id_perencanaan"];
8         $output["title"]=$a["judul"];
9         $output["budget"]=$a["total_pengeluaran"];
10        $output["lama"]=$a["lama_hari"];
11        $output["nama"]=$a["nama"];
12        array_push($response["data"], $output);
13    }else{
14        $output = array();
15        $output["pesan"] = "Tidak ada rekomendasi";
16        array_push($response["data"], $output);
17    }
18 }
19 echo json_encode($response);

```

Gambar 4. 1 Pseudocode algoritma similarity nearest neighbor

Berdasarkan pseudocode pada gambar 4.1 Pseudocode algoritma similarity nearest neighbor, kemudian dirubah menjadi *flowchart* yang kemudian diubah lagi menjadi *flowgraph* yang dapat dilihat pada gambar 4.2 Merubah *flowchart* ke *flowgraph*.

Kesimpulan Pengujian White-Box

Berdasarkan hasil pengujian yang sudah dilakukan diperoleh jumlah region, *cyclomatic complexity*, dan independent path adalah sama besar yaitu 3, sehingga dapat ditarik kesimpulan bahwa algoritma tersebut sudah benar. Dengan kata lain bahwa sistem rekomendasi jamaah diperangkat lunak giving everyday yang dibuat telah layak digunakan.

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 dibuat telah benar. Tetapi jika keluaran yang dihasilkan tidak sesuai dengan kebutuhan fungsionalnya, maka masih terdapat kesalahan pada program aplikasi.

Pengujian dilakukan dengan mencoba semua kemungkinan yang terjadi dan pengujian dilakukan berulang-ulang jika dalam pengujian ditemukan kesalahan maka akan dilakukan penelusuran atau perbaikan untuk memperbaiki kesalahan yang terjadi. Jika telah selesai melakukan perbaikan, maka 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:

Tabel 4. 8 Rancangan pengujian memasukan info kajian

Kelas Uji	Butir Uji	Jenis Pengujian
Memasukan info dan tempat kajian	Menampilkan form info dan tempat kajian	Black box
	Mengisi form info dan tempat kajian	
	Verifikasi data masukan	
	memasukkan data kedalam basis data	

Tabel 4. 9 Rancangan pengujian menambah tumpangan

Kelas Uji	Butir Uji	Jenis Pengujian
Menambah tumpangan	Menampilkan form menambah tumpangan	Black box
	Mengisi form tambah tumpangan	

	Verifikasi data masukan	
	memasukkan data kedalam basis data	

Tabel 4. 10 Rancangan pengujian deteksi wajah

Kelas Uji	Butir Uji	Jenis Pengujian
Mendeteksi wajah	Menampilkan camera	Black box
	Menampilkan jumlah orang	
	Verifikasi data masukan	
	memasukkan data kedalam basis data	

Tabel 4. 11 Rancangan pengujian menghapus info kajian

Kelas Uji	Butir Uji	Jenis Pengujian
Menghapus info kajian	Menampilkan pesan konfirmasi	Black box
	Menyetujui pesan konfirmasi	
	Menghapus data pada basis data	

Tabel 4. 12 Rancangan pengujian mendapatkan rekomendasi tempat

Kelas Uji	Butir Uji	Jenis Pengujian
Mendapatkan rekomendasi tempat	Menampilkan form mendapatkan rekomendasi	Black box
	Mengisi form dapatkan rekomendasi	
	Verifikasi data masukan	
	mencari data masukkan di basis data	

Tabel 4. 13 Rancangan pengujian menumpang

Kelas Uji	Butir Uji	Jenis Pengujian
Menyimpan pengeluaran	Menampilkan form menyimpan pengeluaran	Black box
	Mengisi form simpan pengeluaran	
	Verifikasi data masukan	
	memasukkan data kedalam basis data	

4.2.1.3 Kasus dan Hasil Pengujian Alpha

Pengujian Black Box

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:

Pengujian Membuat Perencanaan

Berikut ini dapat dilihat pengujian dari mengisi nfo kajian yang dapat dilihat pada tabel 4.14 Pengujian mengisi info kajian.

