

**SISTEM PENDUKUNG KEPUTUSAN
PENJADWALAN BERBASIS LOKASI**

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika**



Oleh :

Nama : Tsabit

No. Induk Mahasiswa : 04523073

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2012

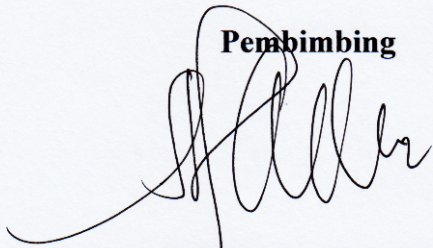
LEMBAR PENGESAHAN PEMBIMBING

**SISTEM PENDUKUNG KEPUTUSAN
PENJADWALAN BERBASIS LOKASI**



البعث الاسلامي

Pembimbing



Dr. Sri Kusumadewi, S.Si, M.T

LEMBAR PENGESAHAN PENGUJI**SISTEM PENDUKUNG KEPUTUSAN
PENJADWALAN BERBASIS LOKASI****TUGAS AKHIR****Oleh :****Nama : Tsabit****No. Induk Mahasiswa : 04523073**

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Jurusan Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 23 April 2012


Tim Penguji

Dr. Sri Kusumadewi, S.Si, M.T
Ketua

Nur Wijyaning Rahayu, S.Kom.
Anggota I

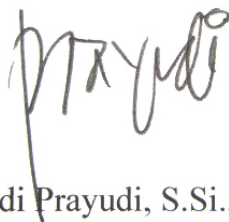
Lizda Iswari, ST., M.Sc.
Anggota II

Tanda Tangan



Mengetahui,

Ketua Jurusan Teknik Informatika



(Yudi Prayudi, S.Si., M.Kom.)

HALAMAN PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Karya yang sederhana ini saya persembahkan kepada kedua Orang Tua tercinta, Adekku Dalilah tersayang, saudara-saudara di Jogja, Surabaya, Gresik dan Lombok, sahabat-sahabatku di Jogja yang selalu memberi semangat untukku.

HALAMAN MOTTO

“Segala sesuatu pasti ada sebab dan akibatnya. Jika kita tidak dapat menemukan/mengerti sebab dan akibatnya, maka itu hanyalah dikarenakan oleh keterbatasan akal manusia.”

*“Kita semua pasti memiliki pilihan dalam segala tindakan kita. **Tidak ada** istilah **tidak ada pilihan** atau **tidak ada jalan keluar**. Rangkaian pilihan-pilihan yang kita ambil itu yang menentukan masa depan kita.”*

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalammualaikum Wr. Wb.

Dengan nama Allah Yang Maha Pengasih Lagi Maha Penyayang. Segala puji hanya untuk Allah Tuhan alam semesta. Ya Allah, limpahkan sholawat dan salam kepada Muhammad yang menjadi penuntun di dunia dan akhirat.

Alhamdulillah, segala puji dan syukur penulis hanya kepada Allah SWT yang telah melimpahkan rahmat, taufik serta hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir ini.

Penulis menyampaikan ucapan terimakasih dan penghargaan yang setinggi-tingginya atas bantuan, bimbingan serta dukungan dari berbagai pihak yang ikut membantu demi kelancaran pelaksanaan Tugas Akhir ini.

1. Bapak Ir. Gumbolo Hadi Susanto, M.Sc. selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Bapak Yudi Prayudi, S.Si., M.Kom. selaku Ketua Jurusan Teknik Informatika Universitas Islam Indonesia.
3. Ibu Dr. Sri Kusumadewi, S.Si, M.T selaku dosen pembimbing. Terima kasih atas segala kesabaran, bantuan, dukungan, semangat dan pengetahuannya, serta kemudahan yang telah diberikan.
4. Ayah, Ibu, Adik yang selalu mendukung dengan semangat dan doa selama menyelesaikan studi.
5. Teman-teman yang selalu memberi semangat untuk penulis agar menyelesaikan Tugas Akhir ini.
6. Semua pihak yang tidak dapat penulis sebutkan satu persatu dalam membantu penyusunan Tugas Akhir ini.

Semoga amal ibadah dan kebaikan yang telah diberikan mendapatkan imbalan yang setimpal dari Allah SWT.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan, oleh sebab itu penulis sangat mengharapkan kritik serta saran yang bersifat membangun

untuk perbaikan di masa mendatang. Semoga Tugas Akhir ini bermanfaat untuk kita semua. Amin.

Wassalamu'alaikum Wr. Wb.

Yogyakarta, Maret 2012

Penyusun

SARI

Saat ini, berbagai permasalahan yang muncul dalam kehidupan sehari-hari dapat diselesaikan dengan memanfaatkan teknologi komputer. Manusia sering dihadapkan kepada berbagai pilihan dalam menjalani aktifitasnya. Salah satunya adalah pemilihan jalur terpendek yang akan dilalui. Banyak manfaat dari pemilihan jalur terpendek yang tepat seperti pada perancangan jaringan komputer, efisiensi waktu tempuh maupun efisiensi bahan bakar kendaraan.

Permutasi adalah penyusunan kembali suatu kumpulan objek dalam urutan yang berbeda dari urutan yang semula. Permutasi dapat memberi hasil berupa jajaran semua kemungkinan urutan objek-objek sehingga dapat diambil suatu keputusan terbaik di antara semua kemungkinan tersebut.

Penggunaan permutasi serta pengujian hasil ke *Google Maps API* dalam aplikasi penentu jalur dengan total jarak terpendek *Penjadwalan Berbasis Lokasi* ini dapat mempertimbangkan semua kemungkinan urutan tempat-tempat yang akan dilalui untuk mendapatkan total jarak tempuh yang terpendek sehingga keputusan yang dihasilkan adalah nyata dan benar-benar dapat menjadi saran kepada pengguna untuk mempersingkat jarak perjalanan.

Aplikasi *Penjadwalan Berbasis Lokasi* ini dapat membantu pengguna untuk menyusun jadwal tugas-tugasnya dengan total jarak perjalanan terpendek serta terhindar dari tabrakan antar tugas sehingga diharapkan dapat memandu pengguna dalam menyelesaikan tugas-tugasnya tersebut dengan baik.

Kata kunci : Permutasi, *Google Maps API*, Jarak terpendek, *Penjadwalan*

DAFTAR ISI

SISTEM PENDUKUNG KEPUTUSAN	i
LEMBAR PENGESAHAN PEMBIMBING	ii
SISTEM PENDUKUNG KEPUTUSAN	ii
LEMBAR PENGESAHAN PENGUJI.....	iii
SISTEM PENDUKUNG KEPUTUSAN	iii
HALAMAN PERSEMBAHAN	iv
HALAMAN MOTTO	v
KATA PENGANTAR.....	vi
SARI	viii
DAFTAR ISI.....	ix
DAFTAR TABEL	xii
DAFTAR GAMBAR.....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	3
1.6 Metodologi Penelitian	3
1.6.1 Metode Pengumpulan Data	3
1.6.2 Metode Pengembangan Sistem	3
1.7 Sistematika Penulisan	4
BAB II LANDASAN TEORI	6
2.1 Permutasi.....	6
2.2 Google Maps Javascript API.....	6
BAB III ANALISIS SISTEM.....	8
3.1 Gambaran Umum Sistem	8

3.2	Analisis Kebutuhan Perangkat Lunak	8
3.2.1	Analisis Kebutuhan Masukan	8
3.2.2	Analisis Kebutuhan Proses.....	9
3.2.3	Analisis Kebutuhan Keluaran	10
3.2.4	Kebutuhan Antarmuka	11
BAB IV PERANCANGAN.....		12
4.1	Perancangan Data Flow Diagram.....	12
4.1.1	Diagram Konteks Penjadwalan Berbasis Lokasi	12
4.1.2	<i>Data Flow Diagram</i> Level 1	12
4.1.3	<i>Data Flow Diagram</i> Level 2 Manipulasi Data Jadwal	14
4.2	Perancangan <i>Flow Chart</i>	15
4.2.1	<i>Flow Chart</i> Keseluruhan Sistem	15
4.2.2	<i>Flow Chart</i> Memasukkan <i>Task</i> Baru	16
4.2.3	<i>Flow Chart</i> Permutasi <i>Tasks</i>	17
4.2.4	<i>Flow Chart</i> Seleksi Set Permutasi Yang Mungkin	18
4.2.5	<i>Flow Chart</i> Seleksi Jadwal Terpendek	20
4.3	Perancangan Tabel Basis Data	21
4.3.1	Struktur Tabel.....	21
4.3.2	Relasi Antar Tabel.....	24
4.4	Perancangan Antarmuka	26
4.4.1	Rancangan Antarmuka <i>Login</i> dan <i>Register</i>	26
4.4.2	Rancangan Antarmuka Ganti <i>Password</i>	27
4.4.3	Rancangan Antarmuka <i>Home</i> dan <i>Overview</i> Jadwal	28
4.4.4	Rancangan Antarmuka <i>Insert Task</i>	29
4.4.5	Rancangan Antarmuka <i>Direction Map</i>	30
4.4.6	Rancangan Antarmuka <i>Task Search</i>	31
4.4.7	Rancangan Antarmuka <i>My Markers</i>	32
4.4.8	Rancangan Antarmuka <i>Add Marker</i>	33
4.4.9	Rancangan Antarmuka <i>Map Settings</i>	34
BAB V IMPLEMENTASI DAN PENGUJIAN.....		35
5.1	Implementasi Sistem	35

5.1.1	Halaman <i>Login Dan Register</i>	35
5.1.2	Halaman <i>Agenda</i>	35
5.1.3	Halaman <i>Tambah Tugas</i>	37
5.1.4	Halaman <i>Edit Tugas</i>	39
5.1.5	Halaman <i>Direction Map</i>	41
5.1.6	Halaman <i>Pencarian Tugas</i>	42
5.1.7	Halaman <i>My Markers</i>	43
5.1.8	Halaman <i>Tambah Marker</i>	44
5.1.9	Halaman <i>Ubah Marker</i>	45
5.1.10	Halaman <i>Map Settings</i>	46
5.1.11	Halaman <i>Ganti Password</i>	47
5.2	<i>Pengujian</i>	48
5.2.1	<i>Penanganan Kesalahan</i>	48
5.2.1.1	<i>Penanganan Kesalahan Input Data Kosong</i>	48
5.2.1.2	<i>Penanganan Kesalahan Input Data Format</i>	48
5.2.2	<i>Penanganan Hasil Tidak Ada</i>	49
5.2.3	<i>Peringatan Aksi</i>	49
5.3	<i>Kasus</i>	49
5.4	<i>Penyelesaian</i>	53
5.5	<i>Kelebihan dan Kelemahan Sistem</i>	55
BAB VI PENUTUP		57
DAFTAR PUSTAKA		59

DAFTAR TABEL

Tabel 4.1 Tabel <i>user</i>	21
Tabel 4.2 Tabel <i>markers</i>	21
Tabel 4.3 Tabel <i>events</i>	22
Tabel 4.4 Tabel permutasi.....	22
Tabel 4.5 Tabel permutasi_mungkin.....	23
Tabel 4.6 Tabel jadwal.....	24
Tabel 5.1 Tabel Data Sampel <i>Markers</i>	50
Tabel 5.2 Tabel Data Sampel Tugas-Tugas.	50
Tabel 5.3 Tabel pengujian ke <i>Google Maps API</i>	53

DAFTAR GAMBAR

Gambar 4.1 Diagram Konteks.....	12
Gambar 4.2 DFD level 1.	13
Gambar 4.3 DFD level 2 Manipulasi Data Jadwal.	14
Gambar 4.4 <i>Flow Chart</i> sistem.	15
Gambar 4.5 <i>Flow Chart</i> Memasukkan <i>Task</i> Baru.	16
Gambar 4.6 <i>Flow Chart</i> Permutasi <i>Tasks</i>	17
Gambar 4.7 <i>Flow Chart</i> Seleksi Set Permutasi Yang Mungkin.	19
Gambar 4.8 <i>Flow Chart</i> Seleksi Jadwal Terpendek.....	20
Gambar 4.9 Relasi antar tabel.	25
Gambar 4.10 Rancangan antarmuka halaman <i>login</i> dan <i>register</i>	26
Gambar 4.11 Rancangan antarmuka halaman ganti <i>password</i>	27
Gambar 4.12 Rancangan Antarmuka <i>Home</i> dan <i>Overview</i> Jadwal.	28
Gambar 4.13 Rancangan Antarmuka <i>Insert Task</i>	29
Gambar 4.14 Rancangan Antarmuka <i>Direction Map</i>	30
Gambar 4.15 Rancangan Antarmuka <i>Task Search</i>	31
Gambar 4.16 Rancangan Antarmuka <i>My Markers</i>	32
Gambar 4.17 Rancangan Antarmuka <i>Add Marker</i>	33
Gambar 4.18 Rancangan Antarmuka <i>Map Settings</i>	34
Gambar 5.1 Halaman <i>Login</i> dan <i>Register</i>	35
Gambar 5.2 Halaman <i>Agenda</i> yang masih kosong.	36
Gambar 5.3 Halaman <i>Agenda</i> yang telah terisi tugas-tugas.	37
Gambar 5.4 Halaman Tambah Tugas (Tahap 1).	38
Gambar 5.5 Halaman Tambah Tugas (Tahap 2).	38
Gambar 5.6 Halaman Tambah Tugas (Tahap 3).	39
Gambar 5.7 Halaman <i>Edit</i> Tugas (Tahap 1).	40
Gambar 5.8 Halaman <i>Edit</i> Tugas (Tahap 2).	41
Gambar 5.9 Halaman <i>Edit</i> Tugas (Tahap 3).	41
Gambar 5.10 Halaman <i>Direction Map</i>	42

Gambar 5.11 Halaman Pencarian Tugas (Tahap 1).	43
Gambar 5.12 Halaman Pencarian Tugas (Tahap 2).	43
Gambar 5.13 Halaman <i>My Markers</i>	44
Gambar 5.14 Halaman Tambah <i>Marker</i>	45
Gambar 5.15 Halaman Ubah <i>Marker</i>	46
Gambar 5.16 Halaman <i>Map Settings</i>	47
Gambar 5.17 Halaman Ganti <i>Password</i>	47
Gambar 5.18 Penanganan kesalahan <i>input</i> data kosong.	48
Gambar 5.19 Penanganan Kesalahan <i>Input Data Format</i>	48
Gambar 5.20 Penanganan Hasil Tidak Ada.	49
Gambar 5.21 Peringatan Aksi.	49
Gambar 5.22 Pengujian kemungkinan set (A,B,C).	51
Gambar 5.23 Pengujian kemungkinan set (A,C,B).	51
Gambar 5.24 Pengujian kemungkinan set (B,A,C).	51
Gambar 5.25 Pengujian kemungkinan set (B,C,A).	51
Gambar 5.26 Pengujian kemungkinan set (C,A,B).	52
Gambar 5.27 Pengujian kemungkinan set (C,B,A).	52
Gambar 5.28 Jadwal untuk penyelesaian kasus.	54
Gambar 5.29 <i>Direction map</i> untuk penyelesaian kasus.	55

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saat ini semakin banyak pekerjaan manusia yang digantikan oleh komputer. Salah satunya ada dalam hal mengambil keputusan. Banyaknya tugas yang harus diselesaikan, membuat manusia membutuhkan rencana dan penjadwalan yang baik dalam melakukan tugas-tugas tersebut agar semuanya dapat diselesaikan secara efektif dan efisien. Rencana dan penjadwalan yang baik juga dapat menentukan keberhasilan tugas-tugas tersebut. Keterbatasan manusia seperti lupa atau perkiraan yang salah semakin memperkuat alasan manusia membutuhkan alat bantu yang menggantikan sebagian tugas mereka.

Saat ini juga telah banyak aplikasi komputer dan perangkat genggam yang memiliki fungsi memudahkan atau menggantikan tugas-tugas manusia. Sebagian pengguna bahkan telah bergantung pada suatu aplikasi tersebut. Mudahnya seseorang mengakses internet juga membuat semakin pesat berkembangnya jenis dan fungsi aplikasi-aplikasi tersebut. Salah satu jenis aplikasi yang saat ini berkembang dan memiliki fungsi sebagai penyusun agenda kegiatan sehari-hari adalah aplikasi penjadwalan. Aplikasi penjadwalan memiliki berbagai fungsi, antara lain sebagai pengingat kegiatan yang akan datang, pengingat tanggal (kalender), agenda kegiatan dan sebagainya. Aplikasi ini mulai dikenal sejak dibenamkannya pada fungsi kalender pada ponsel. Aplikasi penjadwalan ini akan sangat bermanfaat bagi orang yang sibuk dan memiliki banyak kegiatan sehingga tidak mampu mengingat seluruh kegiatan tersebut. Banyak juga pengguna yang memanfaatkan aplikasi ini untuk sekedar mengingatkan ulang tahun teman atau sebagai pengingat kegiatan harian. Melihat masalah-masalah serta ketersediaan teknologi di atas, maka diperlukan pengembangan suatu aplikasi yang dapat menyelesaikan masalah tersebut dengan memanfaatkan teknologi yang ada.

