

UNIVERSITAS GUNADARMA

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI



METODE REKOMENDASI ITEM COLLABORATIVE FILTERING PADA SISTEM E - TRAVEL HARUNI BERBASIS WEB MOBIL MENGGUNAKAN FRAMEWORK LARAVEL

Disusun Oleh :

Nama : Muhammad Aditya Putra
NPM : 1B117042
Jurusan : Sistem Informasi
Pembimbing : Dr. Widya Silfianti, SKom., MMSI

Diajukan Guna Melengkapi Sebagian Syarat
Dalam Mencapai Gelar Sarjana Strata Satu (S1)

JAKARTA

2018

Pernyataan Originalitas dan Publikasi

Saya yang bertanda tangan di bawah ini:

Nama	:	Muhammad Aditya Putra
NPM	:	1B117042
Judul Skripsi	:	Metode Rekomendasi Item Collaborative Filtering Pada Sistem E - Travel Haruni Berbasis Web Mobil Menggunakan Framework Laravel
Tanggal Sidang	:	11 Oktober 2018
Tanggal Lulus	:	11 Oktober 2018

Menyatakan bahwa tulisan di atas merupakan hasil karya saya sendiri dan dapat dipublikasikan sepenuhnya oleh Universitas Gunadarma. Segala kutipan dalam bentuk apapun telah mengikuti kaidah dan etika yang berlaku. Semua hak cipta dari logo serta produk yang disebut dalam buku ini adalah milik masing-masing pemegang haknya, kecuali disebutkan lain. Mengenai isi dan tulisan merupakan tanggung jawab Penulis, bukan Universitas Gunadarma.

Demikianlah pernyataan ini dibuat dengan sebenarnya dan dengan penuh kesadaran.

Jakarta, Oktober 2018

(Muhammad Aditya Putra)

Lembar Pengesahan

KOMISI PEMBIMBING

No.	Nama	Kedudukan
1.	Dr. Widya Silfianti, SKom., MMSI	Ketua
2.	Dr. Rodiah, ST., MM	Anggota
3.	Dr. Trini Saptariani, SKom., MM	Anggota

Tanggal Sidang: 11 Oktober 2018

PANITIA UJIAN

No.	Nama	Kedudukan
1.	Dr. Ravi Ahmad Salim	Ketua
2.	Prof. Dr. Wahyudi Priyono	Sekretaris
3.	Dr. Widya Silfianti, SKom., MMSI	Anggota
4.	Dr. Rodiah, ST., MM	Anggota
5.	Dr. Trini Saptariani, SKom., MM	Anggota

Tanggal Lulus: 11 Oktober 2018

Mengetahui

Pembimbing

Kepala Bagian Sidang Ujian

(Dr. Widya Silfianti, SKom., MMSI)

(Dr. Edi Sukirman, SSi., MM)

Abstrak

Muhammad Aditya Putra1B117042.

Metode Rekomendasi Item Collaborative Filtering Pada Sistem E - Travel Haruni Berbasis Web Mobil Menggunakan Framework Laravel.

Penulisan Ilmiah, Fakultas Ilmu Komputer dan Teknologi Informasi, Jurusan Sistem Informasi, Universitas Gunadarma, 2018.

Kata Kunci : Sistem, Rekomendasi, Item Collaborative Filtering, Traveller, Web Mobile, Laravel

(xiv + 86 + Lampiran)

Kata *traveller* merupakan kata dari bahasa inggris yang berarti penjelajah, wisatawan, pengembara, pelancong, turis. Sebelum melakukan kegiatan ada beberapa persiapan yang harus dilakukan *traveller* diantaranya menentukan tujuan perjalanan, informasi mengenai tempat tujuan, menentukan transportasi yang akan digunakan, menentukan akomodasi ditempat tujuan, menentukan apakah melakukan kegiatan *traveller* sendirian, bersama keluarga atau rombongan dan menentukan panduan menuju tempat yang diinginkan. Persiapan lainnya yang perlu di perhatikan *traveller* adalah tempat menginap, seringkali traveller menemukan kendala dalam menentukan tempat penginapan, salah satu faktor kendala adalah biaya.

Rumusan masalah pada penulisan ini adalah bagaimana membangun sistem rekomendasi web *mobile* yang menerapkan *item collaborative filtering* sehingga para *traveller* mendapatkan rekomendasi informasi tempat wisata, informasi rumah singgah dan informasi untuk mencari teman perjalanan. Tujuan penulisan ini yaitu membuat sistem rekomendasi web *mobile*. Tujuan penulisan ini yaitu membuat sistem rekomendasi web *mobile* yang bisa digunakan oleh para *traveller* untuk mendapatkan rekomendasi informasi seperti informasi tempat wisata, informasi rumah singgah dan informasi untuk mencari teman perjalanan. Metodologi yang digunakan dalam penelitian ini adalah metode *software development live cycle* (SDLC) model *waterfall* diantaranya : analisis definisi, desain sistem, implementasi, pengujian sistem dan pemeliharaan. Sistem rekomendasi yang dibuat hanya menggunakan metode *Item base collaborative filtering*, yaitu menghasilkan rekomendasi berda-

sarkan *item* yang sama yang mereka sukai.

Penelitian ini menyimpulkan bahwa sistem yang dibuat dapat menampilkan informasi rekomendasi tempat wisata dan rumah singgah berdasarkan metode *Item base collaborative filtering*, selain itu sistem ini dapat digunakan *traveller* untuk mencari teman barengan dan berdiskusi tentang pertanyaan yang berkaitan dengan *travelling* atau berwisata.

Daftar Pustaka (2001 - 2018)

Kata Pengantar

Puji syukur kehadiran Allah SWT, karena atas rahmat dan hidayah-Nya penulis dapat menyelesaikan skripsi ini dengan judul **“Metode Rekomendasi Item Collaborative Filtering Pada Sistem E - Travel Haruni Berbasis Web Mobil Menggunakan Framework Laravel”**. Skripsi ini dibuat untuk melengkapi salah satu syarat mencapai jenjang setara sarjana S1 pada program pendidikan di Universitas Gunadarma.

Keberhasilan penulisa dalam menyelesaikan skripsi ini tidak lepas dari bantuan dan dukungan dari berbagai pihak, walaupun banyak hambatan dan kesulitan yang dihadapi, namun hal tersebut menjadi pelajaran dan pengalaman yang berharga yang berkesan bagi penulis. Oleh sebab itu, dalam kesempatan ini penulis ingin mengucapkan terima kasih kepada :

1. Prof. Dr. E.S. Margianti, S.E., M.M, selaku Rektor Universitas Gunadarma.
2. Prof. Dr. rer. nat. Achmad Benny Mutiara, SSi, SKom, selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Gunadarma.
3. Dr. Setia Wirawan, SKom., MMSI, selaku Ketua Jurusan Sistem Informasi, Universitas Gunadarma.
4. Dr. Edi Sukirman, SSi., MM, selaku Bagian Sidang Ujian.
5. Dr. Widya Silfianti, SKom., MMSI, sebagai Pembimbing penulis yang ditengah- tengah aktifitas dan kesibukannya telah membimbing penulis dan memberikan dorongan sehingga penulisan ini dapat diselesaikan.
6. Kedua orang tua dan adik tercinta yang selalu mendukung dan terus memberikan motivasi.
7. Teman-teman kelas 4KA25 yang telah memberikan semangat, do'a dan informasi terhadap hasil penulisan ini.

8. Seluruh rekan seperjuangan di Universitas Gunadarma yang telah banyak membantu penulis.
9. Seluruh rekan kerja di BPK-RI yang telah banyak membantu dan memberi semangat kepada penulis.
10. Semua pihak yang tidak disebutkan yang telah membantu penyelesaian Tugas Akhir ini, penulis ucapkan juga terima kasih atas segala bantuan dan sarannya.

Sebagai manusia biasa yang tak luput dari kesalahan, maka penulis meminta maaf atas segala kekurangan dan keterbatasan dalam penyusunan skripsi ini.

Akhir kata, hanya kepada Tuhan jualah segalanya dikembalikan dan penulis sadari bahwa penulisan ini masih jauh dari sempurna, disebabkan karena berbagai keterbatasan yang penulis miliki. Untuk itu penulis mengharapkan kritik dan saran yang bersifat membangun untuk menjadi perbaikan di masa yang akan datang.

Semoga apa yang ada pada penulisan ilmiah ini dapat bermanfaat bagi kita semua. Amin.

Jakarta, Oktober 2018

(Muhammad Aditya Putra)

Daftar Isi

Halaman Judul	i
Pernyataan Originalitas dan Publikasi	ii
Lembar Pengesahan	iii
Abstrak	iv
Kata Pengantar	vi
Daftar Isi	x
Daftar Tabel	xi
Daftar Gambar	xiii
Daftar Lampiran	xiv
1 Pendahuluan	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penulisan	4
1.5 Metodologi Penelitian	4
1.6 Sistematika Penulisan	6
2 Landasan Teori	8
2.1 Web Mobile	8
2.2 Sistem Rekomendasi	9
2.3 Collaborative Filtering	9
2.4 Sosial Media	10
2.5 Framework Laravel	11

2.6	MySQL	12
2.7	Framework Bootstrap	13
2.8	Php	14
2.9	HTML	15
2.10	JavaScript	15
2.11	CSS	16
2.12	UML (Unified Modelling Language)	16
2.13	Struktur Navigasi	17
2.14	Flowchart	19
2.15	Kuesioner	19
2.16	Pengujian Perangkat Lunak	20
2.17	SDLC	21
2.18	Xampp	22
2.19	Visual Studio Code	23
2.20	Google Chrome	23
2.21	Penelitian Terdahulu	25
3	Metodologi dan Perancangan	29
3.1	Metodologi Penelitian	29
3.2	Analisis Kebutuhan Sistem	32
3.2.1	Analisis Perhitungan <i>Item Based Collaborative Filtering</i>	32
3.2.2	Analisis Kebutuhan Fitur dan Menu	34
3.3	Analisis Data dan Pengumpulan Data	35
3.3.1	Studi Lapangan	35
3.3.2	Studi Pustaka	35
3.4	Analisis Kebutuhan Alat dan Bahan	36
3.4.1	Alat	36
3.4.2	Bahan	37
3.5	Perancangan	38
3.5.1	Rancangan UML (Unified Modeling Language)	38
3.5.2	Rancangan Struktur Navigasi	40
3.5.3	Rancangan Tampilan	42
3.5.4	Rancangan Database	45
4	Implementasi dan Uji Coba	55
4.1	Implementasi Collaborative Filtering	55
4.2	Implementasi Tampilan	60
4.3	Implementasi Database	67

4.4	Pengujian Black Box	73
4.4.1	Rencana Pengujian	73
4.4.2	Kasus dan Hasil Pengujian	74
4.5	Pengujian Fungsionalitas	78
4.6	Pengujian Fungsi Mobilitas	78
5	Penutup	81
5.1	Kesimpulan	81
5.2	Saran	81
	DAFTAR PUSTAKA	83
	LAMPIRAN	87

Daftar Tabel

2.1	Tabel Penelitian Terdahulu	25
3.1	Tabel Sample Rating	33
3.2	Tabel Hasil Perhitungan Prediksi	34
3.3	Tabel users	45
3.4	Tabel galleries	46
3.5	Tabel posts	47
3.6	Tabel post_comments	47
3.7	Tabel post_likes	48
3.8	Tabel barengans	49
3.9	Tabel barengan_comments	50
3.10	Tabel barengan_gabung	51
3.11	Tabel singgahs	51
3.12	Tabel singgah_comments	52
3.13	Tabel infos	53
3.14	Tabel info_comments	54
4.1	Tabel Sample Rating	55
4.2	Tabel Hasil Perhitungan Prediksi	58
4.3	Tabel Rencana Pengujian	73
4.4	Kasus dan hasil pengujian daftar, login dan fitur diskusi	74
4.5	Kasus dan hasil pengujian fitur cari barengan	75
4.6	Kasus dan hasil pengujian fitur rumah singgah	76
4.7	Kasus dan hasil pengujian fitur info wisata	77
4.8	Table Kuesioner	78

Daftar Gambar

3.1	Metodologi Penelitian	29
3.2	Formula Algoritma Pembangkit Rekomendasi.	32
3.3	Flowchart Perhitungan Rekomendasi.	33
3.4	Use Case Diagram E-Travel	38
3.5	Class Diagram E-Travel	39
3.6	Struktur Navigasi E-Travel	41
3.7	Rancangan Tampilan Halaman Utama / Beranda	42
3.8	Rancangan Tampilan Halaman Login	43
3.9	Rancangan Tampilan Website	43
3.10	Rancangan Tampilan Mobile	44
4.1	Formula Algoritma Pembangkit Rekomendasi.	56
4.2	Tampilan Program Hasil Perhitungan	59
4.3	Tampilan Halaman Utama Website	60
4.4	Tampilan Halaman Utama Versi Mobile	61
4.5	Tampilan Program Login Versi Website	62
4.6	Tampilan Program Login Versi Mobile	62
4.7	Tampilan Program Home Versi Website	63
4.8	Tampilan Halaman Home Versi Mobile	63
4.9	Tampilan Halaman Barengan Versi Website	64
4.10	Tampilan Halaman Barengan Versi Website	64
4.11	Tampilan Halaman Singgah Versi Mobile	65
4.12	Tampilan Halaman Singgah Versi Mobile	65
4.13	Tampilan Halaman Info Versi Website	66
4.14	Tampilan Halaman Info Versi Mobile	66
4.15	Pengujian dengan Google Chrome Pada Komputer atau Laptop	79
4.16	Pengujian dengan Mobile Phone atau Handphone	80
5.1	Halaman Beranda Versi Web	106

5.2	Halaman Beranda Versi Mobile	107
5.3	Halaman Login Versi Web	108
5.4	Halaman Login Versi Mobile	108
5.5	Halaman Daftar Versi Web	109
5.6	Halaman Daftar Versi Mobile	109
5.7	Halaman Profile User Versi Web	110
5.8	Halaman Profile User Versi Mobile	110
5.9	Ubah Profile User Versi Web	111
5.10	Ubah Profile User Versi Mobile	111
5.11	Halaman Home Versi Web	112
5.12	Halaman Home Versi Mobile	112
5.13	Halaman Barengan Versi Web	113
5.14	Halaman Barengan Versi Mobile	113
5.15	Halaman Input Barengan Versi Web	114
5.16	Halaman Input Barengan Versi Mobile	114
5.17	Halaman Rumah Singgah Versi Web	115
5.18	Halaman Rumah Singgah Versi Mobile	115
5.19	Halaman Input Rumah Singgah Versi Web	116
5.20	Halaman Input Rumah Singgah Versi Mobile	116
5.21	Halaman Info Wisata Versi Web	117
5.22	Halaman Info Wisata Versi Mobile	117
5.23	Halaman Input Info Wisata Versi Web	118
5.24	Halaman Input Info Wisata Versi Mobile	118
5.25	Halaman Detail Info Wisata Web	119
5.26	Halaman Detail Info Wisata Versi Mobile	119

Daftar Lampiran

Lampiran 1	87
Lampiran 2	106

Bab 1

Pendahuluan

1.1 Latar Belakang

Kegiatan *traveller* semakin lama semakin diminati oleh berbagai kalangan, terutama anak muda. Banyak orang yang melakukan kegiatan *travelling* untuk mengatasi stres rutinitas sehari – hari seperti pekerja kantor, mahasiswa dan lain sebagainya. Berbagai penelitian selama 30 tahun terakhir menunjukkan hasil yang konsisten bahwa individu akan merasa dirinya lebih baik dan lebih bisa mengatasi stres setelah melakukan traveling. Contoh pada penelitian dari American Psychological Association (2013) tujuan utama dari *traveling* adalah untuk keluar dari lingkungan dan rutinitas sehari-hari manusia yang merupakan sumber utama stres yang sering dirasakan. Stres dan jenuh dengan kegiatan sehari-hari, *traveling* bisa jadi salah satu obat yang mujarab.

Kata *traveller* merupakan kata dari bahasa Inggris yang berarti penjelajah, wisatawan, pengembara, pelancong, turis. *Traveller* menunjukan kepada orang yang melakukan kegiatan perjalanan dari satu tempat ke tempat lain untuk berwisata, menambah pengetahuan atau berpetualang dengan kurun waktu yang cukup lama. Kegiatan *traveller* dapat dilakukan sendiri atau dengan beberapa orang namun sangat jarang mengandalkan jasa lain seperti agen travel. Tidak sedikit orang yang menganggap kegiatan *traveller* sebagai hobi dan gaya hidup.

Traveller sudah semakin diminati dan tak sedikit yang menjadikan sebagai gaya hidup. Menurut situs elfaviona.com (2017) sebelum melakukan perjalanan, ada beberapa persiapan yang harus dilakukan *traveller* diantaranya menentukan tujuan perjalanan, informasi mengenai tempat tujuan, menentukan transportasi yang akan digunakan, menentukan akomodasi di-

tempat tujuan, menentukan apakah melakukan kegiatan *traveller* sendiri-an, bersama keluarga atau rombongan dan menentukan panduan menuju tempat yang diinginkan. Persiapan lainnya yang perlu di perhatikan *traveller* adalah tempat menginap, seringkali *traveller* menemukan kendala dalam menentukan tempat penginapan, salah satu faktor kendala adalah biaya. Situs Phinemo.com (2016) menuturkan para *backpacker* atau *traveller* juga tidak jarang meminimalkan budget menginap yang biasanya memakan biaya paling banyak atau bahkan ada juga yang rela tidur di alam terbuka agar lebih hemat.

Platform travel niaga Travelport (2017) mengeluarkan hasil survei perilaku wisatawan global dalam menggunakan alat digital saat merencanakan travel, pemesanan dan melakukan perjalanan. Indonesia menjadi negara di peringkat ke tiga terbesar dalam kategori wisatawan digital. Peringkat pertama diduduki oleh China dan selanjutnya adalah India. 93 persen wisatawan Indonesia menggunakan video dan foto dari media sosial (*review*) ketika merencanakan perjalanan, 71 persen menggunakan pencarian suara, 84 persen masih lebih memilih untuk berkonsultasi pada agen perjalanan. Elmore (2010) menyatakan media sosial adalah bentuk interaksi baru yang memungkinkan semua orang menjadi sumber informasi melalui teknologi tanpa harus mempunyai pengalaman, sehingga memberi kesempatan kepada semua orang untuk bisa menjadi sumber informasi.

Saat penelitian ini dilakukan sudah ada situs yang bisa digunakan *traveller* untuk mencari informasi yang dibutuhkan, seperti situs *tripadvisor.com* bisa digunakan untuk mencari informasi akomodasi, informasi transportasi dan informasi untuk memilih destinasi. Tidak hanya situs *tripadvisor.com*, situs *couchsurfing.com* juga bisa digunakan *traveller* untuk *sharing* pengalaman perjalanan, tips-tips hemat bepergian di destinasi tertentu dan untuk mencari informasi rumah singgah. Namun, menurut pengamatan peneliti konten yang bisa digunakan khusus *traveller* yang berasal indonesia dirasa masih kurang karena situs - situs tersebut berisi informasi dari *traveller* berbagai negara dan saat ini belum ada situs serupa yang bisa digunakan khusus *traveller* indonesia untuk dijadikan sumber informasi yang dibutuhkan.

Berdasarkan latar belakang yang telah diuraikan, maka dibuat sebuah sistem berbasis web mobile di bidang media sosial yang bisa dimanfaatkan para *traveller* indonesia untuk mendapatkan rekomendasi informasi seperti informasi tempat wisata, informasi rumah singgah sebagai solusi meminimalkan budget untuk menginap dan informasi untuk mencari teman perjalanan

dengan menggunakan metode *item collaborative filtering*. Alasan menggunakan metode tersebut adalah kecocokan antara sistem yang akan dibuat dengan metode yang akan dipilih menggunakan perhitungan yang berdasarkan rating yang diberikan pengguna terhadap konten yang tersedia. Pengertian *Collaborative filtering* itu sendiri adalah salah satu teknik yang paling dikenal dalam menghasilkan sistem rekomendasi. Metode *collaborative filtering* dapat dibedakan menjadi *user-based* dan *item-based*. *User based collaborative filtering* menghasilkan rekomendasi untuk pengguna berdasarkan item yang disukai oleh pengguna lain yang sejenis. *Item based collaborative filtering* menghasilkan rekomendasi berdasarkan item yang sama yang mereka sukai (Shambour, et al., 2016). Bahasa yang digunakan dalam pembuatan sistem rekomendasi ini adalah bahasa pemrograman PHP dengan *framework* Laravel dan menggunakan database MySQL.

1.2 Rumusan Masalah

Rumusan masalah pada penulisan ini adalah bagaimana membangun sistem rekomendasi web mobile yang menerapkan *item collaborative filtering* sehingga para *traveller* mendapatkan rekomendasi informasi tempat wisata, informasi rumah singgah dan informasi untuk mencari teman perjalanan.

1.3 Batasan Masalah

Masalah yang akan dibahas pada penulisan ini hanya mengenai pembuatan sistem rekomendasi web mobile *traveller* menggunakan *Framework* Php Laravel, *Framework* Bootstraps dan database MySQL. Aplikasi yang digunakan dalam proses pembuatan antara lain: Sublime Text 3, Xampp dan Google chrome. Fitur menu yang disediakan antara lain: diskusi, cari barengan, info wisata dan rumah singgah.

Pada penelitian ini sistem rekomendasi yang dibuat hanya menggunakan metode *Item base collaborative filtering* sehingga para *traveller* mendapatkan rekomendasi informasi tempat wisata, informasi rumah singgah dan informasi untuk mencari teman perjalanan.

1.4 Tujuan Penulisan

Tujuan penulisan ini yaitu membuat sistem rekomendasi web mobile yang bisa digunakan oleh para *traveller* untuk mendapatkan rekomendasi informasi seperti informasi tempat wisata, informasi rumah singgah dan informasi untuk mencari teman perjalanan.

1.5 Metodologi Penelitian

Metodologi yang digunakan dalam penelitian ini adalah metode *software development live cycle* (SDLC) model *waterfall*. berikut dibawah ini adalah tahapan yang terdapat pada penelitian ini.

1. Analisis Desifinisi

Pada tahap ini adalah proses menganalisa data yang dibutuhkan, adapun analisis tersebut terdiri dari :

(a) Analisa kebutuhan sistem

Tahap pertama menganalisa kebutuhan sistem, pada tahap ini membahas bagaimana analisis perhitungan *item based collaborative filtering* dan menganalisis kebutuhan fitur atau menu yang akan tersedia pada sistem.

(b) Analisa Data dan Pengumpulan Data

Tahap ini dilakukan menganalisis data dan mengumpulkan data dengan teknik studi lapangan yaitu observasi dan wawancara, selain itu pada tahap dilakukan studi pustaka.

(c) Analisis Kebutuhan Alat dan Bahan

Pada bagian ini dilakukan analisa terhadap kebutuhan perangkat keras dan perangkat lunak yang digunakan oleh pengembang dan *traveller*.

2. Desain Sistem

Tahap desain sistem dilakukan untuk merancang sistem yang akan dibuat. Terdapat 4 tahap pada desain sistem, yaitu :

(a) Merancang UML (*unified modeling language*)

Pada tahap merancang UML (*unified modeling language*) dilakukan perancangan *use case diagram* dan perancangan *class diagram*.

(b) Merancang Struktur Navigasi

Struktur navigasi yang dibuat pada tahap ini menggunakan struktur navigasi campuran (*composite*), struktur navigasi dibuat untuk menjelas alus navigasi yang akan diterapkan pada sistem.

(c) Merancang Tampilan

Pada tahap merancang tampilan dibuat untuk gambaran rancangan dari desain atau tampilan yang akan dibuat. Rancangan tampilan yang dibuat antara lain adalah rancangan tampilan halaman beranda atau halaman utama, halaman *login*, halaman tampilan *website* versi *dekstop* atau laptop dan halaman tampilan *mobile*

(d) Merancang Database

Pada tahap ini dilakukan rancangan database untuk sistem yang dibuat. Rancangan tersebut meliputi nama field, tipe data, panjang dari field dan *attribute* tambahan. Database pada sistem ini menggunakan database MySQL dengan nama database “skripsi” dan terdapat 12 tabel yaitu yaitu tabel *users*, *galleries*, *posts*, *post_likes*, *post_comments*, *barengan*, *barengan_comments*, *barengan_gabung*, *infos*, *info_review*, *singgahs* dan tabel *sing_gah_review*.

3. Implementasi

Tahap implementasi sistem dilakukan untuk membangun sistem yang dibuat berdasarkan hasil dari tahap analisis dan desain sistem. Terdapat 3 tahap pada tahap implementasi, yaitu :

(a) Implementasi Sistem Rekomendasi

Pada tahap ini dilakukan implementasi proses perhitungan pada *code* sistem yang dibuat dengan menggunakan metode perhitungan algoritma pembangkit rekomendasi.

(b) Implementasi Tampilan

Implementasi tampilan dibuatkan untuk pembuatan tampilan dari masing - masing halaman dan fitur menu yang tersedia pada sistem.

(c) Implementasi Datatabase.

Pada tahap ini dilakukan untuk membuat database dan tabel berdasarkan rancangan yang sebelumnya dibuat.

4. Pengujian Sistem

terdapat 3 tahap pada tahap pengujian pengujian sistem, yaitu :

- (a) pengujian sistem menggunakan metode *black box*

Tahap ini dilakukan pengujian dengan metode *black box testing* untuk menguji apakah sistem yang dibuat sudah berfungsi sesuai yang diharapkan.

- (b) pengujian fungsionalitas

Pada tahap ini dilakukan dengan cara mengisi kuesioner untuk menguji apakah sistem yang dibuat sudah sesuai, dengan melibatkan 20 responden *traveller*.

- (c) fungsi mobilitas

Pengujian fungsi mobilitas dilakukan untuk menguji apakah sistem yang dibuat dapat berjalan pada perangkat yang diuji. perangkat yang diuji antara lain adalah komputer *dekstop*, laptop dan perangkat *mobile*.

5. Pemeliharaan

Dalam penelitian ini, penulis membatasi pada tahap ini karena sistem yang dibuat belum dirilis.

1.6 Sistematika Penulisan

BAB I PENDAHULUAN

Bab ini di uraikan latar belakang, rumusan masalah, batasan masalah, tujuan dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini berisi penjelasan teori yang berkaitan dalam pembuatan sistem dan teori tentang media atau *software* apa saja yang digunakan untuk membuat aplikasi ini.

BAB III METODOLOGI DAN PERANCANGAN

Bab ketiga membahas tentang metodologi dan perancangan yang digunakan dalam pembuatan sistem

BAB IV IMPLEMENTASI DAN PENGUJIAN

Bab keempat membahas tentang proses implementasi sampai pada tahap pengujian sistem yang dibuat

BAB V PENUTUP

Pada bab ini berisikan kesimpulan dan saran.

Bab 2

Landasan Teori

2.1 Web Mobile

Web mobile atau mobile web merupakan sebuah kumpulan halaman html dengan berbasis browser. yang dapat diakses dengan menggunakan perangkat portable seperti smartphone dan gadget atau tablet. melalui jaringan internet atau telekomunikasi berupa 3G, 4G, atau wifi. Mobile web ini dirancang untuk menampilkan sebuah text atau konten data gambar, audio dan video atau penggabungan dari beberapa konten yang berada pada layar sebuah smartphone dan tentu juga di pengaruhi oleh keterbasan ukuran layar dan jenis layar sentuhnya.

Mobile web mempunyai prinsip dalam sistemnya, yakni *develope once run everywhere* yang dapat berarti bahwa mobile web ini beroperasi dalam lintas platform untuk sekali pengembangan. jadi kesimpulanya mobile web lebih relatif tidak membutuhkan banyak biaya dalam proses pengembangan jika di bandingkan dengan mobile aplikasi di sebabkan dapat berjalannya dengan baik di semua browser dan platform mobile. mobile web juga dapat di distribusikan tanpa harus mendapatkan approve dari pihak tertentu serta pengguna tidak perlu susah payah mencari di toko aplikasi. namun dengan demikian bukan berarti mobile web ini tidak memiliki kekurangan. tetap saja mobile web ini mempunyai kekurangan namanya juga sistem pasti tidak lepas dengan kekurangan meskipun tidak membuat rugi pengguna secara financial atau material. dan kekurangan mobile web ini terdapat pada domain yang dimana pengguna mobile web harus memiliki 2 buah domain dan bila ada gangguan atau masalah maka akan rumit dalam memperbaikinya. (Ayatulloh. 2014).

2.2 Sistem Rekomendasi

Sistem rekomendasi merupakan sebuah perangkat lunak yang bertujuan untuk membantu pengguna dengan cara memberikan rekomendasi kepada pengguna ketika pengguna dihadapkan dengan jumlah informasi yang besar. Rekomendasi yang diberikan diharapkan dapat membantu pengguna dalam proses pengambilan keputusan, seperti barang apa yang akan dibeli, buku apa yang akan dibaca, atau musik apa yang akan didengar, dan lainnya (Ricci, Francesco, Lior Rokach, dan Bracha Shapira, 2011).

Menurut Tang, J., Hu, X. dan Liu (2013) sejak pertama kali ditemukan pada tahun 1990, ada banyak penelitian mengenai sistem rekomendasi dilakukan. Sistem rekomendasi mulai diaplikasikan ke berbagai bidang dengan metode yang berbeda seperti Content Based, Collaborative Filtering, dan Hybrid. Sistem rekomendasi yang menggunakan metode Content Based menggunakan kesamaan produk untuk ditawarkan kepada pengguna. Namun, metode Content Based memiliki kelemahan yaitu ketika fitur konten yang tersedia terbatas, maka akurasi rekomendasi yang dihasilkan cukup rendah (Yuan, Q., Cong, G. & Lin, C.Y., 2014).

Collaborative Filtering adalah metode yang paling sering digunakan untuk membangun sistem rekomendasi. Metode ini bergantung pada riwayat pemilihan atau riwayat penilaian (Su X dan Khoshgoftaar, 2009). Metode Hybrid menggabungkan metode Content Based dan Collaborative Filtering untuk menghasilkan rekomendasi yang lebih baik (Tang et al., 2013).

2.3 Collaborative Filtering

Collaborative filtering merupakan proses penyaringan atau menggunakan opini orang lain (Schafer, J.B., Frankowski, D., Herlocker, J. dan Sen, S., 2007). Collaborative filtering melakukan penyaringan data berdasarkan kemiripan karakteristik sehingga mampu memberikan informasi yang baru kepada konsumen karena system memberikan informasi berdasarkan pola satu kelompok konsumen yang hamper sama. Perbedaan minat pada beberapa anggota kelompok menjadikan sumber informasi baru yang mungkin bermanfaat bagi anggota kelompok lainnya.

Schafer membagi algoritma collaborative filtering ke dalam dua kelas yang berbeda, yaitu :

1. User Based Collaborative Filtering

User based nearest neighbor algorithm menggunakan teknik statistika untuk menemukan sekumpulan pengguna, dikenal sebagai tetangga. Yang memiliki sejarah setuju dengan pengguna yang menjadi sasaran. Setelah sekumpulan tetangga terbentuk sistem menggunakan algoritma yang berbeda untuk menggabungkan kesukaan neighbours untuk menghasilkan predikis atau rekomendasi N-teratas untuk active user. (Sarwar, Karypis, G., Konstan, J. A., and Riedl. J. 2001).

2. Item Base Collaborative Filtering

Item based atau disebut juga Model based, adalah salah satu teknik dari metode sistem rekomendasi yaitu collaborative filtering. Item based memberikan rekomendasi dari kemiripan antar item berdasarkan rating yang diberikan pengguna. Kelebihan dari item based adalah komputasi yang lebih cepat dan ketahanan terhadap perubahan data (Sarwar, et al., 2001).

Metode ini muncul sebagai solusi untuk beberapa permasalahan pada user based collaborative filtering yaitu pada masalah keterbatasan (sparsity) dan skalabilitas serta masalah waktu dan memori.