Tabel 4. 14 Pengujian Mengisi info kajian

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Memasukkan data membuat berita atau info kajian, contoh masukan: Judul : move on	Membuat perencanaan, menyimpan data perencanaan berdasarkan field yang tersedia ke basis data dan menampilkan pesan “Buat perencanaan berhasil”	Dapat mengisi form buat perencanaan, dapat menambah data ke dalam basis data dan menampilkan pesan “Buat	[<input checked="" type="checkbox"/>] Diterima [<input type="checkbox"/>] Ditolak

Waktu : 18:30-20:15 wib Lokasi : masjid al-lathiif Bandung Tanggal mulai : 1 agustus 2018		perencanaan berhasil”	
Kasus dan Hasil Uji (Data Salah)			
Data Masukkan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Semua field di form buat berita tidak di isi atau salah satu field tidak di isi	Menampilkan pesan “Harap isi semua data”	Dapat menampilkan pesan “Harap isi semua data”	[<input checked="" type="checkbox"/>] Diterima [<input type="checkbox"/>] Ditolak

Pengujian Menambah Perencanaan

Berikut ini dapat dilihat pengujian dari menambah tumpangan yang dapat dilihat pada tabel 4.20 Pengujian menambah tumpangan.

Tabel 4. 20 Pengujian Menambah tumpangan

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Memasukkan data menambah tumpangan, contoh masukkan: Nama pengendara : anonim Tujuan : Bandung Biaya tumpangan : gratis / gantian nyetir	Menambah tumpangan jamaah, menyimpan data tumpangan ke dalam basis data dan menampilkan pesan “Menambah tumpangan berhasil”	Dapat mengisi form tambah tumpangan, dapat menambah data ke dalam basis data dan dapat menampilkan pesan “Menambah tumpangan berhasil”	[<input checked="" type="checkbox"/>] Diterima [<input type="checkbox"/>] Ditolak

Tanggal keberangkatan : 10 agustus 2018			
Kasus dan Hasil Uji (Data Salah)			
Data Masukkan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Semua field di form tambah tumpangan tidak di isi atau salah satu field tidak di isi	Menampilkan pesan “Harap isi semua data”	Dapat menampilkan pesan “Harap isi semua data”	[<input checked="" type="checkbox"/>] Diterima [<input type="checkbox"/>] Ditolak

Pengujian Memperbarui Perencanaan

Berikut ini dapat dilihat pengujian deteksi wajah yang dapat dilihat pada tabel 4.21 Pengujian deteksi wajah.

Tabel 4. 21 Pengujian deteksi wajah

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Masukkan gambar atau foto, contoh masukkan: gambar : perkumpulan orang Syarat : tidak membelakangi kamera dan menunduk	Mengisi foto, berhasil mengisi foto dalam basis data dan menampilkan pesan “deteksi wajah berhasil”	Dapat mengisi form input foto kegiatan, dapat mengisi foto di basis data dan dapat menampilkan pesan “Mengisi foto berhasil”	[<input checked="" type="checkbox"/>] Diterima [<input type="checkbox"/>] Ditolak
Kasus dan Hasil Uji (Data Salah)			

Data Masukkan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Semua field di form mengisi foto tidak di isi atau salah satu field tidak di isi	Menampilkan pesan “Harap isi semua data”	Dapat menampilkan pesan “Harap isi semua data”	[<input checked="" type="checkbox"/>] Diterima [<input type="checkbox"/>] Ditolak

Pengujian Menghapus Perencanaan

Berikut ini dapat dilihat pengujian dari menghapus perencanaan yang dapat dilihat pada tabel 4.22 Pengujian menghapus perencanaan.