Pada aplikasi penjadwalan konvensional tidak disertakan fitur yang dapat membantu pengguna menentukan jalur tercepat yang akan dilalui dalam

melaksanakan agenda kegiatannya. Padahal tidak menutup kemungkinan seseorang akan berpindah-pindah tempat dalam melakukan kegiatan-kegiatannya. Bahkan tidak jarang pula, seseorang harus berpindah-pindah tempat yang jarak antar tempat tersebut tidaklah dekat. Masalah lain yang mungkin muncul adalah orang tersebut dihadapkan oleh pilihan rute yang bervariasi, jadi perlu adanya keputusan yang menentukan jalur atau rute mana yang memiliki jarak total terpendek sehingga dibutuhkan waktu sependek mungkin untuk menempuhnya.

1.2 Rumusan Masalah

Setelah melihat latar belakang di atas, maka muncul rumusan permasalahan sebagai berikut :

- a. Sebagian orang yang memiliki aktifitas padat dan mobilitas yang tinggi memerlukan aplikasi yang dapat menyusun serta menentukan rencana tugas-tugas yang efektif dan efisien.
- b. Bagaimana membangun sistem bantu untuk menyusun tugas sehari-hari serta menentukan rute yang efektif dan efisien untuk menyelesaikannya.

1.3 Batasan Masalah

Untuk mengatasi permasalahan dalam penyusunan aplikasi ini, maka penyusun membatasi lingkup permasalahan sebagai berikut :

- a. Jumlah *task* untuk satu hari dibatasi maksimal lima.
- b. Waktu tempuh ke suatu tempat akan menjadi bagian dari durasi tugas yang bersangkutan.

1.4 Tujuan Penelitian

Tujuan penyusunan tugas akhir ini adalah menghasilkan aplikasi yang dapat membantu pengguna melaksanakan serta menyelesaikan tugas sehari-hari secara efektif dan efisien.

1.5 Manfaat Penelitian

Beberapa manfaat dari dikembangkannya aplikasi ini bagi pengguna adalah :

- a. Membantu pengguna menyusun dengan rapi tugas-tugas sehari-hari serta menentukan jadwal yang efektif dan efisien menurut waktu dan jarak. Aplikasi ini akan sangat berguna bagi pengguna yang memiliki aktifitas dan mobilitas yang padat.
- b. Pengguna akan terhindar dari tabrakan tugas-tugas (beberapa tugas dalam satu waktu).

1.6 Metodologi Penelitian

Metodologi penelitian yang digunakan adalah metode pengumpulan data dan pengembangan sistem. Pengembangan sistem ini memerlukan pengambilan data sample untuk pengujian di setiap tahap-tahap pengembangannya.

1.6.1 Metode Pengumpulan Data

Pengumpulan data yang diperlukan menggunakan metode sebagai berikut :

- a. Metode observasi

Metode ini merupakan metode pengumpulan data yang didapatkan dari sumber yang menyediakan data mentah yang nantinya akan diproses dan diseleksi untuk keperluan pengujian maupun keperluan memasukkan data sesungguhnya.

- b. Studi pustaka

Metode ini digunakan sebagai materi tambahan atau penggalian lebih dalam mengenai permasalahan yang muncul.

1.6.2 Metode Pengembangan Sistem

Metode yang digunakan dalam pengembangan sistem, yaitu :

- a. Analisis Masalah

Analisis yang dilakukan untuk menggali masalah-masalah yang muncul sehingga dapat menentukan solusi untuk menyelesaikannya. Dari analisis tersebut akan diperoleh metode yang akan dipakai serta spesifikasi pada sistem yang akan dirancang.

- b. Perancangan

Pemodelan sistem berdasarkan hasil analisa sehingga diperoleh gambaran penyelesaian dari permasalahan yang didapatkan dari tahapan analisa. Gambaran ini akan digunakan sebagai acuan pada tahap implementasi. Pemodelannya akan menggunakan *data flow diagram* dan *flow chart*.

c. Implementasi

Merupakan tahapan implementasi dari perancangan dengan berbagai perangkat lunak untuk desain antarmuka dan *server* lokal.

d. Pengujian

Merupakan tahapan yang dilakukan untuk mengetahui apakah sistem dapat berjalan dengan baik serta sesuai dengan tujuan yang diharapkan dengan cara membandingkan fitur dan kinerja dengan aplikasi penjadwalan konvensional.

e. Analisa Hasil

Merupakan langkah akhir, menganalisis hasil dari sistem yang dibuat apakah telah menyelesaikan permasalahan yang ada.

1.7 Sistematika Penulisan

Untuk mempermudah dalam pembahasan tugas akhir ini, maka dalam penyusunan, penulis membagi pokok permasalahan ke dalam beberapa bab.

Bab 1 Pendahuluan, bab ini membahas tentang latar belakang, rumusan masalah, batasan masalah, tujuan tugas akhir, manfaat tugas akhir, metodologi penelitian, serta sistematika penulisan.

Bab 2 Landasan Teori, bab ini membahas tentang konsep dasar landasan teori yang berkaitan dengan algoritma penyusunan *task*.

Bab 3 Analisis Sistem, bab ini membahas masalah, pemodelan dan analisis kebutuhan perangkat lunak yang akan dibuat.

Bab 4 Perancangan, bab ini membahas implementasi dari rancangan perangkat lunak yang telah diterangkan sebelumnya, termasuk di dalamnya adalah perancangan antarmuka, analisis kebutuhan masukan, analisis kebutuhan keluaran, dan perancangan aliran data.

Bab 5 Implementasi dan Pengujian, bab ini akan membahas langkah selanjutnya setelah terselesaikannya implementasi, kemudian dilakukan pengujian

terhadap sistem informasi tersebut, untuk mengetahui segala kekurangan yang ada pada sistem.

Bab 6 Penutup, bab ini memuat kesimpulan-kesimpulan yang merupakan rangkuman dari hasil analisis kinerja pada bagian sebelumnya dan saran yang perlu diperhatikan berdasarkan keterbatasan yang ditemukan dan asumsi-asumsi yang dibuat selama pembuatan aplikasi Penjadwalan Berbasis Lokasi.

BAB II LANDASAN TEORI

2.1 Permutasi

Jika suatu himpunan A beranggotakan n benda yang berlainan disusun menurut urutan tertentu, maka susunan tersebut disebut permutasi. Misalkan himpunan A beranggota P, Q dan R maka $A = \{P, Q, R\}$. Dari himpunan ini dapat disusun 6 jajaran anggota, yaitu (Negoro & Harahap, 1998):

$$\begin{array}{l} PQR \quad QRP \quad RPQ \\ PRQ \quad QPR \quad RQP \end{array}$$

Jajaran anggota P, Q dan R yang disusun dengan memperhatikan urutannya disebut permutasi.

Contoh lain, suatu himpunan beranggotakan tiga angka $1, 2$ dan 3 akan disusun berpasang-pasang sehingga menghasilkan pasangan-pasangan angka dengan memperhatikan urutannya dan diperoleh hasil sebagai berikut.

$$12 \quad 13 \quad 21 \quad 23 \quad 31 \quad 32$$

Permutasi yang demikian dilambangkan dengan ${}_n P_k$.

$$P_k^n = \frac{n!}{(n-k)!} = \frac{3!}{(3-2)!} = 6$$

2.2 Google Maps Javascript API

Antarmuka pemrograman aplikasi (Inggris: *application programming interface* disingkat *API*) adalah sekumpulan perintah, fungsi, dan protokol yang dapat digunakan oleh *programmer* saat membangun perangkat lunak untuk sistem operasi tertentu. *API* memungkinkan *programmer* untuk menggunakan fungsi standar untuk berinteraksi dengan sistem operasi (Google, 2012).

Pada sistem ini, permasalahan jalur terpendek akan diselesaikan dengan memanfaatkan *service* dari *Google Maps Javascript API*. Semua kemungkinan jalur yang memenuhi syarat akan dijadikan sebagai input ke *Google* yang kemudian akan dipilih sebuah jalur yang paling efektif dari sisi jarak. Keuntungan menggunakan *service* dari *Google* ini banyak, antara lain : *Google Maps*

Javascript API memungkinkan *programmer* menanamkan *Google Maps* di halaman webnya sendiri. *API* ini menyediakan sejumlah utilitas untuk memanipulasi peta (seperti <http://maps.google.com> pada halaman web) dan menambahkan konten ke peta melalui berbagai layanan. Selain itu, layanan tersebut bersifat gratis serta data yang disuguhkan oleh *Google* adalah data nyata dan akurat.

BAB III

ANALISIS SISTEM

3.1 Gambaran Umum Sistem

Sistem pendukung keputusan Penjadwalan Berbasis Lokasi ini adalah aplikasi yang memiliki fungsi utama sebagai penentu rute terpendek yang menghubungkan titik-titik atau tempat-tempat yang akan dikunjungi oleh pengguna dalam tugas-tugasnya dalam satu hari. Ada tiga tahap utama yang akan dilalui oleh pengguna, yaitu :

a. Menentukan titik-titik.

Titik-titik atau tempat-tempat ini adalah salah satu syarat untuk pengguna dapat menyisipkan suatu tugas ke dalam penjadwalannya. Sebelum memulai membuat daftar tugasnya pengguna diharuskan menentukan atau membuat titik (*marker*).

b. Menyisipkan tugas.

Penyisipan tugas adalah pemasukan suatu kegiatan yang direncanakan oleh pengguna untuk akan dilalui di tanggal tertentu. Atribut atau data yang dibutuhkan untuk memasukkan suatu tugas antara lain nama tugas, keterangan tugas, tempat (*marker*), waktu dapat dilakukannya sebuah tugas tersebut serta durasi tugas.

c. Mendapatkan *Agenda*.

Pengguna akan dipandu untuk mendapatkan susunan *agenda* dalam tanggal tertentu yang dipilih oleh pengguna. Susunan acara atau *agenda* tersebut disajikan dalam dua bentuk antara lain berupa daftar tugas atau penjadwalan serta berupa peta petunjuk jalan atau *direction map*.

3.2 Analisis Kebutuhan Perangkat Lunak

3.2.1 Analisis Kebutuhan Masukan

Masukan dari sistem Penjadwalan Berbasis Lokasi ini terdiri dari dua jenis pengguna yaitu *administrator* dan *user*. *Administrator* adalah pengguna yang memiliki wewenang penuh dalam memanipulasi semua data dalam *database*. Sedangkan *user* adalah pengguna yang dapat menambah dan melihat data-data

tugas yang bersangkutan dengan *account*-nya. Analisis kebutuhan masukan berdasarkan jenis pengguna yaitu :

a. *Administrator*

Secara umum tidak ada jenis masukan khusus yang membatasi kemampuan *administrator* dalam memanipulasi *database*. Hampir semua data di *database* dapat di ubah oleh *administrator*.

b. *User*

Masukan *user* adalah masukan data yang dilakukan oleh *user* yang telah terdaftar sebagai pengguna. Jenis-jenis masukan tersebut adalah :

1. Data *account*

Data *account* adalah masukan data yang berhubungan dengan *account* seseorang, bisa berupa informasi data pribadi maupun password untuk proses *login*.

2. Data *marker*

Data *marker* adalah masukan data untuk titik-titik posisi (*markers*) yang dikehendaki oleh *user* untuk keperluan penempatan tugas-tugas (*tasks*) nantinya. Data *marker* yang menjadi masukan berupa nama *marker*, alamat *marker*, koordinat posisi *marker*, dan keterangan lain untuk *marker*.

3. Data *task*

Data *task* adalah masukan data yang dibutuhkan *user* untuk mendefinisikan suatu *task*. *Task* itu sendiri nantinya akan menjadi bagian dari *agenda* pada hari tertentu. Data *task* yang menjadi masukan berupa nama *task*, deskripsi *task*, tanggal, rentang waktu dilaksanakannya, durasi, serta *marker* tempat terlaksananya.

3.2.2 Analisis Kebutuhan Proses

Berikut ini adalah kebutuhan proses dalam sistem *Penjadwalan Berbasis Lokasi* :

a. *Login*

Proses *login* dilakukan oleh *user* sebelum memasuki sistem *Penjadwalan Berbasis Lokasi*. Proses ini dilakukan agar keamanan data dalam sistem dapat terjaga. Proses ini dimulai setelah pengguna memasukkan *username* dan

password. Sesaat setelah pengguna berhasil divalidasi melalui *username* dan *password*-nya, maka sistem akan memberikah hak akses sesuai dengan jenis pengguna.

b. Olah data *marker*

Proses olah data *marker* meliputi: tambah, ubah, dan hapus data *marker*. Data meliputi: nama *marker*, alamat *marker*, koordinat (*latitude*, *longitude*) *marker*, deskripsi *marker*. Pengolahan data ini diperuntukkan kepada *user* dan dapat dilakukan pada halaman *markers*.

c. Olah data *task*

Proses olah data *task* meliputi: tambah, ubah, dan hapus data *task*. Data meliputi: judul *task*, deskripsi/keterangan *task*, waktu *start task*, waktu *end task*, durasi *task*. Pengelolaan data ini diperuntukkan kepada *user* dan dapat dilakukan pada halaman *agenda*.

d. Olah data *agenda*

Proses olah data *agenda* merupakan proses pengurutan *tasks* sehingga menjadi jadwal kegiatan *user* pada hari tertentu. Hasil dari proses data *agenda* ini diperuntukkan kepada *user* sebagai panduan urutan kegiatan sehari-harinya. Hasil dari proses ini disajikan kepada *user* pada halaman *agenda*.

e. Olah data peta (*direction map*)

Proses olah data peta penunjuk jalan (*direction map*) merupakan proses yang memanfaatkan *service* dari *Google Maps API*. Dalam proses ini sistem memberikan masukan berupa *markers* yang selanjutnya akan direspon oleh *Google* berupa rute perjalanan tercepat, jarak tempuh, dan waktu tempuhnya.

3.2.3 Analisis Kebutuhan Keluaran

Keluaran yang dihasilkan oleh sistem *Penjadwalan Berbasis Lokasi* adalah informasi yang ditampilkan kepada pengguna yang telah terdaftar sebagai *user*. Informasi tersebut adalah :

- a. Informasi *markers*
- b. Informasi *agenda*
- c. Penunjuk jalan/rute (*direction map*)

3.2.4 Kebutuhan Antarmuka

Antarmuka sistem ini adalah antarmuka berbasis web. Pada sistem pendukung keputusan ini, informasi yang dimasukkan maupun yang diubah oleh pengguna akan ditampilkan oleh *browser* dan dibagi dalam beberapa bagian utama yaitu :

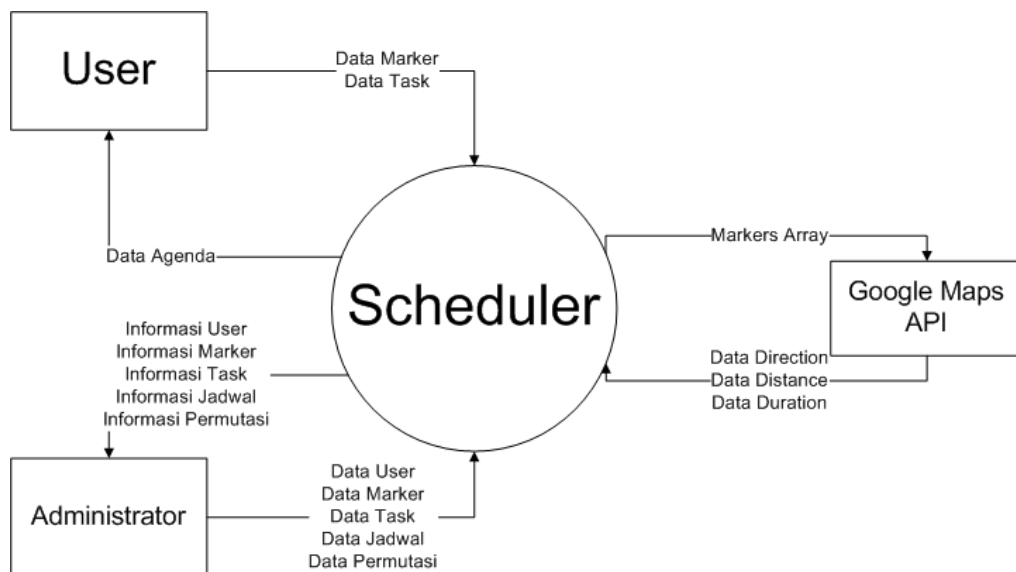
- a. Antarmuka untuk *Login* atau validasi pengguna.
- b. Antarmuka untuk registrasi pengguna.
- c. Antarmuka untuk olah *Agenda*.
- d. Antarmuka untuk olah *Markers*.
- e. Antarmuka untuk informasi *Direction Map*.

BAB IV PERANCANGAN

4.1 Perancangan Data Flow Diagram

4.1.1 Diagram Konteks Penjadwalan Berbasis Lokasi

Penggunaan diagram arus data di sini bertujuan untuk memudahkan dalam melihat arus data dalam sistem. Sistem selalu memiliki sebuah skema, seperti yang ditampilkan pada gambar 4.1. Diagram pada gambar 4.1 ini merupakan gambaran dari seluruh sistem secara umum di mana sistem lokal dapat mengirimkan data yang merupakan *input* dari *user* serta berhubungan dengan *Google Maps API* untuk pengolahan data. Pengolahan dan penyeleksian hasil olahan nantinya akan digunakan untuk menyusun agenda dalam satu hari, perkiraan total jarak yang ditempuh serta perkiraan total durasi waktu yang diperlukan untuk melewati semua titik pemberhentian.