Pada metode item based collaborative filtering melakukan similaritas dengan membentuk suatu model similairtas secara offline yang secara otomatis akan menghemat waktu dan memori yang digunakan untuk perhitungan pada saat pengguna mengakses halaman situs.

2.4 Sosial Media

Sosial media situs jaringan sosial misalnya layanan berbasis web yang memungkinkan bagi setiap individu untuk membangun profil publik ataupun semi publik dalam sistem terbatas, daftar pengguna lain dengan siapa mereka terhubung, dan melihat dan menjelajahi daftar koneksi mereka yang dibuat oleh orang lain dengan suatu sistem. (Henderi, Muhammad Yusup, dan Yuliana Isma Graba . 2007).

menurut Puntoadi (2011) pengguna media sosial berfungsi sebagai berikut :

1. Keunggulan membangun personal branding melalui sosial media adalah tidak mengenal trik atau popularitas semu, karena aduensiilah yang akan menentukan. Berbagai sosial media menjadi media untuk orang yang berkomunikasi, berdiskusi dan bahkan memberikan sebuah popularitas di media sosial.
2. Media sosial memberikan sebuah kesempatan yang berfungsi interaksi lebih dekat dengan konsumen. Media sosial menawarkan *content* komunikasi yang lebih individual. Melalui media sosial pula berbagai para pemasar dapat mengetahui kebiasaan dari konsumen mereka dan melakukan suatu interaksi secara personal serta dapat membangun sebuah ketertarikan yang lebih dalam.

2.5 Framework Laravel

Laravel adalah framework PHP yang dikembangkan pertama kali oleh Taylor Otwell. Walaupun termasuk pemain baru, namun komunitas pengguna laravel sudah berkembang pesat dan mampu menjadi alternatif utama dari sejumlah framework besar seperti CodeIgniter & Yii. Laravel oleh para developer disetarakan dengan CodeIgniter dan FuelPHP namun memiliki keunikan tersendiri dari sisi coding yang lebih ekspresif dan elegan.

Keunggulan Laravel daripada framework lain antara lain:

- Coding yang simple – Tersedia generator yang canggih dan memudahkan, Artisan CLI
- Fitur Schema Builder untuk berbagai database,
- Fitur Migration & Seeding untuk berbagai database,
- Fitur Query Builder yang keren,
- Eloquent ORM yang luar biasa,
- Fitur pembuatan package dan bundle,

Laravel untuk pertama kali dikembangkan sendiri oleh Taylor Otwell. Namun, sampai versi ke-4 sekarang, framework opensource ini dikembangkan bersama oleh komunitas dengan tokoh-tokoh penting selain Otwell adalah

Dayle Rees, Shawn McCool (pembaca Nettuts pasti hafal orang ini), Jeffrey Way, Jason Lewis, Ben Corlett, Franz Liedke, Dries Vints, Mior Muhammad Zaki dan Phil Sturgeon. Mereka adalah kontributor sejumlah framework dan library PHP yang luar biasa. (Ifhtul Emka, 2013).

2.6 MySQL

MySQL adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengolahan datanya. (Arief , 2011)

MySQL dikembangkan oleh perusahaan swedia bernama MySQL AB yang pada saat ini bernama Tcx DataKonsult AB sekitar tahun 1994-1995, namun cikal bakal kodenya sudah ada sejak tahun 1979. Awalnya Tcx merupakan perusahaan pengembang software dan konsultan database, dan saat ini MySQL sudah diambil alih oleh Oracle Corp.

MySQL adalah pasangan serasi dari PHP. MySQL dibuat dan dikembangkan oleh MySQL AB yang berada di Swedia. MySQL dapat digunakan untuk membuat dan mengelola database beserta isinya. Kita dapat memanfaatkan MySQL untuk menambahkan, mengubah dan menghapus data yang berada dalam database. MySQL merupakan sistem manajemen database yang bersifat relational, artinya data-data yang dikelola dalam database akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan menjadi jauh lebih cepat. MySQL memiliki beberapa keistimewaan, antara lain :

1. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka. MySQL di distribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. Multi-user. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. 'Performancetuning', MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL persatuan waktu.

5. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti signed/unsigned integer, float, double, char, text, date, timestamp, dan lainlain.
6. Perintah dan fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (query).
7. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan pembatasan. MySQL mampu menangani basis dalam skala besar, dengan jumlah rekaman(records) lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix socket (UNIX), atau Named Pipes (NT).
10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk didalamnya.
11. Antar Muka. MySQL memiliki antar muka (Interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).
12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan(tool) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
13. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle. (Erudeye, 2015)

2.7 Framework Bootstrap

Bootstrap adalah sebuah library framework CSS yang dibuat khusus untuk bagian pengembangan front-end website. Bootstrap juga merupakan

salah satu framework HTML, CSS dan javascript yang paling populer di kalangan web developer yang digunakan untuk mengembangkan sebuah website yang responsive. Sehingga halaman website nantinya dapat menyesuaikan sesuai dengan ukuran monitor device (desktop, tablet, ponsel) yang digunakan pengguna disaat mengakses website-website dari browser.

Bootstrap merupakan sebuah framework css yang memudahkan pengembangan untuk membangun website yang menarik dan responsif. Tidak konsistensinya terhadap aplikasi individual membuat sulitnya untuk mengembangkan dan pemeliharannya. Bootstrap adalah css tetapi dibentuk dengan LESS, sebuah pre-processor yang memberi fleksibilitas dari css biasa. Bootstrap memberikan solusi rapi dan seragam terhadap solusi yang umum, tugas interface yang setiap pengembang hadapi. Bootstrap dapat dikembangkan dengan tambahan lainnya karena ini cukup fleksibel terhadap pekerjaan design butuhkan.

Bootstrap adalah suatu kerangka kerja untuk membuat suatu aplikasi ataupun situs web yang bersifat responsive dengan mudah, cepat dan gratis. Responsive artinya adalah aplikasi ataupun situs yang kita buat dapat menyesuaikan ukurannya dengan layar perangkat yang kita gunakan seperti personal computer, smartphone dan tab.

Dengan menggunakan bootstrap maka kita mampu membuat suatu situs ataupun aplikasi web dengan menghemat waktu serta mampu berjalan atau beroperasi pada banyak browser (Husen Alatas,2013)

2.8 Php

PHP adalah Bahasa server-side –scripting yang menyatu dengan HTML untuk membuat halaman web yang dinamis (Arief, 2011). Menurut Betha Sidik, dalam bukunya yang berjudul Pemrograman Web Dengan PHP (2012 : 4), menyebutkan bahwa : "PHP merupakan secara umum dikenal dengan sebagai bahasa pemrograman script – script yang membuat dokumen HTML secara on the fly yang dieksekusi di server web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemrograman server side".

Kelebihan PHP dari Bahasa Pemrograman lain antara lain :

1. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak dapat melakukan sebuah kompilasi dalam penggunaannya.

2. Web Server yang mendukung PHP dapat ditemukan dimana-mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan developer yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa open source yang dapat digunakan diberbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah sistem. (Erudeye, 2015)

2.9 HTML

Hyper Text Markup Language (HTML) adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web Internet dan pemformatan hiperteks sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan dalam format ASCII normal sehingga menjadi halaman web dengan perintah-perintah HTML.

Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (Standard Generalized Markup Language), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh World Wide Web Consortium (W3C). HTML dibuat oleh kolaborasi Caillau TIM dengan Bernerslee Robert ketika mereka bekerja di CERN pada tahun 1989 (CERN adalah lembaga penelitian fisika energi tinggi di Jenewa). (Erudeye, 2015)

2.10 JavaScript

JavaScript adalah bahasa pemrograman web yang bersifat Client Side Programming Language. Client Side Programming Language adalah tipe baha-

sa pemrograman yang pemrosesannya dilakukan oleh client. Aplikasi client yang dimaksud merujuk kepada web browser seperti Google Chrome dan Mozilla Firefox.

JavaScript bahasa pemrograman berbasis java yang merupakan interface pembantu dalam pemrograman web. JavaScript populer di internet dan dapat bekerja di sebagian besar penjelajah web populer seperti Internet Explorer (IE), Mozilla Firefox, Netscape dan Opera. Kode JavaScript dapat disisipkan dalam halaman web menggunakan tag SCRIPT dalam file HTML. (Erudeye, 2015)

2.11 CSS

Cascading Style Sheet (CSS) merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. CSS bukan merupakan bahasa pemrograman. Sama halnya styles dalam aplikasi pengolahan kata seperti Microsoft Word yang dapat mengatur beberapa style, misalnya heading, subbab, bodytext, footer, images, dan style lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (file). Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML. (Erudeye, 2015)

2.12 UML (Unified Modelling Language)

UML (Unified Modelling Language) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual (Braun D., Sivils J., Shapiro A., Versteegh J. 2001). UML Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek (Whitten, Bentley, Ditman. 2004).

Sejarah UML sendiri terbagi dalam dua fase; sebelum dan sesudah munculnya UML. Dalam fase sebelum, UML sebenarnya sudah mulai diperkenalkan sejak tahun 1990an namun notasi yang dikembangkan oleh para ahli analisis dan desain berbeda-beda, sehingga dapat dikatakan belum memiliki standarisasi.

Fase kedua; dilandasi dengan pemikiran untuk mempersatukan metode tersebut dan dimotori oleh Object Management Group (OMG) maka pengembangan UML dimulai pada akhir tahun 1994 ketika Grady Booch de-

ngan metode OOD (Object-Oriented Design), Jim Rumbaugh dengan metode OMT (Object Modelling Technique) mereka ini bekerja pada Rational Software Corporation dan Ivar Jacobson dengan metode OOSE (Object-Oriented Software Engineering) yang bekerja pada perusahaan Objectory Rasional.

Sebagai pencetus metode-metode tersebut mereka bertiga berinisiatif untuk menciptakan bahasa pemodelan terpadu sehingga pada tahun 1996 mereka berhasil merilis UML versi 0.9 dan 0.91 melalui Request for Proposal (RFP) yang dikeluarkan oleh OMG (Braun, et.al. 2001).

Kemudian pada Januari 1997 IBM, ObjecTime, Platinum Technology, Ptech, Taskon, Reich Technologies dan Softeam juga menanggapi Request for Proposal (RFP) yang dikeluarkan oleh OMG tersebut dan menyatakan kesediaan untuk bergabung.

Perusahaan-perusahaan ini menyumbangkan ide-ide mereka, dan bersama para mitra menghasilkan UML revisi 1.1. Fokus dari UML versi rilis 1.1 ini adalah untuk meningkatkan kejelasan UML Semantik versi rilis 1.0. Hingga saat ini UML versi terbaru adalah versi 2.0 (<http://www.uml.org/>).

Saat ini sebagian besar para perancang sistem informasi dalam menggambarkan informasi dengan memanfaatkan UML diagram dengan tujuan utama untuk membantu tim proyek berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuat program. (Haviludin. 2011).

2.13 Struktur Navigasi

Struktur Navigasi adalah susunan menu atau hirarki dari suatu situs yang menggambarkan isi dari setiap halaman dan link atau navigasi tiap halaman pada suatu situs web. (Prihatna, 2005) . Struktur navigasi suatu situs web sangat di pengaruhi oleh tujuan dari situs web yang akan dibuat. Struktur navigasi dapat digolongkan menurut kebutuhan akan objek, kemudahan pemakaian, keinteraktifitasannya dan kemudahan membuatnya yang berpengaruh terhadap waktu pembuatan suatu situs web. Bentuk dasar dari struktur navigasi adalah sebagai berikut:

1. Linear (satu alur) merupakan struktur yang hanya mempunyai satu rangkaian cerita yang berurut. Dengan kata lain struktur ini hanya dapat menampilkan satu demi satu tampilan layar secara berurut menurut urutannya. Tampilan yang dapat ditampilkan pada struktur jenis

ini adalah satu halaman sebelumnya atau satu halaman sesudahnya dan tidak dapat menampilkan dua halaman sebelumnya atau dua halaman sesudahnya. Salah satu yang terpenting dari struktur ini adalah tidak diperkenankan terjadinya percabangan.

2. Struktur Hierarchi (bercabang) ini percabangan untuk menampilkan data berdasarkan kriteria tertentu. Tampilan pada menu pertama akan disebut sebagai Master Page (halaman utama kesatu), halaman utama ini akan mempunyai halaman percabangan yang dikatakan Slave Page (halaman pendukung). Jika salah satu halaman pendukung dipilih atau diaktifkan, maka tampilan tersebut akan bernama Master Page (halaman utama kedua), dan seterusnya. Yang terpenting dari struktur penjejakan ini tidak diperkenankan adanya tampilan secara linear.
3. Struktur penjejakan Non Linear (tidak berurut) merupakan pengembangan dari struktur penjejakan Linear. Pada struktur ini diperkenankan membuat penjejakan bercabang. Pemakai bebas menelusuri website tanpa dibatasi oleh suatu rute dimana kontrol navigasi dapat mengakses ke semua halaman manapun. Percabangan yang dibuat pada struktur Non Linear ini berbeda dengan percabangan pada struktur Hierarchi, karena pada percabangan Non Linear ini walaupun terdapat percabangan, tetapi tiap-tiap tampilan mempunyai kedudukan yang sama tidak ada Master Page dan Slave Page.
4. Composite (campuran) atau disebut juga struktur penjejakan bebas merupakan gabungan dari ketiga struktur sebelumnya yaitu Linear, Non Linear dan Hierarchi. Jika suatu tampilan membutuhkan percabangan, maka dapat dibuat percabangan dan bila dalam percabangan tersebut terdapat suatu tampilan yang sama kedudukannya maka dapat dibuat struktur Linear dalam percabangan tersebut. Setiap struktur peta penjejakan seperti yang baru di bahas mempunyai fungsi dan tujuan tersendiri, tidak ada yang lebih baik atau lebih buruk. Penggunaan peta penjejakan bergantung kepada kebutuhan dan tujuan dari web yang hendak dibuat. Semakin kompleks peta penjejakan yang digunakan, maka semakin sulit pula pembuatan page dari peta penjejakan tersebut.

2.14 Flowchart

Adalah Bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. Flowchart merupakan cara penyajian dari suatu algoritma. (Dwiky Andika, 2017).

Tujuan Membuat Flowchat :

1. Menggambarkan suatu tahapan penyelesaian masalah
2. Secara sederhana, terurai, rapi dan jelas
3. Menggunakan simbol-simbol standar

Simbol Flowchart Simbol-simbol yang di pakai dalam flowchart dibagi menjadi 3 kelompok :

1. Flow direction symbols

Digunakan untuk menghubungkan simbol satu dengan yang lain – Di-sebut juga connecting line

2. Processing symbols

Menunjukan jenis operasi pengolahan dalam suatu proses / prosedur Input / Output

3. symbols

Menunjukkan jenis peralatan yang digunakan sebagai media input atau output. (Dwiky Andika, 2017)

2.15 Kuesioner

Kuesioner atau angket adalah teknik pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pernyataan tertulis kepada responden untuk dijawabnya. Angket merupakan teknik pengumpulan data yang efisien jika peneliti tahu dengan pasti variabel yang akan diukur dan tahu apa yang tidak bisa diharapkan dari responden. Angket sebagai teknik pengumpulan data sangat cocok untuk mengumpulkan data dalam jumlah besar.

Uma Sekaran (dalam Sugiyono), mengemukakan beberapa prinsip dalam penulisan angket sebagai teknik pengumpulan data, yaitu prinsip penulisan, pengukuran, dan penampilan fisik.

Prinsip Penulisan Angket. Prinsip ini menyangkut beberapa faktor yaitu isi dan tujuan pertanyaan, bahasa yang digunakan mudah, pertanyaan tertutup terbuka-negatif positif, pertanyaan tidak mendua, tidak menanyakan hal-hal yang sudah lupa, pertanyaan tidak mengarahkan, panjang pertanyaan, dan urutan pertanyaan.

Prinsip Pengukuran. Prinsip pengukuran dalam hal ini adalah Instrumen angket yang digunakan untuk mengumpulkan data penelitian dapat memperoleh data yang valid dan reliabel. Supaya data yang diperoleh benar-benar valid dan reliabel maka perlu diuji dengan uji validitas dan uji reliabilitas terlebih dahulu sebelum diberikan kepada responden.

Penampilan Fisik Angket. Penampilan fisik angket sebagai alat pengumpul data akan mempengaruhi respon atau keseriusan responden dalam mengisi angket. (Sugiyono, 2011)

2.16 Pengujian Perangkat Lunak

merupakan suatu proses investigasi yang dilakukan untuk mendapatkan informasi mengenai kualitas dari suatu produk atau layanan yang sedang diuji, atau lebih spesifiknya software testing adalah proses mengeksekusi suatu program untuk menemukan bug (kesalahan atau cacat lainnya) dari perangkat lunak.

Pengujian perangkat lunak juga memberikan pandangan mengenai perangkat lunak secara obyektif dan independen, yang bermanfaat untuk memahami tingkat resiko pada implementasinya.

Hal ini juga dapat dinyatakan sebagai proses validasi dan verifikasi bahwa sebuah program:

- Memenuhi persyaratan dan kebutuhan teknis yang mendasari perancangan dan pengembangan perangkat lunak tersebut.
- Bekerja seperti yang diharapkan.
- Dapat diterapkan menggunakan karakteristik yang sama.

Ada beberapa pendekatan metode testing

1. Pengujian Perangkat Lunak Metode White Box

White box testing adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam

beberapa kasus pengujian. Secara sekilas dapat diambil kesimpulan white box testing merupakan petunjuk untuk mendapatkan program yang benar secara 100%.

2. Pengujian Perangkat Lunak Metode White-Box

Black box testing adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dianalogikan seperti kita melihat suatu kotak hitam, kita hanya bisa melihat penampilan luarnya saja, tanpa tau ada apa di balik bungkus hitamnya. Sama seperti pengujian black box, mengevaluasi hanya dari tampilan luarnya (interface nya), fungsionalitasnya. tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detilnya (hanya mengetahui input dan output). (Nanda Samsu, 2017).

2.17 SDLC

SDLC (System Development Life Cycle) merupakan siklus yang menggambarkan perangkat lunak yang dibangun (O'Brien, 2009). SDLC juga merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak, yang terdiri dari tahap-tahap:

1. Perencanaan (planning)
2. Analisis (analysis)
3. Desain (design)
4. Implementasi / Koding (implementation)
5. Uji coba (testing)
6. Pengelolaan (maintenance).

Setiap kegiatan dalam SDLC dapat dijelaskan melalui tujuan (purpose) dan hasil kegiatannya (deliverable). SDLC didefinisikan oleh Departemen Kehakiman AS sebagai sebuah proses pengembangan software yang digunakan oleh analyst system, untuk mengembangkan sebuah sistem informasi. SDLC mencakup kebutuhan (requirement), validasi, pelatihan, kepemilikan (user ownership) sebuah sistem informasi yang diperoleh melalui investigasi, analisis, desain, implementasi, dan perawatan software. (O'Brien, 2009)

2.18 Xampp

XAMPP ialah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan campuran dari beberapa program. Yang mempunyai fungsi sebagai server yang berdiri sendiri (localhost), yang terdiri dari program MySQL database, Apache HTTP Server, dan penerjemah ditulis dalam bahasa pemrograman PHP dan Perl.

Nama XAMPP merupakan singkatan dari X (empat sistem operasi), Apache, MySQL, PHP dan Perl. Program ini tersedia di bawah GNU General Public License dan bebas, adalah mudah untuk menggunakan web server yang dapat melayani tampilan halaman web yang dinamis. Jika ingin mendapatkan xampp dapat mendownload langsung dari situs resminya.

Memahami XAMPP :

1. htdoc adalah folder di mana Anda meletakkan file yang akan dijalankan, seperti file PHP, HTML dan script lainnya.
2. phpMyAdmin adalah bagian untuk mengelola database MySQL yang dikomputer. Untuk membukanya, membuka browser dan ketik alamat `http: // localhost / phpMyAdmin`, halaman phpMyAdmin akan muncul.
3. Control Panel yang berfungsi untuk mengelola layanan (service) XAMPP. Seperti stop service (berhenti), atau mulai (mulai).

Componen yang ada dalam Xampp V3.2.2 :

1. Apache 2.4.4 MySQL 6.5.11
2. PHP 7.0.21
3. phpMyAdmin 4.0.4
4. FileZilla FTP Server 0.9.41
5. Tomcat 7.0.41 (with mod_proxy_ajp as connector) Strawberry
6. Perl 5.16.3.1 Portabel
7. XAMPP Control Panel 3.2.1 (hackattack142, 2013)

.

2.19 Visual Studio Code

Visual Studio Code atau biasa disingkat VSCode adalah editor keluaran Microsoft yang berhasil mencuri perhatian banyak developer. Editor teks yang open source ini sudah bisa langsung dipakai bekerja tanpa perlu melakukan banyak konfigurasi. Namun, jika ingin pekerjaan yang dilakukan lebih mudah lagi, VSCode memiliki plugin/extension yang bisa memberikan kemudahan itu untuk para developer. (Takagi Fujimaru, 2018)

Sublime Text Editor pertama kali dirilis pada tanggal 18 januari 2008, dan sekarang versi Sublime Text Editor sudah mencapai versi 3 yang dirilis pada tanggal 29 januari 2013. Sublime Text mendukung operation system seperti Linux, Mac Os X, dan juga windows. Sangat Banyak fitur yang tersedia pada Sublime Text Editor diantaranya minimap, membuka script secara side by side, bracket highlight sehingga tidak bingung mencari pasangannya, kode snippets, drag and drop direktori ke sidebar terasa mirip dengan TextMate untuk Mac OS.

VSCode juga memiliki fitur IntelliSense yang cukup canggih. Fitur ini akan sangat terasa apabila kita mengembangkan aplikasi web yang menggunakan HTML, CSS, JavaScript, JSON, Less, atau SASS. Fitur IntelliSense merupakan salah satu fitur yang membantu kita dalam menulis kode program melalui popup yang muncul secara otomatis pada saat kita sedang mengetik dan menampilkan saran sintaks yang bisa kita pilih. Fitur lain yang tak kalah keren adalah telah integrasinya version control git dalam VSCode. Dengan adanya fitur ini kita dapat dengan mudah melakukan commit, melihat perubahan di repositori master maupun branch, bahkan VSCode akan memberikan penanda jika terjadi konflik pada repositori git kita. Meskipun masih dalam tahap preview namun VSCode sudah memiliki segudang fitur menarik yang menanti untuk dicoba. Jika pembaca menjadi editor baru untuk menemani mengembangkan aplikasi Node.js mungkin VSCode bisa menjadi partner setia. (Lord Voldemort, 2015)

2.20 Google Chrome

Google Chrome adalah sebuah aplikasi peramban yang digunakan untuk menjelajah dunia maya seperti halnya Firefox, Opera ataupun Microsoft Edge. Jika Firefox dikembangkan oleh Mozilla, Google Chrome dibuat dan dirancang oleh Google, perusahaan internet terbesar di dunia yang juga em-

punya Android.

Proyek open source yang digunakan oleh Google disebut Chromium, menggunakan mesin rendering Webkit sampai dengan versi 27 dan dirancang untuk bekerja dengan kecepatan di atas rata-rata namun tetap ringan dijalankan di perangkat desktop dan mobile.

Versi beta pertama Google Chrome lahir pada 2 September 2008, empat setelah berita pertama soal pengembangan aplikasi browser oleh Google merebak. Di versi awal ini Google masih menggunakan mesin rendering Webkit dan baru tersedia untuk perangkat Windows XP. Beberapa bulan mengembara di fase beta, pada 11 Desember 2008 Google resmi merilis Chrome ke publik.

Tak butuh waktu lama bagi Chrome untuk menarik perhatian publik, dan di awal-awal peluncurannya, ia mengklaim 1% pangsa pengguna peramban global, namun jatuh di angka 0,69% pada bulan Oktober 2008. Tapi setelah versi stabilnya dirilis ke publik, tepat di bulan Desember 2008 persentase pengguna Chrome kembali melampaui angka 1%.

Sukses di Windows, Google langsung mengembangkan versi OSX yang dimulai pada awal tahun 2009 dan versi preview pengembangnya dirilis pada 4 Juni di tahun yang sama. Baru di akhir tahun 2009 Google membawa Chrome versi OS X keluar dari fase preview ke versi beta, termasuk untuk versi Linux. Setahun kemudian, pada 25 Mei 2010 Google akhirnya merilis versi stabil yang mendukung semua platform desktop. (Bambang Winarso, 2016

2.21 Penelitian Terdahulu

Tabel 2.1: Tabel Penelitian Terdahulu

No	Artikel	Hasil
1.	Intan Melia, Wina Witanti dan Faiza Renaldi. 2016. Perancangan Sistem Rekomendasi Pendistribusian Kaos Pada Industri Pakai an Jadi Menggunakan Item Based Collaborative Filtering . Bandung: Selisik. ISSN: 2503-2844.	Membuat rancangan sistem rekomendasi menggunakan Metode <i>Item Based Collaborative Filtering</i> dalam memberikan rekomendasi cabang berdasarkan jumlah penjualan dan diklasifikasikan menurut kategori produknya masing-masing serta dapat membantu perusahaan untuk melakukan pendistribusian.
2.	Assaf Arief. 2016. Rancang Bangun Sistem Rekomendasi Pariwisata Mobile Advertising Menggunakan Metode Hybrid Filtering Sebagai Pemberdayaan Masyarakat Usaha Kecil Menengah (UKM) di Pulau Ternate . Ternate : Jurnal PROtek. Vol 03 No. 1.	Membuat aplikasi mobile yang dapat memberikan rekomendasi dengan menggunakan metode <i>Hybrid Filtering</i> untuk pemberdayaan UKM di pulau Ternate
3.	Septian Eka Anggraeni, Irfan Darmawan dan Nia Ambarsari. 2017. Membangun Sistem Rekomendasi Dalam Situs Lest-Travels Berbasis Collaborative Filtering Menggunakan Metode Waterfall . Bandung. ISSN : 2355-9365.	Membuat sistem rekomendasi Lets-travels untuk memudahkan wisatawan mendapatkan rekomendasi tempat mendapatkan rekomendasi baik rekomendasi tempat wisata yang sesuai dengan kategori minat yang disukai oleh wisatawan, rekomendasi tiket bus, tiket kereta dan tiket pesawat serta mendapatkan rekomendasi hotel yang berada di lokasi yang sama dengan tempat wisata yang akan dikunjungi oleh wisatawan. Metode yang digunakan Collaborative Filtering Menggunakan dan Metode Waterfal

No	Artikel	Hasil
4.	Dedy Hidayat Kusuma dan Moh. Nur Shodiq. 2017. Sistem Rekomendasi Destinasi Pariwisata Menggunakan Metode Hibrid Case Based Reasoning dan Location Based Service Sebagai Pemandu Wisatawan di Banyuwangi . Banyuwangi : Jurnal INTENSIF. Vol.1, No.1 ISSN: 2549-6824	Menghasilkan sebuah program aplikasi mobile berbasis android yang mampu memberikan rekomendasi destinasi pariwisata dengan mempertimbangkan kriteria preferensi yang dipilih pengguna serta lokasi pengguna yang diperoleh dari koordinat aktif GPS perangkat yang digunakan
5.	Norman B. Anderson, et al. 2013. Stress in America Missing the Health Care Connection . American Psychological Association.	Penelitian tersebut membahas tentang stress yang terjadi di amerika. Pembahasan dalam penelitian yang dilakukan dibagi kedalam beberapa kriteria, seperti <i>stress</i> berdasarkan <i>gender</i> , <i>stress</i> berdasarkan generasi, <i>stress</i> berdasarkan wilayah, <i>stress</i> berdasarkan kota dan dampak dari <i>stress</i> .
6.	Haiyang Zhang, et al. 2017 A trust-enriched approach for item-based collaborative filtering recommendations . IEEE	Penelitian tersebut membahas penggunaan hubungan kepercayaan antara pengguna untuk meningkatkan akurasi rekomendasi dari pemindaian kolaboratif berbasis item (CF). Hasil penelitian tersebut menegaskan bahwa metode yang diusulkan membantu meningkatkan hasil prediksi.

Pada bagian ini peneliti mengawali dengan menelaah penelitian terdahulu yang memiliki keterkaitan serta relevansi dengan penelitian yang dilakukan. Dengan demikian, peneliti mendapatkan rujukan pendukung, pelengkap serta pembanding yang memadai sehingga penulisan penelitian ini lebih memadai. Hal ini dimaksudkan untuk memperkuat penelitian yang sedang dilakukan dengan didukung penelitian yang ada. Penelitian yang terdahulu yang dijadikan sumber memiliki beberapa perbedaan cara pandang mengenai objek-objek tertentu, sehingga meskipun terdapat kesamaan maupun perbedaan adalah suatu hal yang wajar dan dapat disinergikan untuk saling melengkapi. Berikut dibawah ini hasil dari penelitian terdahulu.

Berdasarkan penelitian yang dilakukan oleh Intan Melia, Wina Witanti

dan Faiza Renaldi pada tahun 2016 dengan judul “Perancangan Sistem Rekomendasi Pendistribusian Kaos Pada Industri Pakaian Jadi Menggunakan Item Based Collaborative Filtering” membahas bagaimana merancang sistem yang dapat dijadikan bahan implementasi sistem guna rekomendasi pendistribusian mengenai pemilihan cabang yang memiliki tingkat pendistribusian tinggi. Penelitian tersebut menyatakan ketika merating suatu item, maka nilai rating tersebut akan dibandingkan dengan nilai rating dari pengguna lainnya, atau item lainnya yang mempunyai kemiripan rating dengan item tersebut. Dari situlah kemudian sistem akan membuat suatu rekomendasi berdasarkan item-item yang telah diberi rating.

Berdasarkan penelitian yang dilakukan oleh Assaf Arief pada tahun 2016 dengan judul “Membuat aplikasi mobile yang dapat memberikan rekomendasi dengan menggunakan metode Hybrid Filtering untuk pemberdayaan UKM di pulau Ternate” membahas bagaimana membangun aplikasi mobile yang dapat memberikan rekomendasi dengan menggunakan metode *Hybrid Filtering* untuk pemberdayaan UKM di pulau Ternate. Dalam penelitian tersebut dikatakan *trend* penggunaan perangkat mobile oleh wisatawan global menjadi perhatian dalam mempromosikan objek-objek wisata daerah di pulau Ternate. Perkembangan jumlah informasi yang begitu banyak di dunia maya menyebabkan sulitnya menemukan informasi tepat yang sesuai dengan selera/preferensi dan minat calon wisatawan.

Penelitian lainnya yang dilakukan oleh Septian Eka Anggraeni, Irfan Darmawan dan Nia Ambarsari pada tahun 2107. yang berjudul “Membangun Sistem Rekomendasi Dalam Situs Lest-Travels Berbasis Collaborative Filtering Menggunakan Metode Waterfall”. Penelitian tersebut menghasilkan sistem rekomendasi Lets-travels yang dibuat untuk memudahkan wisatawan mendapatkan rekomendasi tempat mendapatkan rekomendasi baik rekomendasi tempat wisata yang sesuai dengan kategori minat yang disukai oleh wisatawan, rekomendasi tiket bus, tiket kereta dan tiket pesawat serta mendapatkan rekomendasi hotel yang berada di lokasi yang sama dengan tempat wisata yang akan dikunjungi oleh wisatawan. Metode yang digunakan Collaborative Filtering Menggunakan dan Metode Waterfall. Penelitian tersebut menyatakan bidang wisata saat ini semakin berkembang bahkan wisata telah menjadi salah satu kebutuhan manusia. Hal tersebut menjadikan sektor pariwisata mengalami pertumbuhan dengan cepat sehingga harus didukung dengan layanan dan fasilitas yang layak untuk wisatawan terutama dalam pemberian informasi.

