Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

Team project ©2017
Dony Pratidana S. Hum | Bima Agus Setyawan S. IIP

BAB III

METODOLOGI DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Metodologi yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Telaah Literatur

   Pada tahap ini, dilakukan pencarian referensi dan pembelajaran mengenai topik dan materi yang diperlukan untuk implementasi *Scatter Search* ke dalam sebuah sistem penjadwalan dan cara kerja *mobile web*. Telaah literatur dilakukan dengan mempelajari buku, jurnal, dan artikel terkait mengenai algoritma *Scatter Search* dan *mobile web.*

   Dari proses ini, didapatkan pemahaman mengenai cara kerja algoritma *Scatter Search*, cara implementasinya ke dalam sebuah sistem penjadwalan, dan bagaimana cara membangun sebuah sistem dengan basis *mobile web.*

2. Pengumpulan Data dan Informasi

   Metode pengumpulan data yang dilakukan untuk sistem penjadwalan ini adalah dengan observasi dan wawancara. Proses observasi dilakukan dengan mengamati cara kerja manajemen dalam menerima jadwal, membuat jadwal, dan proses logika yang diambil saat memberikan kegiatan kepada anggota.

   Wawancara dilakukan dengan bertanya langsung kepada manajer grup JKT48. Dari proses wawancara tersebut, didapatkan data mengenai sumber daya yang ada (anggota grup JKT48), jenis kegiatan yang digunakan untuk membuat jadwal, dan syarat-syarat yang harus dipenuhi dalam pembuatan jadwal tersebut.

3. Perancangan

Pada tahap ini, dilakukan perancangan *Scatter Search* untuk sebuah aplikasi penjadwalan. Dimulai dari diagram umum aplikasi, yaitu gambaran umum mengenai aplikasi penjadwalan. Kemudian dibuat *flowchart* implementasi algoritma ke dalam proses kerja sistem, *Entity Relationship Diagram* untuk menjelaskan pemakaian sistem, dan perancangan umum *User Interface* yang akan dibuat agar sistem ini dapat diakses dengan menggunakan *mobile web*.

4. Implementasi

Tahap selanjutnya adalah implementasi program ke dalam bahasa PHP agar dapat diakses oleh sistem berbasis *mobile web*. Langkah pertamanya adalah dengan mempersiapkan modul-modul yang dibutuhkan dalam sebuah aplikasi penjadwalan seperti modul *input* data, *ouput* data, dan modul penjadwalan itu sendiri yang menggunakan algoritma *Scatter Search*

Kemudian disiapkan juga *database* yang dibutuhkan oleh sistem agar dapat bekerja. *Database* tersebut berisi data member, data kegiatan, dan jadwal.

Dalam kaitannya agar dapat diakses dengan *mobile web*, maka *user interface* dirancang agar lebih kompak dan mudah untuk digunakan walaupun hanya dibuka melalui layar perangkat yang *mobile*.

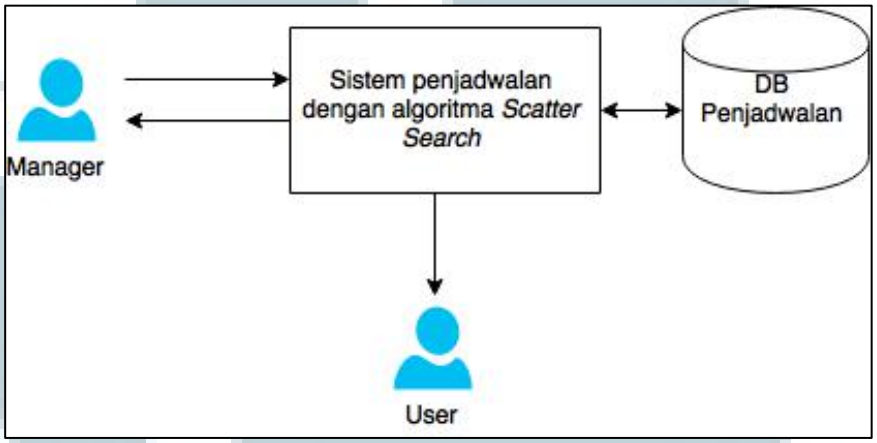
5. Uji Coba dan Evaluasi

Pengujian hasil implementasi algoritma *Scatter Search* ke dalam aplikasi penjadwalan berbasis *mobile web* ini akan dilakukan dengan menghitung waktu komputasi yang dibutuhkan untuk menghasilkan sebuah jadwal.

3.1 Perancangan Sistem

Berikut akan diuraikan perancangan terhadap implementasi *Scatter Search* ke dalam sebuah aplikasi penjadwalan.