Gambar 4.1 Diagram Konteks

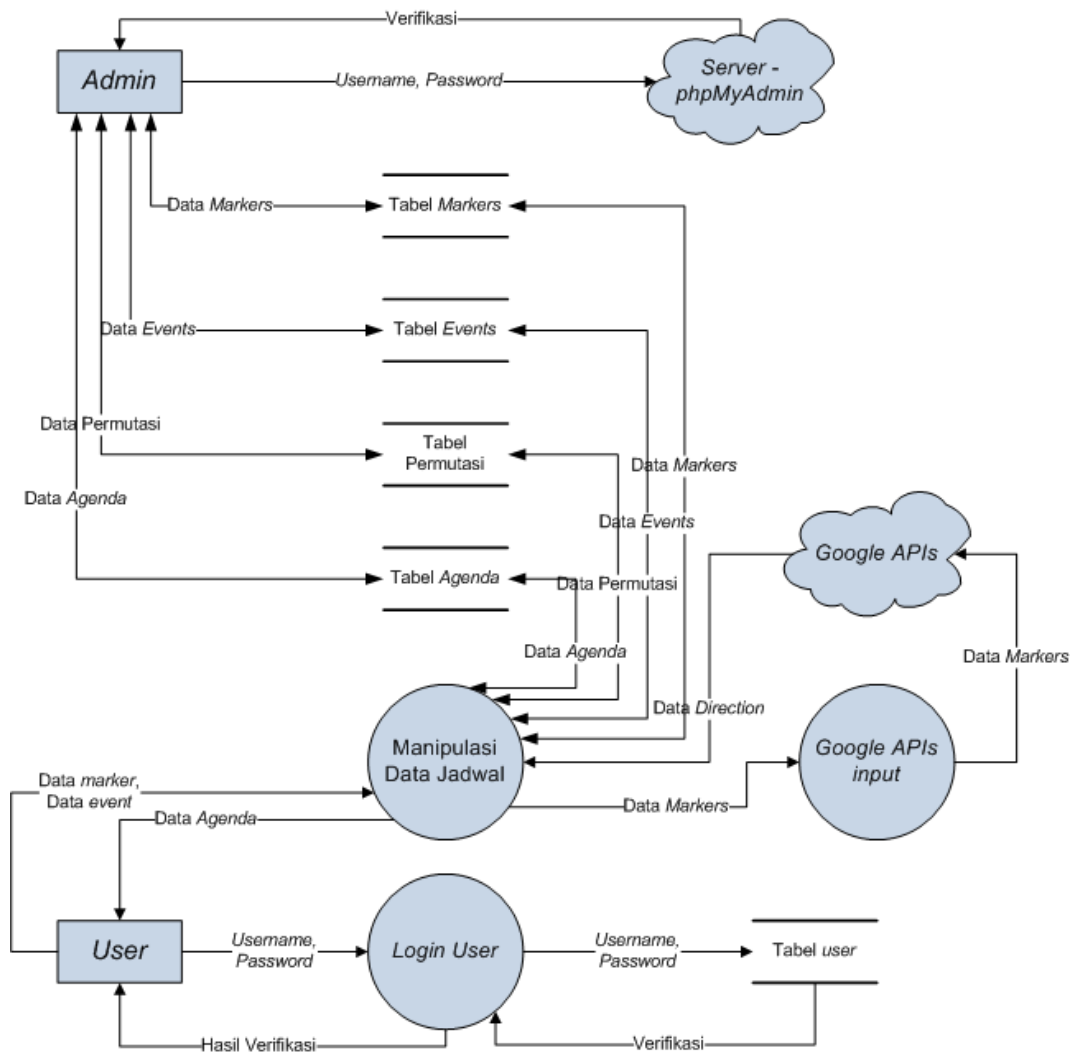
4.1.2 Data Flow Diagram Level 1

Pada DFD level 1 ini, menggambarkan semua proses yang terjadi dalam sistem. DFD level 1 ini terdiri dari 5 proses yaitu :

- a. Proses manipulasi data oleh *administrator*
- b. Proses *user login*

- c. Proses *input* untuk *Google Maps API*
- d. Proses yang terjadi di *Google*
- e. Proses manipulasi data Jadwal.

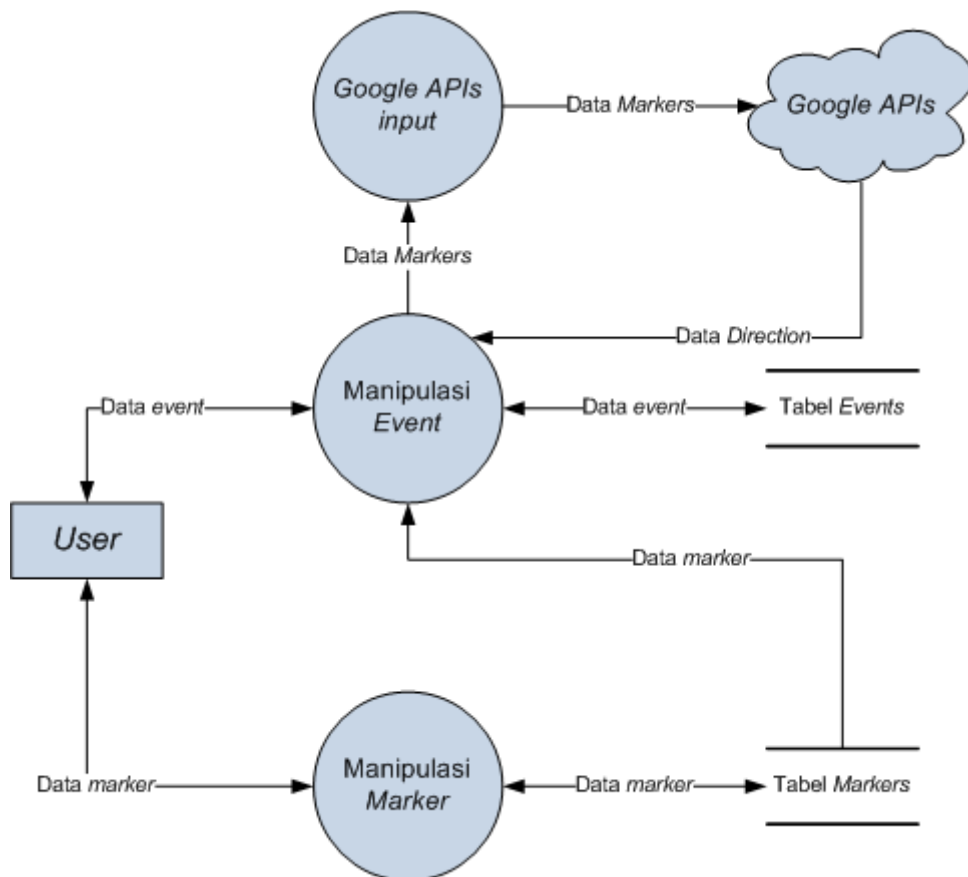
Dari kelima proses di atas yang terjadi di dalam aplikasi ini hanya 3 proses yaitu proses login *user*, proses input *Google Maps API*, dan proses manipulasi data Jadwal, karena kedua proses lainnya terjadi di luar sistem. Proses aliran data dapat dilihat pada gambar 4.2.



Gambar 4.2 DFD level 1.

4.1.3 Data Flow Diagram Level 2 Manipulasi Data Jadwal

Pada DFD untuk memanipulasi data jadwal terdiri dari 2 proses. Penjelasan dari proses manipulasi data jadwal dapat dilihat pada diagram gambar 4.3 di bawah ini.

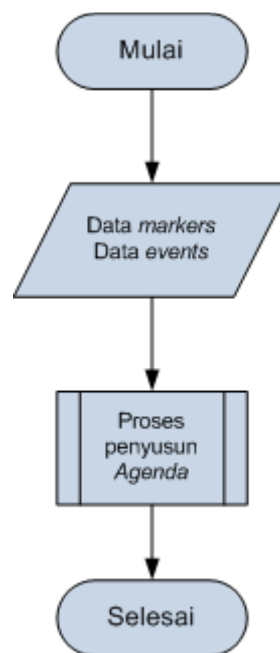


Gambar 4.3 DFD level 2 Manipulasi Data Jadwal.

4.2 Perancangan *Flow Chart*

4.2.1 *Flow Chart* Keseluruhan Sistem

Alur proses sistem dari *input*, proses inti sampai dengan hasil digambarkan sebagai sebuah diagram alir yang dapat dilihat pada gambar 4.4.

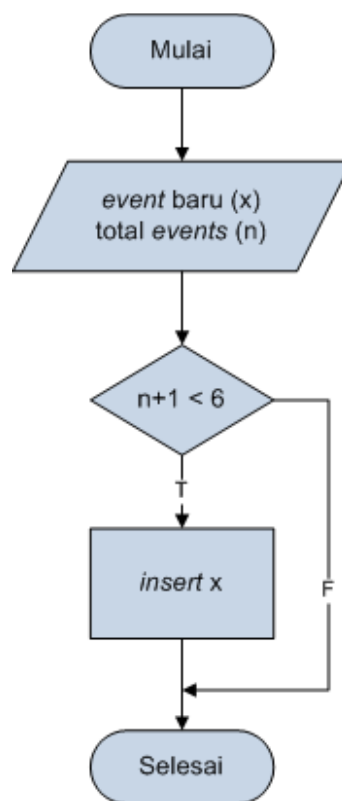


Gambar 4.4 *Flow Chart* sistem.

Proses penyusunan *Agenda* terbagi menjadi proses-proses yang lebih spesifik yaitu proses permutasi *tasks*, proses seleksi permutasi *tasks* yang mungkin, dan proses seleksi jadwal tercepat.

4.2.2 Flow Chart Memasukkan Task Baru

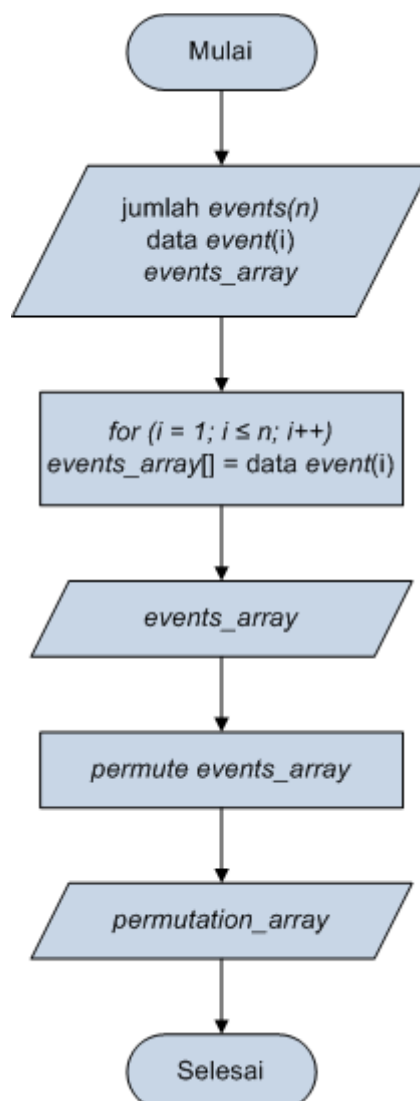
Gambar 4.5 menjelaskan bahwa inialisasi awal untuk proses memasukkan *task* baru adalah data *task* baru itu sendiri yang dihasilkan dari form *insert new task* yang diisi oleh *user*, serta jumlah *tasks* yang sudah ada untuk hari terpilih. Jumlah *tasks* pada aplikasi ini untuk satu hari dibatasi maksimal lima (tidak termasuk titik berangkat dan titik pulang) dikarenakan semakin banyaknya kemungkinan (himpunan anggota) dalam permutasi seiring dengan semakin banyaknya *tasks* dalam satu hari.



Gambar 4.5 Flow Chart Memasukkan Task Baru.

4.2.3 Flow Chart Permutasi Tasks

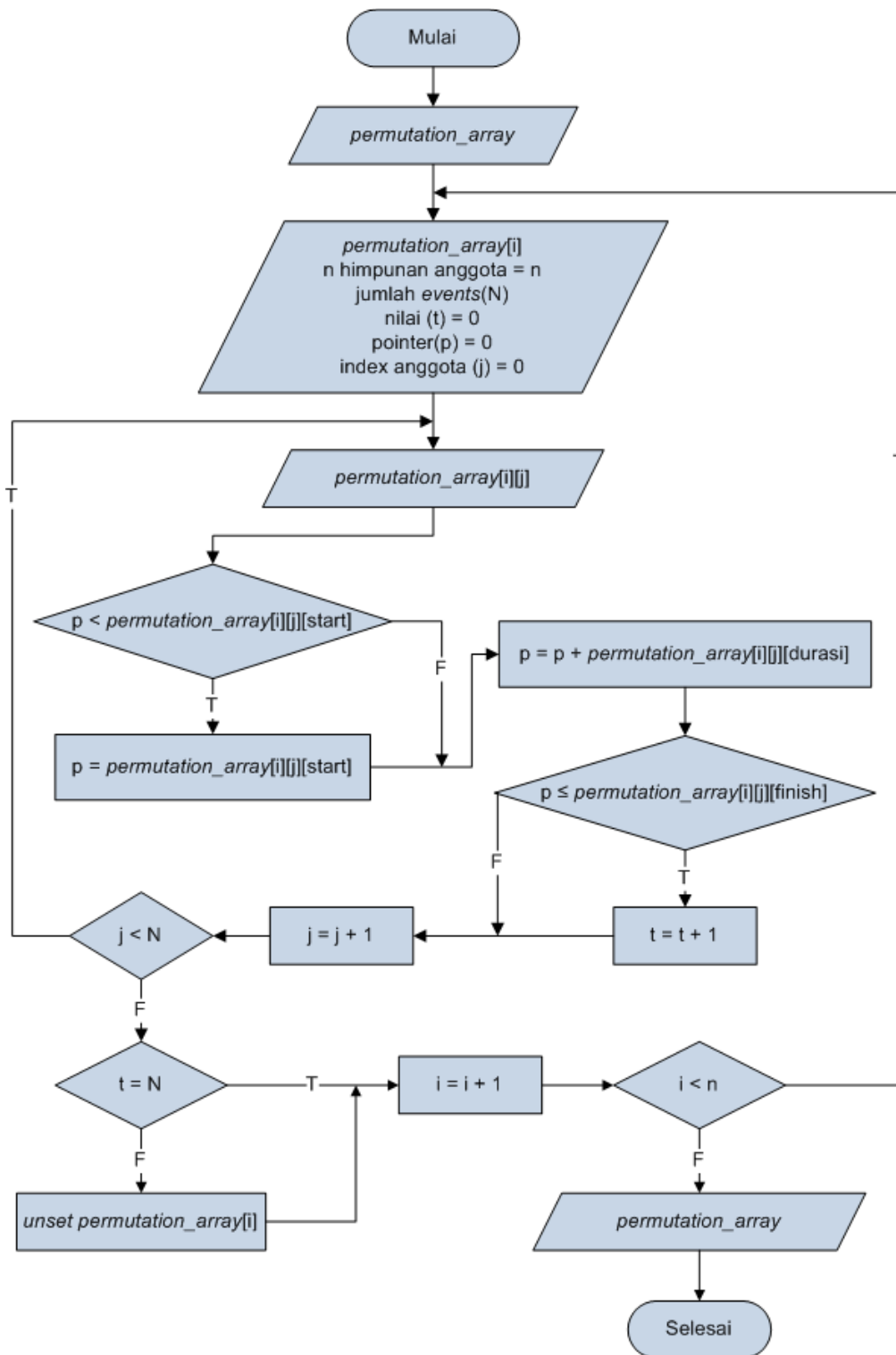
Proses pertama dalam rangkaian proses penyusunan *Agenda* adalah proses pembuatan permutasi dari seluruh *tasks* yang ada dalam satu hari yang bersangkutan ke dalam sebuah *array* agar nantinya mudah untuk diproses lebih lanjut. Inisialisasi awal dari proses ini adalah jumlah *tasks* hari terpilih, data *tasks* hari yang terpilih, *tasks array* untuk menampung data *tasks*, dan *array* semua permutasi yang akan digunakan untuk menampung hasil dari keseluruhan proses permutasi ini. *Flow chart* permutasi *tasks* ditunjukkan pada gambar 4.6.



Gambar 4.6 *Flow Chart* Permutasi *Tasks*.