Tabel 4. 22 Pengujian Menghapus Info Kajian

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Memilih Info kajian yang akan dihapus	Menampilkan pesan konfirmasi hapus, meghapus data pada basis data dan menampilkan pesan “Hapus info kajian berhasil”	Dapat menampilkan konfirmasi hapus, dapat menghapus data di basis data dan dapat menampilkan pesan “Hapus info kajian berhasil”	[<input checked="" type="checkbox"/>] Diterima [<input type="checkbox"/>] Ditolak

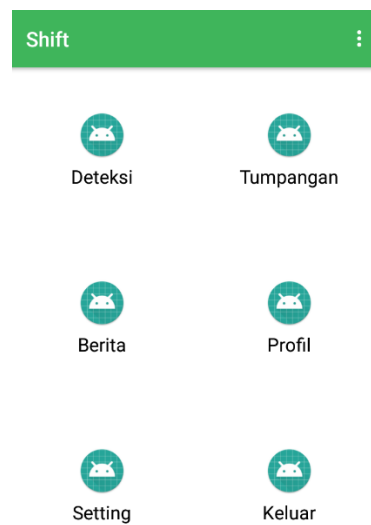
Pengujian Mendapatkan Rekomendasi

Berikut ini dapat dilihat pengujian dari mendapatkan rekomendasi yang dapat dilihat pada tabel 4.23 Pengujian mendapatkan rekomendasi.

Tabel 4. 23 Pengujian Mendapatkan Rekomendasi tempat

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Memasukkan data mendapatkan rekomendasi, contoh masukkan: lokasi : Bandung masjid al-lathiif	Mendapatkan rekomendasi tempat untuk jamaah berdasarkan field yang tersedia, menampilkan pesan “Rekomendasi tempat di temukan”	Dapat menampilkan rekomendasi jamaah dan dapat menampilkan pesan “Rekomendasi di temukan”	[<input checked="" type="checkbox"/>] Diterima [<input type="checkbox"/>] Ditolak

Kasus dan Hasil Uji (Data Salah)			
Data Masukkan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
Semua field di form dapatkan rekomendasi tidak di isi atau salah satu field tidak di isi	Menampilkan pesan "Harap isi semua data"	Dapat menampilkan pesan "Harap isi semua data"	[<input checked="" type="checkbox"/>] Diterima [<input type="checkbox"/>] Ditolak



Gambar 4. 4 Hasil pengujian rekomendasi data uji

Kesimpulan Pengujian Alpha

Berdasarkan pengujian yang telah dilakukan, maka kesimpulan yang didapatkan bahwa poses-proses pada pembangunan perangkat lunak giving everyday telah melalui tahap perbaikan dan sudah dimaksimalkan bergitupun secara fungsional sudah menghasilkan output yang diharapkan sehingga sistem

sudah layak untuk digunakan dan dari hasil pengujian rekomendasi yang dilakukan, metode *cased-based reasoning* yang sudah diimplementasikan menghasilkan rekomendasi yang baik bila nilai *similarity*-nya lebih dari atau sama dengan jumlah dari nilai bobot budget dengan nilai bobot asal *jamaah* yaitu sebesar 0,79 dan menghasilkan rekomendasi yang buruk bila nilai *similarity*-nya kurang dari 0,79. Hasil rekomendasi yang baik inilah yang akan direkomendasikan ke pengguna sebagai rekomendasi yang sesuai dengan biaya dan asal *jamaah*.

4.2.2 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif yang dilakukan secara langsung oleh pengguna yang nantinya akan menggunakan perangkat lunak *Giving everyday* ini di antaranya adalah admin *Giving everyday* dan *jamaah* yang menggunakan aplikasi. Penelitian ini dilakukan di Indonesia dengan menggunakan teknik kuisisioner untuk admin dan *jamaah*. Hal ini dilakukan agar dapat mengetahui sejauh mana aplikasi yang dibangun ini dapat membantu dan dapat menyelesaikan masalah yang sudah dijelaskan pada perumusan masalah.

4.2.2.1 Skenario Pengujian Beta

Kuisisioner Pengujian merupakan teknik pengolahan data yang merupakan langkah dalam mengolah data yang telah didapat untuk dijadikan hasil penelitian sehingga dapat ditarik kesimpulan. Berikut ini adalah pertanyaan kuisisioner yang akan diajukan kepada *jamaah* yang nantinya akan menggunakan aplikasi ini. Pertanyaannya adalah sebagai berikut:

Apakah anda setuju aplikasi *Giving everyday* dapat membantu anda mencari informasi tumpangan dan tempat?