Berdasarkan penelitian yang dilakukan oleh Dedy Hidayat Kusuma dan Moh. Nur Shodiq pada tahun 2017 yang berjudul “Sistem Rekomendasi Destinasi Pariwisata Menggunakan Metode Hibrid Case Based Reasoning dan Location Based Service Sebagai Pemandu Wisatawan di Banyuwangi”. Penelitian tersebut bertujuan untuk menghasilkan sebuah program aplikasi mobile berbasis android yang mampu memberikan rekomendasi destinasi pariwisata dengan mempertimbangkan kriteria preferensi yang dipilih pengguna serta lokasi pengguna yang diperoleh dari koordinat aktif GPS perangkat yang digunakan. Metode yang digunakan penelitian tersebut adalah metode hibrid case based reasoning dan location based service. Penelitian tersebut menyatakan informasi wisata Banyuwangi yang tersedia dalam bentuk cetakan maupun yang bisa diakses secara online masih mengharuskan wisatawan untuk memilah dan memilih sendiri sesuai dengan minat dan preferensinya sehingga wisatawan memerlukan adanya saran atau rekomendasi.

Berdasarkan penelitian yang dilakukan oleh Norman B. Anderson, et al. pada tahun 2013. yang berjudul “Stress in America Missing the Health Care Connection” membahas tentang stress yang terjadi di amerika. Penelitian tersebut menyatakan tujuan utama dari traveling adalah untuk keluar dari lingkungan dan rutinitas sehari-hari manusia yang merupakan sumber utama stress yang sering dirasakan. Stress dan jenuh dengan kegiatan sehari-hari, traveling bisa jadi salah satu obat yang mujarab.

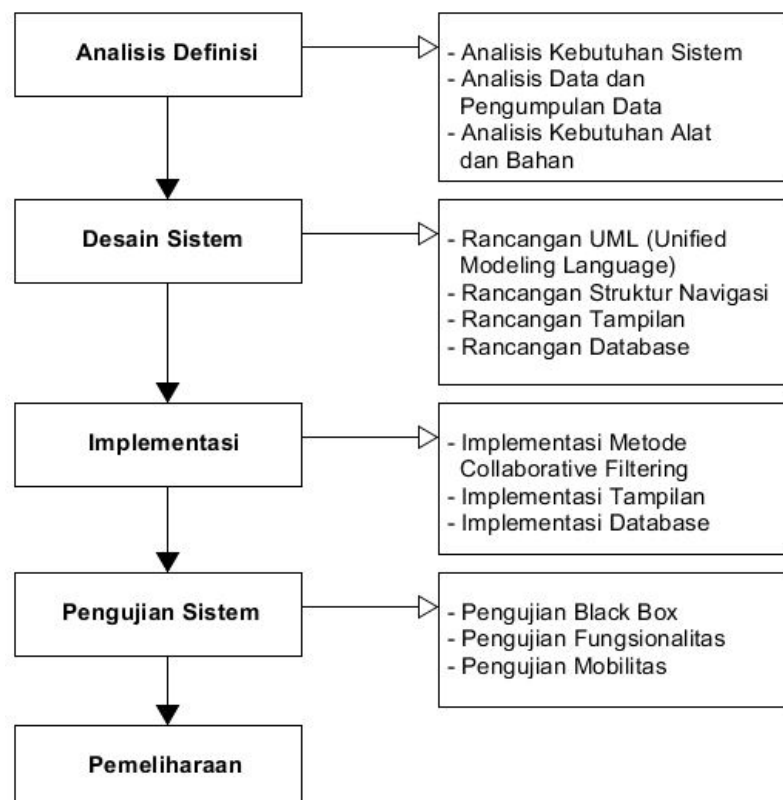
Berdasarkan Penelitian Haiyang Zhang, et al. pada tahun 2017 yang berjudul “A trust-enriched approach for item-based collaborative filtering recommendations” membahas tentang *item-based collaborative filtering* Penelitian tersebut menyatakan *item-based collaborative filtering* adalah salah satu pendekatan yang paling berhasil yang digunakan oleh sistem rekomendasi. Konsep dasar di baliknya adalah merekomendasikan item tersebut kepada pengguna yang mirip dengan item lain yang diminati pengguna. Hasil penelitian tersebut menunjukkan bahwa metode yang digunakan menunjukkan bahwa pendekatan yang diusulkan mencapai akurasi yang lebih baik. Evaluasi kinerja metode yang diusulkan dilakukan dengan membandingkan hasil dengan *collaborative filtering* berbasis item.

Bab 3

Metodologi dan Perancangan

3.1 Metodologi Penelitian

Metodologi yang digunakan pada penelitian ini menggunakan metode *Software Development Live Cycle (SDLC)* model *waterfall*. Alasan menggunakan metode ini karena model *waterfall* melakukan pendekatan secara sistematis dan berurutan dalam membangun suatu sistem. Berikut tahap-tahap yang digunakan menjadi acuan atau prosedur penelitian yang penulis lakukan.



Gambar 3.1: Metodologi Penelitian

Sistem yang dihasilkan akan berkualitas baik, dikarenakan pelaksanaannya secara bertahap sehingga tidak terfokus pada tahapan tertentu. Tahapan yang dilakukan dalam model *waterfall* meliputi analisis definisi, desain sistem, implementasi, pengujian sampai tahap pemeliharaan, berikut dibawah ini penulis uraikan :

1. Analisis Definisi

Terdapat 3 tahap utama dalam analisis definisi, yaitu menganalisa kebutuhan sistem, menganalisa kebutuhan data, dan tahap terakhir menganalisa kebutuhan alat dan bahan.

Pada tahap menganalisa kebutuhan sistem dilakukan analisa perhitungan terhadap metode rekomendasi yang digunakan, yaitu *item based collaborative filtering* dengan menggunakan metode perhitungan algoritma pembangkit rekomendasi agar memberikan rekomendasi informasi yang sesuai untuk *traveller*. Pada tahap ini juga dilakukan analisa untuk menentukan fitur atau menu apa saja yang akan dibuat pada sistem ini. Tahap selanjutnya yaitu menganalisa kebutuhan data. Pada tahap menganalisa kebutuhan data dilakukan dengan studi lapangan seperti observasi dan wawancara, selain itu juga dilakukan studi pustaka. Tahap terakhir dalam menganalisa kebutuhan sistem yaitu tahap menganalisa kebutuhan alat dan bahan. Didalam analisa kebutuhan alat dilakukan analisa terhadap kebutuhan perangkat keras dan perangkat lunak yang digunakan oleh pengembang dan *traveller*.

2. Desain Sistem

Desain sistem atau perancangan sistem penelitian terdiri dari 4 bagian, yaitu merancang UML (*unified modeling language*) , merancang struktur navigasi, merancang tampilan sampai pada tahap merancang *database*.

Pada tahap merancang UML (*unified modeling language*) dilakukan perancangan *use case diagram* dan *class diagram*, pada tahap merancang struktur navigasi, struktur navigasi yang digunakan adalah struktur navigasi *composite* (campuran). Tahap selanjutnya dalam desain sistem yaitu merancang tampilan. Tahap merancang tampilan dilakukan perancangan tampilan halaman beranda atau halaman utama, halaman login, halaman tampilan website dan halaman tampilan mobile. Tahap terakhir yaitu merancang *database*, pada tahap ini *database* yang

digunakan adalah database MySQL dengan nama database yaitu “skripsi”. terdapat 12 tabel yang digunakan dalam perancangan tabel, yaitu tabel users, galleries, posts, post_likes, post_comments, barengan, barengan_comments, barengan_gabung, infos, info_review, singgahs dan tabel singgah_review

3. Implementasi

Implementasi dilakukan untuk membangun sistem yang dibuat berdasarkan hasil dari tahap analisis dan desain sistem. Terdapat 3 tahap utama pada implementasi, yaitu implementasi sistem rekomendasi, implementasi tampilan dan implementasi *datatabase*

Implementasi sistem rekomendasi dilakukan untuk mengimplementasi proses perhitungan dengan menggunakan metode perhitungan algoritma pembangkit rekomendasi pada *code* sistem yang dibuat. Tahap selanjutnya yaitu implementasi tampilan. Pada tahap implementasi tampilan dilakukan untuk menjelaskan tampilan masing - masing halaman dan fitur menu yang tersedia pada sistem. Tahap terakhir yaitu implementasi *database*. Implementasi *database* dilakukan dengan menggunakan fitur pembuatan database dari framework laravel yaitu fitur *datatabase migration* yang mengubah *code* menjadi tabel *database*.

4. Pengujian Sistem

Pengujian sistem pada penelitian ini terdapat 3 tahap pengujian, yaitu pengujian sistem menggunakan metode *black box*, pengujian fungsionalitas dan fungsi mobilitas.

Pengujian dengan metode *black box* terdiri dari rencana pengujian, kasus dan hasil pengujian. Pada tahap pengujian fungsionalitas dilakukan dengan melibatkan 20 responden *traveller* untuk mengisi kuesioner. Tahap pengujian fungsi mobilitas dilakukan pengujian sistem pada perangkat komputor dan perangkat mobile untuk menguji apakah sistem dapat berjalan pada perangkat yang diuji dan menguji apakah tampilan sistem yang dibuat telah responsive terhadap layar dari perangkat yang diuji.

5. Pemeliharaan

Tahap pemeliharaan dilakukan setelah sistem dirilis. Dalam penelitian ini, penulis membatasi pada tahap ini karena sistem yang dibuat belum dirilis.

3.2 Analisis Kebutuhan Sistem

3.2.1 Analisis Perhitungan *Item Based Collaborative Filtering*

Tahap ini dilakukan untuk mengetahui bagaimana setiap langkah proses perhitungan *item based collaborative filtering* dengan menggunakan formula *algoritma* pembangkit rekomendasi, tujuannya untuk memberikan rekomendasi yang sesuai, khususnya rekomendasi info wisata dan rumah singgah. Alasan menggunakan metode *item based collaborative filtering* karena kecocokan antara sistem yang dibuat dan metode yang akan digunakan terhadap perhitungan yang digunakan berdasarkan rating yang diberikan pengguna terhadap konten yang tersedia. Formula perhitungan yang digunakan dalam metode ini adalah

$$R = \left(\frac{\sum B \times \text{rating} \times 100}{\text{Max} \times \text{MaxB}} \right) / C$$

Gambar 3.2: Formula Algoritma Pembangkit Rekomendasi.

R = Rekomendasi

B = Nilai ulasan konten (1 jika memberi ulasan & 0 jika tidak memberi ulasan).

Rating = Jumlah rating yang diberikan oleh traveller terhadap konten

100 = Nilai maksimum 100, nilai rekomendasi tertinggi adalah 100.

Max = Nilai maksimum rating adalah 5

MaxB = Nilai maksimum ulasan yang diberikan traveller

C = Jumlah *traveller* yang mempunyai pola rekomendasi yang sama dengan *traveller* yang *login* pada sistem.

Berikut dibawah ini diagram alir untuk menentukan rekomendasi



Gambar 3.3: Flowchart Perhitungan Rekomendasi.

Proses pertama yang dilakukan adalah memasukan data yang akan dihitung dengan *sample* 5 data konten, 5 data *traveller* dan 10 data rating. Tabel berikut ini menunjukkan jumlah rating yang diberikan oleh *traveller* untuk masing - masing konten

Tabel 3.1: Tabel Sample Rating

Content	T1	T2	T3	T4	T5
A	0	0	0	1 ⁴	1 ⁴
B	0	0	1 ⁵	0	1 ⁵
C	1 ⁵	0	1 ⁴	0	0
D	1 ⁵	1 ⁵	0	0	0
E	1 ⁴	0	1 ²	0	0

A - E = Konten atau data yang ada pada sistem

T1 - T5 = Traveller yang telah terdaftar pada sistem

Angka 1 dan 0 = Jika traveller telah memberi rating pada konten maka ditandai dengan angka 1 dan jika traveller tidak pernah memberi ulasan ditandai dengan angka 0

Angka superskrip (1-5) = rating yang diberikan traveller terhadap konten

Setelah dilakukan pemberian rating terhadap konten, tahap selanjutnya yaitu menghitung prediksi rekomendasi dengan menggunakan algoritma pembangkit rekomendasi. Pada perhitungan dibawah ini data *traveller* yang digunakan sebagai contoh perhitungan ialah traveller dengan id 2. Jika *traveller* login kedalam sistem maka perhitungan pembangkit rekomendasi adalah sebagai berikut :

$$(\text{Konten A}) = ((1*4*100/5*1) + ((1*4*100/5*1)) / 4 = 40$$

$$(\text{Konten B}) = ((1*5*100/5*1) + ((1*5*100/5*1)) / 4 = 50$$

$$(\text{Konten C}) = ((1*5*100/5*1) + ((1*4*100/5*1)) / 4 = 45$$

$$(\text{Konten D}) = ((1*5*100/5*1) + ((1*5*100/5*0)) / 4 = 25$$

$$(\text{Konten E}) = ((1*4*100/5*1) + ((1*2*100/5*1)) / 4 = 30$$

Tahap berikutnya adalah menampilkan hasil perhitungan dalam bentuk tabel. Sebelumnya jika dilihat dari data tabel *sample rating*, *traveller* 2 sudah memberi rating pada konten D, sehingga sistem akan merekomendasikan konten kepada *traveller* 2 seperti di bawah ini

Tabel 3.2: Tabel Hasil Perhitungan Prediksi

Konten	Nilai Prediksi
Konten B	50
Konten C	45
Konten A	40
Konten E	30
Konten D	25

Dapat dilihat bahwa nilai prediksi rekomendasi tertinggi jika *traveller* dengan id 2 masuk ke dalam sistem adalah konten B dengan nilai 50 dan nilai terendah adalah konten E yaitu 30.

3.2.2 Analisis Kebutuhan Fitur dan Menu

Pada tahap ini juga dilakukan analisis untuk menentukan fitur atau menu apa saja yang akan dibuat pada sistem ini. Analisis dilakukan dengan mempertimbangkan kebutuhan rekomendasi informasi *traveller* yaitu rekomendasi tempat wisata, rekomendasi rumah singgah dan kebutuhan pendukung informasi lainnya seperti berdiskusi dan mencari teman barengan untuk melakukan perjalanan. berikut dibawah ini hasil analisa menu yang akan tersedia didalam sistem yang akan dibuat.

1. Menu “Home” digunakan untuk *traveller* berdiskusi.

2. Menu “Barengan” digunakan *traveller* untuk mencari barengan dalam berwisata.
3. Menu “Singgah” digunakan *traveller* untuk mendapatkan rekomendasi dan mencari rumah singgah gratis saat dalam perjalanan.
4. Menu “Info” digunakan *traveller* untuk mendapatkan rekomendasi dan mencari informasi tempat wisata yang menarik.

3.3 Analisis Data dan Pengumpulan Data

3.3.1 Studi Lapangan

1. Observasi

Observasi yang dilakukan penulis yaitu dengan mengamati situs - situs yang berhubungan dengan kegiatan wisata atau *traveller* seperti situs CouchSurfing.com dan situs yang penggunanya dapat memberi ulasan review atau rating seperti situs zomato.com dan tokopedia.com. Hasil observasi yang didapat gambaran interaksi situs - situs tersebut dengan *traveller* kemudian dijadikan sebagai referensi dalam pembuatan sistem.

2. Wawancara

Proses wawancara dilakukan kepada para *traveller* secara langsung di-tempat biasanya terdapat *traveller* seperti terminal bus, stasiun kereta dan tempat wisata. Hasil yang didapatkan adalah data - data yang diperlukan untuk merancang struktur database dan alur sistem yang akan dibuat.

3.3.2 Studi Pustaka

Penulis melakukan studi pustaka sebagai bahan tambahan untuk melengkapi kekurangan-kekurangan data yang diperoleh dari *observasi* dan wawancara. Pengumpulan data dengan studi pustaka mengambil sumber-sumber data melalui studi pustaka buku, jurnal dan *website*. Referensi dalam penyusunan skripsi yang digunakan penulis dapat dilihat pada daftar pustaka.

3.4 Analisis Kebutuhan Alat dan Bahan

3.4.1 Alat

Berikut dibawah ini hasil analisis dari kebutuhan alat yang digunakan berupa perangkat keras dan perangkat lunak.

1. Perangkat Keras Untuk Pengembang

Perangkat keras untuk pengembang adalah laptop yang digunakan selama proses pembuatan sistem. Laptop yang digunakan mempunyai sistem operasi Windows 10 pro dengan processor Intel Core i7- 5600, kapasitas Ram 8 GB dan Ruang penyimpanan atau hardisk sebesar 1TB.

2. Perangkat Keras Untuk Pengguna

Perangkat keras untuk pengguna adalah spesifikasi minimal perangkat komputer atau laptop dan perangkat *mobile* yang dapat digunakan oleh *traveller* untuk menjalankan sistem ini. Sistem ini dapat berjalan pada laptop atau komputer dengan spesifikasi minimal mempunyai sistem operasi Windows XP, processor intel pentium 4 dengan ram 1 GB. Pada perangkat mobile spesifikasi minimal yang dibutuhkan adalah Ram 512MB, processor ARMv7 dan sistem operasi Android Jelly Bean.

3. Perangkat Lunak Untuk Pengembang

Perangkat lunak untuk pengembang adalah aplikasi *software* yang digunakan selama proses pembuatan sistem. Perangkat lunak yang digunakan oleh pengembang adalah Sublime Text Versi 3 yang digunakan untuk proses pengkodean atau *coding*, Xampp Versi 3.2.2 digunakan sebagai *server* database lokal selama proses pembuatan sistem dan Browser Google Chrome digunakan untuk menjalankan sistem yang dibuat.

4. Perangkat Lunak Untuk Pengguna

Perangkat lunak untuk pengguna adalah perangkat lunak yang digunakan untuk menjalankan sistem ini. Perangkat lunak yang dibutuhkan untuk menjalankan sistem ini adalah aplikasi browser seperti Google Chrome dan Mozilla Firefox.

3.4.2 Bahan

Dalam penelitian ini bahan yang digunakan adalah :

1. Data yang diperoleh melalui studi literatur penelitian yang masih memiliki keterkaitan dengan penulisan yang dilakukan.
2. Data yang diperoleh melalui aplikasi atau situs serupa yang sebelumnya sudah ada seperti situs couchsurfing.com (situs dan layanan jejaring sosial wisatawan), situs zomato.com (situs pencarian restoran) dan situs tokopedia.com yang dijadikan referensi dalam pembuatan sistem.

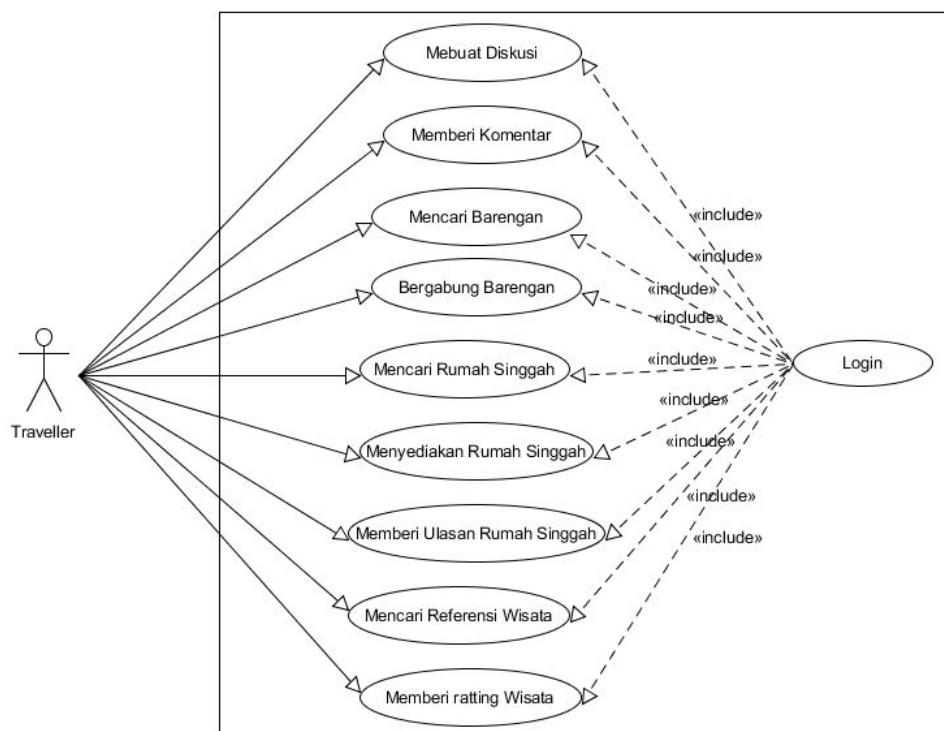
3.5 Perancangan

3.5.1 Rancangan UML (Unified Modeling Language)

Metode UML yang digunakan dalam perancangan sistem ini antara lain *use case diagram* dan *class diagram*. Perancangan UML tersebut dijelaskan sebagai berikut:

1. Use Case Diagram

Rancangan *use case diagram* pada sistem ini terdiri dari fungsi login, fungsi membuat diskusi, fungsi memberi komentar, fungsi mencari barengan, fungsi bergabung barengan, fungsi mencari rumah singgah, fungsi menyediakan rumah singgah, memberi ulasan rumah singgah, fungsi mencari referensi wisata dan fungsi memberi rating wisata.

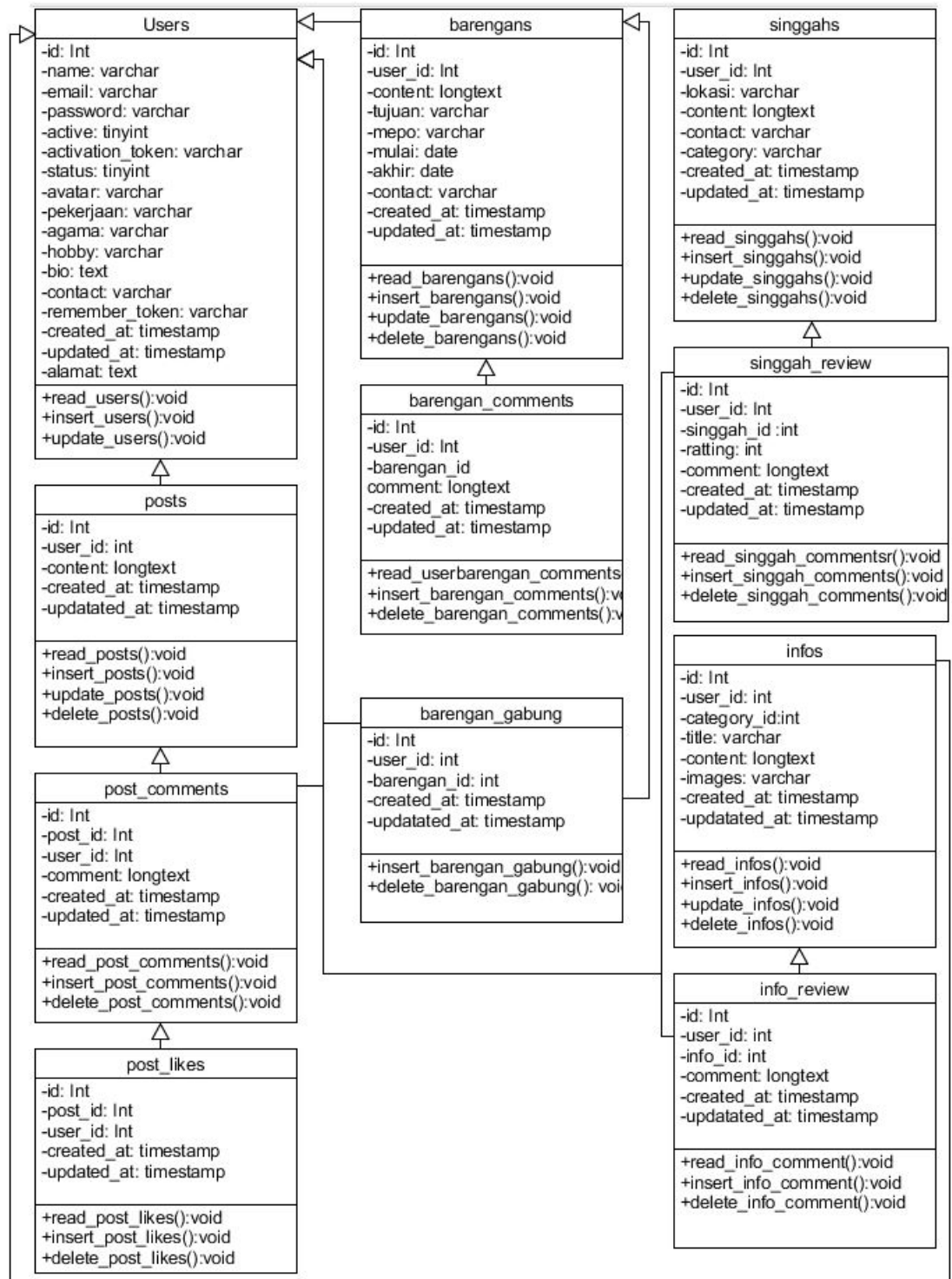


Gambar 3.4: Use Case Diagram E-Travel

Use case diagram diatas menjalankan semua fungsi - fungsi yang harus dimiliki dalam rancangan sistem. *Traveller* sebagai pengguna sistem ini dapat melakukan semua proses fungsi yang tersedia pada sistem

ini, dengan kata lain *traveller* dapat mencari dan memberi informasi pada sistem ini.

2. Class Diagram



Gambar 3.5: Class Diagram E-Travel

Class diagram yang digunakan dalam perancangan sistem diatas menjelaskan hubungan antara *variable* dan *class*. Pada *class* User, *variable* yang ada digunakan untuk melakukan fungsi melihat *traveller*, menambah *traveller* dan mengubah informasi tentang *traveller*.

Class posts digunakan untuk melakukan fungsi diskusi seperti melihat *post* atau kiriman diskusi, menambah diskusi, mengubah diskusi dan menghapus diskusi. *Class* post_comments digunakan untuk melakukan fungsi komentar pada diskusi seperti melihat komentar, menambah komentar dan menghapus komentar

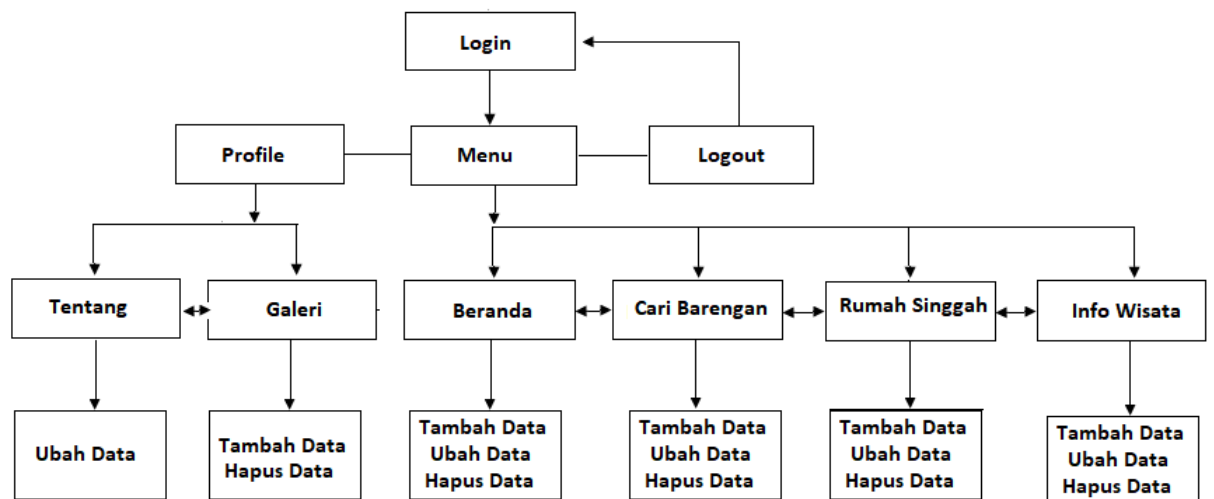
Class barengans digunakan untuk melakukan fungsi pencarian barengan seperti menambah data barengan baru, mengubah data dan menghapus data barengan. *Class* barengan_comments digunakan untuk melakukan fungsi komentar pada data kiriman barengan seperti melihat komentar, menambah komentar dan menghapus komentar. *Class* barengan_gabungs digunakan untuk bergabung pada kiriman barengan.

Class singgahs digunakan untuk melakukan fungsi pencarian rumah singgah seperti menambah data rumah singgah baru, mengubah data dan menghapus data rumah singgah. *Class* singgah_review digunakan untuk melakukan fungsi memberikan ulasan atau rating pada data kiriman singgah seperti melihat, menambah dan menghapus ulasan atau rating

Class infos digunakan untuk melakukan fungsi pencarian info wisata seperti menambah data info wisata baru, mengubah dan menghapus data info wisata. *Class* info_review digunakan untuk melakukan fungsi pemberian ulasan pada data info wisata seperti melihat, menambah dan menghapus ulasan atau rating

3.5.2 Rancangan Struktur Navigasi

Struktur navigasi yang digunakan dalam “Metode Rekomendasi Item Collaborative Filtering Pada Sistem E - Travel Haruni Berbasis Web Mobil Menggunakan Framework Laravel” adalah navigasi composite (campuran). Dengan menggunakan struktur navigasi tersebut dapat memberikan kebebasan dalam berinteraksi dengan halaman lainnya. Adapun perancangan struktur navigasi seperti dibawah ini.



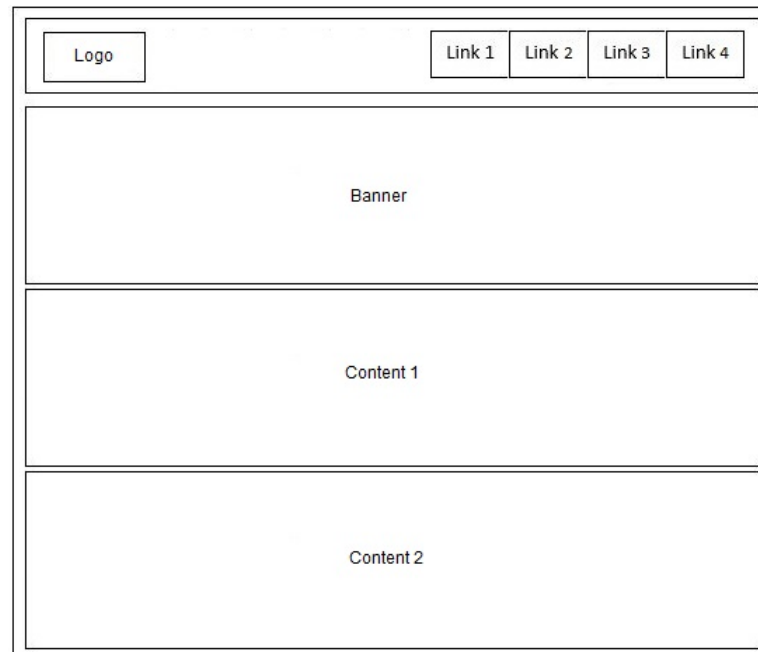
Gambar 3.6: Struktur Navigasi E-Travel

Struktur navigasi diatas menjelaskan alur dari sistem yang dibuat. Jika *traveller* melakukan login pada sistem ini maka akan terdapat 3 navigasi yaitu profile, menu, dan logout. Jika *traveller* memilih navigasi profile maka akan terdapat 2 navigasi yaitu tentang dan galeri. Jika *traveller* memilih navigasi menu akan terdapat pilihan menu yaitu beranda, caribarengan, rumah singgah dan info wisata. Jika *traveller* memilih navigasi logout maka akan keluar dari sistem dan kembali pada navigasi login.

3.5.3 Rancangan Tampilan

Rancangan tampilan pada penulisan ini menjelaskan

1. Rancangan Tampilan Halaman Utama / Beranda



Gambar 3.7: Rancangan Tampilan Halaman Utama / Beranda

Rancangan tampilan utama atau beranda adalah tampilan utama ketika *traveller* mengakses website, berikut dibawah ini keterangan dari isi rancangan tampilan halaman utama atau beranda.

Link 1 digunakan untuk mengakses halaman utama, *Link 2* digunakan untuk mengakses bagian *content 1*, *link 3* merupakan akses untuk bagian *content 2*, *link 4* merupakan akses untuk halaman login.