3.2.1 Diagram Umum Aplikasi



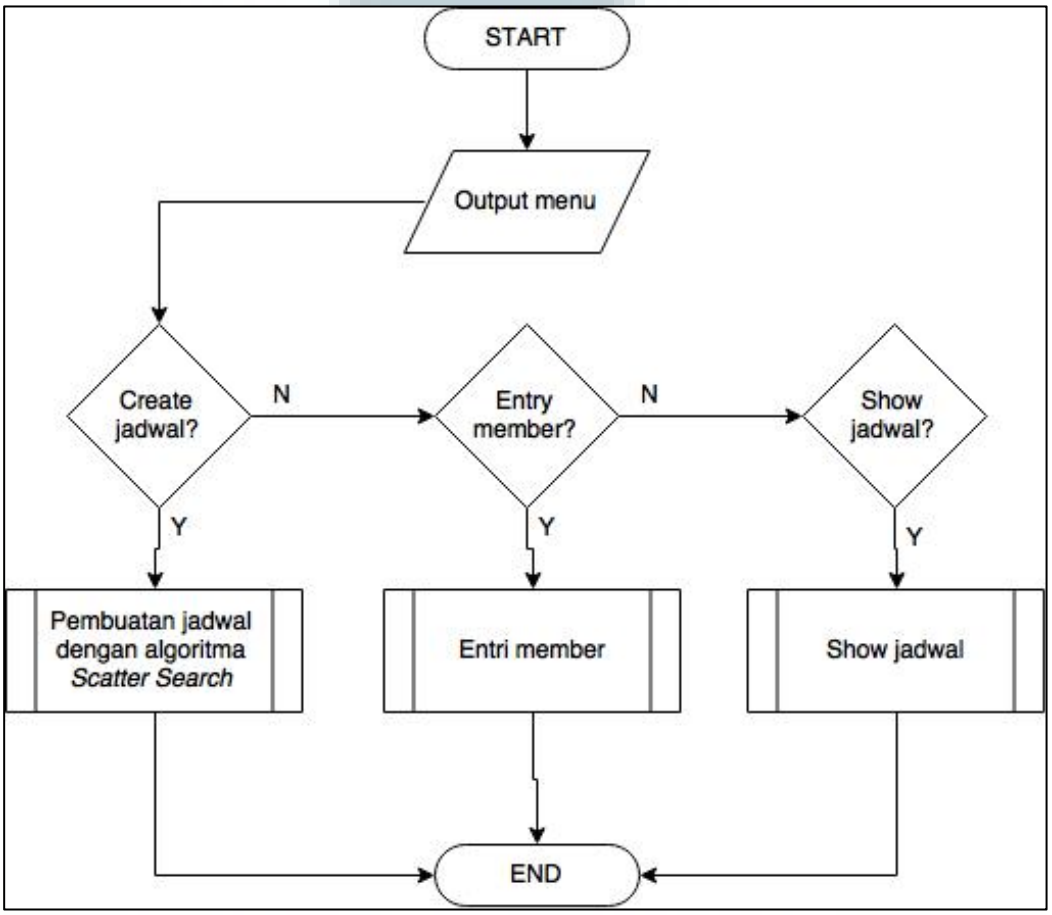
Gambar 3.1 Diagram umum aplikasi

Aplikasi ini akan menerima *input* dari manager berupa data sumber daya dan kegiatan yang harus dilakukan untuk kemudian dimasukkan dalam *database.* Data tersebut akan dicek *availability*-nya terlebih dahulu (apakah bentrok atau tidak dengan jadwal yang sudah ada). Bila tidak bentrok, data yang telah dimasukkan kemudian akan diolah dengan menggunakan algoritma *Scatter Search* untuk menghasilkan sebuah jadwal yang dapat ditampilkan untuk pengguna.

3.2.2 Flowchart

Berikut akan digambarkan *flowchart* mengenai proses aplikasi penjadwalan dengan menggunakan algoritma *Scatter Search*.

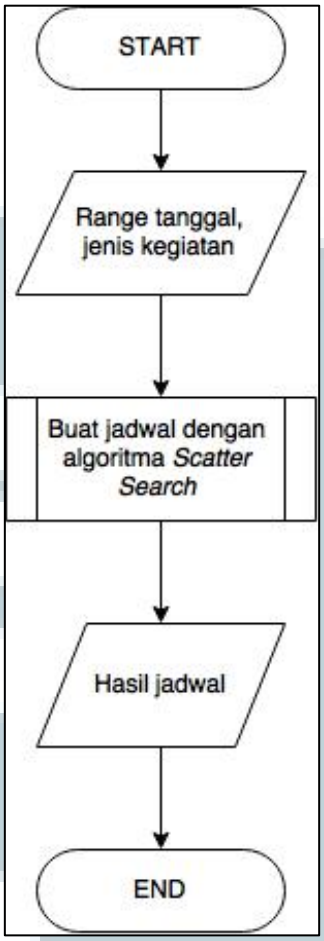
Proses dimulai dengan sebuah halaman awal yang prosesnya dijelaskan melalui gambar 3.2. Akan muncul *menu* di mana pengguna bisa memilih fitur yang diinginkan.