4.2.4 *Flow Chart* Seleksi Set Permutasi Yang Mungkin

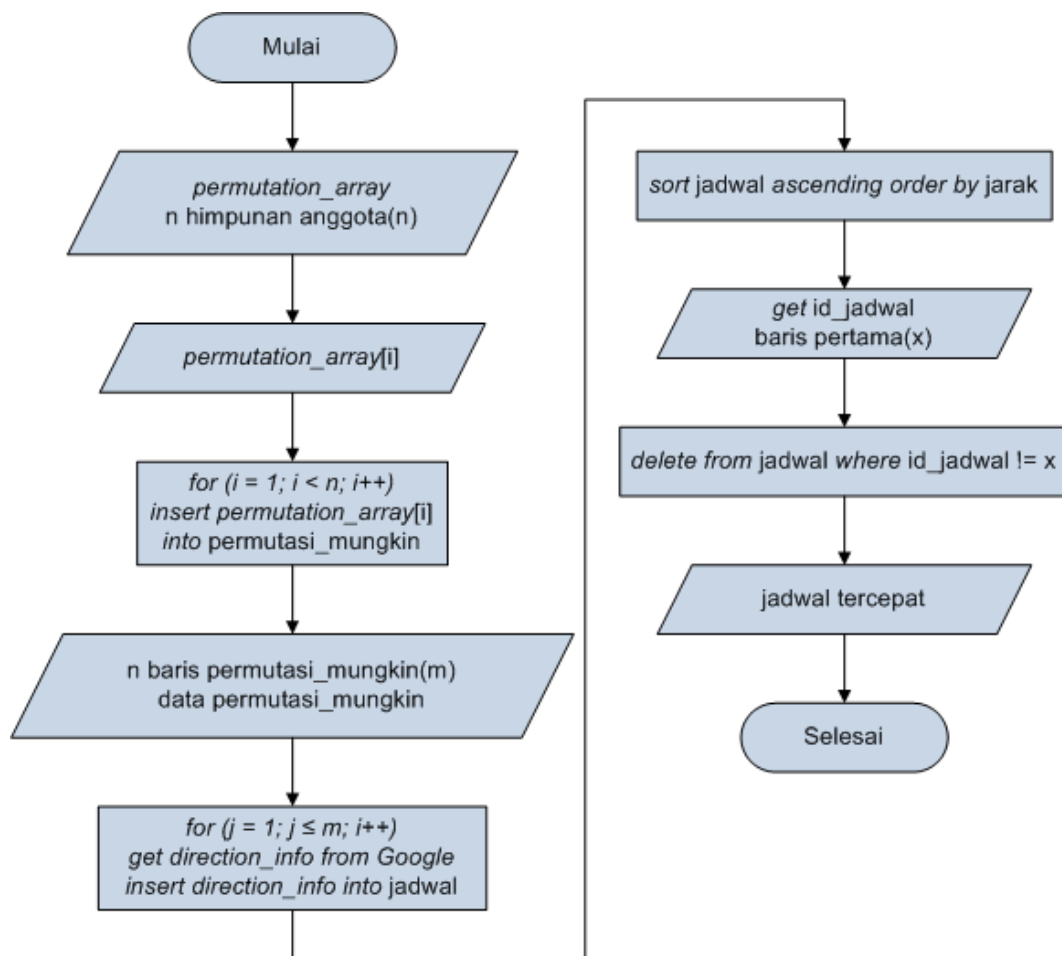
Proses selanjutnya dalam rangkaian penyusunan *agenda* adalah penyeleksian himpunan anggota *permutation_array*. Pada tahap ini seleksi diperlukan untuk membuang(*unset*) himpunan anggota permutasi yang tidak memungkinkan untuk dilakukan. Inisialisasi awal proses ini adalah *permutation_array*. Saat inisialisasi, *permutation_array* berisi semua hasil permutasi dari proses sebelumnya. Data turunan dari *array* tersebut adalah himpunan-himpunan anggota permutasi yang merupakan elemen-elemen dari *permutation_array*. Data yang dibutuhkan untuk inisialisasi berikutnya adalah jumlah anggota himpunan (n), jumlah *tasks* (N), nilai penjejak (t), pointer (p), dan index anggota himpunan (j). Terdapat dua proses rekursif dalam keseluruhan proses ini yaitu proses rekursif tingkat himpunan anggota permutasi dan proses rekursif tingkat anggota himpunan. *Flow chart* seleksi set permutasi yang mungkin ditunjukkan pada gambar 4.7.



Gambar 4.7 *Flow Chart* Seleksi Set Permutasi Yang Mungkin.

4.2.5 Flow Chart Seleksi Jadwal Terpendek

Tahap terakhir dari rangkaian penyusunan *agenda* adalah proses pemilihan data jadwal yang memiliki total jarak paling pendek. Pada diagram alir di bawah ini ditunjukkan pemilihan jadwal tercepat yaitu dengan cara pengurutan yang kemudian diambil baris teratasnya. Data yang dibutuhkan untuk inisialisasi adalah *permutation_array* dan jumlah himpunan anggota dalam *array* tersebut. *Flow chart* seleksi jadwal terpendek ditunjukkan pada gambar 4.8.



Gambar 4.8 *Flow Chart* Seleksi Jadwal Terpendek.

4.3 Perancangan Tabel Basis Data

4.3.1 Struktur Tabel

Sistem ini akan menggunakan basis data relasional. Ada beberapa tabel basis data yang diperlukan dalam sistem ini, tabel-tabel tersebut antara lain:

1. Tabel *user*

Tabel *user* diperlukan untuk menyimpan data akun pengguna termasuk data *login*. Struktur dari tabel tema dapat dilihat pada tabel 4.1.

Tabel 4.1 Tabel *user*.

nama field	tipe	panjang	keterangan
id_user	integer	11	id unik <i>user</i>
name	varchar	128	nama <i>user</i>
email	varchar	64	<i>user email</i>
password	varchar	32	<i>user password</i>
confirmcode	varchar	32	kode konfirmasi registrasi
lat_center	float	10,6	<i>default latitude</i>
lng_center	float	10,6	<i>default longitude</i>
zoom	integer	2	<i>default zoom level</i>

2. Tabel *markers*

Tabel *markers* diperlukan untuk menyimpan data *markers* pengguna. Struktur dari tabel *markers* dapat dilihat pada tabel 4.2.

Tabel 4.2 Tabel *markers*.

nama field	tipe	panjang	keterangan
id_marker	integer	11	id unik <i>marker</i>
name	varchar	60	nama <i>marker</i>
address	varchar	80	alamat <i>marker</i>
lat	float	10,6	koordinat <i>marker latitude</i>
lng	float	10,6	koordinat <i>marker longitude</i>
type	varchar	30	tipe <i>marker</i>

id_user	integer	11	<i>foreign key id_user</i>
description	varchar	255	deskripsi <i>marker</i>

3. Tabel *events*

Tabel *user* diperlukan untuk menyimpan data *task* pengguna. Struktur dari tabel *events* dapat dilihat pada tabel 4.3.

Tabel 4.3 Tabel *events*.

nama field	tipe	panjang	keterangan
id_event	integer	8	id unik <i>task</i>
id_user	integer	11	<i>foreign key id_user</i>
id_marker	integer	10	<i>foreign key id_marker</i>
title_event	varchar	50	judul <i>task</i>
detail_event	text	-	penjelasan <i>task</i>
start_event	integer	10	waktu mulai <i>task</i>
end_event	integer	10	waktu berakhir <i>task</i>
durasi_event	integer	10	durasi <i>task</i>
status_event	integer	1	status <i>task</i>
status_remove_event	integer	1	status penghapusan <i>task</i>
status_edit_event	integer	1	status perubahan <i>task</i>

4. Tabel permutasi

Tabel permutasi diperlukan untuk menyimpan hasil permutasi dari seluruh tugas-tugas pada hari terpilih. Struktur dari tabel permutasi dapat dilihat pada tabel 4.4.

Tabel 4.4 Tabel permutasi.

nama field	tipe	panjang	keterangan
id_permut	integer	10	id unik himpunan anggota
id_user	integer	11	<i>foreign key id_user</i>
date_permut	integer	10	tanggal sasaran permutasi

element1	integer	8	elemen pertama anggota
element2	integer	8	elemen kedua anggota
element3	integer	8	elemen ketiga anggota
element4	integer	8	elemen keempat anggota
element5	integer	8	elemen kelima anggota
element6	integer	8	elemen keenam anggota
element7	integer	8	elemen ketujuh anggota

5. Tabel permutasi_mungkin

Tabel permutasi_mungkin diperlukan untuk menyimpan himpunan anggota permutasi yang memungkinkan untuk dilakukan. Struktur dari tabel permutasi_mungkin dapat dilihat pada tabel 4.5.

Tabel 4.5 Tabel permutasi_mungkin.

nama field	tipe	panjang	keterangan
id_permut_m	integer	10	id unik himpunan anggota
id_user	integer	11	<i>foreign key</i> id_user
date_permut_m	integer	10	tanggal sasaran permutasi
element_m_1	integer	8	elemen pertama anggota
element_m_2	integer	8	elemen kedua anggota
element_m_3	integer	8	elemen ketiga anggota
element_m_4	integer	8	elemen keempat anggota
element_m_5	integer	8	elemen kelima anggota
element_m_6	integer	8	elemen keenam anggota
element_m_7	integer	8	elemen ketujuh anggota
status_permut_m	integer	1	status himpunan anggota

6. Tabel jadwal

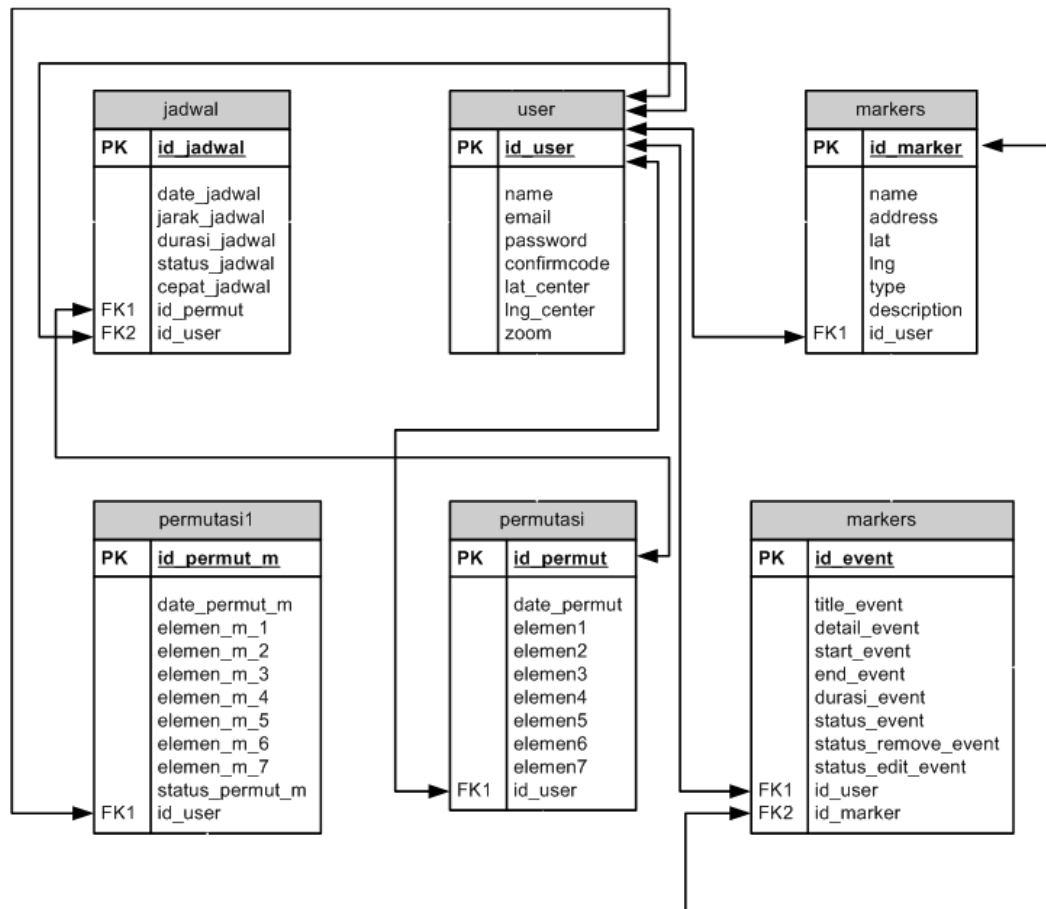
Tabel jadwal diperlukan untuk menyimpan data-data jadwal seperti tanggal, jarak dan durasi *agenda* pada hari terpilih. Struktur dari tabel jadwal dapat dilihat pada tabel 4.6.

Tabel 4.6 Tabel jadwal.

nama field	tipe	panjang	keterangan
id_jadwal	integer	11	id unik jadwal
id_permut	integer	10	<i>foreign key</i> id_permut
id_user	integer	11	<i>foreign key</i> id_user
date_jadwal	integer	10	tanggal jadwal
jarak_jadwal	integer	10	jarak tempuh <i>agenda</i>
durasi_jadwal	integer	10	durasi tempuh <i>agenda</i>
status_jadwal	integer	1	status jadwal
cepat_jadwal	integer	1	status jadwal tercepat

4.3.2 Relasi Antar Tabel

Basis data merupakan kumpulan data yang saling berhubungan satu dengan yang lainnya. Data tersebut disimpan dalam sebuah *file* atau tabel yang memiliki keterangan atau atribut tertentu yang apabila direlasikan akan menghasilkan informasi yang lebih kompleks. Keberadaan relasi tabel dapat memudahkan dalam pemeliharaan data dan menghindarkan dari kerangkapan data (*redundancy*) sehingga data yang disimpan dan dihasilkan akan lebih efektif. Relasi antar tabel aplikasi *Penjadwalan Berbasis Lokasi* ditunjukkan pada gambar 4.9.



Gambar 4.9 Relasi antar tabel.

4.4 Perancangan Antarmuka

Antarmuka yang akan dirancang adalah hanya antarmuka dari sisi *user*. Antarmuka untuk *administrator* sudah disediakan oleh *phpMyAdmin*.

4.4.1 Rancangan Antarmuka *Login* dan *Register*

Halaman ini adalah halaman pertama yang akan dijumpai oleh pengguna saat mengunjungi sistem ini. Terdapat dua *form* dalam rancangan halaman pada gambar 4.10 yaitu *form* untuk registrasi dan *form* untuk *login*. Di bawah *form* untuk *login* terdapat *link* lupa *password* yang mengarahkan pengguna ke dialog *password recovery*.

Header	
<p>First name : <input style="width: 100%;" type="text" value="First name"/></p> <p>Last name : <input style="width: 100%;" type="text" value="Last name"/></p> <p>Email : <input style="width: 100%;" type="text" value="Email"/></p> <p>Password : <input style="width: 100%;" type="text" value="Password"/></p> <p style="text-align: center;"><input type="button" value="Sign Up"/></p>	<p>Email : <input style="width: 100%;" type="text" value="email"/></p> <p>Password : <input style="width: 100%;" type="text" value="password"/></p> <p style="text-align: center;"><input type="button" value="Login"/></p> <p>Forgot Password?</p>
Footer	

Gambar 4.10 Rancangan antarmuka halaman *login* dan *register*.

4.4.2 Rancangan Antarmuka Ganti *Password*

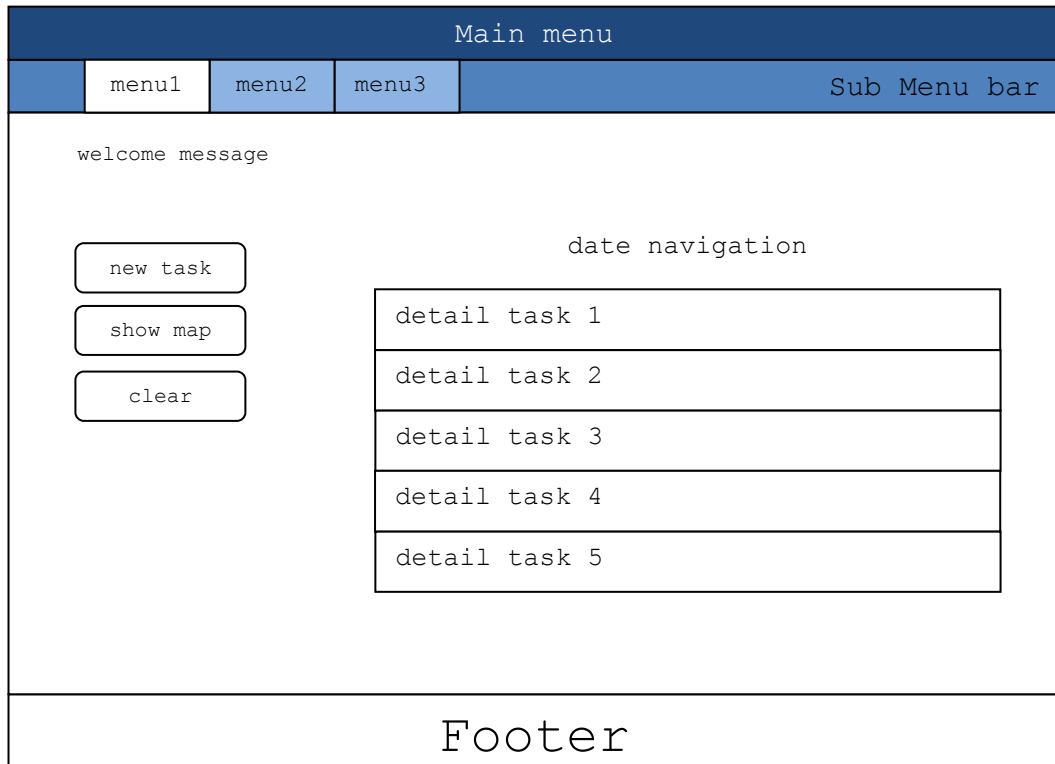
Halaman ini berfungsi untuk mengganti *password* pada akun pengguna yang bersangkutan. Seorang pengguna diharuskan berhasil *logged in* terlebih dahulu untuk dapat menggunakan fungsi penggantian *password* ini. Rancangan antarmuka ganti *password* ditunjukkan pada gambar 4.11.