Banner adalah gambar yang dibuat untuk mempercantik tampilan, *content 1* berisi data tentang apa saja yang tersedia pada *website* ini, *content 2* berisi form pendaftaran akun baru.

1. Rancangan Tampilan Halaman Login

A wireframe diagram of a login page. It consists of a large rectangular container. Inside, there are two stacked rectangular input fields. The top field is labeled 'email' and the bottom field is labeled 'Password'. Below these fields, on the right side, is a rectangular button labeled 'Login'.

Gambar 3.8: Rancangan Tampilan Halaman Login

Halaman login digunakan untuk masuk kedalam sistem dengan cara memasukkan email dan password, email dan password didapatkan traveller saat mendaftarkan akun ke dalam sistem

2. Rancangan Tampilan Website

A wireframe diagram of a website layout. The layout is enclosed in a large rectangular frame. At the top, there is a horizontal header bar containing five links: 'Menu Utama', 'Link 1', 'Link 2', 'Link 3', and 'Link 4', followed by a separate box containing 'Link 5'. On the left side, there is a vertical sidebar area containing a box labeled 'Sidebar' and two smaller boxes below it labeled 'Link 6' and 'Link 7'. The main body of the page is a large rectangular area labeled 'Content'.

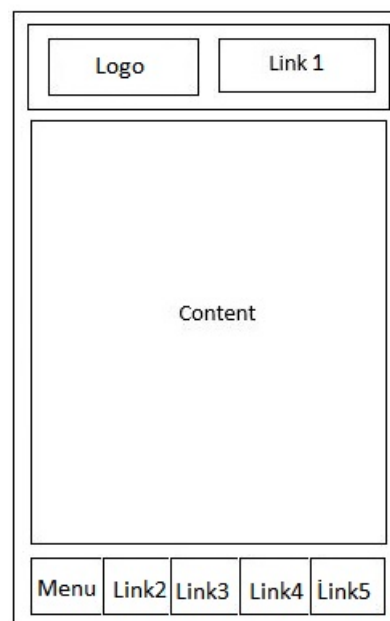
Gambar 3.9: Rancangan Tampilan Website

Rancangan halaman *website* dimaksud adalah *website* pada umumnya yang dibuka di dalam laptop atau komputer. berikut dibawah ini keterangan dari isi rancangan tampilan website.

Link 1 merupakan akses untuk halaman home, *link 2* merupakan akses untuk halaman cari barengan, *link 3* merupakan akses untuk halaman rumah singgah. *Link 4* merupakan akses untuk halaman info wisata. *Link 5* merupakan navigasi menu untuk *logout* dan melihat *profile*, *link 6* merupakan akses untuk kehalaman tentang dan *link 7* merupakan akses untuk kehalaman galeri.

Sidebar pada halaman home berisi data *profile* dari *traveller*, pada halaman cari barengan berisi data jadwal kalender yang telah *traveller* buat. pada halaman singgah berisi data yang *traveller* buat dihalaman rumah singgah, dan pada halamna info wisata berisi data informasi wisata yang *traveller* buat. Content berisikan semua kiriman *traveller* data pada masing - masing halaman.

3. Rancangan Tampilan Mobile



Gambar 3.10: Rancangan Tampilan Mobile

Pada rancangan tampilan mobile dibuat jika *website* di buka dalam perangkat mobile dengan menggunakan aplikasi browser.

Link 1 merupakan navigasi menu untuk *logout* dan melihat *profile*, *link 2* merupakan akses untuk halaman home, *link 3* merupakan akses untuk halaman cari barengan. *Link 4* merupakan akses untuk halaman rumah singgah dan *link 5* merupakan akses untuk halaman info wisata.

Sidebar pada halaman home berisi data profile dari *traveller*, pada halaman cari barengan berisi data jadwal kalender yang telah *traveller* buat. pada halaman singgah berisi data yang *traveller* buat di halaman rumah singgah, dan pada halaman info wisata berisi data informasi wisata yang *traveller* buat. *Content* berisikan semua kiriman *traveller* data pada masing - masing halaman.

3.5.4 Rancangan Database

Database yang digunakan adalah database MySQL dengan nama database “skripsi”. Didalam database tersebut terdapat 12 tabel yaitu tabel users, galleries, posts, post_likes, post_comments, barengan, barengan_comments, barengan_gabung, infos, info_review, singgahs dan tabel singgah_review

Adapun spesifikasi dari masing – masing table seperti dibawah ini :

1. Tabel users

Tabel users digunakan untuk menyimpan data *traveller*. seperti nama, email, password, status, foto, pekerjaan, agama, hobi, bio, kontak dan alamat. Berikut dibawah ini spesifikasi tabel users :

Tabel 3.3: Tabel users

NO	NAMA FIELD	TIPE	PANJANG	KETERANGAN
1	id	Int	4	Primary Key, AI
2	name	Varchar	25	-
3	email	Varchar	30	Foreign Key
4	password	Varchar	100	md5
5	active	tinyint	1	-
6	activation_token	Varchar	100	-
7	avatar	Varchar	100	-
8	pekerjaan	Varchar	25	-
9	agama	Varchar	20	-
10	hobby	Varchar	30	-
11	bio	text	-	-
12	contact	Varchar	12	-
13	alamat	text	-	-

Field id pada tabel ini mempunyai panjang sebesar 4 dengan tipe data integer. *Field* id tersebut maksimal menyimpan data sebanyak 4 digit yaitu 9999 data *traveller* dan dibuat sebagai primary key atau kunci tabel dengan pengisian otomatis yaitu auto increment (AI).

Pada *field* name digunakan untuk menyimpan data nama *traveller* dengan tipe data varchar dan panjang 25. Kolom email digunakan untuk menyimpan data email *traveller* dengan tipe data varchar, panjang 30 dan dibuat foreign key atau kunci kedua karena email bersifat unik.

Field password bertipe data varchar dengan panjang 100 dan dibuat MD5 untuk melindungi password tersebut supaya tidak mudah dibaca.

Pada *field* active digunakan untuk menyimpan data status *traveller* yang sudah melakukan verifikasi email dengan tipe data tinyint dan panjang 1, sedangkan *field* activation_token digunakan untuk menyimpan data aktivasi token email dengan panjang karakter 100 bertipe data varchar. *Field* avatar digunakan untuk menyimpan path atau lokasi dari file foto dengan tipe data varchar dan panjang 100. *Field* agama digunakan untuk menyimpan data agama *traveller* dengan tipe data varchar dan panjang 20. *Field* hobby dibuat untuk menyimpan data hobi *traveller* dengan panjang 30 dan bertipe data varchar. Pada *field* contact digunakan untuk menyimpan contact *traveller* dengan panjang 12 dan bertipe data varchar. *Field* terakhir yaitu *field* alamat, *field* ini digunakan untuk menyimpan data alamat *traveller* dengan tipe data text.

2. Tabel galleries

Pada tabel galleries digunakan untuk menyimpan data galeri atau foto dari *traveller*. Berikut dibawah ini spesifikasi tabel galleries :

Tabel 3.4: Tabel galleries

NO	NAMA FIELD	TIPE	PANJANG	KETERANGAN
1	id	Int	5	Primary Key, AI
2	user_id	Int	4	-
3	images	Varchar	100	-
4	caption	Longtext	-	-

Field id pada tabel ini mempunyai panjang sebesar 5 dengan tipe data integer. *Field* id tersebut maksimal menyimpan data sebanyak 5 digit yaitu 99999 data foto *traveller* dan dibuat sebagai primary key atau kunci tabel dengan pengisian otomatis yaitu auto increment (AI).

Pada *field* user_id digunakan untuk menyimpan data *traveller*, *field* tersebut dijadikan sebagai relasi dengan tabel users. Tipe data yang digunakan adalah integer dengan panjang 4. panjang tipe data tersebut dibuat sesuai dengan *field* relasinya yaitu *field* id pada tabel users.

Pada *field* images digunakan untuk menyimpan path atau lokasi dari file gambar dengan tipe data varchar dan panjang 100. *Field* caption digunakan untuk menyimpan data deskripsi foto *traveller* dengan tipe data longtext.

3. Tabel posts

Tabel posts digunakan untuk menyimpan data diskusi *traveller*. Berikut dibawah ini spesifikasi tabel posts :

Tabel 3.5: Tabel posts

NO	NAMA FIELD	TIPE	PANJANG	KETERANGAN
1	id	Int	5	Primary Key, AI
2	user_id	Int	4	-
3	content	Longtext	-	-

Field id pada tabel ini mempunyai panjang sebesar 5 dengan tipe data integer. *Field* id tersebut maksimal menyimpan data sebanyak 5 digit yaitu 99999 data *post* atau kiriman diskusi *traveller* dan dibuat sebagai primary key atau kunci tabel dengan pengisian otomatis yaitu auto increment (AI).

Pada *field* user_id digunakan untuk menyimpan data *traveller*, *field* tersebut dijadikan sebagai relasi dengan table users. Tipe data yang digunakan adalah integer dengan panjang 4. panjang tipe data tersebut dibuat sesuai dengan *field* relasinya yaitu *field* id pada tabel users.

Field content digunakan untuk menyimpan data deskripsi kiriman diskusi *traveller* dengan tipe data longtext.

4. Tabel post_comments

Pada tabel post_comments digunakan untuk menyimpan data komentar diskusi *traveller*. berikut dibawah ini spesifikasi dari tabel post_comments :

Tabel 3.6: Tabel post_comments

NO	NAMA FIELD	TIPE	PANJANG	KETERANGAN
1	id	Int	5	Primary Key, AI
2	post_id	Int	5	-
3	user_id	Int	4	-
4	comment	Longtext	-	-

Field id pada tabel ini mempunyai panjang sebesar 5 dengan tipe data integer. *Field* id tersebut maksimal menyimpan data sebanyak 5 digit yaitu 99999 kiriman komentar diskusi *traveller* dan dibuat sebagai primary key atau kunci tabel dengan pengisian otomatis yaitu auto increment (AI).

Pada *field* post_id digunakan untuk menyimpan data *post* atau kiriman diskusi *traveller*, *field* tersebut dijadikan sebagai relasi dengan tabel posts. Tipe data yang digunakan adalah integer dengan panjang 5. panjang tipe data tersebut dibuat sesuai dengan *field* relasinya yaitu *field* id pada tabel posts.

Pada *field* user_id digunakan untuk menyimpan data *traveller*, *field* tersebut dijadikan sebagai relasi dengan table users. Tipe data yang digunakan adalah integer dengan panjang 4. panjang tipe data tersebut dibuat sesuai dengan *field* relasinya yaitu *field* id pada tabel users.

Field comment digunakan untuk menyimpan data deskripsi komentar dengan tipe data longtext.

5. Tabel post_likes

Pada tabel post_likes digunakan untuk menyimpan data user yang menyukai kiriman diskusi *traveller*. berikut dibawah ini spesifikasi dari tabel post_likes :

Tabel 3.7: Tabel post_likes

NO	NAMA FIELD	TIPE	PANJANG	KETERANGAN
1	id	Int	5	Primary Key, AI
2	post_id	Int	5	-
3	user_id	Int	4	-

Field id pada tabel ini mempunyai panjang sebesar 5 dengan tipe data integer. *Field* id tersebut maksimal menyimpan data sebanyak 5 digit yaitu 99999 data *traveller* dan dibuat sebagai primary key atau kunci tabel dengan pengisian otomatis yaitu auto increment (AI).

Pada *field* post_id digunakan untuk menyimpan data *post* atau kiriman diskusi *traveller*, *field* tersebut dijadikan sebagai relasi dengan tabel posts. Tipe data yang digunakan adalah integer dengan panjang 5. panjang tipe data tersebut dibuat sesuai dengan *field* relasinya yaitu *field* id pada tabel posts.

Pada *field* *user_id* digunakan untuk menyimpan data *traveller*, *field* tersebut dijadikan sebagai relasi dengan table *users*. Tipe data yang digunakan adalah integer dengan panjang 4. panjang tipe data tersebut dibuat sesuai dengan *field* relasinya yaitu *filed* *id* pada tabel *users*.

6. Tabel barengans

Tabel barengans digunakan untuk menyimpan data kiriman cari barengan *traveller*. Berikut dibawah ini spesifikasi tabel barengans :

Tabel 3.8: Tabel barengans

NO	NAMA FIELD	TIPE	PANJANG	KETERANGAN
1	id	Int	5	Primary Key, AI
2	user_id	Int	4	-
3	content	Longtext	-	-
4	tujuan	Varchar	30	-
5	mepo	Varchar	30	-
6	mulai	date	-	-
7	akhir	date	-	-
8	contact	Varchar	12	-

Field *id* pada tabel ini mempunyai panjang sebesar 5 dengan tipe data integer. *Field* *id* tersebut maksimal menyimpan data sebanyak 5 digit yaitu 99999 data kiriman barengan *traveller* dan dibuat sebagai primary key atau kunci tabel dengan pengisian otomatis yaitu auto increment (AI).

Pada *field* *user_id* digunakan untuk menyimpan data *traveller*, *field* tersebut dijadikan sebagai relasi dengan table *users*. Tipe data yang digunakan adalah integer dengan panjang 4. panjang tipe data tersebut dibuat sesuai dengan *field* relasinya yaitu *filed* *id* pada tabel *users*.

Field *content* digunakan untuk menyimpan data deskripsi kiriman barengan *traveller* dengan tipe data longtext. *Field* *tujuan* digunakan untuk menyimpan data tujuan kiriman barengan dengan panjang 30 dan tipe data varchar. *Field* *mepo* digunakan untuk menyimpan data *meeting point* kiriman barengan dengan panjang 30 dan tipe data varchar. Pada *field* *mulai* dan *akhir* digunakan untuk menyimpan tanggal mulai dan tanggal berakhir rencana wisata *traveller* dengan type data date. *Field* *contact* digunakan untuk menyimpan data kontak dari *traveller* dengan tipe data varchar dan panjang 12.

7. Tabel `barengan_comments`

Pada tabel `barengan_comments` digunakan untuk menyimpan data komentar dari *traveller*. berikut dibawah ini spesifikasi dari tabel `barengan_comments`

Tabel 3.9: Tabel `barengan_comments`

NO	NAMA FIELD	TIPE	PANJANG	KETERANGAN
1	<code>id</code>	Int	5	Primary Key, AI
2	<code>user_id</code>	Int	4	-
3	<code>barengan_id</code>	Int	5	-
4	<code>comment</code>	Long Text	-	-

Field `id` pada tabel ini mempunyai panjang sebesar 5 dengan tipe data integer. *Field* `id` tersebut maksimal menyimpan data sebanyak 5 digit yaitu 99999 kiriman komentar *traveller* dan dibuat sebagai primary key atau kunci tabel dengan pengisian otomatis yaitu auto increment (AI).

Pada *field* `post_id` digunakan untuk menyimpan data *barengan traveller*, *field* tersebut dijadikan sebagai relasi dengan tabel `posts`. Tipe data yang digunakan adalah integer dengan panjang 5. panjang tipe data tersebut dibuat sesuai dengan *field* relasinya yaitu *field* `id` pada tabel `posts`.

Pada *field* `user_id` digunakan untuk menyimpan data *traveller*, *field* tersebut dijadikan sebagai relasi dengan table `users`. Tipe data yang digunakan adalah integer dengan panjang 4. panjang tipe data tersebut dibuat sesuai dengan *field* relasinya yaitu *field* `id` pada tabel `users`.

Field `comment` digunakan untuk menyimpan data deskripsi komentar dengan tipe data `longtext`.

8. Tabel `barengan_gabung`

Pada tabel `barengan_gabung` digunakan untuk menyimpan data *traveller* yang bergabung dengan *cari barengan*. berikut dibawah ini spesifikasi dari tabel `barengan_gabung`

Tabel 3.10: Tabel barengan_gabung

NO	NAMA FIELD	TIPE	PANJANG	KETERANGAN
1	id	Int	5	Primary Key, AI
2	user_id	Int	4	-
3	barengan_id	Int	5	-

Field id pada tabel ini mempunyai panjang sebesar 5 dengan tipe data integer. *Field* id tersebut maksimal menyimpan data sebanyak 5 digit yaitu 99999 kiriman barengan gabungan *traveller* dan dibuat sebagai primary key atau kunci tabel dengan pengisian otomatis yaitu auto increment (AI).

Pada *field* user_id digunakan untuk menyimpan data *traveller*, *field* tersebut dijadikan sebagai relasi dengan table users. Tipe data yang digunakan adalah integer dengan panjang 4. panjang tipe data tersebut dibuat sesuai dengan *field* relasinya yaitu *field* id pada tabel users.

Pada *field* barengan_id digunakan untuk menyimpan data barengan *traveller*, *field* tersebut dijadikan sebagai relasi dengan tabel barengans. Tipe data yang digunakan adalah integer dengan panjang 5. panjang tipe data tersebut dibuat sesuai dengan *field* relasinya yaitu *field* id pada tabel barengans.

9. Tabel singgahs

Tabel singgahs digunakan untuk menyimpan data kiriman penyedia rumah singgah *traveller*. Berikut dibawah ini spesifikasi tabel singgahs :

Tabel 3.11: Tabel singgahs

NO	NAMA FIELD	TIPE	PANJANG	KETERANGAN
1	id	Int	5	Primary Key, AI
2	user_id	Int	4	-
3	lokasi	Varchar	30	-
4	content	Longtext	-	-
5	contact	Varchar	12	-

Field id pada tabel ini mempunyai panjang sebesar 5 dengan tipe data integer. *Field* id tersebut maksimal menyimpan data sebanyak 5 digit yaitu 99999 data kiriman rumah singgah *traveller* dan dibuat sebagai

primary key atau kunci tabel dengan pengisian otomatis yaitu auto increment (AI).

Pada *field* *user_id* digunakan untuk menyimpan data *traveller*, *field* tersebut dijadikan sebagai relasi dengan table *users*. Tipe data yang digunakan adalah integer dengan panjang 4. panjang tipe data tersebut dibuat sesuai dengan *field* relasinya yaitu *field* *id* pada tabel *users*.

Field lokasi digunakan untuk menyimpan data lokasi rumah singgah *traveller* dengan tipe data *varchar* dan panjang 30. *Field* *content* digunakan untuk menyimpan data deskripsi kiriman rumah singgah dengan tipe data *longtext*. *Field* *contact* digunakan untuk menyimpan data kontak dari *traveller* dengan tipe data *varchar* dan panjang 12.

10. Tabel *singgah_reviews*

Pada tabel *singgah_reviews* digunakan untuk menampung data ulasan rumah singgah *traveller*. berikut dibawah ini spesifikasi dari tabel *singgah_reviews* :

Tabel 3.12: Tabel *singgah_comments*

NO	NAMA FIELD	TIPE	PANJANG	KETERANGAN
1	<i>id</i>	Int	5	Primary Key, AI
2	<i>user_id</i>	Int	4	-
3	<i>singgah_id</i>	Int	5	-
4	<i>rating</i>	Int	1	-
5	<i>comment</i>	Long Text	-	-

Field *id* pada tabel ini mempunyai panjang sebesar 5 dengan tipe data integer. *Field* *id* tersebut maksimal menyimpan data sebanyak 5 digit yaitu 99999 kiriman pencarian rumah singgah *traveller* dan dibuat sebagai primary key atau kunci tabel dengan pengisian otomatis yaitu auto increment (AI).

Pada *field* *user_id* digunakan untuk menyimpan data *traveller*, *field* tersebut dijadikan sebagai relasi dengan table *users*. Tipe data yang digunakan adalah integer dengan panjang 4. panjang tipe data tersebut dibuat sesuai dengan *field* relasinya yaitu *field* *id* pada tabel *users*.

Pada *field* *singgah_id* digunakan untuk menyimpan data id singgah *traveller*, *field* tersebut dijadikan sebagai relasi dengan tabel *singga*hs. Tipe data yang digunakan adalah integer dengan panjang 5. panjang tipe

data tersebut dibuat sesuai dengan *field* relasinya yaitu *filed* id pada tabel singgahs.

Field rating digunakan untuk menyimpan data jumlah rating yang *traveller* berikan dengan tipe data int dan panjang 1, karena rating yang berikan hanya berjumlah 1 digit yaitu 1 sampai 5. *Field* comment dibuat untuk menyimpan data deskripsi ulasan yang *traveller* berikan dengan tipe data longtext.

11. Tabel infos

Tabel infos digunakan untuk menyimpan data kiriman informasi wisata. Berikut dibawah ini spesifikasi tabel infos :

Tabel 3.13: Tabel infos

NO	NAMA FIELD	TIPE	PANJANG	KETERANGAN
1	id	Int	5	Primary Key, AI
2	user_id	Int	4	-
3	content	Longtext	-	-
4	images	Varchar	50	-

Field id pada tabel ini mempunyai panjang sebesar 5 dengan tipe data integer. *Field* id tersebut maksimal menyimpan data sebanyak 5 digit yaitu 99999 data kiriman informasi wisata *traveller* dan dibuat sebagai primary key atau kunci tabel dengan pengisian otomatis yaitu auto increment (AI).

Pada *field* user_id digunakan untuk menyimpan data *traveller*, *field* tersebut dijadikan sebagai relasi dengan table users. Tipe data yang digunakan adalah integer dengan panjang 4. panjang tipe data tersebut dibuat sesuai dengan *field* relasinya yaitu *filed* id pada tabel users.

Field content digunakan untuk menyimpan data deskripsi kiriman informasi wisata dengan tipe data longtext. Pada *field* images digunakan untuk menyimpan path atau lokasi dari file gambar dengan tipe data varchar dan panjang 100.

12. Tabel info_reviews

Pada tabel info_reviews digunakan untuk menampung data ulasan informasi wisata. berikut dibawah ini spesifikasi dari tabel info_reviews :

Tabel 3.14: Tabel info_comments

NO	NAMA FIELD	TIPE	PANJANG	KETERANGAN
1	id	Int	5	Primary Key, AI
2	user_id	Int	4	-
3	info_id	Int	5	-
4	rating	Int	1	-
5	comment	Long Text	-	-

Field id pada tabel ini mempunyai panjang sebesar 5 dengan tipe data integer. *Field* id tersebut maksimal menyimpan data sebanyak 5 digit yaitu 99999 kiriman informasi wisata *traveller* dan dibuat sebagai primary key atau kunci tabel dengan pengisian otomatis yaitu auto increment (AI).

Pada *field* user_id digunakan untuk menyimpan data *traveller*, *field* tersebut dijadikan sebagai relasi dengan table users. Tipe data yang digunakan adalah integer dengan panjang 4. panjang tipe data tersebut dibuat sesuai dengan *field* relasinya yaitu *field* id pada tabel users.

Pada *field* info_id digunakan untuk menyimpan data id info *traveller*, *field* tersebut dijadikan sebagai relasi dengan tabel infos. Tipe data yang digunakan adalah integer dengan panjang 5. panjang tipe data tersebut dibuat sesuai dengan *field* relasinya yaitu *field* id pada tabel infos.

Field rating digunakan untuk menyimpan data jumlah rating yang *traveller* berikan dengan tipe data int dan panjang 1, karena rating yang berikan hanya berjumlah 1 digit yaitu 1 sampai 5. *Field* comment dibuat untuk menyimpan data deskripsi ulasan yang *traveller* berikan dengan tipe data longtext.

Bab 4

Implementasi dan Uji Coba

4.1 Implementasi Collaborative Filtering

Pada proses implementasi *Collaborative Filtering* menggunakan metode *itembased collaborative filtering* dilakukan dengan beberapa tahapan berikut :

1. Memberikan rating untuk masing - masing konten oleh *traveller*.

Pada tahap ini dilakukan rekomendasi dengan *sample* 5 data konten yang ada didalam info wisata, 5 data *traveller* dan 10 data rating. Tabel berikut ini menunjukkan jumlah rating yang diberikan oleh *traveller* untuk masing - masing konten

Tabel 4.1: Tabel Sample Rating

Judul Content	Traveller 1	Traveller 2	Traveller 3	Traveller 4	Traveller 5
Patai Pink Lombok	0	0	0	1 ⁴	1 ⁴
Gunung Merbabu	0	0	1 ⁵	0	1 ⁵
Curug Cikaracak	1 ⁵	0	1 ⁴	0	0
Pantai Ora	1 ⁵	1 ⁵	0	0	0
Gunung Rinjani	1 ⁴	0	1 ²	0	0

T1 - T5 = Traveller yang telah terdaftar pada sistem

Angka 1 dan 0 = Jika traveller telah memberi rating pada konten maka ditandai dengan angka 1 dan jika traveller tidak pernah memberi ulasan ditandai dengan angka 0

Angka superskrip (1-5) = rating yang diberikan traveller terhadap konten

Setelah dilakukan pemberian rating terhadap konten, tahap selanjutnya yaitu menghitung prediksi rekomendasi dengan menggunakan algoritma pembangkit rekomendasi.

2. Perhitungan algoritma pembangkit rekomendasi pada sistem

Pada tahap ini formula yang digunakan untuk perhitungan adalah sebagai berikut :

$$R = \left(\frac{\sum B \times \text{rating} \times 100}{\text{Max} \times \text{MaxB}} \right) / C$$

Gambar 4.1: Formula Algoritma Pembangkit Rekomendasi.

R = Rekomendasi

B = Nilai ulasan konten (1 jika memberi ulasan & 0 jika tidak memberi ulasan).

Rating = Jumlah rating yang diberikan oleh traveller terhadap konten

100 = Nilai maksimum 100, nilai rekomendasi tertinggi adalah 100.

Max = Nilai maksimum rating adalah 5

MaxB = Nilai maksimum ulasan yang diberikan traveller

C = Jumlah *traveller* yang mempunyai pola rekomendasi yang sama dengan *traveller* yang login pada sistem.

Pada perhitungan dibawah ini data *traveller* yang digunakan sebagai contoh perhitungan ialah traveller dengan id 2. Jika *traveller* login ke dalam sistem maka perhitungan pembangkit rekomendasi adalah sebagai berikut :

```

1 <?php namespace App;
2 use Illuminate\Database\Eloquent\Model;
3
4 class Info extends Model {
5     public function getPerhitunganAttribute()
6     {
7         if ($this->attributes['user_id'] == auth()->id()) {
8             $rating = 0;
9         }
10        else {
11            $rating = $this->attributes['rating'];
12        }

```

```

13             $hitung = 1 * $rating * 100 / 5 * 1;
14             return $hitung;
15         }
16
17     public function getRekomendasiAttribute()
18     {
19         $rating = $this->infocomment->pluck('perhitungan')->sum
16         ();
20         $user = User::all()->pluck('id')->count();
21         $ratinguserlain = $user;
22         return $rating / $ratinguserlain;
23     }
24 }

```

baris 7 - 12 adalah proses pengecekan apakah traveller sudah pernah memberi rating atau ulasan pada content, jika pernah maka data tidak akan dihitung, jika tidak pernah maka data akan dihitung.

baris 13 adalah proses perhitungan adalah proses perhitungan dari formula $B * rating * 100 / Max + Max B$

baris 14 adalah hasil dari proses perhitungan dari ulasan yang ditampung kedalam variable \$hitung

baris 19 adalah proses penjumlahan dari masing - masing perhitungan rating yang ada didalam setiap content kemudian ditampung kedalam variable \$rating

baris 20 - 21 adalah proses untuk mencari nilai C pada formula atau Jumlah traveller yang mempunyai pola rekomendasi yang sama dengan traveller yang login pada sistem, kemudian data tersebut ditampung kedalam variable \$user

baris 22 adalah proses terakhir dari perhitungan, yaitu hasil dari formula $B * rating * 100 / Max + Max B$ dibagi dengan C

Setelah dilakukan proses perhitungan pada sistem, tahap berikutnya adalah menampilkan hasil perhitungan berdasarkan urutan rekomendasi.

3. Mengurutkan hasil perhitungan berdasarkan nilai prediksi pada sistem

Berikut dibawah ini adalah code yang digunakan untuk mengurutkan hasil perhitungan berdasarkan nilai prediksi pada sistem

```

1 <?php namespace App\Http\Controllers;
2 class InfoController extends Controller {

```

```

3      public function index(){
4          $data = Info::with('infocomment','info','likecek')
5              ->filtered()
6              ->get()
7              ->sortByDesc('rekomendasi');
8      return view('info',compact('data'));
9  }
10 }

```

baris 4 - 6 adalah proses menampilkan data konten yang ada pada sistem

baris 7 adalah proses mengurutkan data berdasarkan perhitungan prediksi yang ditampung kedalam variable \$data

Setelah dilakukan proses perhitungan pada sistem dan proses pengurutan data berdasarkan hasil perhitungan, tahap terakhir adalah menampilkan hasil pada sistem.

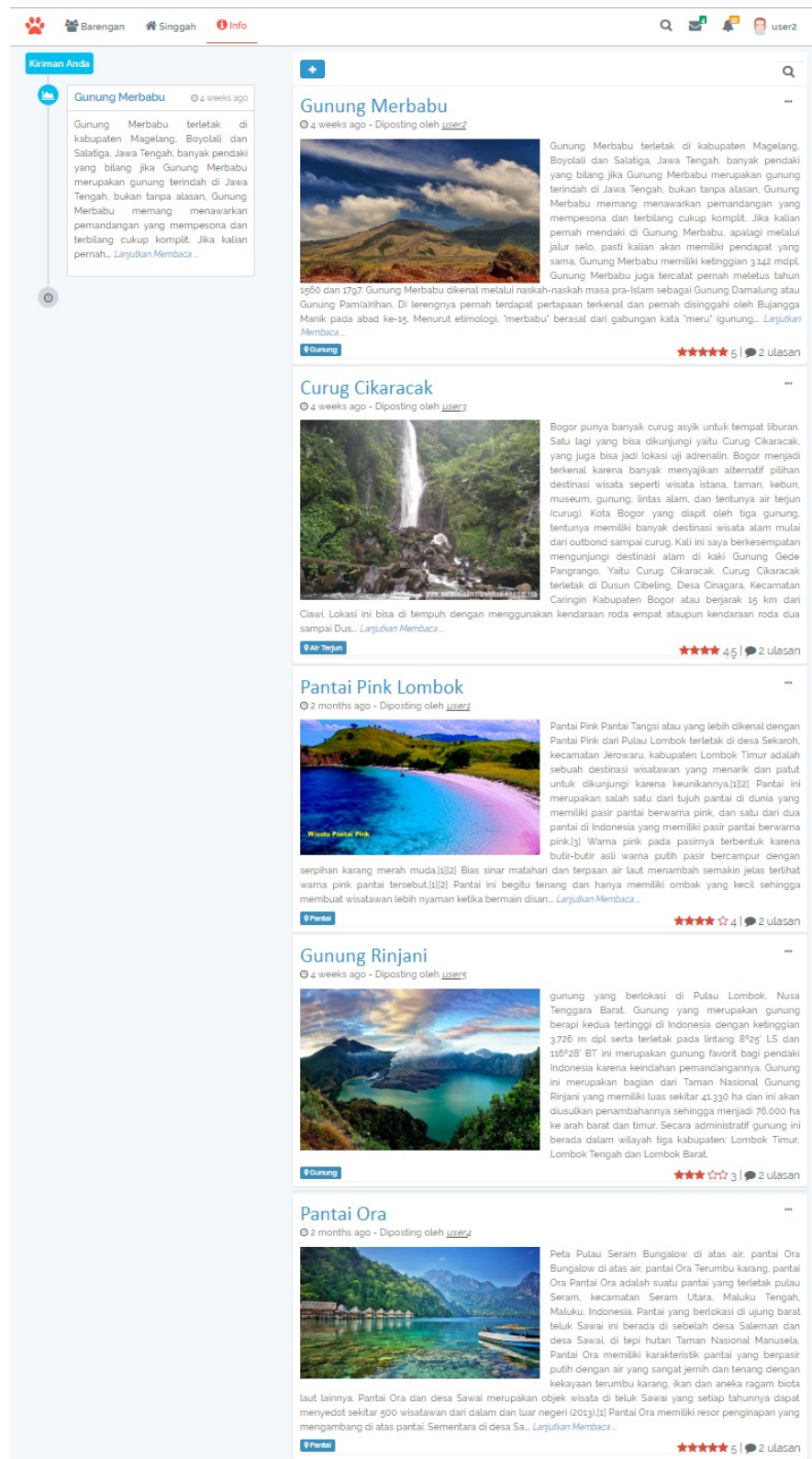
4. Menampilkan hasil perhitungan prediksi pada sistem

Pada tahap ini adalah menampilkan hasil prediksi traveller, data *traveller* yang akan digunakan sebagai contoh adalah *traveller* dengan id 2. Sebelumnya jika dilihat dari data tabel *sample rating traveller 2* sudah memberi rating pada konten Pantai Ora, sehingga sistem akan merekomendasikan konten kepada *traveller 2* seperti tabel di bawah ini.