Apakah anda setuju aplikasi *Giving everyday* dapat membantu anda untuk membuat perencanaan tumpangan dan tempat?

Apakah anda setuju aplikasi *Giving everyday* dapat membantu anda mengontrol pengeluaran selama kegiatan tumpangan dan tempat?

Apakah anda setuju hasil rekomendasi *jamaah* atau tempat sesuai dengan yang anda harapkan?

Apakah anda setuju aplikasi *Giving everyday* membantu anda untuk berbagi pengalaman *jamaah* atau pengguna?

Jawaban untuk kuisisioner :

Sangat Setuju

Setuju

Cukup Setuju

Tidak Setuju

Sangat Tidak Setuju

Penghitungan kuisisioner 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 Giving everyday yang dibuat untuk membantu *jamaah* dalam melakukan kegiatan *jamaah*. Berikut adalah langkah – langkah penelitian untuk menghitung Hasil Kuisisioner.

Pada kuisisioner ini, memberikan lima kategori nilai pilihan jawaban beserta skor dari masing – masing jawaban yang dapat dilihat pada Tabel 4.30 kategori nilai.

Tabel 4. 37 Kategori Nilai

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

Untuk menghitung jumlah maksimum skor kriterium jumlah pertanyaan = 6, dan jumlah responden = 30 yaitu dengan rumus :

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

Maka nilai dari kuesioner yang dibuat adalah $5 \times 30 = 150$

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

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

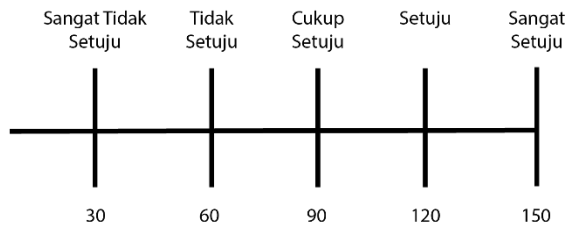
Keterangan:

P : Nilai presentase yang dicari

Total nilai : Jumlah frekuensi dikalikan nilai yang ditetapkan jawaban

Skor ideal : Nilai tertinggi dilakukan dengan jumlah sampel

Skor yang telah didapat selanjutnya dimasukkan kedalam bentuk *interval rating scale* sebagai berikut :



Gambar 4. 9 Interval Rating Scale

4.2.2.2 Penghitungan Hasil Kuesioner

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

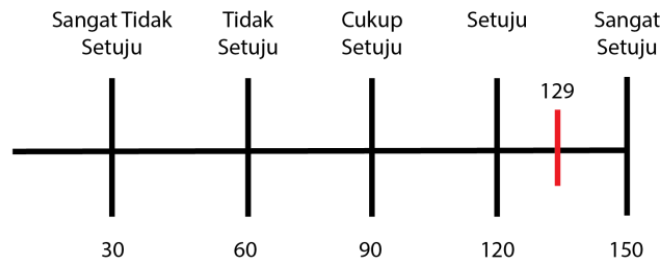
Apakah anda setuju aplikasi Giving everyday dapat membantu anda mencari informasi tumpangan dan tempat?

Tabel 4. 38 Kuesioner pertanyaan 1

Kategori Jawaban	Skor	Frekuensi Jawaban	Jumlah Skor	Nilai Presentase (%)
Sangat Setuju	5	11	55	$129 / (30 \times 5) \times 100 = 86\%$
Setuju	4	17	68	
Cukup Setuju	3	2	6	
Tidak Setuju	2	0	0	
Sangat Tidak Setuju	1	0	0	
Jumlah		30	129	

Berdasarkan perhitungan diatas, jumlah skor yang didapat adalah sebanyak 129 skor dengan skala kategori jawaban diantara setuju dan sangat setuju. Sedangkan hasil dari nilai presentasi responden adalah 86% dari nilai yang diharapkan sebesar

100%. Maka dapat disimpulkan bahwa aplikasi ini dapat membantu mencari informasi jamaah atau tempat. Secara kontinuum dapat digambarkan seperti pada gambar 4.10.