Header	
Old Password :	<input type="text" value="old password"/>
New Password :	<input type="text" value="new password"/>
Re-enter New Password :	<input type="text" value="new password again"/>
<input type="button" value="Submit"/>	
Footer	

Gambar 4.11 Rancangan antarmuka halaman ganti *password*.

4.4.3 Rancangan Antarmuka *Home* dan *Overview Jadwal*

Halaman ini akan tertampil saat pengguna berhasil *login*. Fungsi utama halaman *overview* jadwal adalah untuk menampilkan informasi urutan tugas-tugas yang telah disusun oleh sistem kepada pengguna. Rancangan antarmuka *home* ditunjukkan pada gambar 4.12.



Gambar 4.12 Rancangan Antarmuka *Home* dan *Overview Jadwal*.

4.4.4 Rancangan Antarmuka *Insert Task*

Halaman ini berisi form yang digunakan pengguna untuk pemasukan data *task* baru. Semua *field* yang dibutuhkan sebagai pemasukkan *task* baru harus diisi oleh pengguna. Sesaat setelah pengguna menekan tombol *submit*, data *task* yang baru akan diproses yang selanjutnya untuk keperluan penyusunan jadwal yang baru dengan disertakannya *task* baru yang baru saja dimasukkan. Rancangan antarmuka tambah tugas ditunjukkan pada gambar 4.13.

Main menu

menu1 menu2 menu3 Sub Menu bar

welcome message

new task Task name : task name

show map Task detail : task detail

clear At : marker ▼

Time

From : hh mm

To : hh mm

Duration : mm minutes

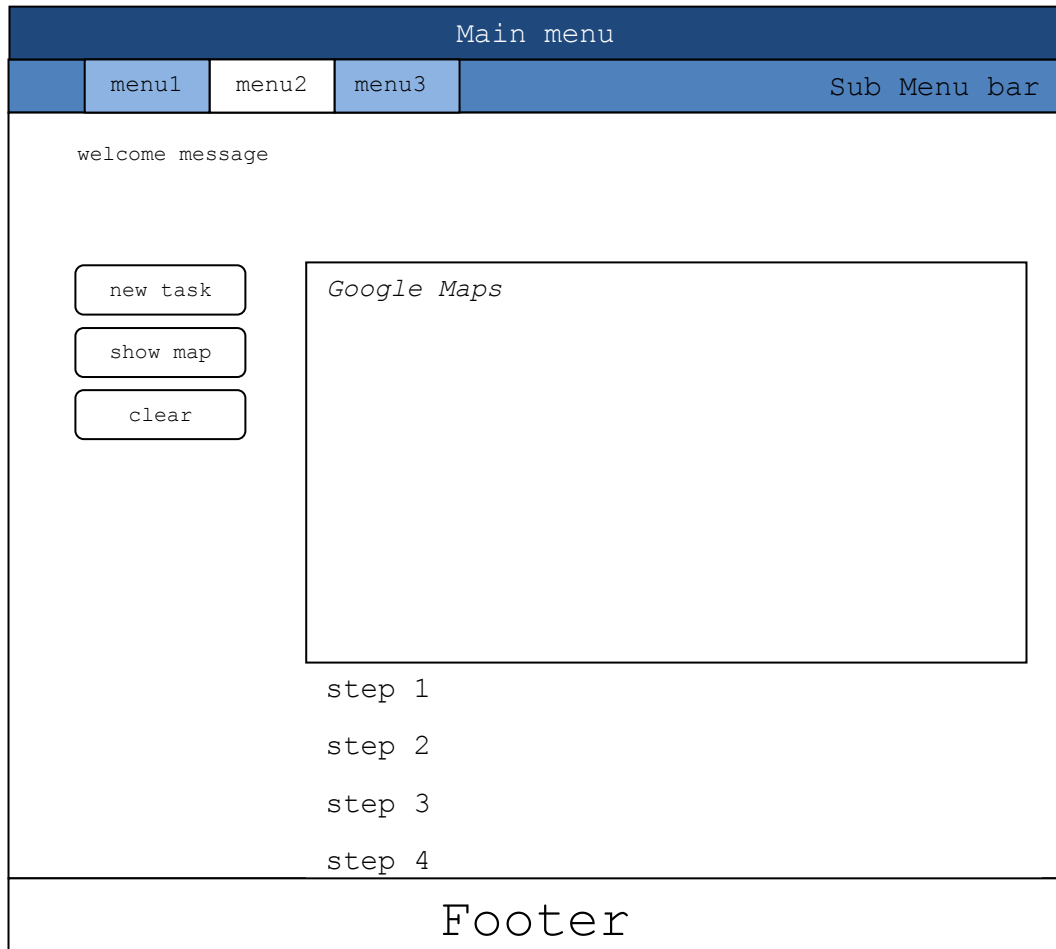
submit

Footer

Gambar 4.13 Rancangan Antarmuka *Insert Task*.

4.4.5 Rancangan Antarmuka *Direction Map*

Halaman ini menampilkan visualisasi rute dalam peta *Google*. Pada halaman ini juga akan ditampilkan detail urutan petunjuk jalan berupa teks dari *marker* tugas pertama hingga *marker* tugas terakhir pada jadwal hari terpilih. Rancangan antarmuka *direction map* ditunjukkan pada gambar 4.14.



Gambar 4.14 Rancangan Antarmuka *Direction Map*.

4.4.6 Rancangan Antarmuka *Task Search*

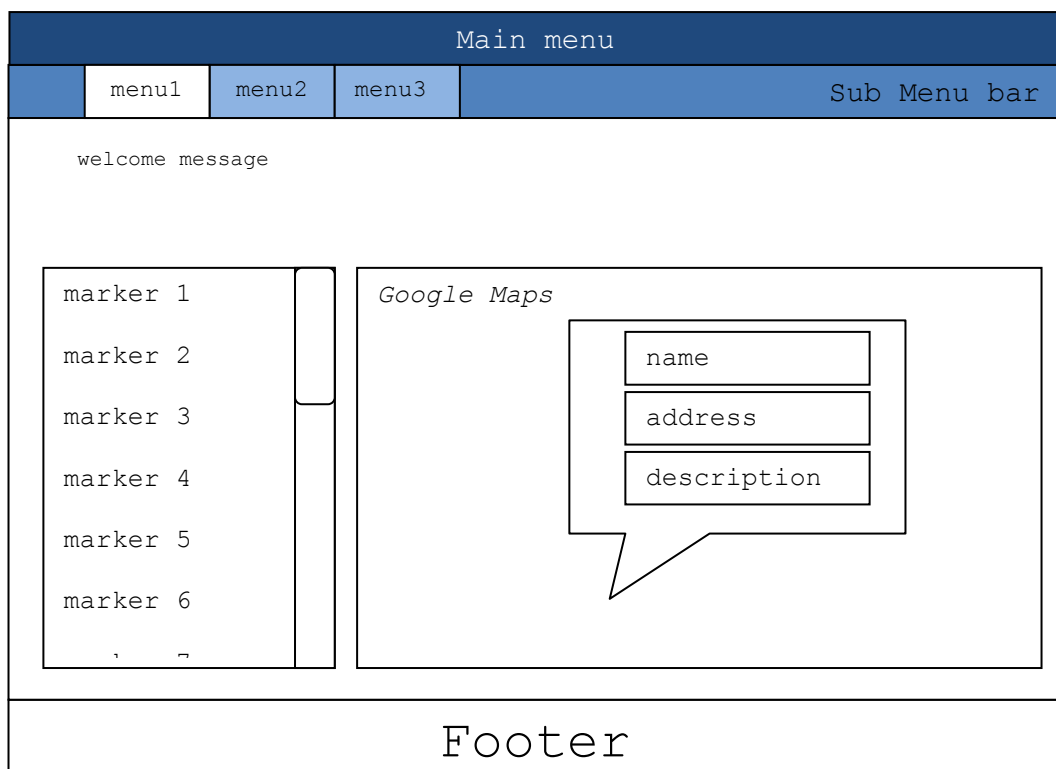
Halaman ini memiliki fungsi untuk mencari *tasks* berdasarkan *keyword* yang dimasukkan. Pencarian *task* ini dapat seleksi dengan lebih spesifik dengan mengisi *drop-down marker field*. Rancangan antarmuka pencarian tugas ditunjukkan pada gambar 4.15.

The wireframe illustrates the layout of the *Task Search* interface. It features a dark blue header with the text "Main menu". Below the header is a "Sub Menu bar" containing three buttons labeled "menu1", "menu2", and "menu3". The main content area starts with a "welcome message". On the left side, there are three buttons: "new task", "show map", and "clear". In the center, the text "task search :" is followed by a "keyword field" and a "search" button. Below the keyword field is a "marker field" with a dropdown arrow icon. The footer area contains the text "Footer".

Gambar 4.15 Rancangan Antarmuka *Task Search*.

4.4.7 Rancangan Antarmuka *My Markers*

Halaman ini menyajikan daftar *markers* yang telah dimasukkan oleh pengguna. Sistem menyajikan *markers* tersebut dalam dua bentuk yaitu dengan *list* dan *markers* di *Google Maps*. Keduanya bentuk tersebut saling berhubungan sehingga jika salah satu *marker* dalam *list* sebelah kiri di-klik, maka *info window* untuk *marker* bersangkutan akan tertampil. Rancangan antarmuka *my markers* ditunjukkan pada gambar 4.16.



Gambar 4.16 Rancangan Antarmuka *My Markers*.

4.4.8 Rancangan Antarmuka *Add Marker*

Halaman ini berfungsi untuk menambah atau memasukkan suatu *marker* untuk pengguna bersangkutan. Pada rancangan antarmuka tambah *marker* gambar 4.17 terdapat dua metode untuk menambahkan suatu *marker*, yaitu cara manual dengan memasukkan koordinat serta informasi lain di form sebelah kiri maupun dengan cara meng-klik langsung pada *Google Maps*. Cara pertama memerlukan data tambahan dari pengguna yaitu koordinat posisi *marker* dalam bentuk *latitude* dan *longitude*, judul *marker* serta deskripsi *marker* jika diperlukan, sedangkan cara kedua yaitu dengan menunjuk langsung sebuah titik pada peta dengan satu kali klik kiri lalu satu kali klik kiri lagi untuk memunculkan *info window* sehingga *user* dapat melengkapinya dengan atribut *marker* yang bersangkutan.

The wireframe shows a web interface for adding markers. At the top is a dark blue 'Main menu' bar containing three buttons: 'menu1', 'menu2', and 'menu3'. To the right of these buttons is a 'Sub Menu bar'. Below the menu is a 'welcome message'. The main content area is split into two columns. The left column, titled 'manually insert', contains five text input fields labeled 'latitude', 'longitude', 'name', 'address', and 'description', with a 'submit' button at the bottom. The right column, titled 'Google Maps', contains three text input fields labeled 'name', 'address', and 'description', and a 'save' button. A speech bubble points from the 'save' button to the 'Google Maps' section. At the bottom of the page is a 'Footer' bar.

Gambar 4.17 Rancangan Antarmuka *Add Marker*.

4.4.9 Rancangan Antarmuka *Map Settings*

Halaman ini memiliki fungsi untuk menentukan di mana titik tengah *Google Maps* yang ditampilkan pada halaman *My Markers* dan *Add Marker* serta tingkat *zoom*-nya. Rancangan antarmuka *map settings* ditunjukkan pada gambar 4.18.

Main menu			
menu1	menu2	menu3	Sub Menu bar
welcome message			
default zoom :			
marker field			▼
map center :			
latitude			
longitude			
save			
Footer			

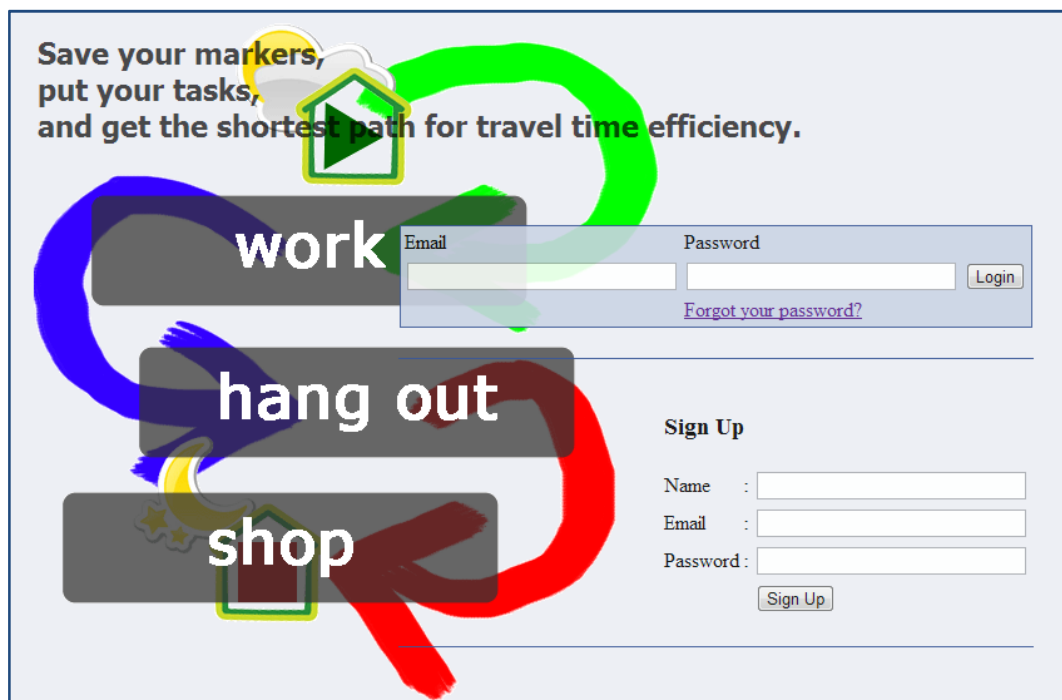
Gambar 4.18 Rancangan Antarmuka *Map Settings*.

BAB V IMPLEMENTASI DAN PENGUJIAN

5.1 Implementasi Sistem

5.1.1 Halaman *Login Dan Register*

Halaman ini merupakan halaman pertama yang akan dilihat pengguna saat mengunjungi alamat *web* aplikasi ini. Halaman ini berisi *form* untuk *login* dan *register*. Bagi pengguna yang lupa dengan *password* mereka, terdapat *link* ke *forgot password* untuk mengirim *password* baru ke alamat *email* pengguna bersangkutan. Gambar 5.1 menunjukkan halaman *login* dan *register* aplikasi *Penjadwalan Berbasis Lokasi*.



Gambar 5.1 Halaman *Login dan Register*.

5.1.2 Halaman *Agenda*

Halaman *Agenda* adalah halaman *home* untuk *user* yang berhasil masuk. Halaman ini akan mengarahkan *user* untuk menentukan titik berangkat dan titik pulang jika dalam hari yang dipilih belum ada *agenda* yang terbentuk. Halaman *agenda* yang masih kosong ditunjukkan pada gambar 5.2. Halaman *agenda* akan

menampilkan susunan acara dalam satu hari atau 24 jam pada tanggal terpilih. Halaman *agenda* yang telah terisi susunan tugas-tugas ditunjukkan pada gambar 5.3.