Tabel 4.2: Tabel Hasil Perhitungan Prediksi

Konten	Nilai Prediksi
Gunung Merbabu	50
Curug Cikaracak	45
Patai Pink Lombok	40
Gunung Rinjani	30
Pantai Ora	25

Berikut dibawah adalah hasil dari perhitungan yang sudah diterapkan pada sistem website.



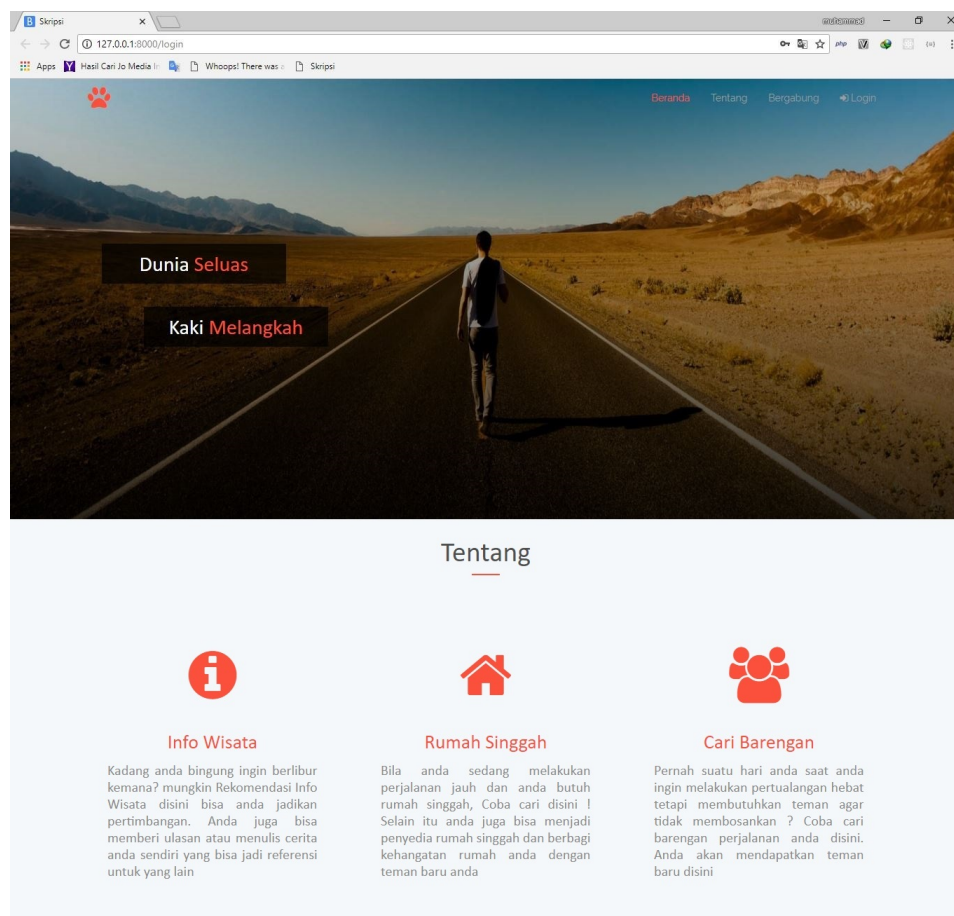
Gambar 4.2: Tampilan Program Hasil Perhitungan

Dapat dilihat bahwa nilai prediksi rekomendasi tertinggi jika *traveller* dengan id 2 masuk ke dalam sistem adalah konten Gunung Merbabu dengan nilai 50 dan nilai terendah adalah konten Pantai Ora yaitu 25. rendahnya hasil perhitungan konten Pantai Ora dikarenakan *traveller* 2 sudah pernah memberikan ulasan pada konten tersebut.

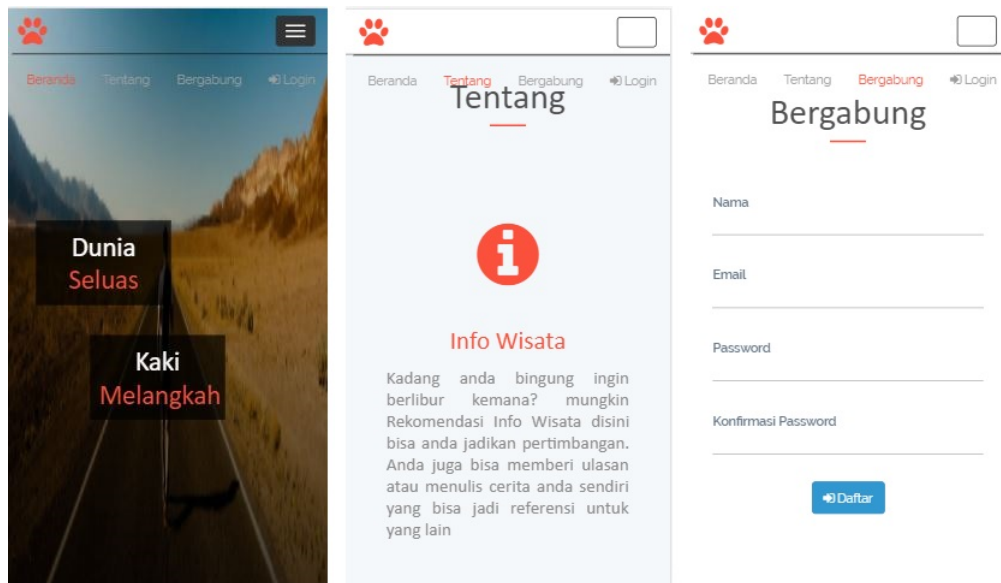
4.2 Implementasi Tampilan

Implementasi tampilan menjelaskan tentang tampilan masing – masing halaman dan menu yang tersedia pada sistem ini, sehingga diketahui apakah sistem telah dibuat sesuai dengan yang direncanakan. Pada tampilan dibawah masing - masing terdapat tampilan versi website yang akses dengan menggunakan perangkat laptop atau komputer dan versi mobile ialah sistem website yang diakses dengan menggunakan perangkat mobile.

1. Tampilan Halaman Utama



Gambar 4.3: Tampilan Halaman Utama Website



Gambar 4.4: Tampilan Halaman Utama Versi Mobile

Pada tampilan utama terdapat 4 menu yaitu utama yaitu :

(a) Menu Beranda

Menu Beranda dibuat hanya sebagai banner website untuk meperindah tampilan website.

(b) Menu Tentang

Menu Tentang dibuat untuk menampilkan informasi apa saja yang bisa dilakukan pada website ini.

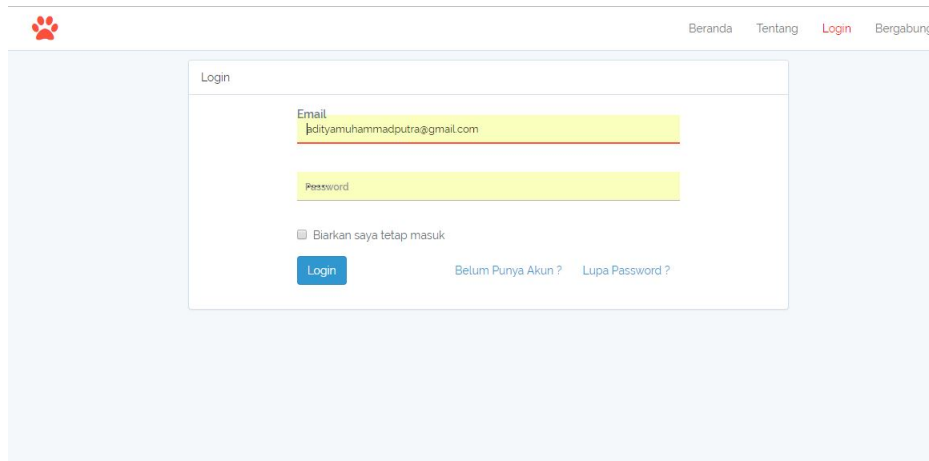
(c) Menu Bergabung

Menu Bergabung dibuat untuk *traveller* baru yang akan mendaftar pada website ini.

(d) Menu Login

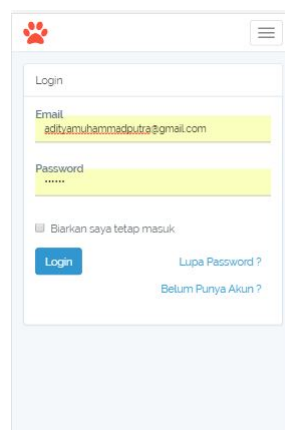
Menu Login dibuat untuk *traveller* masuk kedalam website.

2. Tampilan Halaman Login



The screenshot shows a desktop browser window with a red paw print logo in the top left corner. The top navigation bar contains links: Beranda, Tentang, Login (highlighted in red), and Bergabung. The main content area features a white login box with the title 'Login'. Inside the box, there are two input fields: 'Email' with the value 'adityamuhammadputra@gmail.com' and 'Password' with masked characters. Below the password field is a checkbox labeled 'Biarkan saya tetap masuk'. At the bottom of the box are three elements: a blue 'Login' button, a blue link 'Belum Punya Akun?', and a blue link 'Lupa Password?'.

Gambar 4.5: Tampilan Program Login Versi Website

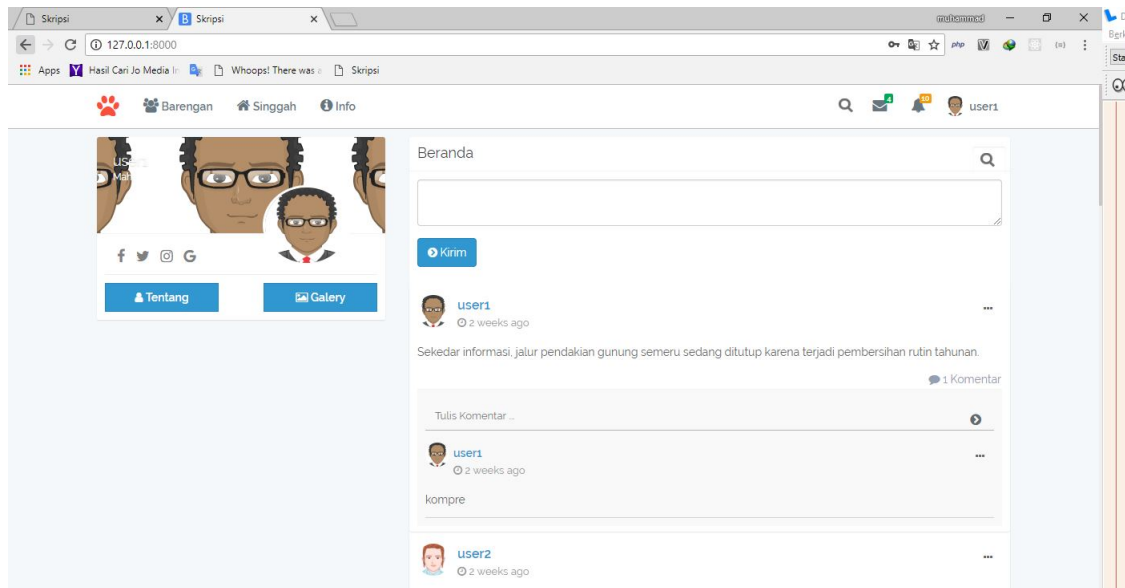


The screenshot shows a mobile browser window with a red paw print logo in the top left corner and a hamburger menu icon in the top right. The login box is centered and contains the same fields and links as the desktop version: 'Email' (adityamuhammadputra@gmail.com), 'Password' (masked), a 'Biarkan saya tetap masuk' checkbox, a blue 'Login' button, and blue links for 'Lupa Password?' and 'Belum Punya Akun?'.

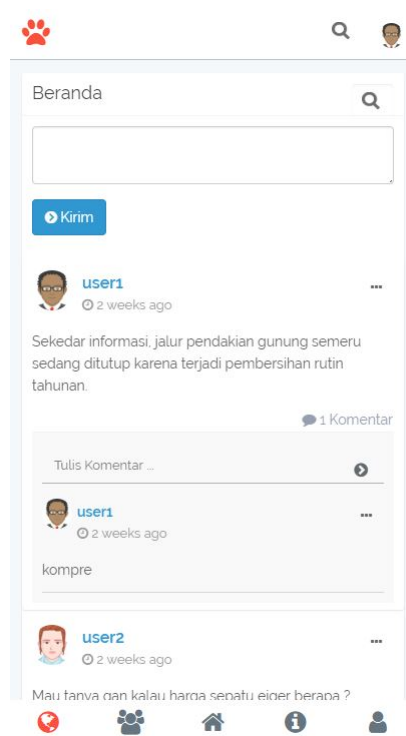
Gambar 4.6: Tampilan Program Login Versi Mobile

Pada tampilan login dibuat *traveller* masuk kedalam sistem dengan memasukkan email dan password yang telah *traveller* daftarkan.

3. Tampilan Halaman Home



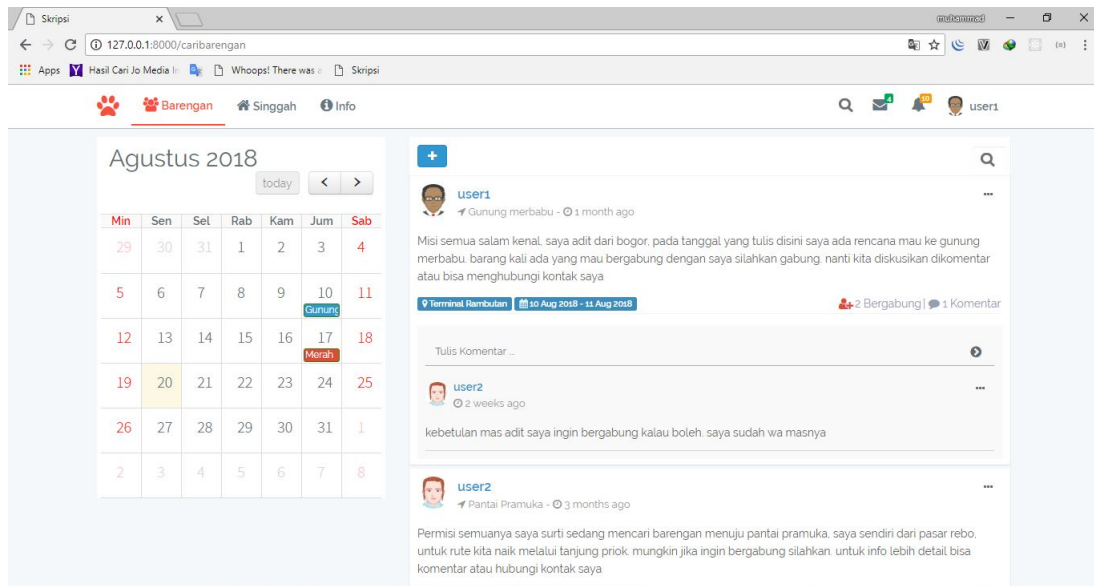
Gambar 4.7: Tampilan Program Home Versi Website



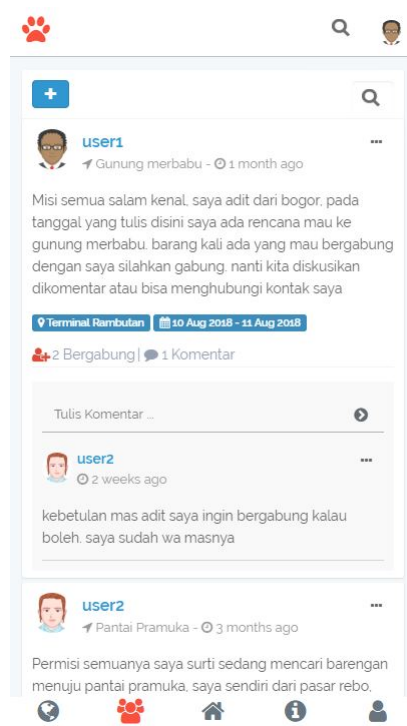
Gambar 4.8: Tampilan Halaman Home Versi Mobile

Pada tampilan halaman home dibuat untuk traveller berdiskusi, seperti berdiskusi tentang pertanyaan yang berkaitan dengan travelling atau berwisata.

4. Tampilan Halaman Barengan



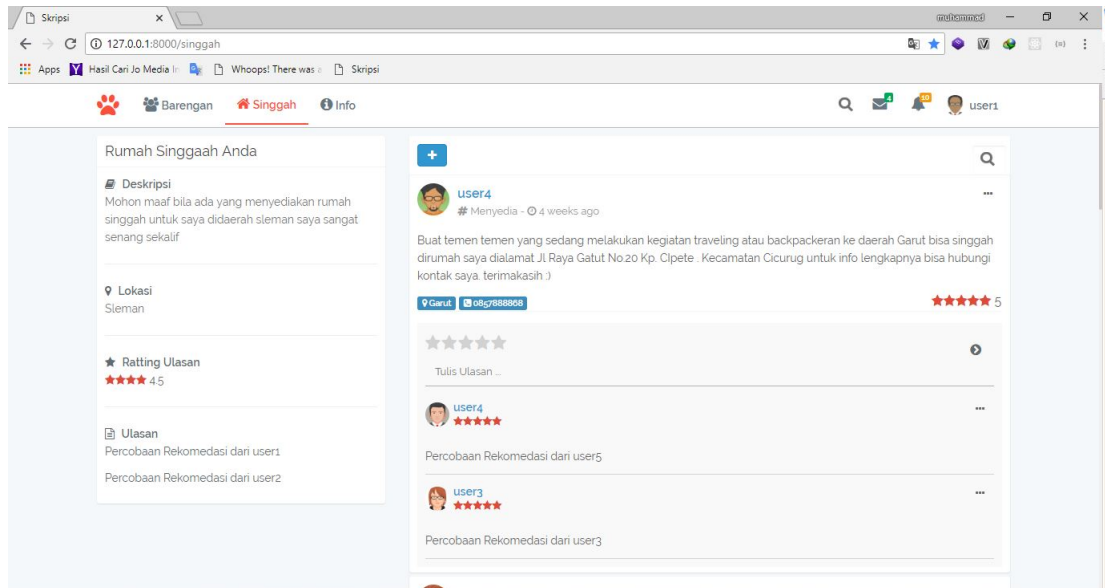
Gambar 4.9: Tampilan Halaman Barengan Versi Website



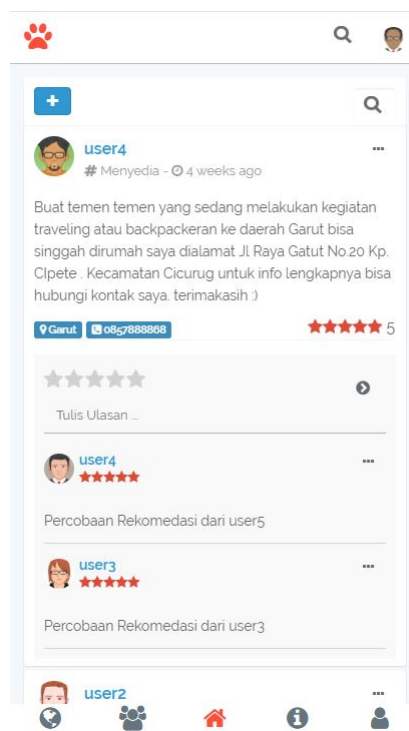
Gambar 4.10: Tampilan Halaman Barengan Versi Website

Pada Halaman Barengan dibuat untuk menampilkan informasi *traveller* yang mencari barengan pada website.

5. Tampilan Halaman Singgah



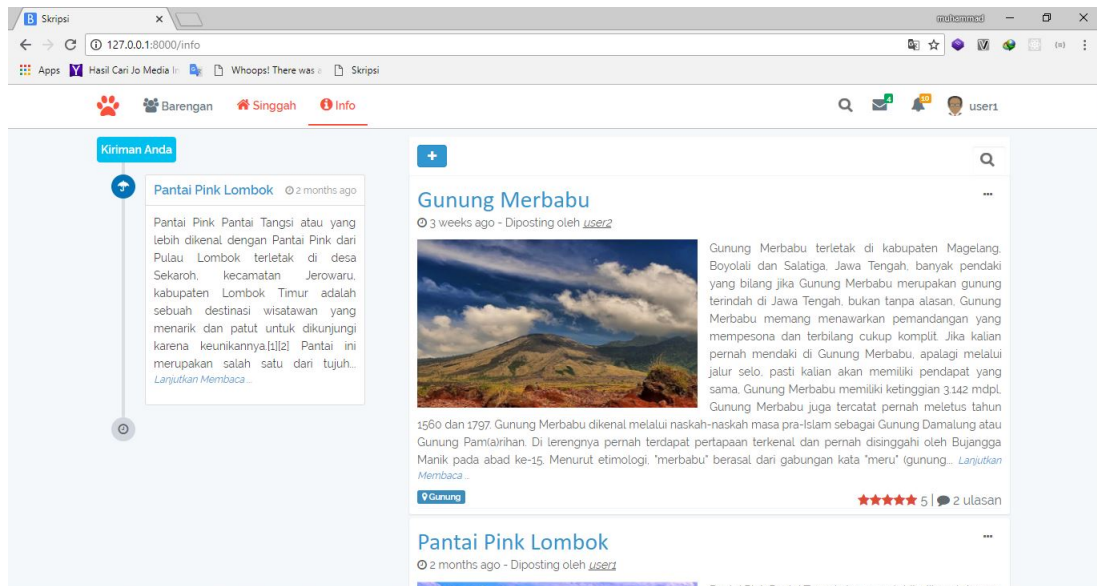
Gambar 4.11: Tampilan Halaman Singgah Versi Mobile



Gambar 4.12: Tampilan Halaman Singgah Versi Mobile

Pada Hamalan Singgah dibuat untuk menampilkan informasi *traveller* yang menyediakan rumah singgah. Pada halaman ini *traveller* juga dapat mencari rumah singgah.

6. Tampilan Halaman Info Wisata



Gambar 4.13: Tampilan Halaman Info Versi Website



Gambar 4.14: Tampilan Halaman Info Versi Mobile

Pada Halaman Info Wisata dibuat untuk menampilkan informasi rekomendasi wisata yang tersedia pada website ini.

4.3 Implementasi Database

Pembuatan database dilakukan dengan menggunakan database MySQL. Implementasi database dilakukan dengan menggunakan fitur database *migrations laravel* yang mengubah kode menjadi tabel. Berikut dibawah ini kode dalam pembuatan masing - masing tabel.

1. Pembuatan Tabel users

```
<?php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
class CreateUsersTable extends Migration
{
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name',25);
            $table->string('email',30)->unique();
            $table->string('password',100);
            $table->boolean('active',1)->default(false);
            $table->string('activation_token',100)->nullable();
            $table->boolean('status',1)->default(false);
            $table->string('avatar',100)->nullable()->default('
                avatars/change-user-male.png');
            $table->string('pekerjaan',25)->nullable();
            $table->string('agama',20)->nullable()->default('belum
                anda isi');
            $table->string('hobby',30)->nullable()->default('belum
                anda isi');
            $table->text('bio')->nullable()->default('belum anda isi
                ');
            $table->string('contact',12)->nullable()->default('belum
                anda isi');
            $table->rememberToken();
            $table->timestamps();});
    }

    public function down()
    {
        Schema::dropIfExists('users');
    }
}
```

2. Pembuatan Table galleries

```
<?php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
class CreateGalleriesTable extends Migration {
    public function up()
```

```

{
    Schema::create('galleries', function (Blueprint $table) {
        $table->increments('id');
        $table->integer('user_id',4)->unsigned();
        $table->string('images',100)->nullable();
        $table->longtext('caption');
        $table->timestamps();
        $table->foreign('user_id')->references('id')->on('users')->
            onDelete('CASCADE');    });
    }
    public function down()
    {
        Schema::dropIfExists('galleries');
    }
}

```

3. Pembuatan Tabel post

```

<?php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
class CreatePostsTable extends Migration {
    public function up()
    {
        Schema::create('posts', function (Blueprint $table) {
            $table->increments('id');
            $table->integer('user_id',4)->unsigned();
            $table->longtext('content');
            $table->timestamps();
            $table->foreign('user_id')->references('id')->on('users')->
                onDelete('CASCADE');    });
    }
    public function down()
    {
        Schema::dropIfExists('posts');
    }
}

```

4. Pembuatan Tabel post_comments

```

<?php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
class CreatePostCommentsTable extends Migration {
    public function up()
    {
        Schema::create('post_comments', function (Blueprint
            $table) {
            $table->increments('id');
            $table->integer('user_id',4)->unsigned();
            $table->integer('post_id',5)->unsigned();
            $table->longtext('content');
            $table->timestamps();
            $table->foreign('user_id')->references('id')->on('users')->
                onDelete('CASCADE');    });
    }
}

```

```

        $table->foreign('post_id')->references('id')->on('posts
            ')->onDelete('CASCADE');
    }
    public function down()
    {
        Schema::dropIfExists('post_comments');
    }
}

```

5. Pembuatan Table post_likes

```

<?php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
class CreatePostLikesTable extends Migration {
    public function up()
    {
        Schema::create('post_likes', function (Blueprint $table)
        {
            $table->increments('id');
            $table->integer('user_id',4)->unsigned();
            $table->integer('post_id',5)->unsigned();
            $table->timestamps();
            $table->foreign('user_id')->references('id')->on('users')->
                onDelete('CASCADE'); });
            $table->foreign('post_id')->references('id')->on('posts
                ')->onDelete('CASCADE');
        }
    public function down()
    {
        Schema::dropIfExists('post_likes');
    }
}

```

6. Pembuatan Tabel barengans

```

<?php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
class CreateBarengansTable extends Migration {
    public function up()
    {
        Schema::create('barengans', function (Blueprint $table)
        {
            $table->increments('id');
            $table->integer('user_id',4)->unsigned();
            $table->longtext('content');
            $table->string('tujuan',20);
            $table->string('mepo',20);
            $table->date('mulai');
            $table->date('akhir');
            $table->string('contact',12);
            $table->timestamps();
            $table->foreign('user_id')->references('id')->on('users')->
                onDelete('CASCADE'); });
    }
}

```

```

    }
    public function down()
    {
        Schema::dropIfExists('barengans');
    }
}

```

7. Pembuatan Tabel barengan comment

```

<?php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
class CreateBarenganCommentsTable extends Migration {
    public function up()
    {
        Schema::create('barengan_comments', function (Blueprint
            $table) {
                $table->increments('id');
                $table->integer('user_id',4)->unsigned();
                $table->integer('barengan_id',5)->unsigned();
                $table->longtext('comment');
                $table->timestamps();
            $table->foreign('user_id')->references('id')->on('users')->
                onDelete('CASCADE');
                $table->foreign('barengan_id')->references('id')->on('
                    posts')->onDelete('CASCADE'); });
        }
    public function down()
    {
        Schema::dropIfExists('barengan_comments');
    }
}

```

8. Permbutan Tabel barengan_gabungs

```

<?php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
class CreateBarenganGabungsTable extends Migration {
    public function up()
    {
        Schema::create('barengan_gabungs', function (Blueprint
            $table) {
                $table->increments('id');
                $table->integer('user_id',4)->unsigned();
                $table->integer('barengan_id',5)->unsigned();
                $table->timestamps();
            $table->foreign('user_id')->references('id')->on('users')->
                onDelete('CASCADE');
                $table->foreign('barengan_id')->references('id')->on('
                    posts')->onDelete('CASCADE'); });
        }
    public function down()
    {

```

```

        Schema::dropIfExists('barengan_gabungs');
    }
}

```

9. Pembuatan Tabel singgahs

```

<?php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
class CreateSinggahsTable extends Migration {
    public function up()
    {
        Schema::create('Singgahs', function (Blueprint $table) {
            $table->increments('id');
            $table->integer('user_id',4)->unsigned();
            $table->integer('barengan_id',5)->unsigned();
            $table->longtext('lokasi');
            $table->longtext('content');
            $table->string('contact',12);
            $table->timestamps();
            $table->foreign('user_id')->references('id')->on('users')->
                onDelete('CASCADE'); });
    }
    public function down()
    {
        Schema::dropIfExists('singgahs');
    }
}

```

10. Pembuatan Table singgah_reviews

```

<?php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
class CreateSinggahReviewsTable extends Migration {
    public function up()
    {
        Schema::create('singgah_reviews', function (Blueprint
            $table) {
            $table->increments('id');
            $table->integer('user_id',4)->unsigned();
            $table->integer('barengan_id',5)->unsigned();
            $table->string('rating',1);
            $table->text('comment');
            $table->timestamps();
            $table->foreign('user_id')->references('id')->on('users')->
                onDelete('CASCADE'); });
    }
    public function down()
    {
        Schema::dropIfExists('singgah_reviews');
    }
}

```

11. Pembuatan Tabel Infos

```
<?php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
class CreateInfosTable extends Migration {
    public function up()
    {
        Schema::create('infos', function (Blueprint $table) {
            $table->increments('id');
            $table->integer('user_id',4)->unsigned();
            $table->string('title',30);
            $table->longtext('content');
            $table->string('images',100)->nullable();
            $table->timestamps();
            $table->foreign('user_id')->references('id')->on('users')->
                onDelete('CASCADE'); });
    }
    public function down()
    {
        Schema::dropIfExists('infos');
    }
}
```

12. Pembuatan Table info_reviews

```
<?php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
class CreateInfoReviewsTable extends Migration {
    public function up()
    {
        Schema::create('info_reviews', function (Blueprint
            $table) {
            $table->increments('id');
            $table->integer('user_id',4)->unsigned();
            $table->integer('info_id',5)->unsigned();
            $table->string('rating',1);
            $table->text('comment');
            $table->timestamps();
            $table->foreign('user_id')->references('id')->on('users')->
                onDelete('CASCADE'); });
    }
    public function down()
    {
        Schema::dropIfExists('info_reviews');
    }
}
```

4.4 Pengujian Black Box

Pengujian pada sistem ini menggunakan metode pengujian *black box*, pengujian ini dilakukan dengan cara memberikan sejumlah *input* pada sistem yang kemudian diproses sesuai dengan kebutuhan fungsinya untuk melihat apakah sistem yang kemudian diproses sesuai dengan fungsi sistem tersebut. Apabila dari *input* yang diberikan proses menghasilkan *output* yang sesuai dengan fungsinya, maka aplikasi yang bersangkutan telah benar, tetapi jika *output* yang dihasilkan tidak sesuai dengan fungsinya, maka masih terdapat kesalahan pada sistem.