Gambar 4. 10 Interval rating scale pertanyaan 1

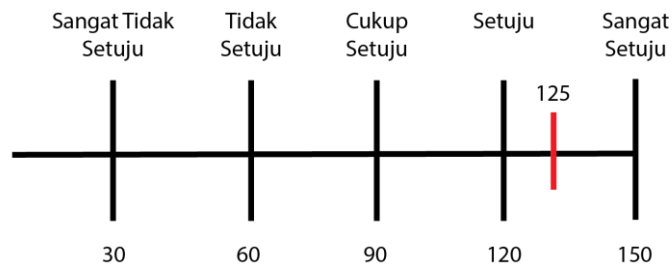
Apakah anda setuju aplikasi Giving everyday dapat membantu anda untuk membuat perencanaan jamaah atau tempat?

Tabel 4. 39 Kuesioner pertanyaan 2

Kategori Jawaban	Skor	Frekuensi Jawaban	Jumlah Skor	Nilai Presentase (%)
Sangat Setuju	5	8	40	$125 / (30 \times 5) \times 100 = 83,3\%$
Setuju	4	19	76	
Cukup Setuju	3	3	9	
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 diharapkan sebesar 100%. Maka dapat disimpulkan bahwa aplikasi ini dapat membantu membuat perencanaan jamaah atau tempat. Secara kontinuum dapat digambarkan seperti pada gambar 4.11.



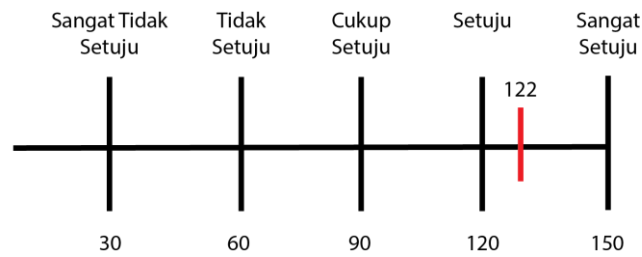
Gambar 4. 11 Interval rating scale pertanyaan 2

Apakah anda setuju aplikasi Giving everyday dapat membantu anda mengontrol pengeluaran selama kegiatan tumpangan dan tempat?

Tabel 4. 40 Kuesioner pertanyaan 3

Kategori Jawaban	Skor	Frekuensi Jawaban	jumlah Skor	Nilai Presentase (%)
Sangat Setuju	5	5	30	$122 / (30 \times 5) \times 100 = 81,3\%$
Setuju	4	20	80	
Cukup Setuju	3	4	12	
Tidak Setuju	2	0	0	
Sangat Tidak Setuju	1	0	0	
jumlah		30	122	

Berdasarkan perhitungan diatas, jumlah skor yang didapat adalah sebanyak 122 skor dengan skala kategori jawaban diantara setuju dan sangat setuju. Sedangkan hasil dari nilai presentasi responden adalah 81,3% dari nilai yang diharapkan sebesar 100%. Maka dapat disimpulkan bahwa aplikasi ini dapat membantu mengontrol pengeluaran selama kegiatan tumpangan dan tempat. Secara kontinuum dapat digambarkan seperti pada gambar 4.12.



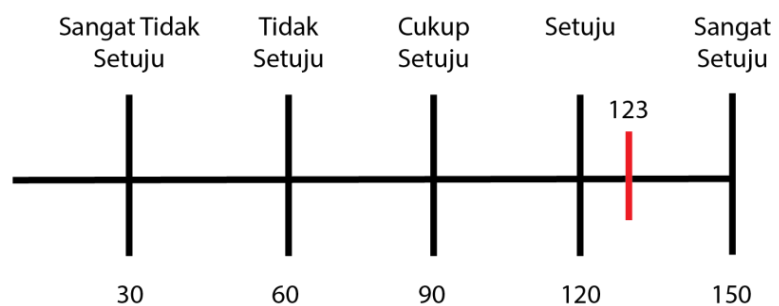
Gambar 4. 12 Interval rating scale pertanyaan 3

Apakah anda setuju hasil rekomendasi tumpangan dan tempat sesuai dengan yang anda harapkan?