Agenda Markers Account Settings Logout

Tasks Map Search

Welcome, Dzaky!
Today : [Wednesday 7th of March 2012](#)
Showing : Wednesday 7th of March 2012

Mar 7 2012

Go

[New task](#) [< previous](#) | 7th of March 2012 | [next >](#)

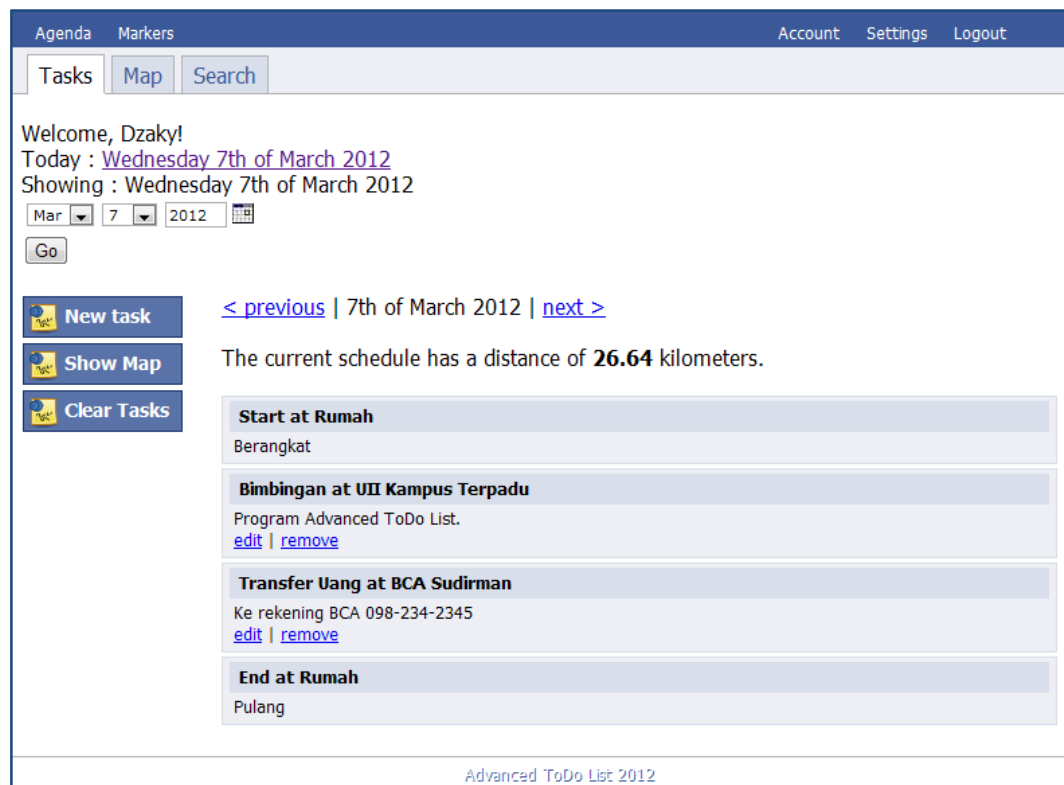
Specify your start point Specify your end point

Save Start-End Points

There is no schedule on showing date!
Before insert new task, please specify your start and end point in form above.

Advanced ToDo List 2012

Gambar 5.2 Halaman *Agenda* yang masih kosong.



Gambar 5.3 Halaman *Agenda* yang telah terisi tugas-tugas.

5.1.3 Halaman Tambah Tugas

Setelah *user* menentukan titik berangkat dan pulang, maka langkah selanjutnya untuk membuat jadwal adalah mulai menyisipkan atau menambah tugas. Proses pemasukan tugas terdiri dari tiga tahap. Tahap pertama adalah memasukkan atribut-atribut tugas seperti nama tugas, detail tugas, tempat, waktu dan durasi, ditunjukkan pada gambar 5.4. Tahap kedua proses pemasukan tugas baru adalah *review* jadwal sebelumnya oleh sistem kepada *user*, ditunjukkan pada gambar 5.5. Tahap terakhir pemasukan tugas baru adalah *review* jadwal baru setelah tugas baru dimasukkan, ditunjukkan pada gambar 5.6. Dalam tahap ketiga tersebut, *user* diberi pilihan untuk tetap melanjutkan proses pemasukan tugas baru dan sistem akan menggantikan jadwal sebelumnya dengan jadwal yang baru atau membatalkan proses pemasukan tugas baru dan mempertahankan jadwal yang sudah ada.

Agenda Markers Account Settings Logout

Tasks Map Search

Welcome, Dzaky!
 Today : [Wednesday 7th of March 2012](#)
 Showing : Wednesday 7th of March 2012

Mar 7 2012

Go

New task **Event Title** : Nonton

Show Map **Event Detail** : GHOST RIDER : SPIRIT OF VENGEANCE 3D

Clear Tasks **At** : Cinema XXI

Time
 From : 18 00
 To : 22 00

Duration : 120 minutes

Insert Event

Advanced ToDo List 2012

Gambar 5.4 Halaman Tambah Tugas (Tahap 1).

Agenda Markers Account Settings Logout

Tasks Map Search

Welcome, Dzaky!
 Today : [Wednesday 7th of March 2012](#)
 Showing : Wednesday 7th of March 2012

Mar 7 2012

Go

New task New task successfully inserted!
 Previous schedule can be reached by 26.64 kilometers.

Show Map Inserted task :

Clear Tasks **Nonton at 19**
 GHOST RIDER : SPIRIT OF VENGEANCE 3D

Process now

Advanced ToDo List 2012

Gambar 5.5 Halaman Tambah Tugas (Tahap 2).



Gambar 5.6 Halaman Tambah Tugas (Tahap 3).

5.1.4 Halaman *Edit* Tugas

Tugas-tugas yang telah dimasukkan oleh *user* dapat di-*edit* atribut-atributnya. Terdapat dua metode perubahan tugas pada halaman ini. Metode yang pertama adalah *edit* dengan tidak mempengaruhi jadwal yang sudah terbentuk. Metode pertama ini terbatas hanya untuk perubahan nama tugas dan detail tugas. Metode kedua yaitu dengan memperbolehkan *user* mengubah seluruh atribut-atribut tugas yang bersangkutan. Metode kedua ini akan mempengaruhi jadwal sehingga setelah *user* melakukan perubahan pada metode dua, sistem akan mengarahkan *user* untuk melakukan *schedule update* atau pembaruan jadwal. Gambar 5.7 menunjukkan halaman yang menampilkan kedua metode *edit* tugas di atas. Gambar 5.8 dan gambar 5.9 menunjukkan sistem mengarahkan *user* untuk memperbarui jadwal yang sudah ada.

Agenda Markers Account Settings Logout

Tasks Map Search

Welcome, Dzaky!
Today : [Wednesday 7th of March 2012](#)
Showing : Wednesday 7th of March 2012

Mar 7 2012

Go

New task

Show Map

Clear Tasks

Basic edit (skip schedule update)

Event Title : Nonton

Event Detail : GHOST RIDER : SPIRIT OF VENGEANCE 3D

Submit

Schedule affected edit

Event Title : Nonton

Event Detail : GHOST RIDER : SPIRIT OF VENGEANCE 3D

Place : Cinema XXI

Time

From : 18 00

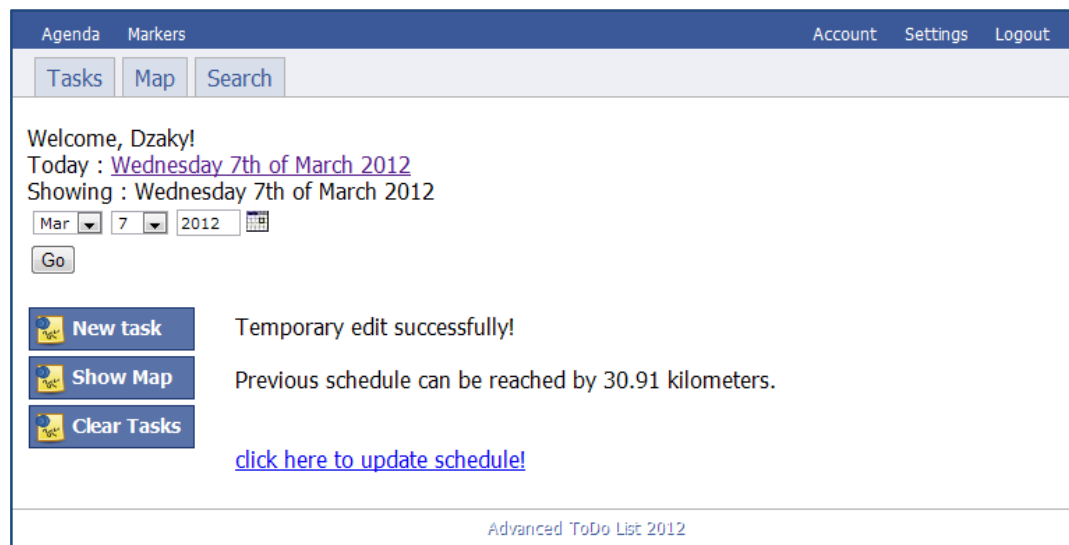
To : 23 00

Duration : 120 minutes

Submit

Advanced ToDo List 2012

Gambar 5.7 Halaman *Edit* Tugas (Tahap 1).



Gambar 5.8 Halaman *Edit* Tugas (Tahap 2).



Gambar 5.9 Halaman *Edit* Tugas (Tahap 3).

5.1.5 Halaman Direction Map

Halaman ini berisi visualisasi rute perjalanan jadwal *user* pada *Google Maps* beserta detail langkah-langkah perjalanannya. Pada peta terdapat tombol-tombol untuk *zoom in* atau *zoom out* serta terdapat tiga pilihan tampilan permukaan peta antara lain peta jalan, citra satelit dan gabungan antara peta jalan dan citra satelit (*hybrid*). Tampilan halaman *Direction Map* ditunjukkan pada gambar 5.10.

Welcome, Dzaky!
 Today : [Friday 27th of April 2012](#)
 Showing : Wednesday 7th of March 2012
 Mar 7 2012
 Go

New task
 Show Map
 Clear Tasks

Map Satellite Hybrid

Start at RumahQu - Unknown road

8.5 km (about 14 mins)

1. Head southeast toward Jalan Mentawai	140 m
2. Continue straight onto Jalan Timor Timor	350 m
3. Turn left onto Jalan Kaliurang	7.7 km
4. Turn left	350 m

Gambar 5.10 Halaman *Direction Map*.

5.1.6 Halaman Pencarian Tugas

Saat seorang *user* telah memiliki banyak tugas-tugas baik pada tanggal yang telah terlalui maupun yang akan datang tentunya dibutuhkan fasilitas yang memungkinkan suatu *user* dapat menemukan suatu tugas. Halaman ini berfungsi untuk melakukan pencarian tugas-tugas berdasarkan nama tugas, detail tugas maupun keduanya. *Form* pencarian ini ditunjukkan pada gambar 5.11. Hasil pencarian tugas-tugas ini disusun dimulai dari yang paling akhir menurut tanggalnya. Halaman yang menampilkan hasil pencarian tugas-tugas ditunjukkan pada gambar 5.12.



Gambar 5.11 Halaman Pencarian Tugas (Tahap 1).

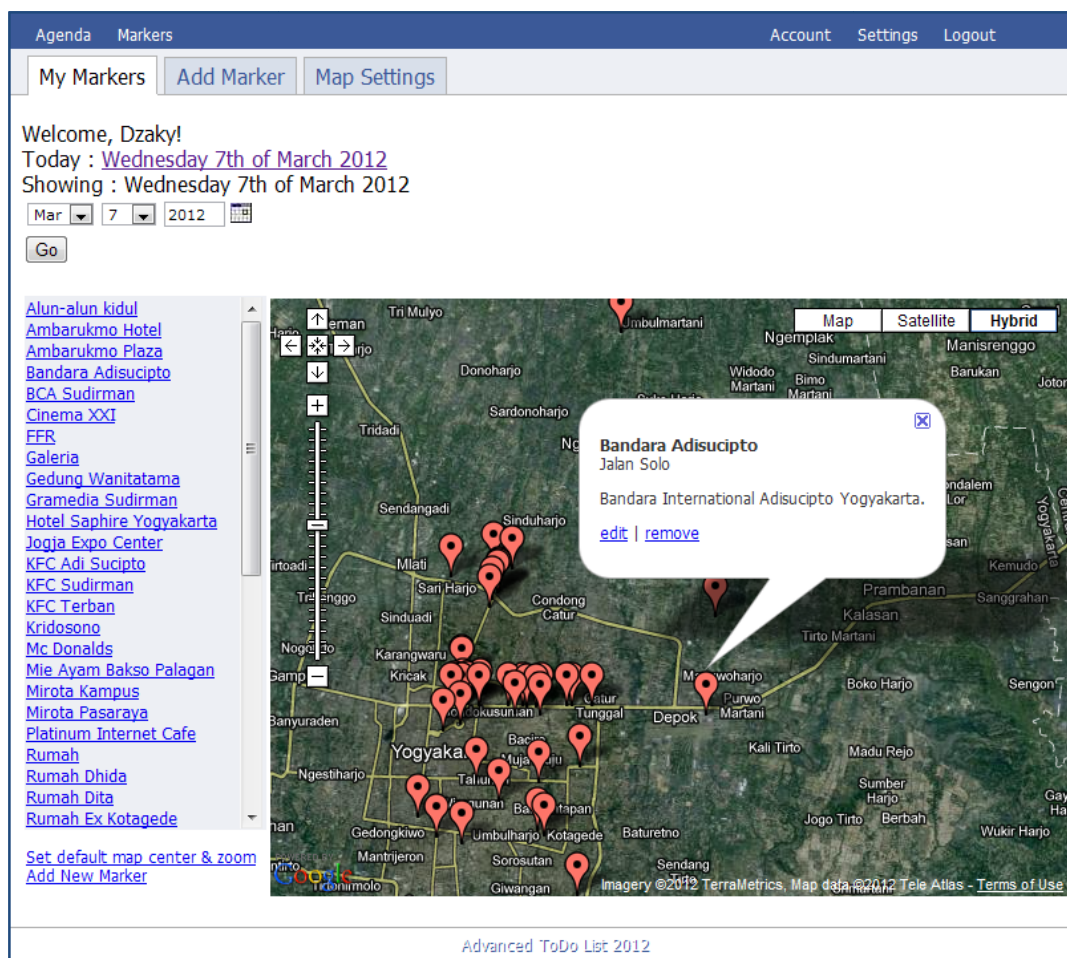


Gambar 5.12 Halaman Pencarian Tugas (Tahap 2).

5.1.7 Halaman *My Markers*

Halaman *My Markers* berisi semua *markers* yang telah dimasukkan oleh *user* yang bersangkutan. Terdapat dua bagian dalam halaman ini yaitu bagian daftar sebelah kiri dan peta sebelah kanan. Pada bagian kiri menunjukkan daftar seluruh

markers yang dimiliki *user* dan pada bagian kanan adalah visualisasi *markers* pada *Google Maps* yang dilambangkan dengan ikon merah. Kedua bagian di atas saling berhubungan oleh perintah memunculkan *info window* pada peta. Hubungannya adalah jika *user* memilih salah satu *marker* pada bagian kiri, maka akan ditampilkan *marker info window* yang bersangkutan. Pada halaman ini juga terdapat *link* untuk menuju ke halaman *atur map center* dan *map zoom* untuk *user* yang menghendaki mengubah titik tengah peta serta perbesarannya. Halaman *My Markers* ditunjukkan pada gambar 5.13.

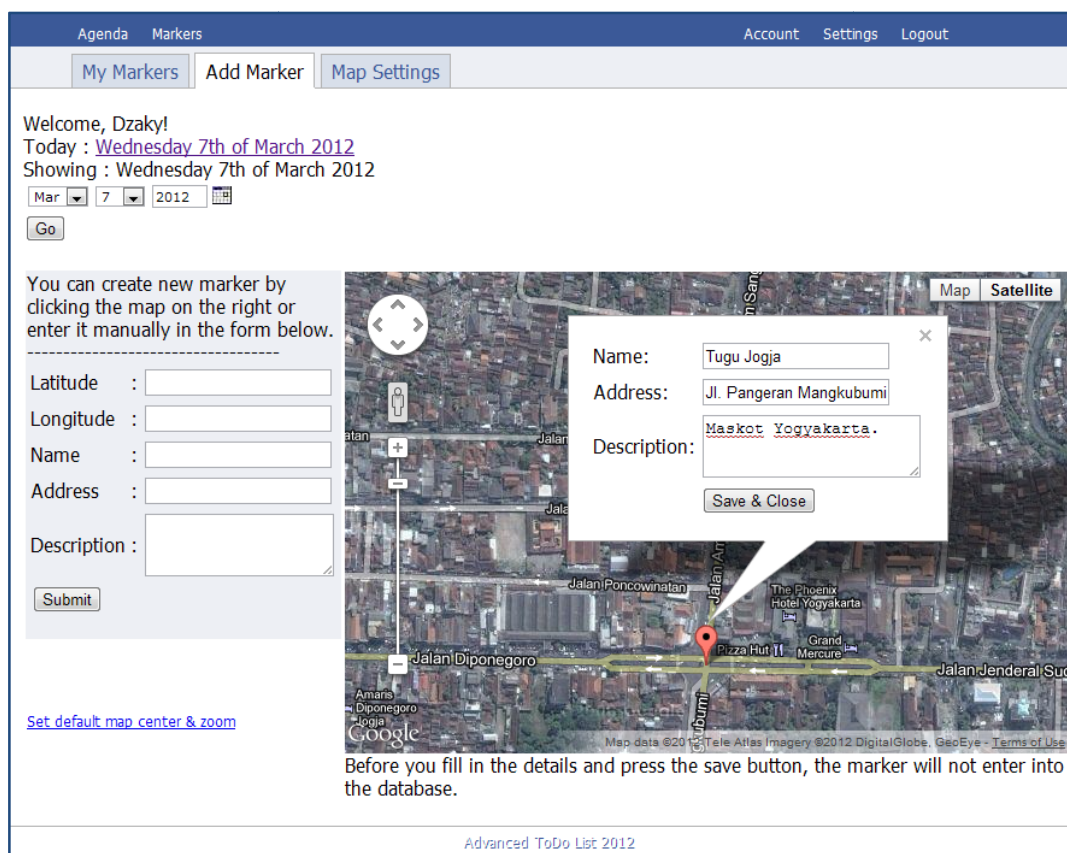


Gambar 5.13 Halaman *My Markers*.