4.4.1 Rencana Pengujian

Rencana pengujian selengkapnya dapat dilihat pada tabel berikut :

Tabel 4.3: Tabel Rencana Pengujian

No.	Kriteria Uji	Detail Pengujian	Jenis Pengujian
1	Daftar Anggota	<i>Traveller</i> Membuat Akun Baru	<i>Black Box</i>
2	Login User	<i>Traveller</i> Melakukan Login	<i>Black Box</i>
3	Fitur Diskusi	Tambah Diskusi Baru	<i>Black Box</i>
		Ubah Data Diskusi	<i>Black Box</i>
		Hapus Data Diskusi	<i>Black Box</i>
		Pencarian Data Diskusi	<i>Black Box</i>
		Tambah Komentar Diskusi	<i>Black Box</i>
		Hapus Komentar Diskusi	<i>Black Box</i>
4	Fitur Cari Barengan	Tambah Data Cari Barengan Baru	<i>Black Box</i>
		Ubah Data Cari Barengan	<i>Black Box</i>
		Hapus Data Cari Barengan	<i>Black Box</i>
		Pencarian Data Barengan	<i>Black Box</i>
		Tambah Data Komentar	<i>Black Box</i>
		Hapus Data Komentar	<i>Black Box</i>
5	Fitur Rumah Singgah	Tambah Data Rumah Singgah	<i>Black Box</i>
		Ubah Data Rumah Singgah	<i>Black Box</i>
		Hapus Data Rumah Singgah	<i>Black Box</i>
		Pencarian Data Rumah Singgah	<i>Black Box</i>
		Tambah Data Ulasan	<i>Black Box</i>
		Hapus Data Ulasan	<i>Black Box</i>
6	Fitur Info Wisata	Tambah Data Info Wisata	<i>Black Box</i>
		Ubah Data Info Wisata	<i>Black Box</i>
		Hapus Data Info Wisata	<i>Black Box</i>
		Pencarian Data Info Wisata	<i>Black Box</i>
		Tambah Data Ulasan	<i>Black Box</i>
		Hapus Data Ulasan	<i>Black Box</i>

4.4.2 Kasus dan Hasil Pengujian

Berdasarkan rencana pengujian yang telah disusun, maka dapat dilakukan pengujian sebagai berikut.

Tabel 4.4: Kasus dan hasil pengujian daftar, login dan fitur diskusi

<i>Requirement</i>	Skenario Uji	Hasil yang Diharapkan	Hasil
Daftar User Baru	jika benar	Data berhasil tersimpan dan bisa digunakan login	Sesuai
	jika salah	Tampil peringatan pada setiap kesalahan	Sesuai
Input Login	jika benar	Tampil halaman home	Sesuai
	jika salah	Tampil peringatan kesalahan	Sesuai
Tambah Diskusi Baru	jika benar	Data Tersimpan ke database dan tampil di halaman diskusi	Sesuai
	jika salah	Data tidak tersimpan dan tidak tampil di halaman diskusi	Sesuai
Ubah Data Diskusi	jika benar	Data di dalam database berubah dan tampil di halaman diskusi	Sesuai
	jika salah	Data tidak akan berubah tetapi tetap ditampilkan	Sesuai
Hapus Data Diskusi	jika benar	Data di dalam database dan di halaman akan terhapus	Sesuai
	jika salah	Data di dalam database dan di halaman tidak akan terhapus	Sesuai
Pencarian Data Diskusi	jika benar	Data pencarian akan tampil di halaman diskusi	Sesuai
	jika salah	Data pencarian tidak akan tampil dan tampil pemberitahuan	Sesuai
Tambah komentar	jika benar	Data tersimpan ke database dan tampil kolom komentar	Sesuai
	jika salah	Data tidak tersimpan dan tidak tampil di kolom komentar	Sesuai
Hapus komentar	jika benar	Data di dalam database dan di halaman akan terhapus	Sesuai
	jika salah	Data di dalam database dan di halaman tidak akan terhapus	Sesuai
Hapus Data Cari Barengan	jika benar	Data di dalam database dan di halaman akan terhapus	Sesuai
	jika salah	Data di dalam database dan di halaman tidak akan terhapus	Sesuai

Tabel 4.5: Kasus dan hasil pengujian fitur cari barengan

<i>Requirement</i>	Skenario Uji	Hasil yang Diharapkan	Hasil
Tambah Data Cari Barengan	jika benar	Data Tersimpan ke database dan tampil di halaman barengan	Sesuai
	jika salah	Data tidak tersimpan dan tidak tampil di halaman barengan	Sesuai
Ubah Data Cari Barengan	jika benar	Data di dalam database berubah dan tampil di halaman barengan	Sesuai
	jika salah	Data tidak akan berubah tetapi tetap ditampilkan	Sesuai
Hapus Data Cari Barengan	jika benar	Data di dalam database dan di halaman akan terhapus	Sesuai
	jika salah	Data di dalam database dan di halaman tidak akan terhapus	Sesuai
Pencarian Data Cari Barengan	jika benar	Data pencarian akan tampil di halaman Cari Barengan	Sesuai
	jika salah	Data pencarian tidak akan tampil dan tampil pemberitahuan	Sesuai
Tambah komentar Cari Barengan	jika benar	Data tersimpan ke database dan tampil kolom komentar	Sesuai
	jika salah	Data tidak tersimpan dan tidak tampil di kolom komentar	Sesuai
Hapus komentar Cari Barengan	jika benar	Data di dalam database dan di halaman akan terhapus	Sesuai
	jika salah	Data di dalam database dan di halaman tidak akan terhapus	Sesuai

Tabel 4.6: Kasus dan hasil pengujian fitur rumah singgah

<i>Requirement</i>	Skenario Uji	Hasil yang Diharapkan	Hasil
Tambah Data Rumah Siggah	jika benar	Data Tersimpan ke database dan tampil di halaman singgah	Sesuai
	jika salah	Data tidak tersimpan dan tidak tampil di halaman singgah	Sesuai
Ubah Data Rumah Siggah	jika benar	Data di dalam database berubah dan tampil di halaman singgah	Sesuai
	jika salah	Data tidak akan berubah tetapi tetap ditampilkan	Sesuai
Hapus Data Rumah Siggah	jika benar	Data di dalam database dan di halaman akan terhapus	Sesuai
	jika salah	Data di dalam database dan di halaman tidak akan terhapus	Sesuai
Pencarian Data Rumah Siggah	jika benar	Data pencarian akan tampil di halaman rumah singgah	Sesuai
	jika salah	Data pencarian tidak akan tampil dan tampil pemberitahuan	Sesuai
Tambah ulasan Rumah Siggah	jika benar	Data tersimpan ke database dan tampil kolom ulasan	Sesuai
	jika salah	Data tidak tersimpan dan tidak tampil di kolom ulasan	Sesuai
Hapus ulasan Rumah Siggah	jika benar	Data di dalam database dan di halaman akan terhapus	Sesuai
	jika salah	Data di dalam database dan di halaman tidak akan terhapus	Sesuai

Tabel 4.7: Kasus dan hasil pengujian fitur info wisata

<i>Requirement</i>	Skenario Uji	Hasil yang Diharapkan	Hasil
Tambah Data Info Wisata	jika benar	Data Tersimpan ke database dan tampil di halaman info	Sesuai
	jika salah	Data tidak tersimpan dan tidak tampil di halaman info	Sesuai
Ubah Data Info Wisata	jika benar	Data di dalam database berubah dan tampil di halaman info	Sesuai
	jika salah	Data tidak akan berubah tetapi tetap ditampilkan	Sesuai
Hapus Data Info Wisata	jika benar	Data di dalam database dan di halaman akan terhapus	Sesuai
	jika salah	Data di dalam database dan di halaman tidak akan terhapus	Sesuai
Pencarian Data Info Wisata	jika benar	Data pencarian akan tampil di halaman info wisata	Sesuai
	jika salah	Data pencarian tidak akan tampil dan tampil pemberitahuan	Sesuai
Tambah ulasan Info Wisata	jika benar	Data tersimpan ke database dan tampil kolom ulasan	Sesuai
	jika salah	Data tidak tersimpan dan tidak tampil di kolom ulasan	Sesuai
Hapus ulasan Info Wisata	jika benar	Data di dalam database dan di halaman akan terhapus	Sesuai
	jika salah	Data di dalam database dan di halaman tidak akan terhapus	Sesuai

4.5 Pengujian Fungsionalitas

Pada tahap pengujian fungsionalitas dengan melibatkan 25 responden *traveller* secara acak untuk mengisi kuesioner yang bertujuan untuk mengetahui pendapat responden tentang fungsi sistem ini. Dalam kuesioner ini terdapat 5 pertanyaan. Adapun hasil dari kuesioner sebagai berikut.

Tabel 4.8: Table Kuesioner

NO	Pertanyaan	TS	S	SS
1	Apakah sistem yang telah dibuat mudah digunakan oleh <i>traveller</i> ?	3	14	8
2	Apakah sistem yang telah dibuat sesuai dengan kebutuhan para <i>traveller</i> ?	2	15	8
3	Apakah <i>design</i> sistem atau tampilan yang dibuat cukup menarik ?	3	14	8
4	Apakah fasilitas dari masing - masing menu dan fitur yang terdapat dalam sistem memenuhi kebutuhan informasi <i>traveller</i> ?	0	12	13
5	Apakah <i>traveller</i> terbantu dengan adanya sistem ini ?	1	10	14

- TS = Tidak Setuju.
- S = Setuju
- SS = Sangat Setuju

Hasil yang diperoleh dari kuesioner adalah :

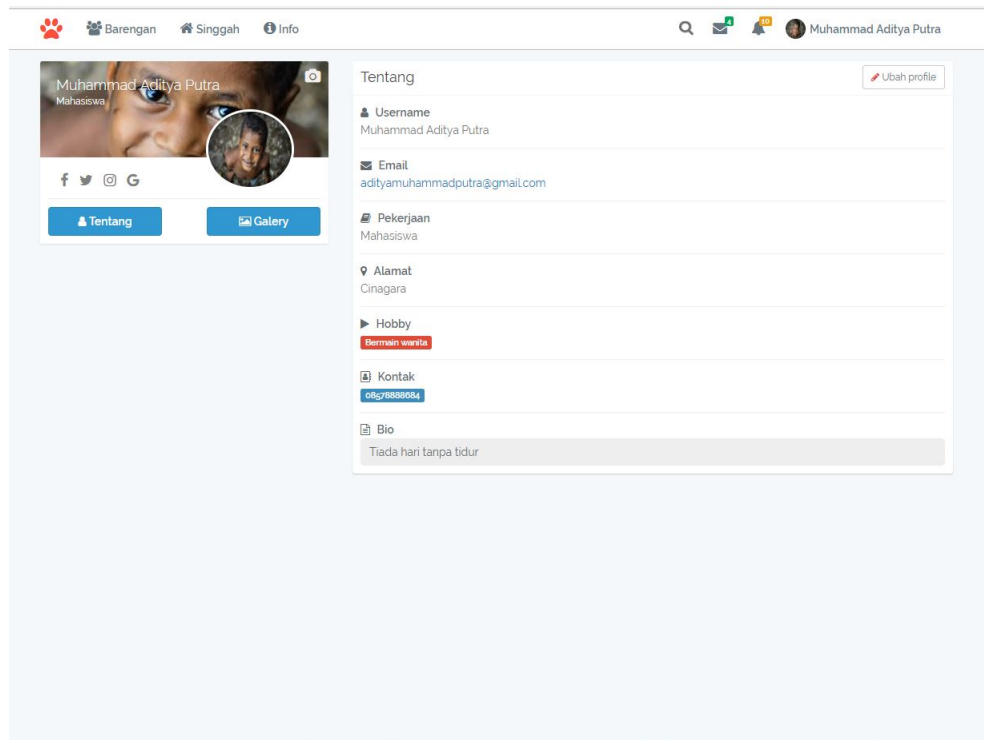
22 responden menyatakan bahwa sistem yang telah dibuat mudah digunakan, 23 responden menyatakan sistem dibuat sudah sesuai dengan kebutuhan *traveller*, 22 responden menyatakan *design* sistem atau tampilan yang dibuat cukup menarik, 25 responden menyatakan fasilitas dari masing - masing menu dan fitur yang terdapat dalam sistem memenuhi kebutuhan informasi dan 24 responden menyatakan terbantu dengan adanya sistem ini.

4.6 Pengujian Fungsi Mobilitas

Pengujian fungsi mobilitas dibuat untuk memastikan sistem berjalan pada beberapa aplikasi browser komputer ataupun mobile. Dibawah ini pengujian sistem dengan aplikasi browser komputer atau laptop dan mobile phone:

- Pada Komputer atau Laptop

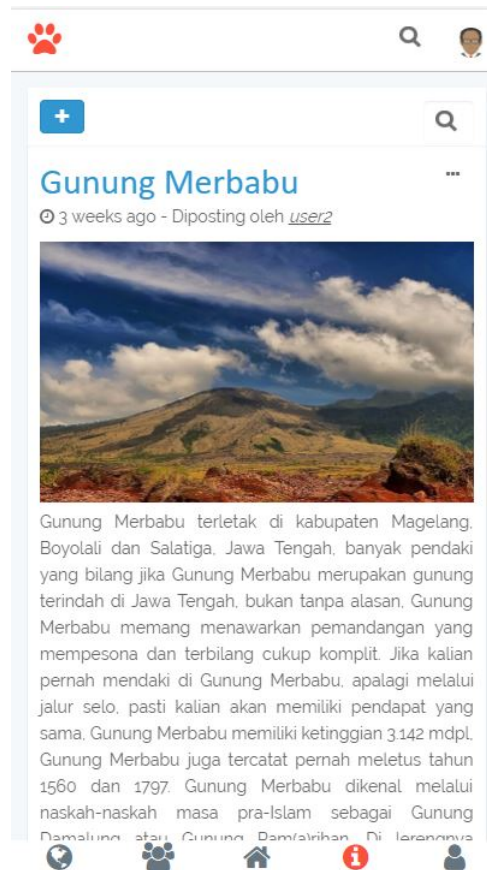
Berikut dibawah ini hasil pengujian sistem dengan aplikasi browser yang ada pada komputer atau laptop dengan menggunakan aplikasi browser Google Chrome



Gambar 4.15: Pengujian dengan Google Chrome Pada Komputer atau Laptop

- Pada Mobile Phone Atau Handphone

Berikut dibawah ini hasil pengujian sistem pada Mobile Phone atau Handphone



Gambar 4.16: Pengujian dengan Mobile Phone atau Handphone

Dengan demikian dapat disimpulkan bahwa sistem website ini dapat berjalan di berbagai platform dengan responsive, baik itu dengan menggunakan komputer desktop atau laptop dan mobile phone atau handphone

Bab 5

Penutup

5.1 Kesimpulan

Sistem rekomendasi *web mobile traveller* menggunakan metode *collaborative filtering* dan *framework* laravel berhasil dibuat dan selesai diuji coba. Diharapkan dengan adanya sistem ini dapat membantu dan memfasilirasi *traveller* dalam menyiapkan kebutuhan rekomendasi informasi sebelum melakukan perjalanan.

Berdasarkan hasil yang didapat dari penelitian yang dilakukan dalam penyusunan skripsi ini, kesimpulan yang diperoleh adalah sistem ini dapat menampilkan rekomendasi informasi bersarkan metode *collaborative filtering* seperti informasi tempat wisata, informasi rumah singgah dan informasi untuk mencari teman perjalanan.

5.2 Saran

Sistem rekomendasi *web mobile traveller* menggunakan metode *collaborative filtering* dan *framework* laravel masih jauh dari kata sempurna. Agar sistem yang dibuat dapat berjalan lebih baik lagi, peneliti memberikan beberapa saran seperti perlunya layanan yang bisa menghubungkan sistem dengan akun sosial media lainnya seperti facebook, intagram dan twittter, perlunya *userguide* yang dapat di unduh pada halaman website agar dapat digunakan user sebagai petunjuk penggunaan website, perlunya peningkatan untuk keamanan akun pengguna agar sistem yang dibuat tidak rentan terkena serangan *hacking*, perbaikan untuk beberapa tampilan yang rasa belum

responsive dengan *mobile* dan sebaiknya ditambah beberapa fitur baru secara berkala agar *traveller* tidak merasa bosan dalam menggunakan sistem yang telah dibuat.

DAFTAR PUSTAKA

Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl. J, (2001), *Analysis Of Recommendation Algorithms For E-Commerce*. In Proceedings of the ACM EC'00 Conference, Minneapolis, MN. Pp. 158-167.

Braun D., Sivils J., Shapiro A., Versteegh J, (2001), *Object Oriented Analysis and Design Team*. Kennesaw State University CSIS, Spring, 4650.

Schafer, J.B., Frankowski, D., Herlocker, J. dan Sen, S, (2007), *Collaborative Filtering Recommender System*, Springer-Verlag, Berlin, Heidelberg.

Abdul Kadir, (2008), *Dasar Pemrograman Web Dinamis dengan menggunakan PHP(Revisi)*, ANDI, Yogyakarta.

Su, X. & hoshgoftaar, T.M, (2009), *A survey of collaborative filtering techniques*, Advances in artificial intelligence, 4.

Arief, M.Rudianto, (2011), *Pemrograman Web Dinamis Menggunakan Php dan Mysql*, Yogyakarta: ANDI.

Danis Puntodi, (2011), *Menciptakan Penjualan via Social Media*, Jakarta: Elex Media Komputindo.

Ricci, Francesco, Lior Rokach, dan Bracha Shapira, (2011), *Introduction to recommender systems handbook*, Recommender systems handbook : 1-35.

Betha Sidik, (2012), *Pemrograman Web dengan PHP*, Bandung: Informatika.

Alatas, Husein.,(2013), *Responsive Web Design dengan PHP & Bootstrap*, Yogyakarta : Lokomedia.

Norman B. Anderson, et al. (2013), *Stress in America Missing the Health Care Connection*, American Psychological Association

Tang, J., Hu, X. dan Liu, H., (2013), *Social recommendation: a review. Social Network Analysis and Mining*, 3(4), 1113-1133.

Umka Web, (2013), *Pengertian Laravel*, diambil dari: <http://emka.web.id/tutorial/tutorial-laravel/2013/tutorial-laravel-1-apa-itu-laravel/>, diakses 06 Juni 2018.

Sugiyono, (2014), *Metode Penelitian Pendidikan Pendekatan Kuantitatif, Kualitatif, dan R&D*, Bandung: Alfabeta.

Dadan Hamdani, (2015), *Belajar Laravel Untuk Pemula*, Jakarta, Tanpa Penerbit.

Lord Voldet, (2015), *Visual Studio Code, Editor Baru dari Microsoft Untuk Windows, OS X, dan Linux*, diambil dari: <https://www.codepolitan.com/visual-studio-code-editor-baru-dari-microsoft-untuk-windows-os-x-dan-linux>, diakses pada : 19 Mei 2018.

Reservasi.com, (2015), *71 Persen Traveler Gunakan Sumber Informasi Wisata Secara Online*, diambil dari: <http://blog.reservasi.com/mengejutkan-ternyata-71-traveler-gunakan-sumber-informasi-wisata-secara-online/>, diakses 21 Desember 2017.

Priyanto Hidayatullah dan Jauhari Khairul Kawistara, (2015), *Pemrograman Web(HTML/CSS/JavaScript/Power Designer/XAMPP/PHP/CodeIgniter/Jquery)*, Bandung: Informatika.

Assaf Arief, (2016), *Rancang Bangun Sistem Rekomendasi Pariwisata Mobile Advertising Menggunakan Metode Hybrid Filtering Sebagai Pemberdayaan Masyarakat Usaha Kecil Menengah (UKM) di Pulau*

Ternate, Ternate : Jurnal PROtek, Vol 03 No. 1.

Intan Melia, Wina Witanti dan Faiza Renaldi, (2016), ***Perancangan Sistem Rekomendasi Pendistribusian Kaos Pada Industri Pakaian Jadi Menggunakan Item Based Collaborative Filtering***. Bandung: Selisik, ISSN: 2503- 2844.

Bambang Winarso, (2016), ***Apa Itu Google Chrome dan Sepenggal Sejarahnya***, diambil dari
: <https://dailysocial.id/post/apa-itu-google-chrome>, diakses 20 Desember 2017.

Dudung, (2016), ***Pengertian, Komponen Dan Fungsi XAMPP Lengkap Dengan Penjelasannya***, diambil dari:
<http://www.dosenpendidikan.com/pengertian-komponen-dan-fungsi-xampp-lengkap-dengan-penjasannya/>, diakses pada 01 Agustus 2018.

Haiyang Zhang, et al, (2017), ***A trust-enriched approach for item-based collaborative filtering recommendations***, IEEE

Septian Eka Anggraeni, Irfan Darmawan dan Nia Ambarsari, (2017), ***Membangun Sistem Rekomendasi Dalam Situs Lest-Travels Berbasis Collaborative Filtering Menggunakan Metode Waterfall*** . Bandung, ISSN : 2355- 9365.

Dedy Hidayat Kusuma dan Moh. Nur Shodiq, (2017), ***Sistem Rekomendasi Destinasi Pariwisata Menggunakan Metode Hibrid Case Based Reasoning dan Location Based Service Sebagai Pemandu Wisatawan di Banyuwangi***, Banyuwangi : Jurnal INTENSIF. Vol.1, No.1 ISSN: 2549-6824

idprogrammer.com, (2017), ***Pengertian Mobile Web dan Mobile Aplikasi***, diambil dari:
<https://idprogrammer.com/pengertian-mobile-web-dan-mobile-aplikasi/>, diakses 14 Desember 2017.

Mila Sari, (2017), ***Survei Travelport: Indonesia Peringkat 3 untuk Wisatawan Digital Dunia***, diambil dari:

<https://ekonomi.kompas.com/read/2017/11/14/200000726/survei-travelport--indonesia-peringkat-3-untuk-wisatawan-digital-dunia>, diakses 14 maret 2018.

Dwiky Andika, (2017), *Pengertian Flowchart*, diambil dari: <https://www.it-jurnal.com/pengertian-flowchart/>, dikases pada 20 Agustus 2017.

Elfaviona.com, (2017), *Persiapan Traveling Yang Wajib Diperhatikan Saat Keluar Kota*. diambil dari: <http://www.elfaviona.com/persiapan-traveling-ke-luar-kota.html>, diakses 26 Desember 2017.

Bab.la, (Tanpa Tahun), *Traveller Bahasa Indonesia Terjemahan*, diambil dari: <https://www.babla.co.id/bahasa-inggris-bahasa-indonesia/traveller>, diakses 20 Desember 2017.

artikelsiana.com, (2017), *Pengertian Media Sosial, Fungsi, Karakteristik, Jenis & Dampak Media Sosial*, diambil dari: <pengertian-media-sosial-fungsi.html>, diakses 28 Agustus 2018.

nanda samsu dhuha, (2017), *Apa yang dimaksud dengan Pengujian perangkat lunak atau Software Testing*, diambil dari : <https://www.dictio.id/t/apa-yang-dimaksud-dengan-pengujian-perangkat-lunak-atau-software-testing/15136/2>, diakses pada: 15 Agustus 2018.

LAMPIRAN 1

- Script Routing Laravel

```
Auth::routes();
Route::get('auth/activate', 'Auth\ActivationController@activate');
Route::get('auth/activate/resend', 'Auth\
    ActivationResendController@showResendForm')->;
Route::post('auth/activate/resend', 'Auth\ActivationResendController@resend');
Route::get('/', 'HomeController@index');
Route::get('/beranda', function () { return view('layouts.beranda'); });
Route::resource('/home/{id}/comment', 'HomeCommentController')->only(['store'])
    ;
Route::resource('/homecomment', 'HomeCommentController')->only(['destroy']);
Route::resource('/caribarengan', 'CariBarenganController')->except(['show', '
    create']);
Route::resource('/caribarengan/{id}/comment', 'CariBarenganCommentController')
    ->only(['store'])
Route::resource('/caribarengancomment', 'CariBarenganCommentController')->only
    (['destroy'])
Route::resource('/caribarengan/gabung', 'CariBarenganGabungController')->only
    (['store', 'show']);
Route::resource('/singgah', 'SinggahController')->except(['create']);
Route::resource('/singgah/{id}/comment', 'SinggahCommentController')->only(['
    store']);
Route::resource('/singgah/like', 'SinggahLikeController')->only(['store', 'show
    ']);
Route::resource('/singgahcomment', 'SinggahCommentController')->only(['destroy'
    ]);
Route::resource('/info', 'InfoController')->except(['create']);
Route::resource('/info/{id}/comment', 'InfoCommentController')->only(['store'])
    ;
Route::resource('/infocomment', 'InfoCommentController')->only(['destroy']);
Route::resource('/info/like', 'InfoLikeController')->only(['store', 'show']);
Route::get('/tentang', 'TentangController@index');
Route::post('tentang', 'TentangController@update');
```

- Script Koneksi Database dan Environment

```
APP_NAME=HaruniSkripsi
APP_ENV=local
APP_KEY=base64: cc2VamFx++d7Zw8XbDUhYNs+deUbS34aELdZx0wDCfE=
APP_DEBUG=true
APP_LOG_LEVEL=debug
```

```

APP_URL=http://localhost:8000
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=skripsi D
B_USERNAME=root
DB_PASSWORD=

MAIL_DRIVER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=8aa6d7eff2f2d4
MAIL_PASSWORD=3114df3d988014
MAIL_ENCRYPTION=null

FILESYSTEM_DRIVER=public

```

• Script Master Menu

```

<div class="collapse navbar-collapse pull-left f-color1" style="display: block
!important;">
    <div id="sidebar-menu">
        <ul class="f-color1 nav navbar-nav active-menu hidden-xs">
            <li><a href="{{route('caribarengan.index')}}"><i class="fa
fa-users fa-size-15"></i> Barengan</a></li>
            <li><a href="{{ route('singgah.index') }}"><i class="fa fa-home fa-
size-15"></i> Singgah</a></li>
            <li><a href="{{ route('info.index') }}"><i class="fa fa-info
-circle fa-size-15"></i> Info</a></li>
        </ul>
        <nav class="icon-bar nav active-menu visible-xs bottom" >
            <li><a href="{{route('home')}}"><i class="fa fa-globe"></i></a></
li>
            <li><a href="{{route('caribarengan.index')}}"><i class="fa fa-users
"></i></a> </li>
            <li><a href="{{ route('singgah.index') }}"><i class="fa fa-home"></i
></a> </li>
            <li><a href="{{ route('info.index') }}"><i class="fa fa-info-circle
"></i></a> </li>
            <li><a href="#"><i class="fa fa-user"></i></a> </li>
        </nav>
    </div>
</div>

```

• Script Halaman Beranda, Tenang, Login dan Bergabung

```

<section id="intro" class="intro-section">
    <div class="container">
        <div class="row">
            <div class="col-lg-3 col-md-3 col-sm-4 col-xs-6">
                <p class="p3">Dunia <a class="p1">Seluas</a></p><br>
                <p class="p4">Kaki <a class="p1">Melangkah</a></p>
            </div>
        </div>
    </div>

```

```

</div>
</section>
<section id="about" class="about-section">
  <div class="container">
    <div class="row">
      <div class="col-lg-12">
        <h2 class="p5">Tentang</h2>
        <hr class="hr">
        <div class="row">
          <div class="col-md-4 col-sm-12">
            <div class="row">
              <div class="col-md-12 col-xs-12">
                <p class="icon-custom"><i class="fa fa-info-circle"></i></p>
              </div>
            </div>
          </div>
          <div class="row">
            <div class="col-md-12 col-sm-12 col-xs-12">
              <h3 class="textabout"><p class="p7 text-center">Info Wisata</p>
                Kadang anda bingung ingin berlibur kemana? mungkin Rekomendasi
                Info Wisata disini bisa anda jadikan pertimbangan. Anda juga
                bisa memberi ulasan atau menulis cerita anda sendiri yang bisa
                jadi referensi untuk yang lain </h3>
              </div>
            </div>
          </div>
          <div class="col-md-4 col-sm-12">
            <div class="row">
              <div class="col-md-12 col-xs-12">
                <p class="icon-custom"><i class="fa fa-home"></i></p>
              </div>
            </div>
          </div>
          <div class="row">
            <div class="col-md-12 col-sm-12 col-xs-12">
              <h3 class="textabout">
                <p class="p7 text-center">Rumah Singgah</p>
                Bila anda sedang melakukan perjalanan jauh dan anda butuh rumah
                singgah, Coba cari disini ! Selain itu anda juga bisa menjadi
                penyedia rumah singgah dan berbagi kehangatan rumah anda dengan
                teman baru anda
              </h3>
            </div>
          </div>
          <div class="col-md-4 col-sm-12">
            <div class="row">
              <div class="col-md-12 col-xs-12">
                <p class="icon-custom"><i class="fa fa-users"></i></p>
              </div>
            </div>
          </div>
          <div class="row">
            <div class="col-md-12 col-sm-12 col-xs-12">
              <h3 class="textabout"><p class="p7 text-center">Cari Barengan</p>
                Pernah
                suatu hari anda saat anda ingin melakukan pertualangan hebat tetapi
                membutuhkan teman agar tidak membosankan ? Coba cari barengan
                perjalanan anda disini. Anda akan mendapatkan teman baru disini</h3>
            </div>
          </div>
        </div>
      </div>
    </div>
  </section>

```

```

        </div>
    </div>
</div>

</div>

</div>
</div>
</div>
</section>

<form class="form-horizontal col-md-8 col-md-offset-2" method="POST" action
    = "{{ route('login') }}" >
    {{ csrf_field() }}
    <div class="form-group{{ $errors->has('email') ? ' has-error' : '' }}">
        <div class="styled-input">
            <input type="email" class="input" id="email" name="email" value="{{ old('
                email') }}" required autofocus/>
            <label>Email</label>
            <span></span>
            @if ($errors->has('email'))
                <span class="help-block">
                    <strong>{{ $errors->first('email') }}</strong>
                </span>
            @endif
        </div>
    </div>
    <div class="form-group{{ $errors->has('password') ? ' has-error' : '' }}">
        <div class="styled-input">
            <input type="password" class="input" id="password" name="password" value
                = "{{ old('password') }}" required />
            <label>Password</label>
            <span></span>
            @if ($errors->has('password'))
                <span class="help-block">
                    <strong>{{ $errors->first('password') }}</strong>
                </span>
            @endif
        </div>
    </div>
    <div class="form-group">
        <div class="">
            <div class="checkbox">
                <label>
                    <input type="checkbox" name="remember" {{ old('remember') ? 'checked' :
                        '' }}> Biarkan saya tetap masuk
                </label>
            </div>
        </div>
    </div>
    <div class="form-group">
        <div class="">
            <button type="submit" class="btn btn-primary">Login</button>
            <a class="btn btn-link pull-right" href="{{ route('password.request') }}">
                Lupa Password ? </a>
        </div>
    </div>

```



```

    </div>
</form>

<form class="form-horizontal" method="POST" action="{ route('register') }">
    {{ csrf_field() }}
    <div class="col-md-8 col-md-offset-2">
        <div class="form-group{{ $errors->has('name') ? ' has-error' : '' }}">
            <div class="styled-input">
                <input type="text" class="input" id="name" name="name" value="{ old('name') }" required autofocus/>
                <label>Nama</label>
                <span></span>
                @if ($errors->has('name'))
                    <span class="help-block">
                        <strong>{{ $errors->first('name') }}</strong>
                    </span>
                @endif
            </div>
            <div class="form-group{{ $errors->has('email') ? ' has-error' : '' }}">
                <div class="styled-input">
                    <input type="email" class="input" id="email" name="email" value="{ old('email') }" required autofocus/>
                    <label>Email</label>
                    <span></span>
                    @if ($errors->has('email'))
                        <span class="help-block">
                            <strong>{{ $errors->first('email') }}</strong>
                        </span>
                    @endif
                </div>
            </div>
            <div class="form-group{{ $errors->has('password') ? ' has-error' : '' }}">
                <div class="styled-input">
                    <input type="password" class="input" id="password" name="password" value="{ old('password') }" required />
                    <label>Password</label>
                    <span></span>
                    @if ($errors->has('password'))
                        <span class="help-block">
                            <strong>{{ $errors->first('password') }}</strong>
                        </span>
                    @endif
                </div>
            </div>
            <div class="form-group{{ $errors->has('password') ? ' has-error' : '' }}">
                <div class="styled-input">
                    <input type="password" class="input" id="password-confirm" name="password_confirmation" required />
                    <label>Konfirmasi Password</label>
                    <span></span>
                </div>
            </div>
            <div class="form-group">
                <div class="">
                    <button type="submit" class="btn btn-primary">
                        Bergabung
                    </button>
                </div>
            </div>
        </div>
    </form>