Tabel 4. 41 Kuesioner pertanyaan 4

Kategori Jawaban	Skor	Frekuensi Jawaban	Jumlah Skor	Nilai Presentase (%)
Sangat Setuju	5	7	35	$(123 / (30 \times 5)) \times 100 = 82\%$
Setuju	4	19	76	
Cukup Setuju	3	4	12	
Tidak Setuju	2	0	0	
Sangat Tidak Setuju	1	0	0	
Jumlah		30	123	

Berdasarkan perhitungan diatas, jumlah skor yang didapat adalah sebanyak 123 skor dengan skala kategori jawaban diantara setuju dan sangat setuju. Sedangkan hasil dari nilai presentasi responden adalah 82% dari nilai yang diharapkan sebesar 100%. Maka dapat disimpulkan bahwa aplikasi ini dapat merekomendasikan tumpangan dan tempat sesuai dengan kebutuhan pengguna. Secara kontinum dapat digambarkan seperti pada gambar 4.13.



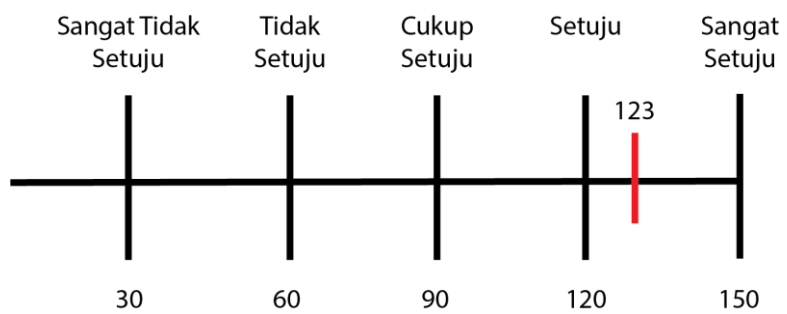
Gambar 4. 13 Interval rating scale pertanyaan 2

Apakah anda setuju aplikasi Giving everyday membantu anda untuk berbagi pengalaman tumpangan dan tempat?

Tabel 4. 42 Kuesioner pertanyaan 5

Kategori Jawaban	Skor	Frekuensi Jawaban	Jumlah Skor	Nilai Presentase (%)
Sangat Setuju	5	9	45	$123 / (30 \times 5)) \times 100 = 82\%$
Setuju	4	16	64	
Cukup	3	4	12	
Tidak Setuju	2	1	2	
Sangat Tidak Setuju	1	0	0	
Jumlah		30	123	

Berdasarkan perhitungan diatas, jumlah skor yang didapat adalah sebanyak 123 skor dengan skala kategori jawaban diantara setuju dan sangat setuju. Sedangkan hasil dari nilai presentasi responden adalah 82% dari nilai yang diharapkan sebesar 100%. Maka dapat disimpulkan bahwa aplikasi ini dapat membantu membagi pengalaman jamaah atau tempat. Secara kontinuum dapat digambarkan seperti pada gambar 4.14.



Gambar 4. 14 Interval rating scale pertanyaan 5

4.2.2.3 Kesimpulan Hasil Pengujian Beta

Berdasarkan hasil pengujian beta, dapat diambil kesimpulan bahwa :

Aplikasi dapat membantu pengguna untuk mencari informasi tempat.

Aplikasi dapat membantu pengguna untuk membuat perencanaan menumpang.

Aplikasi dapat membantu pengguna mengontrol pemilihan penumpang.

Aplikasi dapat merekomendasikan tempat sesuai dengan yang dibutuhkan pengguna.