5.1.8 Halaman Tambah *Marker*

Halaman tambah *marker* memungkinkan *user* untuk menambah *marker* atau penanda pada akun mereka. Ada dua metode untuk menambahkan penanda ini yaitu dengan cara manual dan cara klik pada peta. Pada cara manual, *user*

diharuskan mengetahui koordinat *latitude* dan *longitude* penanda yang akan dimasukkan. Pada metode klik pada peta, *user* hanya diharuskan mengisi nama, alamat dan deskripsi *marker*, sedangkan untuk koordinatnya sudah ditentukan saat *user* mengklik peta. Pada halaman ini juga terdapat *link* untuk menuju ke halaman atur *map center* dan *map zoom* untuk *user* yang menghendaki mengubah titik tengah peta serta perbesarannya. Halaman tambah *marker* ditunjukkan pada gambar 5.14.



Gambar 5.14 Halaman Tambah *Marker*.

5.1.9 Halaman Ubah *Marker*

Halaman ubah *marker* berfungsi untuk mengubah beberapa atribut dari suatu *marker*. Hanya nama, detail dan deskripsi *marker* yang dapat diubah oleh *user*. Sistem tidak memperbolehkan pengubahan *latitude* dan *longitude* suatu *marker* karena akan mempengaruhi posisi *marker* tersebut sehingga akan mempengaruhi juga jadwal yang menggunakannya. Jika *user* tetap akan merubahnya maka hal itu

hanya dapat dilakukan dengan menghapusnya lalu membuat *marker* yang baru. Halaman ubah *marker* ditunjukkan pada gambar 5.15.

Agenda Markers Account Settings Logout

My Markers Add Marker Map Settings

Welcome, Dzaky!
 Today : [Wednesday 7th of March 2012](#)
 Showing : Wednesday 7th of March 2012
 Mar 7 2012
 Go

Nama / Name : Bandara International Adis
 Alamat / Address : Jl. Solo Km.9 Yogyakarta
 Deskripsi / Description : Telp : +62 (0274) 512144, 560108, 560
 Fax : +62 (0274) 560155

Submit

[Set default map center & zoom](#) Anda tidak dapat mengubah koordinat latitude dan longitude suatu marker. Jika anda ingin melakukannya, hapus terlebih dahulu marker bersangkutan lalu buatlah yang baru. Penghapusan suatu marker akan berdampak ikut terhapusnya jadwal-jadwal yang di dalamnya terdapat marker bersangkutan.
[Add New Marker](#)

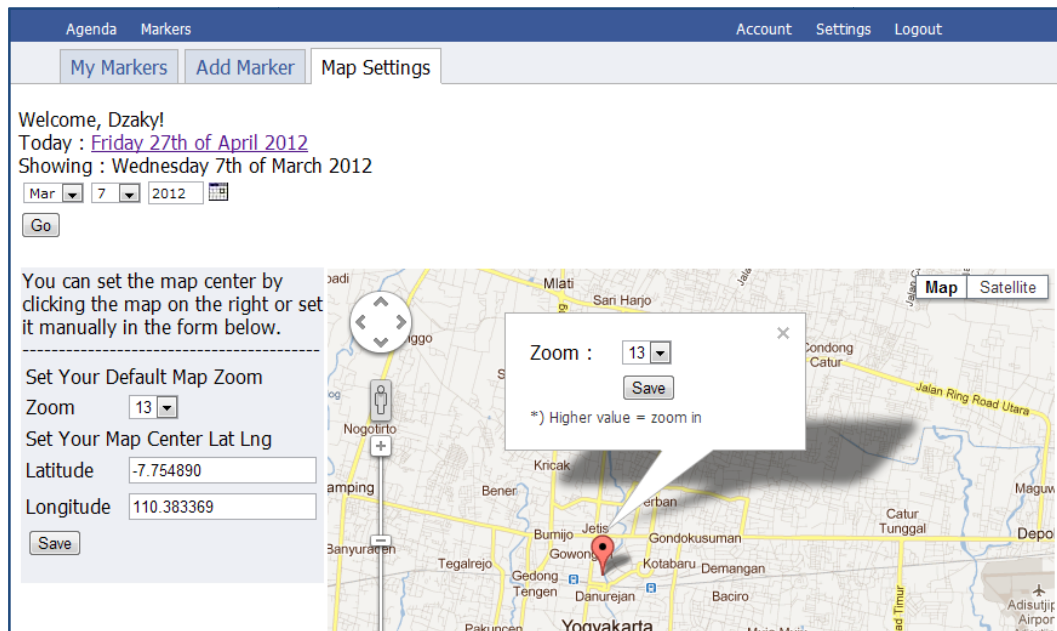
You can not change the latitude and longitude coordinates of a marker. If you want to do this, first remove the relevant marker, and remake new ones. Removal of a marker will affects deletion of schedules in which there is relevant marker.

Advanced ToDo List 2012

Gambar 5.15 Halaman Ubah *Marker*.

5.1.10 Halaman Map Settings

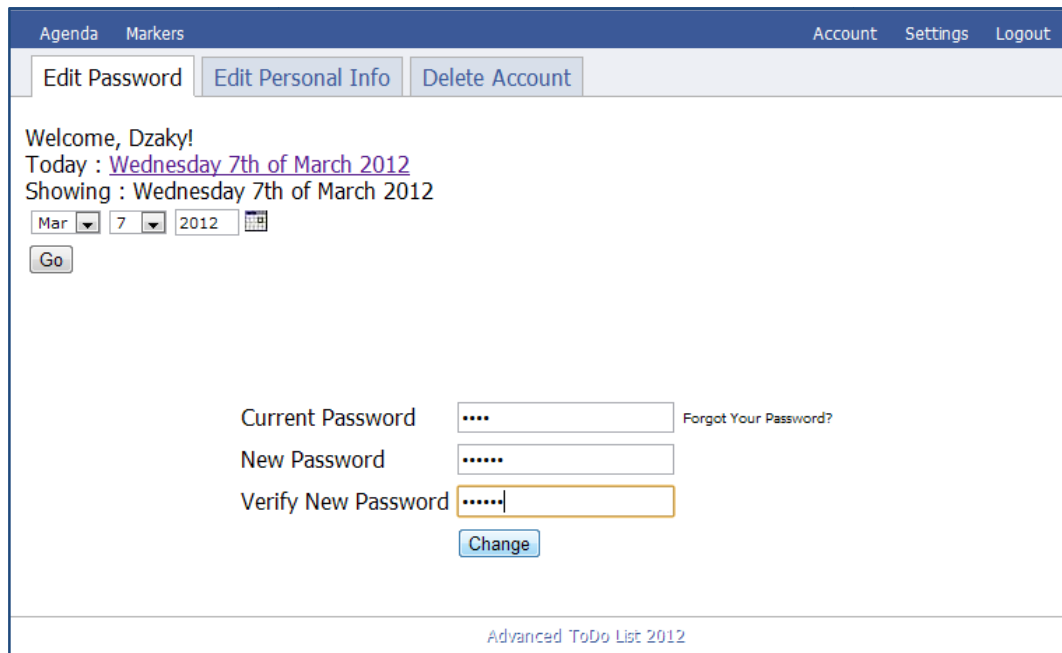
Halaman *Map Settings* adalah halaman yang berfungsi untuk merubah pengaturan titik tengah *Google Maps* serta tingkat *zoom*-nya. Pengaturan ini akan mempengaruhi tampilan *Google Maps* yang ada pada halaman *my markers*, tambah *marker* serta halaman *map settings* itu sendiri. Pengaturan ini dapat dilakukan dengan dua cara yaitu dengan input manual dan dengan klik pada peta. Halaman *map settings* ditunjukkan pada gambar 5.16.



Gambar 5.16 Halaman *Map Settings*.

5.1.11 Halaman Ganti *Password*

Halaman ini berfungsi untuk mengubah *password* akun pengguna. Setiap pengguna dapat mengubah *password* akun mereka setelah berhasil masuk (*logged in*). Halaman ganti *password* ditunjukkan pada gambar 5.17.



Gambar 5.17 Halaman Ganti *Password*.

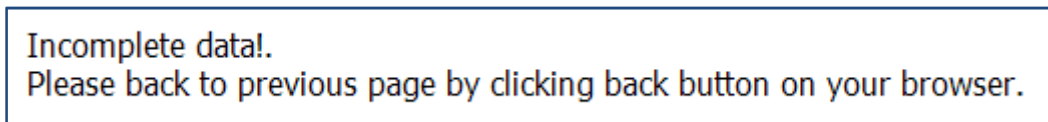
5.2 Pengujian

5.2.1 Penanganan Kesalahan

Aplikasi ini dibuat untuk mudah dimengerti oleh pengguna. Jika terdapat kesalahan-kesalahan pemasukan data ataupun proses yang akan dilakukan, maka sistem akan memberikan tanggapan kepada pengguna berupa pesan kesalahan yang dilakukan oleh pengguna ketika memberikan masukkan data.

5.2.1.1 Penanganan Kesalahan *Input Data Kosong*

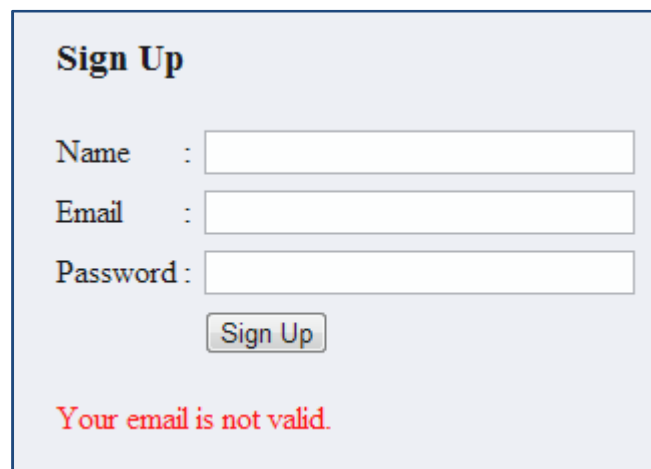
Penanganan kesalahan *input* ini ditampilkan untuk menunjukkan kesalahan yang terjadi ketika salah satu atau beberapa *field* pada suatu *form* belum diisi oleh *user*. Penanganan kesalahan ini terdapat pada setiap proses *input* data yang dilakukan oleh *user*. Gambar 5.18 adalah contoh yang menunjukkan penanganan kesalahan oleh sistem jika salah satu atau beberapa *field* belum terisi atau kosong.



Gambar 5.18 Penanganan kesalahan *input* data kosong.

5.2.1.2 Penanganan Kesalahan *Input Data Format*

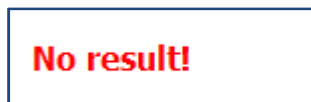
Penanganan kesalahan *input* ini ditampilkan untuk menunjukkan kesalahan yang terjadi ketika salah satu atau beberapa *field* pada suatu *form* diisi dengan format atau tipe data yang salah. Gambar 5.19 adalah contoh yang menunjukkan penanganan kesalahan *input* data format.

A screenshot of a "Sign Up" form. The form has a light blue background and a blue border. It contains three input fields: "Name", "Email", and "Password". Below the "Email" field, there is a "Sign Up" button. At the bottom of the form, there is a red error message: "Your email is not valid." The "Name" and "Password" fields are empty, while the "Email" field is also empty, suggesting the error occurred after an attempt to submit the form.

Gambar 5.19 Penanganan Kesalahan *Input Data Format*.

5.2.2 Penanganan Hasil Tidak Ada

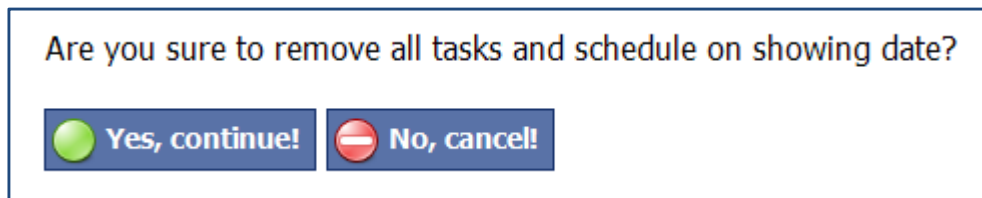
Penanganan hasil tidak ada adalah penanganan kesalahan oleh sistem saat sistem yang melakukan pencarian tugas-tugas atas perintah *user* namun tidak menemukan hasil. Penanganan kesalahan ini hanya terdapat pada halaman pencarian tugas. Gambar 5.20 adalah aksi dari sistem saat tidak menemukan hasil pencarian tugas.



Gambar 5.20 Penanganan Hasil Tidak Ada.

5.2.3 Peringatan Aksi

Peringatan aksi adalah pilihan yang dimunculkan oleh sistem kepada *user* untuk melanjutkan atau membatalkan proses yang sedang berlangsung. Pada gambar 5.21 menunjukkan peringatan aksi *user* saat ingin menghapus seluruh tugas beserta jadwal pada hari terpilih untuk melanjutkan penghapusan atau membatalkannya.



Gambar 5.21 Peringatan Aksi.

5.3 Kasus

Pengujian manual dilakukan dengan menghitung serta mempertimbangkan secara manual proses pemilihan jalur terpendek sebuah jadwal dengan beberapa tugas di dalamnya dalam satu hari. Data sampel masukkan *markers* dan tugas-tugas yang dibutuhkan untuk pengujian manual ini dijabarkan pada tabel 5.1 dan tabel 5.2.

Tabel 5.1 Tabel Data Sampel *Markers*.

Nama <i>Marker</i>	Alamat <i>Marker</i>	<i>Latitude</i>	<i>Longitude</i>
Rumah	Jalan Kaliurang km 6.8	-7.748063	110.387459
Kampus UII Terpadu	Jalan Kaliurang km 14.5	-7.687537	110.415749
Tugu Yogyakarta	Jalan Pangeran Mangkubumi	-7.782947	110.367050
Ambarukmo Plaza	Jalan Laksamana Adisucipto	-7.783064	110.401619

Tabel 5.2 Tabel Data Sampel Tugas-Tugas.

Nama Tugas	Keterangan	<i>Marker</i>	Waktu	Durasi
Bimbingan	Program & BAB V	Kampus UII Terpadu	10:00-12:00	60
Jalan-Jalan	Ambil Foto Tugu	Tugu Yogyakarta	08:00-16:00	60
Belanja	Belanja April	Ambarukmo Plaza	10:00-20:00	120

Proses penentuan rute terpendek secara manual dibagi atas tahapan-tahapan berikut ini :

a. Proses permutasi

Proses pertama dalam pengolahan data adalah perhitungan kemungkinan-kemungkinan dengan permutasi. Untuk memudahkan perhitungan permutasinya maka tugas-tugas akan dilambangkan dengan karakter-karakter sebagai berikut :

A = Bimbingan

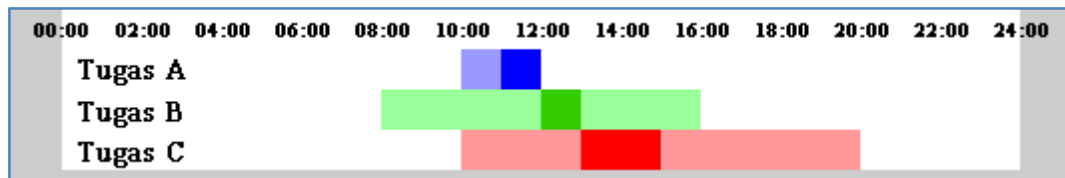
B = Jalan-jalan

C = Belanja

Permutasi dari {A, B, C} ada enam set yaitu (A,B,C), (A,C,B), (B,A,C), (B,C,A), (C,A,B), (C,B,A).

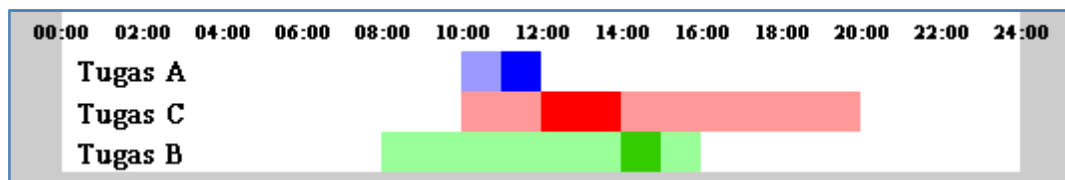
b. Pengujian set-set hasil permutasi

Untuk mempermudah penentuan kemungkinan suatu set hasil permutasi untuk dijadikan jadwal, maka set-set tersebut akan dijabarkan pada gambar 5.22 sampai gambar 5.27.



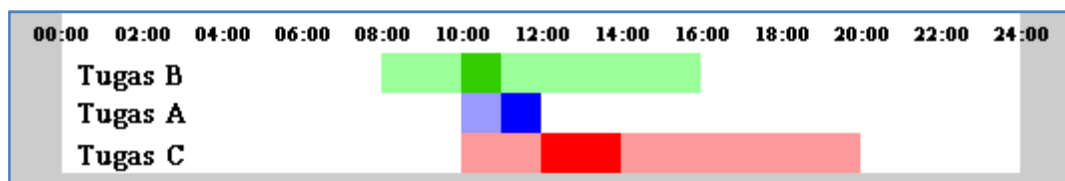
Gambar 5.22 Pengujian kemungkinan set (A,B,C).

Menurut gambar 5.22, set (A,B,C) adalah **mungkin** untuk dijadikan jadwal.