```

```

        </button>
    </div>
</div>
</div>
</form>

```

• Script Controller Beranda, Tentang, Login dan Bergabung

```

//file ControllerBeranda.php
Route::get('/beranda', function () {
    return view('layouts.beranda'); }
);

//file TentangController.php
class TentangController extends Controller {
    public function __construct(){
        $this->middleware('auth');
    }
    public function index(){
        return view('tentang.tentang');
    }
    public function update(Request $request){
        if ($request->file('avatar')) {
            if ($request->user()->avatar){
                Storage::delete($request->user()->avatar);
            }
            $filename = 'avatars/'.str_slug($request->user()->email, '') . '.' .
                $request->avatar->getClientOriginalExtension();
            $pathfoto = Image::make($request->file('avatar'))->resize(200,200)->save(
                storage_path('/app/public/') . $filename);
            $pathfoto;
            $request->user()->update(['avatar' => $filename]);
        }
        else {
            $request->user()->update([
                'name' => request('name'),
                'pekerjaan' => request('pekerjaan'),
                'agama' => request('agama'),
                'hobby' => request('hobby'),
                'bio' => request('bio'),
                'contact' => request('contact'),
                'alamat' => request('alamat')]);
        }
        return redirect()->back()->withInfo('Data anda berhasil di perbaharui ');
    }
    public function edit(Post $id){
        $datakategori = Category::all();
        return view('post.edit', compact('id', 'datakategori'));
    }
    public function getUserImage($filename){
        $myFile = Storage::disk('public')->get($filename);
        return view('layouts.master', compact('myFile'));
    }
}

```

```
//file LoginController.php
class LoginController extends Controller {
    use AuthenticatesUsers;
    protected $redirectTo = '/';
    public function __construct(){
        $this->middleware('guest')->except('logout');
    }
    protected function validateLogin(Request $request){
        $this->validate($request,[
            $this->username()=>['required','string',Rule::exists('users')->where(
                function($query){
                    $query->where('active',true);})],
            'password' =>'required|string',[], $this->validationError());}
    public function validationError(){
        return[$this->username().'exists' => 'Silahkan cek kembali akun anda atau
            Email anda mungkin belum diaktivasi'];
    }
}

//file RegisterController.php

class RegisterController extends Controller {
    use RegistersUsers;
    protected $redirectTo = '/';
    public function __construct(){
        $this->middleware('guest');}
    protected function validator(array $data){
        return Validator::make($data,['name' => 'required|string|max:255','email'=>
            'required|string|email|max:255|unique:users','password' => 'required|string|
            min:6|confirmed',]);
    }
    protected function create(array $data)
    {
        return User::create(['name'=>$data['name'],
            'email' => $data['email'],'password'=>bcrypt($data['password']),'active'=>
            false,'activation_token' => str_random(190)]);}
    protected function registered(Request $request, $user){
        event(new UserActivationEmail($user));
        $this->guard()->logout();
        return redirect()->route('login')->withSuccess('Anda berhasil daftar,
            Silahkan cek email untuk aktivasi akun');
    }
}
```

• Script Model User, Beranda, Barengan, Singgah dan Info

```
//file user.php

class User extends Authenticatable {
    use Notifiable;
    protected $fillable =
        ['name', 'email', 'password','active', 'activation_token', 'status', 'avatar
            ', 'pekerjaan', 'agama', 'hobby', 'bio',
            'contact','alamat'];
    protected $hidden = [ 'password', 'remember_token', ];
}
```

```

        // public function scopeByActivationColumns(BUILDER $builder, $email,
        $token){ // return $builder->where('email', $email)->where('
        activation_token', $request->token) // } public function user
        () { return $this->belongsTo(User::class); }
    }

//file post.php

class Post extends Model {
protected $fillable = ['user_id','content'];
public function user(){
return $this->belongsTo(User::class);}
public function comments(){
return $this->hasMany(PostComment::class);}
public function likes(){
return $this->hasMany(PostLikes::class);}
public function scopeFiltered($query){
$query->when(request('q'),function($query){$query->where(function($query){
$params='%'.request('q').'%';
$query->where('id', 'like', $param)->orWhere('content', 'like', $param);
});
});
}

//file barengan.php

class Barengan extends Model {;
protected $fillable = ['user_id','content','tujuan','mepo','mulai','akhir',
,'contact'];
public function user(){return $this->belongsTo(User::class);}
public function barengancomments(){return $this->hasMany(BarenganComment::
class);}
public function barengangabung() {return $this->hasMany(BarenganGabung::
class);}
public function barengangabungcek(){return $this->hasMany(BarenganGabung::
class)->where('user_id',auth()->id());}
public function calendar(){return $this->belongsTo(Calendar::class);}
public function getMulaiAttribute($value){return $this->attributes['mulai
'] = (new Carbon($value))->format('d M Y');}
public function getAkhirAttribute($value){return $this->attributes['akhir
'] = (new Carbon($value))->format('d M Y');}
public function scopeFiltered($query){$query->when(request('q'), function
($query){$query->where(function
($query){$param='%'.request('q').'%';$query->where('id','like', $param)
->orWhere('tujuan', 'like', $param)->orWhere('mepo', 'like', $param);
});
});
} }

//file post.php

class Post extends Model {
protected $fillable = ['user_id','content'];
public function user(){
return $this->belongsTo(User::class); }
public function comments(){return $this->hasMany(PostComment::class);}
public function likes()

```

```

{
    return $this->hasMany(PostLikes::class);}
public function scopeFiltered($query){
    $query->when(request('q'), function ($query) {$query->where(function ($query
    ) {$param =
    '%'.request('q').'%';$query->where('id', 'like', $param)->orWhere('content',
    'like', $param);
});
});
}

//file singgah.php

public function getRekomendasiAttribute(){
    $rating=$this->singgahcomment->
    pluck('perhitungan')->sum();
    $user = User::all()->pluck('id')->count();$ratinguserlain = $user - 1;
    return $rating/$ratinguserlain;
}
public function scopeFiltered($query){
    $query->when(request('q'), function (
    $query){
    $query->where(function ($query) {$param = '%' . request('q') . '%';
    $query->where('id', 'like', $param)->orWhere('lokasi', 'like', $param)
    ->orWhere('category', 'like', $param)->orWhere('content', 'like', $param);
});
});
}

//file Info.php
class Info extends Model {
    protected $fillable = ['user_id','category_id','title','content','images'];
    protected $appends = ['icon','rekomendasi'];
    public function user() {
        return $this->belongsTo(User::class);
    }
    public function category() {
        return $this->belongsTo(Category::class);
    }
    public function infocomment(){
        return $this->hasMany(InfoComment::class);
    }
    public function infolike(){
        return $this->hasMany(InfoLikes::class);
    }
    public function likecek(){
        return $this->hasMany(InfoLikes::class)->where('user_id',auth()->id());
    }
    public function getRekomendasiAttribute(){
        $rating = $this->infocomment->pluck('perhitungan')->sum();
        $user = User::all()->pluck('id')->count();
        $ratinguserlain = $user - 1;return $rating / $ratinguserlain;
    }
    public function getIconAttribute($value){
        $category = $this->attributes['category_id'];
    }
}

```

• Script Halaman Home

```
//file Welcome.blade.php

<div class = "row">
    <div class="hidden-xs hidden-sm">
        @include('layouts.profile')
    </div>
    <div class="col-md-8"><div class="box"><div class="box-header with-border">
        <h3 class="box-title">Beranda</h3><div class="box-tools pull-right">
            <form method="get" class="custom-search navbar-form navbar-left">
                <input class="form-control" type="text" id="q" name="q" value="{{
                    request()->get('q') }}"placeholder="Cari Kiriman">
            </form>
        </div>
    </div>
    <div id="modal-form" style="padding:10px;"><form method="POST">
        {{csrf_field() }}{{ method_field('POST') }}
        <input type="hidden" name="id" id="id"><div class="form-group has-feedback {{
            $errors->has('content') ? 'has-error' : '' }}">
            <textarea name
                ="content" id="content" class="form-control" onclick="addForm()"></
                textarea>@if($errors->has('content'))
            <span class="help-block"><p>{{ $errors->first('content') }}
        </p>
    </span> @endif </div> <div class="form-group">
        <button type="submit" class="btn btn-primary"><i class="fa fa-chevron-circle-
            right"></i> Kirim</button>
    </div>
</form> </div>
<!-- /.box-header -->
<div id="contact-table" class="infinite-scroll">
    @if($datapost->count() > 0)
    @foreach($datapost as $d)
    <div class="box box2">
        <div class="box-body box-body-custom">
            <div class="post"><div class="user-block">
                
                <span class="username"><!-- $email = Crypt::encrypt($d->user->email)-->
                <a href="{{ action('ProfileController@show', $d->user->email) }}">{{ $d->
                    user->name }} </a>
                <div class="btn-group pull-right custom-curret nav-right1"><button type="
                    button"
class="btn btn-primary-outline dropdown-toggle" data-toggle="dropdown" aria-
    haspopup="true" aria-expanded="false">
    <span class="fa fa-ellipsis-h"></span>
    </button>
    <ul class="dropdown-menu">@if(Auth::user()->name==$d->user->name )
    <li><a onclick="editForm('{{ $d->id }}')" class="pointer-jempol"> <i class="fa
        fa-pencil-square-o"></i> Edit Kiriman</a></li>
        <li><a onclick="deleteData('{{ $d->id }}')"
class="pointer-jempol"> <i class="fa fa-times-circle"></i> Hapus Kiriman
    </a></li>
        <li role="separator" class="divider"
"></li> @endif <li><a href="#" class="pointer-jempol"><i class="fa fa-
    exclamation-circle">
    </i> Laporkan Kiriman</a></li></ul></div>
    </span>
```

```

    <span class="description"><i class="fa fa-clock-o"></i> {{ $d->created_at->
        diffForHumans() }}</span>
</div><p>
{{ $d->content }}
</p>
    <div class="box box-default box-costum-collapse"><div class="box-header with
        -border" style="padding:0px;">
        <table class="pull-right box-hp"><tr>
            <td class="mailbox-star" data-value="{{ $d->id }}"><i class="fa fa-star text-red"></i>
            <i class="fa fa-star"></i>
        @endif
    @endif
    </td>
    <td class="btn-nopadding btn btn-box-tool"><a onclick="showlike('{{ $d->id }}'
        ">{{ $d->likes->count() }}
    Menyukai</a> </td>
    @include('layouts.form.formLike')
    <td class="btn-nopadding btn btn-box-tool" data-widget="collapse"> |
    <i class="fa fa-comment"></i>
    {{ $d->comments->count() }} Komentar</td>
</tr>
</table>
</div>
<div class="box-body" style="padding:0px;">
    <div class="box-komentar">
    @include('layouts.form.formComment')
    <div id="box-komentar">
    @foreach($d->comments as $c)
    <div class="komentar-post">
    <div class="user-block">
    
    <span class="username usernamekomentar">
    <a href="{{ action('ProfileController@show', $c->user->email) }}">{{ $c->
        user->name }} </a>
    <div class="btn-group custom-curret nav-right-koment pull-right">
    <button type="button" class="btn btn-primary-outline dropdown-toggle" data-
        toggle="dropdown"
    aria-haspopup="true" aria-expanded="false">
    <span class="fa fa-ellipsis-h"></span></button>
    <ul class="dropdown-menu">
    @if(Auth::user()->name == $c->user->name )
    <li><a onclick="deleteComment('{{ $c->id }}')" class="pointer-jempol"><i
        class="fa fa-times-circle">
    </i> Hapus Komentar</a></li>
    <li role="separator" class="divider"></li>
    @endif
    <li><a href="#" class="pointer-jempol"><i class="fa fa-exclamation-circle
        "></i>
    Laporkan komentar</a></li>
    </ul>
    </div>
    </span>
    <span class="description descriptionkomentar"><i class="fa fa-clock-o"></i>
        {{ $c->created_at->diffForHumans() }}</span>
    </div>

```

```

        <p>{{ $c->comment }}</p>
    </div>
    @endforeach
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
    @endforeach
    @else
<div class="box box2"> <div class="box-body box-body-custom text-center">
    <h3><i class="fa fa-warning text-info"></i> Data yang anda cari tidak ada
    <i class="fa fa-warning text-info"></i></h3>
    <p>Silahkan masukan kata kunci lain</p>
</div>
</div>
</div>
    @endif
    {{ $datapost->links() }}
</div>
    <!-- /.box-body -->
</div>
</div>
</div>

```

• Script Controller Home

```

//file HomeController.php
class HomeController extends Controller {
    public function __construct(){
        $this->middleware('auth');
    }
    public function index(){
        $datapost = Post::with('likes')->latest()->filtered()->paginate(10);
        $datapost;
        return view('welcome', compact('datapost'));
    }
    public function store(Request $request){
        $data = ['user_id' =>auth()->id(), 'content' => $request['content']];
        Post::create($data);
        Session::flash('success', 'Kiriman Ditambah'); return View::make('layouts/
        partials/_alertajax');
    }
    public function edit($id){
        $data = Post::find($id); return $data;
    }
    public function update(Request $request, $id){
        $data = Post::find($id); $data->content = $request['content'];
        $data->update();
        Session::flash('info', 'Kiriman Rubah');
        return View::make('layouts/partial/_alertajax');
    }
    public function destroy($id){
        Post::destroy($id);
    }
}

```



```

    Session::flash('error', 'Kiriman Dihapus');
    return View::make('layouts/partials/_alertajax');
}
}

```

• Script Halaman Barengan

```

<div class="modal fade" id="modal-form" tabindex="-1" role="dialog" aria-
    labelledby="exampleModalLabel">
<div class="modal-dialog" role="document">
<div class="modal-content">
    <form method="post" data-toogle="validator" class="form-horizontal" id="form
        " enctype="multipart/form-data">
    {{csrf_field()}}{{method_field('POST')}}
    <div class="modal-header">
    <button type="button" class="close" data-dismiss="modal" aria-label="Close
        ">
        <span aria-hidden="true">&times;</span>
    </button>
    <h4 class="modal-title warna" id="modal-title"></h4>
    </div>
<div class="modal-body">
<input type="hidden" name="id" id="id">
<div class="form-group">
    <div class="styled-input">
        <input type="text" class="input" name="tujuan" id="tujuan" required/>
        <label>Tempat Tujuan</label>
        <span></span>
    </div>
</div>
<div class="form-group">
    <div class="styled-input">
        <input type="text" class="input" name="mepo" id="mepo" required/>
        <label>Tempat Meeting Point</label>
        <span></span>
    </div>
</div>
<div class="form-group">
    <div class="styled-input50">
        <input type="date" class="input" name="mulai" id="mulai" required/>
        <label class="labeldate">Tanggal Mulai</label><span></span>
    </div>
</div>
<div class="form-group">
    <div class="styled-input50">
        <input type="date" class="input" name="akhir" id="akhir" required/>
        <label>Tanggal Selesai</label>
        <span></span>
    </div>
</div>
<div class="form-group">
    <div class="styled-input">
        <input type="text" class="input" name="contact" id="contact" required/>
        <label>Kontak</label>
        <span></span>
    </div>
</div>

```

```

</div>
</div>
<div class="form-group">
  <div class="styled-input wide">
    <textarea required name="content" class="textarea" id="content"></textarea>
    <label>Deskripsi</label>
    <span></span>
  </div>
</div>
  <div class="modal-footer">
    <button type="button" class="btn btn-default btn-save" data-dismiss="
      modal"> <i class="fa fa-times-circle"></i> Batal</button>
    <button type="submit" class="btn btn-primary"><i class="fa fa-chevron-
      circle-right"></i> Kirim</button>
  </div>
</div>
</form>
</div>
</div>
</div>

```

• Script Controller Barengan

```

//file CariBarenganController.php
class CariBarenganController extends Controller {
  public function __construct(){
    $this->middleware('auth');
  }
  public function index(Request $request){
    $barengan=Barengan::with('barengancomments','barengangabungcek')->withCount
      ('barengangabung')
    ->orderBy('barengangabung_count','desc')->filtered()->get();
    $barengan;return view('caribarengan',compact('barengan'));
  }
  public function store(Request $request){
    $data = ['user_id' =>auth()->id(),'tujuan' => $request['tujuan'],'mepo' =>
      $request['mepo'],'mulai' => $request['mulai'],
      'akhir' => $request['akhir'],'contact' => $request['contact
     '],'content' => $request['content']];
    Barengan::create($data);
    Session::flash('success','Kiriman Ditambah');
    return View::make('layouts/partials/_alertajax');
  }
  public function edit($id){
    $caribarengan = Barengan::find($id);
    return $caribarengan;
  }
  public function update(Request $request, $id){
    $caribarengan = Barengan::find($id);
    $caribarengan->tujuan =
    $request['tujuan'];
    $caribarengan->mepo = $request['mepo'];
    $caribarengan->mulai = $request['mulai'];
    $caribarengan->akhir = $request['akhir'];
    $caribarengan->contact = $request['contact'];
  }
}

```

```

$caribarengan->content = $request['content'];
$caribarengan->update();
Session::flash('info', 'Kiriman Rubah');
return View::make('layouts/partial/_alertajax');
}
public function destroy($id){
    Barengan::destroy($id);
    Session::flash('error', 'Kiriman Dihapus');
    return View::make('layouts/partial/_alertajax');
}
}

```

• Script Halaman Singgah

```

//File SinggahBlade.php
<div class="row"> <div class="">
    <div class="col-md-4 hidden-xs hidden-sm">
        <div class="box box-primary">
            <div class="box-header with-border"> <h3 class="box-title">Rumah
                Singgah Anda</h3> </div>
            @if ($singgahsaya->count() > 0)
            @foreach($singgahsaya as $s)
                <div class="box-body"><strong><i class="fa fa-book margin-r-5"></i>
                    Deskripsi</strong>
                <p class="text-muted"> {{ $s->content }}</p>
                <hr>
                <strong><i class="fa fa-map-marker margin-r-5"></i> Lokasi</
                    strong>
                <p class="text-muted">{{ $s->lokasi }}</p>
                <hr>
                <strong><i class="fa fa-star margin-r-5"></i> Ratting Ulasan</
                    strong>
                <p>
                    {!! str_repeat('<i class="fa fa-star text-red" aria-hidden="
                        true"></i>', $s->singgahcomment->pluck('ratting')->avg() )
                        !!}
                    {!! str_repeat('<i class="fa fa-star-o text-red" aria-hidden="
                        true"></i>',
                        5 - $s->singgahcomment->pluck('ratting')->avg() ) !!}{{ $s->
                        singgahcomment->pluck('ratting')->avg() }}
                </p>
                <hr>
                <strong><i class="fa fa-file-text-o margin-r-5"></i> Ulasan</strong>
                @foreach($s->singgahcomment as $sc)<p>
                    {{ $sc->comment }} </p>
                @endforeach </div><!-- /.box-body --></div> @endforeach @else
                <div class="alert alert-info alert-light" role="">
                <i class="icon fa fa-info"></i>
                <button type="button" class="close" data-dismiss="alert" aria-label="Close
                    "></button>
                Anda tidak memiliki kiriman di halaman ini, Silahkan buat kiriman anda
                </div>
            @endif
        </div>
    </div>
</div>

```

```

<div class="col-md-8">
  <div class="box box-atas"> <div class="box-header box1 with-border"><h3 class
    ="box-title">
    <button class="btn btn-primary pull-right" onclick="addForm()"> <i class="fa
      fa-plus"></i> </button>
    </h3> <div class="box-tools pull-right">
    <form method="get" class="custom-search navbar-form navbar-left">
    <input class="form-control" type="text" id="q" name="q" value="{{request()->
      get('q')}}" placeholder="Cari Rumah Singgah">
    </form>
  </div>
</div>
<div>
  <div id="contact-table">
@foreach($singgah as $d) <div class="box box2">
<div class="box-body box-body-custom"> <div class="post"> <div class="user-
  block">
  
  <span class="username"> <?php $email = Crypt::encrypt($d->user->email) ?>
  <a href="{{ action('ProfileController@show', $d->user->email) }}">{{ $d->user
    ->name }} </a>
  <div class="btn-group pull-right custom-curret nav-right1">
    <button type="button" class="btn btn-primary-outline dropdown-toggle"
      data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
  <span class="fa fa-ellipsis-h"></span> </button><ul class="dropdown-menu">
    @if(Auth::user()->name == $d->user->name )
    <li><a onclick="editForm('{{ $d->id }}')" class="pointer-jempol">
    <i class="fa fa-pencil-square-o"></i> Edit Kiriman</a></li>
    <li><a onclick="deleteData('{{ $d->id }}')" class="pointer-jempol">
    <i class="fa fa-times-circle"></i> Hapus Kiriman</a></li>
    <li role="separator" class="divider"></li>@endif <li><a href="#" class="
      pointer-jempol"><i class="fa fa-exclamation-circle">
    </i> Laporkan Kiriman</a></li></ul>
  </div> </span><span class="description">
    <i class="fa fa-hashtag" title="Sebagai"></i>
    {{ $d->category }} - <i class="fa fa-clock-o"></i>
    {{ $d->created_at->diffForHumans() }}</span>
  </div> <p> {{ $d->content }}
    </p>
    <div class="box box-default box-costum-collapse">
    <div class="box-header with-border" style="padding:0px;">
    <a class="label label-primary" title="Kota"><i class="fa fa-
      map-marker"></i> {{ $d->lokasi }}</a>
    <a class="label label-primary" title="Kontak"><i class="fa fa-
      phone-square"></i> {{ $d->contact }}</a>
    <a href="{{ action('SinggahController@show', $d) }}"><table
      class="pull-right">
    <tr>
    <td>
    <!! str_repeat('<i class="fa fa-star text-red" aria-
      hidden="true"></i>',
      $d->singgahcomment->pluck('rating')->avg() ) !!>
    <!! str_repeat('<i class="fa fa-star-o text-red"
      aria-hidden="true"></i>',
      5 - $d->singgahcomment->pluck('rating')->avg() ) !!>
    {{ $d->singgahcomment->pluck('rating')->avg() }}|

```

```
<i class="fa fa-comment"></i>
    {{ $d->singgahcomment->count() }} ulasan</td>
        </tr>
    </table>
</a>
</div>
</div>
</div>
</div>
@endforeach
</div>
</div> @include('layouts.form.formSinggah')
@push('scripts')
```

- Script Controller Singgah

```
//File SinggahController.php

class SinggahController extends Controller {
public function __construct(){
$this->middleware('auth');}
public function index()
{$singgah = Singgah::with
('singgahcomment','singgahlike','likecek')
->withCount('singgahlike')
->filtered()
->get()
->sortByDesc('rekendasi');
$singgahsaya=Singgah::with('singgahcomment','singgahlike','likecek')
->withCount('singgahlike')
->orderBy('singgahlike_count','desc')
->where('user_id',auth()->id())->get();
return view('singgah',compact('singgah','singgahsaya'));
}

public function store(Request $request) {
$data = [ 'user_id' =>auth()->id(),
'category' => $request['category'],
'lokasi' => $request['lokasi'],
'contact' => $request['contact'],
'content' => $request['content'],
];
Singgah::create($data);
Session::flash('success','Kiriman Ditambah');
return View::make('layouts/partials/_alertajax');
public function edit($id) {
$singgah = Singgah::find($id);
return $singgah;}
public function show($id) {
$d = Singgah::find($id);
// $datasaya = Singgah::where('user_id', auth()->id())->get();
$datasaya = Singgah::with('singgahcomment','singgahlike','likecek')
->withCount('singgahlike')
->orderBy('singgahlike_count','desc')
}
```

```

->where('user_id', auth()->id())
->get();
return view('singgah-detail', compact('d', 'datasaya')); public function
    update(Request $request, $id)
{
    $singgah = singgah::find($id);
    $singgah->category =
    $request['category'];
    $singgah->lokasi =
    $request['lokasi'];
    $singgah->contact = $request['contact'];
    $singgah->content = $request['content'];
    $singgah->update();
    Session::flash('info', 'Kiriman Rubah');
    return View::make('layouts/partial/_alertajax');    }
    public function destroy($id){
    Singgah::destroy($id);
    Session::flash('error', 'Kiriman Dihapus');
    return View::make('layouts/partial/_alertajax');    } }

```

• Script Halaman Info

```

//File info.blade.php
<div id="contact-table">
    @foreach($data as $d)
    <div class="box box2">
        <div class="box-body box-body-custom">
            <div class="post">
                <div class="title-info">
                    <a href="{{ action('InfoController@show', $d) }}" class="title">{{ $d
                    ->title}} </a>
                <div class="btn-group pull-right custom-curret nav-right1">
                    <button type="button" class="btn btn-primary-outline dropdown-toggle
                    " data-toggle="dropdown"
                    aria-haspopup="true" aria-expanded="false">
                    <span class="fa fa-ellipsis-h"></span></button>
                <ul class="dropdown-menu">
                    @if(Auth::user()->name == $d->user->name )
                    <li><a onclick="editForm('{{ $d->id }}')" class="pointer-jempol">
                        <i class="fa fa-pencil-square-o"></i> Edit Kiriman</a></li>
                    <li><a onclick="deleteData('{{ $d->id }}')" class="pointer-jempol">
                        <i class="fa fa-times-circle"></i> Hapus Kiriman</a></li>
                    <li role="separator" class="divider"></li>@endif
                    <li><a href="#" class="pointer-jempol"><i class="fa fa-exclamation-
                    circle"></i> Laporkan Kiriman</a></li>
                </ul>
            </div>
        </div>
        <div class="info-by">
            <i class="fa fa-clock-o"></i> {{ $d->created_at->diffForHumans() }} -
            Diposting oleh
            <a href="{{ action('ProfileController@show', $d->user->email) }}">{{
            $d->user->name }}</a>
        </div>
    </div>
</div>
<div class="col-md-6 img-info">
    

```

```
</div>
    <p class="text-justify">
        {{ str_limit(strip_tags($d->content), 800) }}
        @if (strlen(strip_tags($d->content)) > 800)
        <a href="{ { action('InfoController@show', $d) } }" ><i class="readmore">
            Lanjutkan Membaca ...</i></a>
        @endif
    </p>
</div class="box box-default box-costum-collapse">
    <div class="box-header with-border" style="padding:0px;">
        <a class="label label-primary" title="Tempat Meeting Point"><i
            class="fa fa-map-marker"></i>
            {{ $d->category->name }}</a>
        <a href="{ { action('InfoController@show', $d) } }" >
    <table class="pull-right">
    <tr>
        <td>
            {!! str_repeat('<i class="fa fa-star text-red" aria-
                hidden="true"></i>', $d->infocomment->pluck('
                    ratting')->avg() ) !!}
                                {!!str_repeat
                                    ('<i class="fa fa-star-o text-red" aria-hidden="
                                        true"></i>',
                                    5 - $d->infocomment->pluck('ratting')->avg() ) !!}{{ $d
                                        ->infocomment->pluck('ratting')->avg()}}
                                <i class="fa fa-comment"></i> {{ $d->infocomment->
                                    count() }} ulasan</td>
        </td>
    </tr>
    </table>
    </a>
</div>
</div>
</div>
</div>
@endforeach
</div>
```

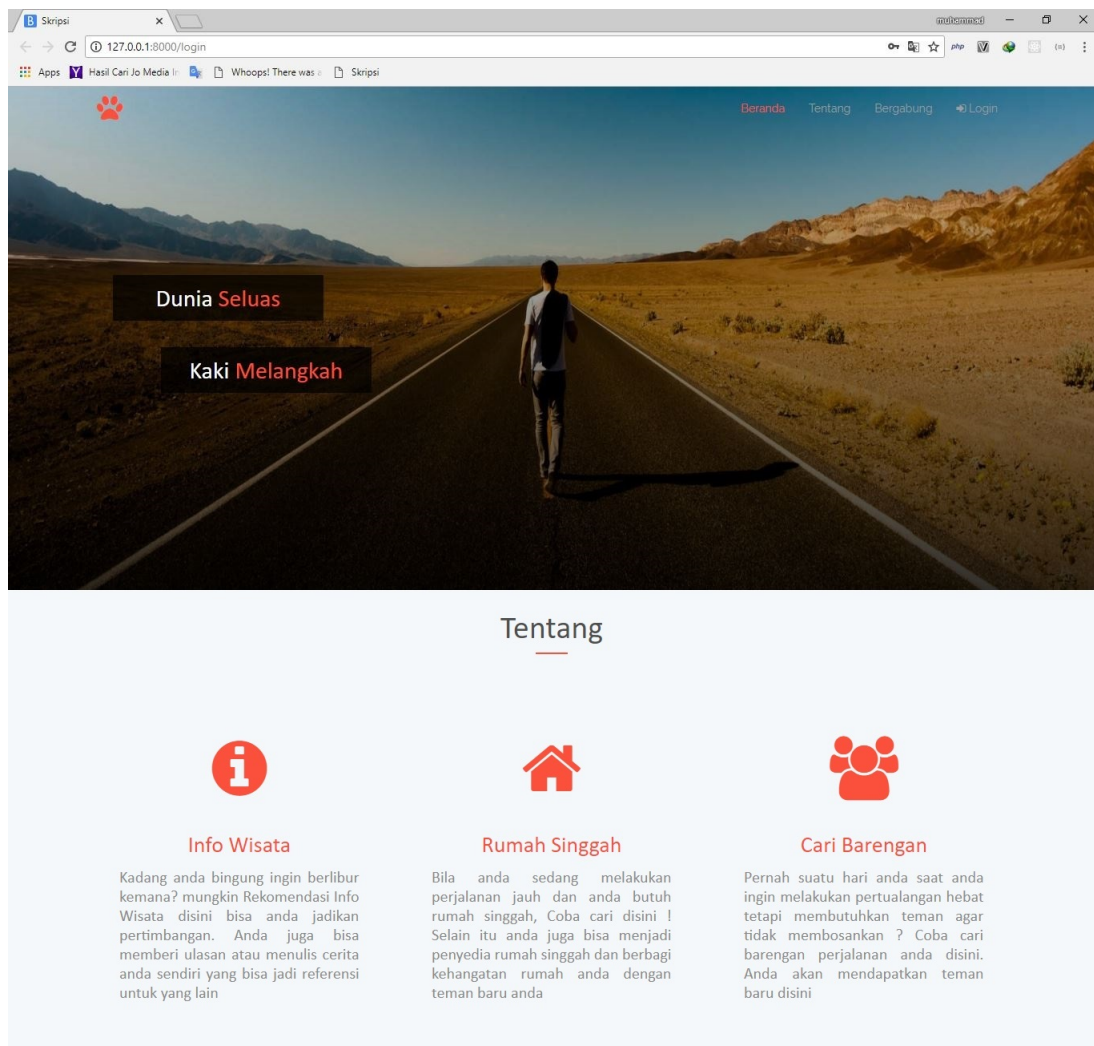
- Script Controller Info

```
//file InfoController.php
class InfoController extends Controller {
    public function __construct(){
        $this->middleware('auth');
    }

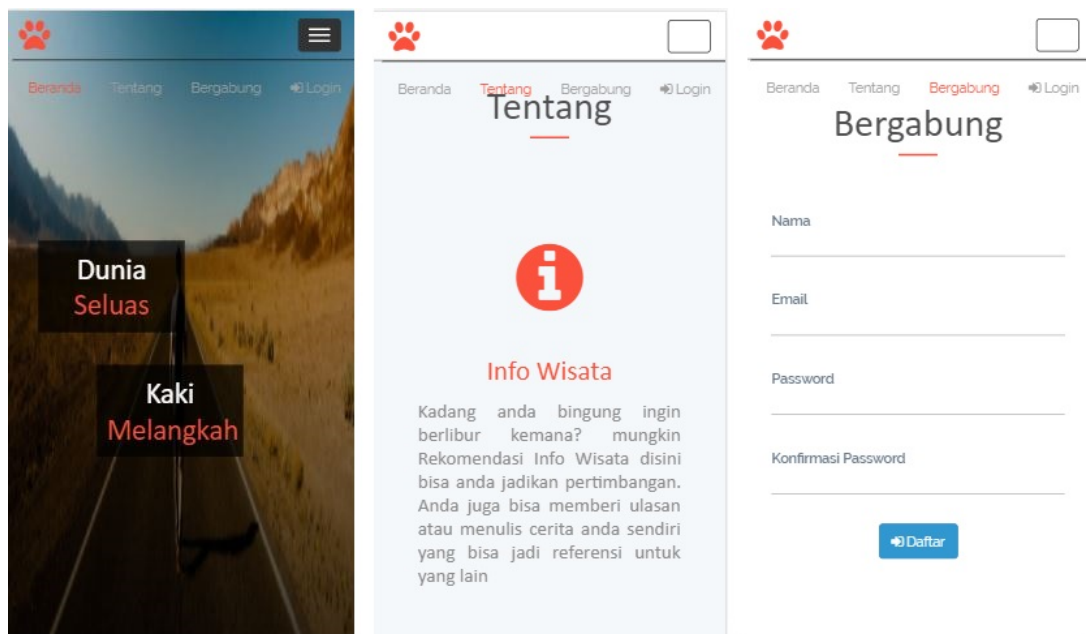
    public function index(){
        $data = Info::with('infocomment','infolike','likecek')
        ->withCount('infolike')
        ->orderBy('infolike_count', 'desc')
        ->filtered()->get()->sortByDesc('rekomendasi');
        $datasaya = Info::where('user_id',auth()->id())->get();
        return view('info',compact('data','datasaya'));
    }
}
```