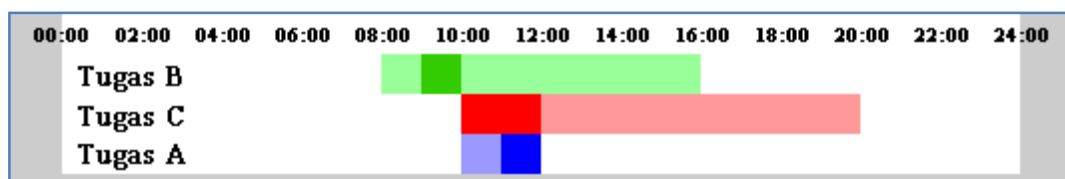
Gambar 5.23 Pengujian kemungkinan set (A,C,B).

Menurut gambar 5.23, set (A,C,B) adalah **mungkin** untuk dijadikan jadwal.



Gambar 5.24 Pengujian kemungkinan set (B,A,C).

Menurut gambar 5.24, set (B,A,C) adalah **mungkin** untuk dijadikan jadwal.



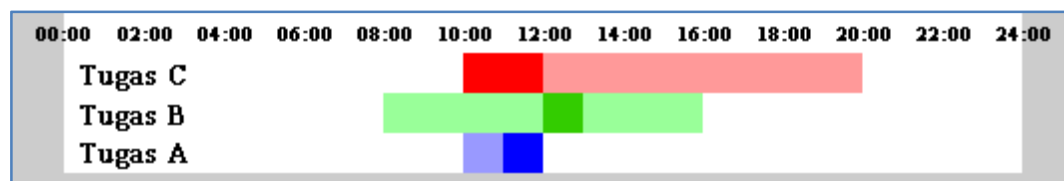
Gambar 5.25 Pengujian kemungkinan set (B,C,A).

Menurut gambar 5.25, set (B,C,A) adalah **tidak mungkin** untuk dijadikan jadwal karena tugas C dan tugas A mengalami tabrakan.



Gambar 5.26 Pengujian kemungkinan set (C,A,B).

Menurut gambar 5.26, set (C,A,B) adalah **tidak mungkin** untuk dijadikan jadwal karena tugas C dan tugas A mengalami tabrakan.



Gambar 5.27 Pengujian kemungkinan set (C,B,A).

Menurut gambar 5.27, set (C,B,A) adalah **tidak mungkin** untuk dijadikan jadwal karena tugas C dan tugas A mengalami tabrakan.

Pada tahap ini terdapat tiga set hasil permutasi yang tidak memungkinkan untuk dijadikan jadwal dan tereliminasi, sehingga tersisa tiga set lainnya yaitu (A,B,C), (A,C,B) dan (B,A,C).

c. Penyisipan titik berangkat dan titik pulang

Langkah berikutnya adalah menyisipkan titik berangkat di awal dan titik pulang di akhir dengan *marker* Rumah pada setiap set hasil permutasi yang mungkin sehingga menjadi :

(S,A,B,C,F), (S,A,C,B,F), (S,B,A,C,F).

S = Berangkat

F = Pulang

d. Pengujian ke *Google Maps API*

Langkah selanjutnya dalam penentuan rute terpendek sebagai jadwal adalah menguji set-set yang tersisa ke *Google Maps API*. *String* urutan tugas-tugas setiap set yang tersisa sebagai masukan untuk *Google* ditunjukkan pada tabel 5.3.

Tabel 5.3 Tabel pengujian ke *Google Maps API*.

Set	<i>String</i> untuk masukkan <i>Google Maps API</i>	Jarak dari <i>Google</i>
(S,A,B,C,F)	From -7.748063, 110.387459 To -7.687537, 110.415749 To -7.782947, 110.367050 To -7.783064, 110.401619 To -7.748063, 110.387459	33937 meter
(S,A,C,B,F)	From -7.748063, 110.387459 To -7.687537, 110.415749 To -7.783064, 110.401619 To -7.782947, 110.367050 To -7.748063, 110.387459	32221 meter
(S,B,A,C,F)	From -7.748063, 110.387459 To -7.782947, 110.367050 To -7.687537, 110.415749 To -7.783064, 110.401619 To -7.748063, 110.387459	40719 meter

e. Pemilihan set permutasi terpendek dan penentuan jadwal

Dari tabel 5.3 diketahui bahwa set **(S,A,C,B,F)** memiliki jarak terpendek yaitu **32221 meter**, maka urutan tugas-tugas dalam jadwal yang akan dibuat akan berdasarkan urutan elemen dalam set tersebut. Urutan tugas-tugasnya menjadi :

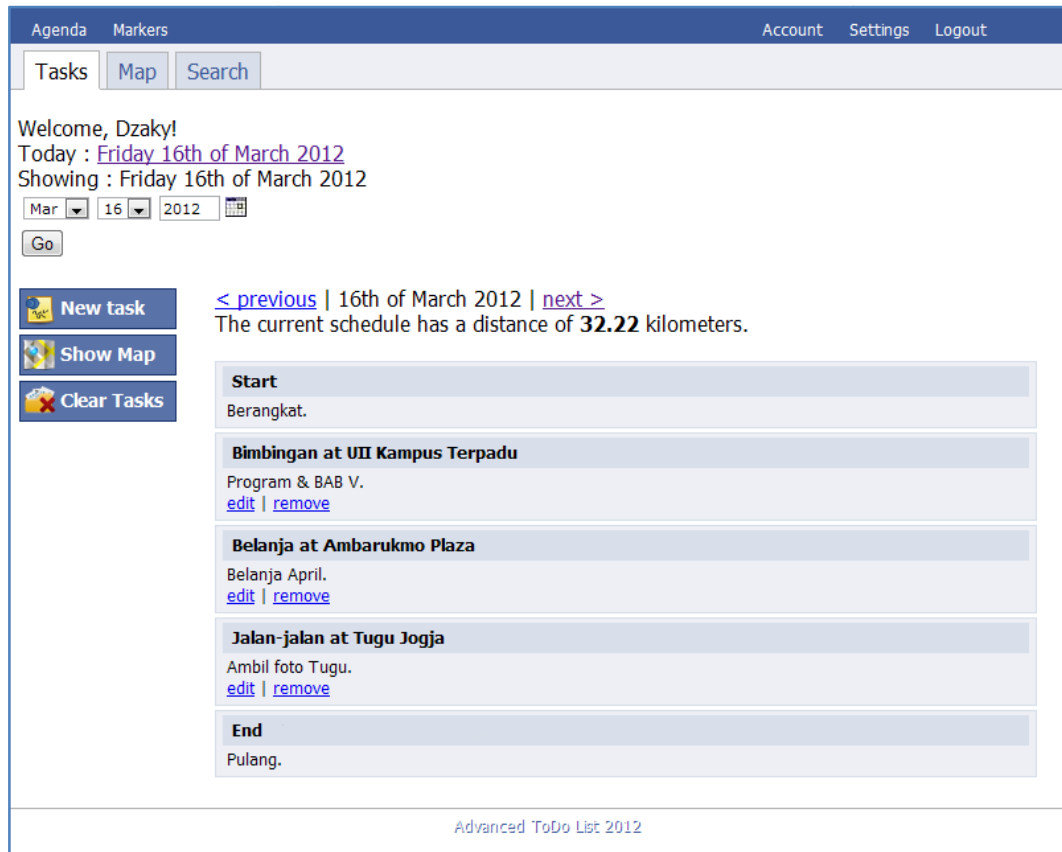
- S : Berangkat di Rumah.
- A : Bimbingan di UII Kampus Terpadu selama 60 menit.
- C : Belanja di Ambarukmo Plaza selama 120 menit.
- B : Jalan-jalan di Tugu Jogja selama 60 menit.
- F : Pulang di Rumah.

5.4 Penyelesaian

Pengujian pada sistem menggunakan data yang sama dengan pengujian manual agar dapat diketahui apakah hasil dari kerja sistem sesuai dengan pengujian secara manual. Proses pengujian sistem dimulai dari pemasukkan data *marker* satu per satu berlanjut dengan penentuan titik berangkat serta titik pulang dan diakhiri dengan penyisipan tugas-tugas. Pada gambar 5.28 dan gambar 5.29, jadwal yang dihasilkan oleh sistem memiliki jarak tempuh **32.22 kilometer** dengan urutan tugas sebagai berikut :

- a. Berangkat di Rumah.
- b. Bimbingan di UII Kampus Terpadu.

- c. Belanja di Ambarukmo Plaza.
- d. Jalan-jalan di Tugu Jogja.
- e. Pulang di Rumah.



Agenda Markers Account Settings Logout

Tasks Map Search

Welcome, Dzaky!
Today : [Friday 16th of March 2012](#)
Showing : Friday 16th of March 2012

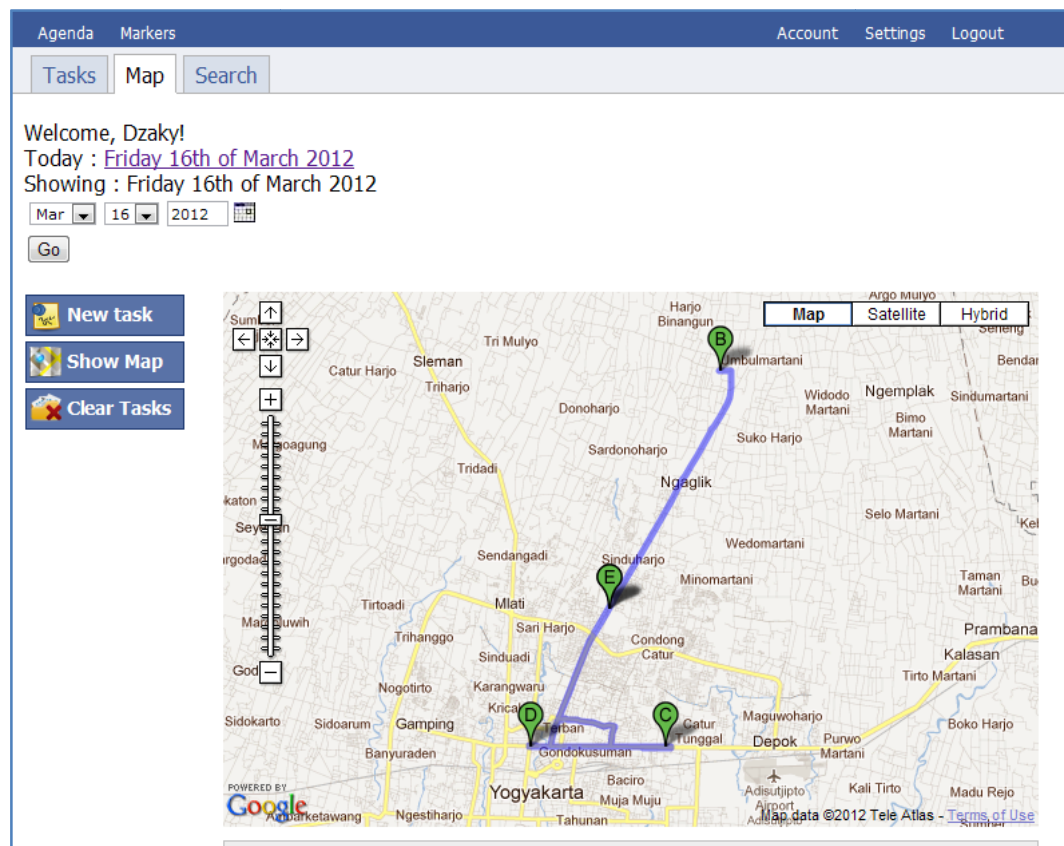
Mar 16 2012

[< previous](#) | 16th of March 2012 | [next >](#)
The current schedule has a distance of **32.22** kilometers.

Start
Berangkat.
Bimbingan at UII Kampus Terpadu
Program & BAB V. edit remove
Belanja at Ambarukmo Plaza
Belanja April. edit remove
Jalan-jalan at Tugu Jogja
Ambil foto Tugu. edit remove
End
Pulang.

Advanced ToDo List 2012

Gambar 5.28 Jadwal untuk penyelesaian kasus.



Gambar 5.29 *Direction map* untuk penyelesaian kasus.

5.5 Kelebihan dan Kelemahan Sistem

Tujuan dibangunnya sebuah aplikasi pasti memiliki fungsi menjadi kelebihan dibandingkan dengan aplikasi serupa maupun yang sudah ada. Berikut ini beberapa kelebihan yang dimiliki aplikasi ini :

- Pengguna tidak hanya dapat mengetahui jarak antara dua tempat, tetapi beberapa tempat sekaligus sehingga lebih memudahkan penentuan rute terpendek dengan melalui tempat-tempat yang harus dilalui.
- Aplikasi ini adalah aplikasi berbasis *web* sehingga semua perangkat yang memiliki *browser* yang mendukung *javascript*, akan dapat menjalankannya dengan baik.

Dengan keterbatasan pengetahuan dan analisa, dalam pembangunannya tentunya tidak terlepas dari kekurangan atau kelemahan sistem. Berikut ini beberapa kekurangan atau kelemahan yang dimiliki aplikasi ini :

- a. Jumlah tugas maksimal yang dapat diproses dalam satu hari adalah lima. Pembatasan jumlah ini berkaitan dengan pertimbangan beban komputasi yang semakin tinggi seiring dengan semakin banyaknya tugas yang dimasukkan.
- b. Aplikasi ini tidak memiliki pengingat atau *reminder* untuk memberi notifikasi mengenai tugas-tugasnya kepada pengguna.
- c. Sistem ini dibuat hanya untuk menyusun jadwal yang berisi tugas-tugas dalam rentang satu hari atau 24 jam saja.
- d. Zona waktu yang menjadi patokan dalam sistem aplikasi ini adalah zona waktu GMT +7 atau WIB.

BAB VI

PENUTUP

Berdasarkan hasil analisis dan implementasi yang telah dilakukan, dapat disimpulkan bahwa :

1. Permutasi dapat diimplementasikan untuk menentukan semua kemungkinan jajaran elemen-elemen sebuah himpunan dengan memperhatikan urutannya. Urutan elemen pada kemungkinan-kemungkinan tersebut dimanfaatkan untuk menentukan semua kemungkinan urutan tugas-tugas dalam satu hari. Selanjutnya kemungkinan-kemungkinan tersebut akan diseleksi lebih lanjut untuk mendukung sebuah keputusan penyusunan urutan tugas-tugas dengan jarak tempuh terpendek.
2. Semakin banyak tugas-tugas yang dimasukkan, maka akan semakin banyak kemungkinan yang ada sehingga memberatkan kinerja serta lalu lintas data internet oleh komputer *server*.

Mengingat berbagai keterbatasan yang dialami penulis, maka penulis menyarankan untuk pengembangan penelitian di masa yang akan datang sebagai berikut :

1. Pada aplikasi *Penjadwalan Berbasis Lokasi* ini jadwal yang terbentuk adalah untuk satu hari atau 24 jam saja sehingga akan lebih fleksibel jika dapat membuat jadwal yang bisa melampaui satu hari.
2. Peningat atau *reminder* pada aplikasi-aplikasi *Penjadwalan* adalah bagian yang penting untuk mengingatkan pengguna tentang tugas-tugasnya sehingga akan lebih baik jika di masa mendatang dikembangkan fungsi pengingatnya.
3. Seorang pengguna mungkin akan memasukkan marker yang sangat banyak, sehingga dibutuhkan pengelompokan *markers* tersebut untuk memudahkan manajemen dan pencariannya.
4. Fungsi utama aplikasi ini adalah meminimalisir total jarak tempuh pengguna dalam satu hari sehingga akan lebih lengkap jika ditambahkan pula fungsi penghitung rata-rata konsumsi bahan bakar yang tentunya sesuai skala

konsumsi bahan bakar menurut jenis kendaraannya yang dimasukkan atau dipilih oleh pengguna.

5. Berbagi atau *share* adalah hal favorit untuk sebagian besar pengguna internet sehingga akan lebih menarik jika terdapat fungsi yang memungkinkan pengguna untuk berbagi informasi tugas maupun *marker*. Fungsi berbagi ini juga bisa dimanfaatkan saat beberapa pengguna berkumpul atau memiliki tugas yang sama sehingga memudahkan dalam membuat sebuah acara bersama.
6. Bagi pengguna dengan mobilitas tinggi, pelanggaran batas jumlah tugas dalam satu hari mungkin menjadi penting mengingat pada aplikasi ini hanya dibatasi 5 tugas saja dalam satu hari. Pembatasan ini didasarkan atas sifat permutasi yang progresif. Sifat progresif di sini artinya semakin banyaknya tugas yang dimasukkan dalam satu hari, maka akan semakin banyak pula kemungkinan yang dihasilkan dari perhitungan permutasi pada hari tersebut sehingga akan memberatkan kinerja komputer dan lalu lintas data antara *hosting server* dan *Google server*.

DAFTAR PUSTAKA

Google. (2012). *Google Maps JavaScript API v3*. Retrieved Maret 6, 2012, from Google Developers: <http://code.google.com/apis/maps/documentation/javascript/>

Negoro, S. T., & Harahap, B. (1998). *Ensiklopedia Matematika*. Bogor: Ghalia Indonesia.