LAMPIRAN 2

- Halaman Beranda

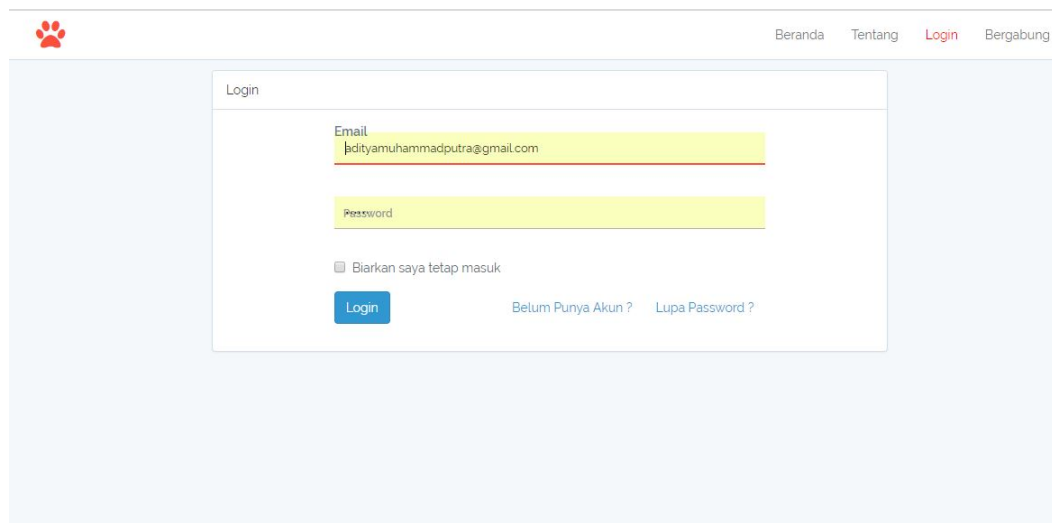


Gambar 5.1: Halaman Beranda Versi Web



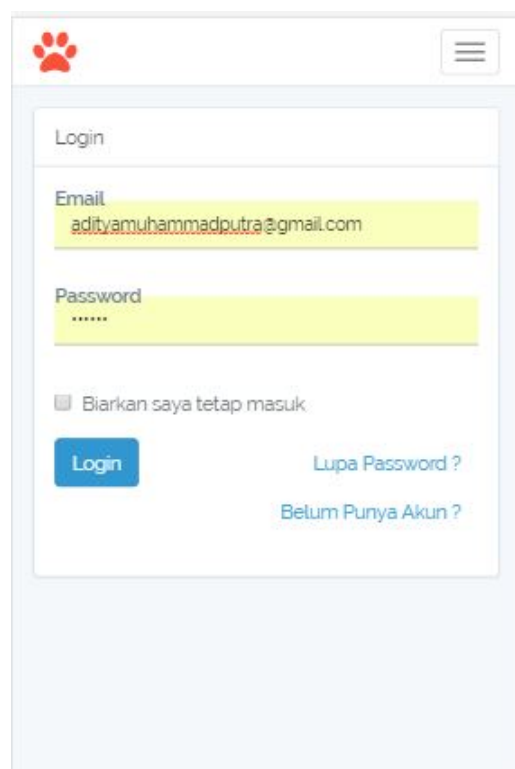
Gambar 5.2: Halaman Beranda Versi Mobile

- Halaman Login



The screenshot shows a web browser interface. At the top left is a red paw print logo. To the right of the logo are navigation links: "Beranda", "Tentang", "Login" (highlighted in red), and "Bergabung". Below the navigation bar is a light blue rectangular area containing a white login form. The form is titled "Login" and has two input fields: "Email" with the value "adityamuhammadputra@gmail.com" and "Password" which is empty. Below the password field is a checkbox labeled "Biarkan saya tetap masuk". At the bottom of the form are three elements: a blue "Login" button, a link "Belum Punya Akun ?", and a link "Lupa Password ?".

Gambar 5.3: Halaman Login Versi Web



The screenshot shows a mobile application interface. At the top left is a red paw print logo. At the top right is a hamburger menu icon. Below the header is a light blue rectangular area containing a white login form. The form is titled "Login" and has two input fields: "Email" with the value "adityamuhammadputra@gmail.com" and "Password" which contains six asterisks. Below the password field is a checkbox labeled "Biarkan saya tetap masuk". At the bottom of the form are three elements: a blue "Login" button, a link "Lupa Password ?", and a link "Belum Punya Akun ?".

Gambar 5.4: Halaman Login Versi Mobile

- Halaman Daftar

127.0.0.1:8000/register

Beranda Tentang Login **Bergabung**

Bergabung

Nama

Email

Password

Konfirmasi Password

Bergabung

Gambar 5.5: Halaman Daftar Versi Web

Bergabung

Nama

Email

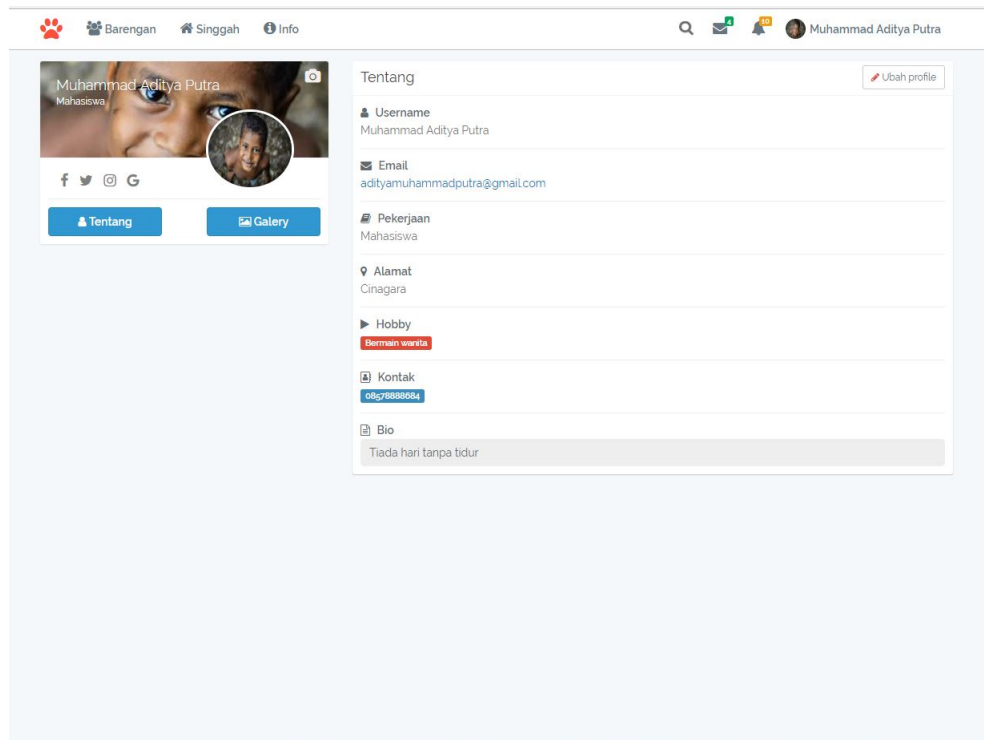
Password

Konfirmasi Password

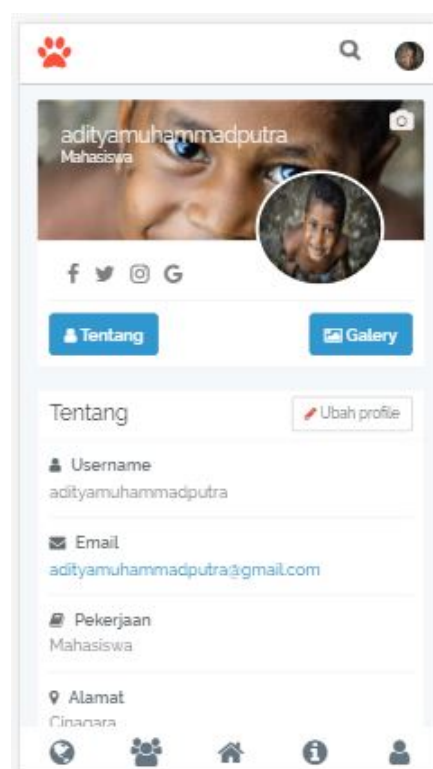
Bergabung

Gambar 5.6: Halaman Daftar Versi Mobile

- Halaman Profile User



Gambar 5.7: Halaman Profile User Versi Web



Gambar 5.8: Halaman Profile User Versi Mobile

- Halaman Ubah Profile User

The screenshot shows a web browser window with the URL `127.0.0.1:8000/tentang`. A modal form for updating a user profile is open. The form fields are as follows:

Field	Value
Nama	adityamuhammadputra
Pekerjaan	Mahasiswa
Agama	islam
Hobby	Berenang
Kontak	08577888684
Alamat	Kp Cinagara Rt. 02, Caringin, Kab. Bogor, Jawa Barat
Bio	Tiada Hari Tanpa Tidur

Buttons: `Batal` (left), `Update` (right).

Gambar 5.9: Ubah Profile User Versi Web

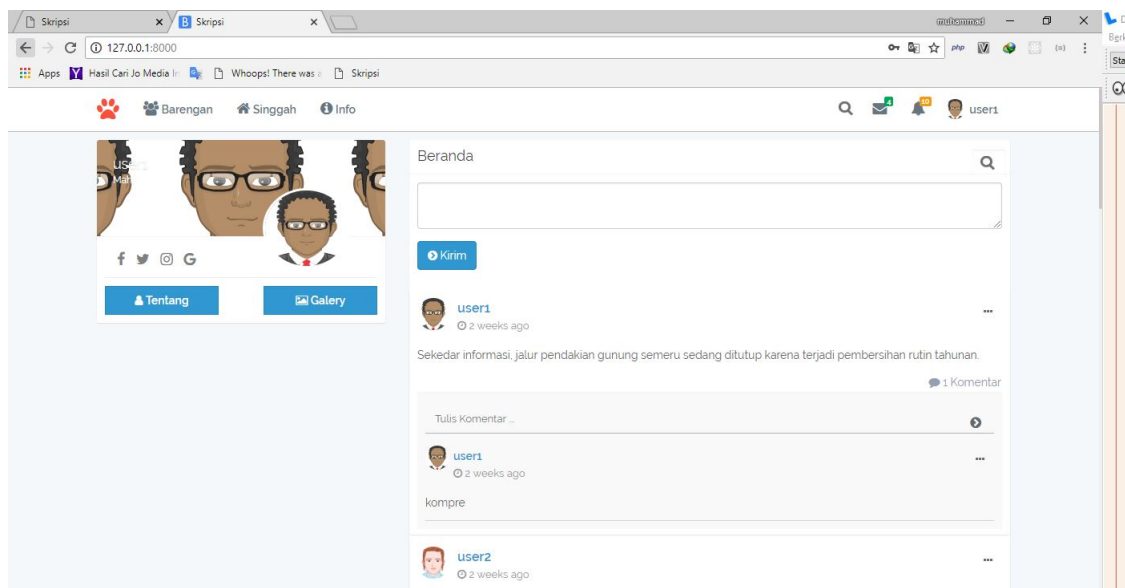
The screenshot shows a mobile view of the same profile update form. The layout is vertical and clean. The fields and their values are identical to the web version:

Field	Value
Nama	adityamuhammadputra
Pekerjaan	Mahasiswa
Agama	islam
Hobby	Berenang
Kontak	08577888684
Alamat	Kp Cinagara Rt. 02, Caringin, Kab. Bogor, Jawa E
Bio	Tiada Hari Tanpa Tidur

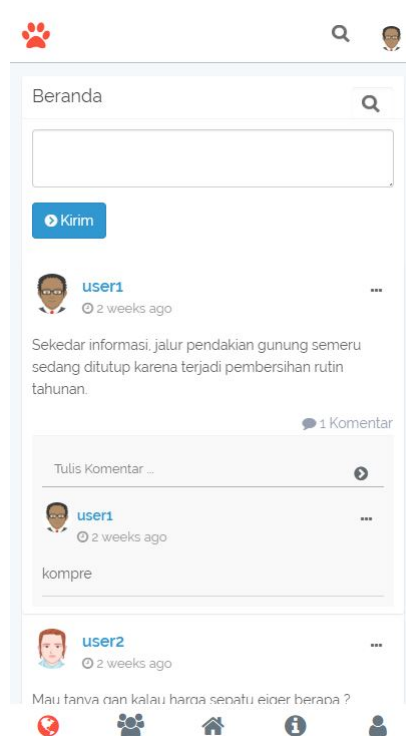
Buttons: `Batal` (left), `Update` (right).

Gambar 5.10: Ubah Profile User Versi Mobile

- Halaman Home

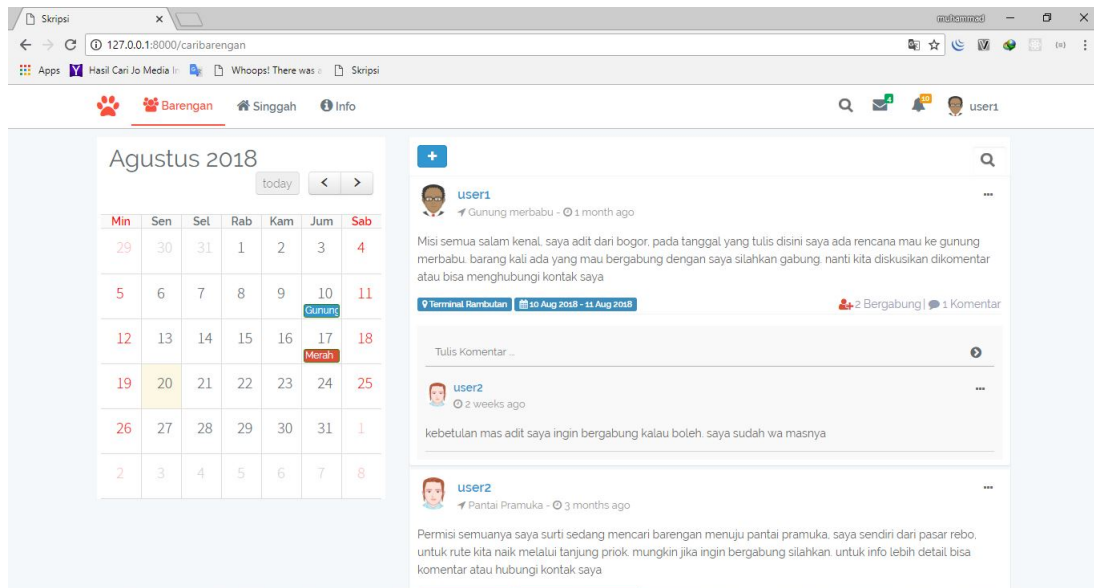


Gambar 5.11: Halaman Home Versi Web

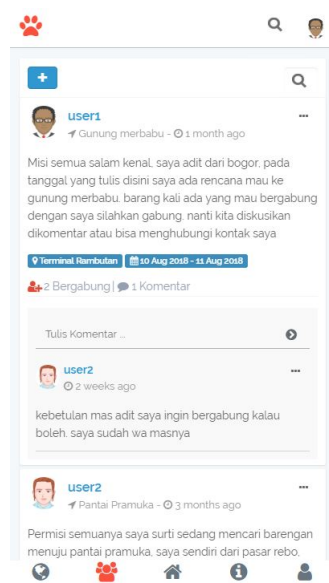


Gambar 5.12: Halaman Home Versi Mobile

- Halaman Barengan

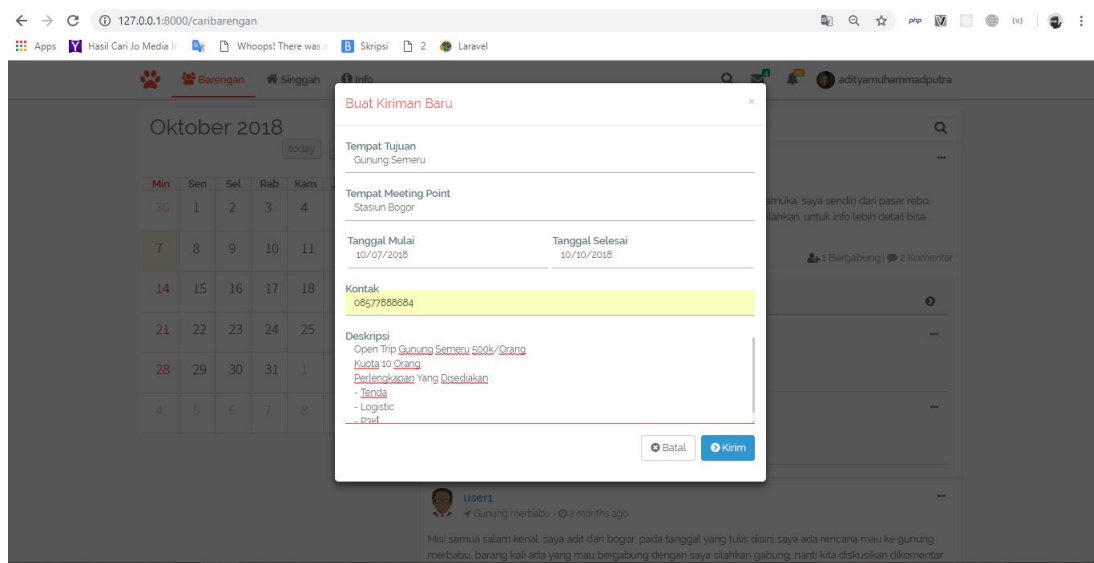


Gambar 5.13: Halaman Barengan Versi Web

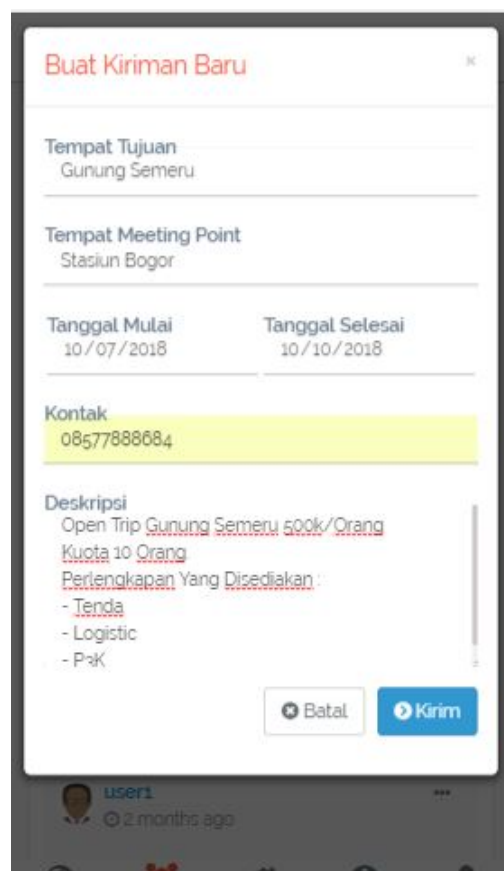


Gambar 5.14: Halaman Barengan Versi Mobile

- Halaman Input Barengan

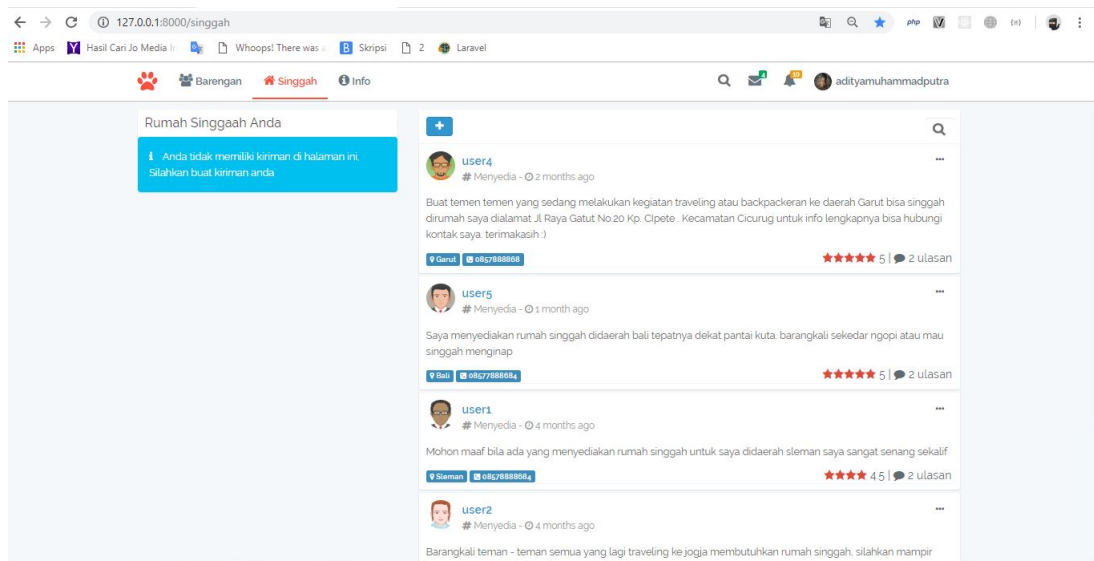


Gambar 5.15: Halaman Input Barengan Versi Web



Gambar 5.16: Halaman Input Barengan Versi Mobile

- Halaman Rumah Singgah

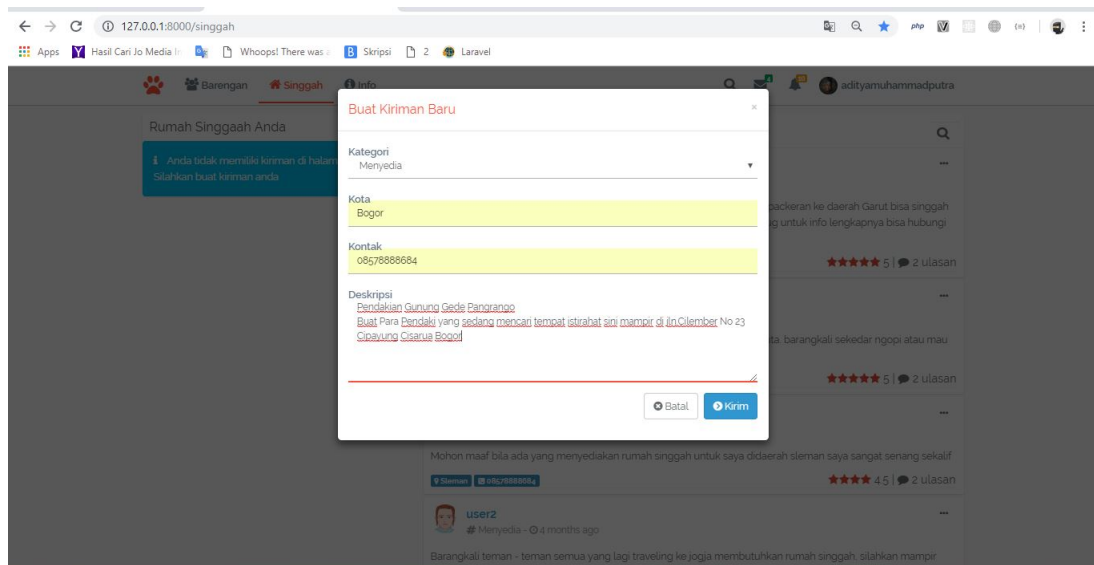


Gambar 5.17: Halaman Rumah Singgah Versi Web

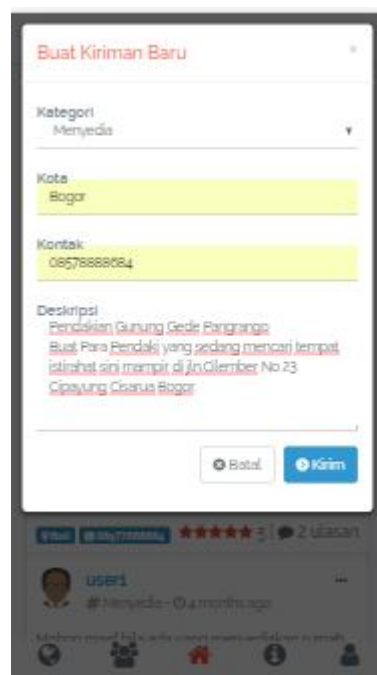


Gambar 5.18: Halaman Rumah Singgah Versi Mobile

- Halaman Input Rumah Singgah

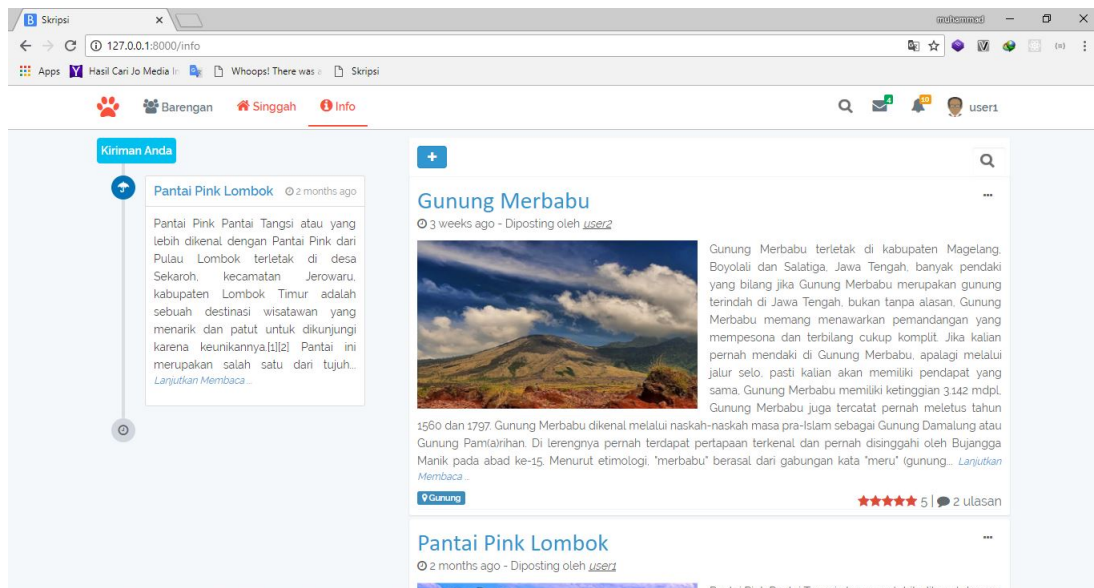


Gambar 5.19: Halaman Input Rumah Singgah Versi Web



Gambar 5.20: Halaman Input Rumah Singgah Versi Mobile

- Halaman Info Wisata

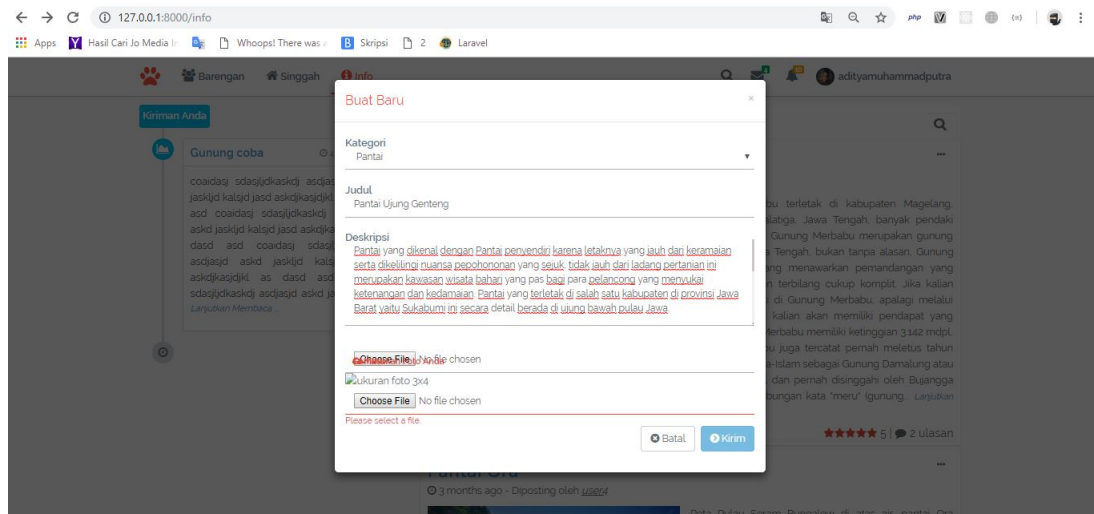


Gambar 5.21: Halaman Info Wisata Versi Web



Gambar 5.22: Halaman Info Wisata Versi Mobile

- Halaman Input Info Wisata



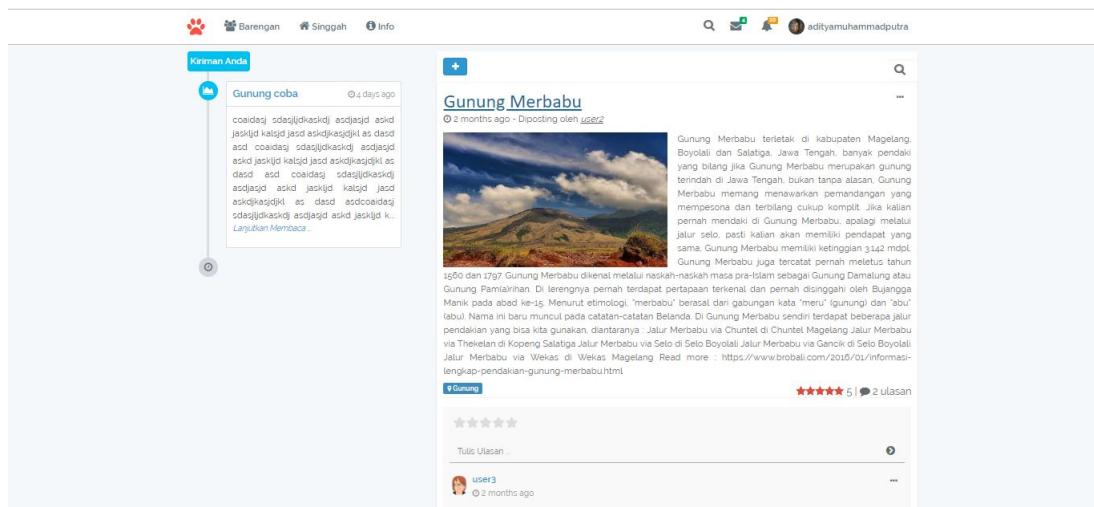
Gambar 5.23: Halaman Input Info Wisata Versi Web

- Halaman Input Info Wisata



Gambar 5.24: Halaman Input Info Wisata Versi Mobile

- Halaman Detail Info Wisata



Gambar 5.25: Halaman Detail Info Wisata Web



Gambar 5.26: Halaman Detail Info Wisata Versi Mobile