Gambar 3.2 *Flowchart* halaman awal aplikasi penjadwalan

Ada tiga menu utama yang akan ditampilkan, yaitu pembuatan jadwal, entri *member* untuk memperbaharui data yang ada, dan *show* jadwal untuk menunjukkan jadwal yang sudah ada.

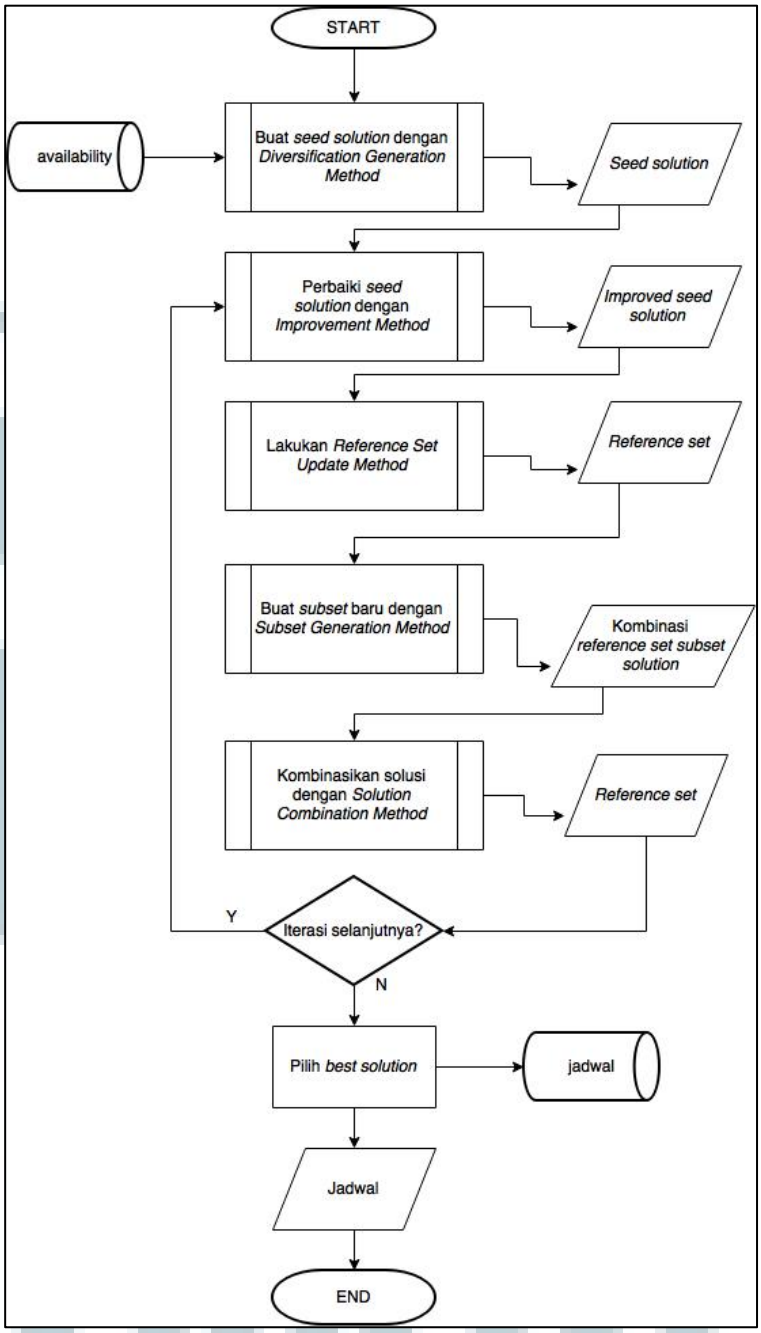
Proses untuk pembuatan jadwal dengan algoritma *Scatter Search* ditunjukkan melalui gambar 3.3.



Gambar 3.3 *Flowchart* pembuatan jadwal dengan algoritma *Scatter Search*

Dari *flowchart* di gambar 3.3, dapat dilihat bahwa *input* berupa *range* tanggal kegiatan yang diinginkan dan jenis kegiatannya. Sistem akan mengolah data tersebut untuk memilih jadwal yang terbaik dengan menggunakan algoritma *Scatter Search*.

Proses pembuatan jadwal dengan menggunakan algoritma *Scatter Search* sendiri dijelaskan melalui gambar 3.4.



Gambar 3.4. *Flowchart* pembuatan jadwal dengan algoritma *Scatter Search*

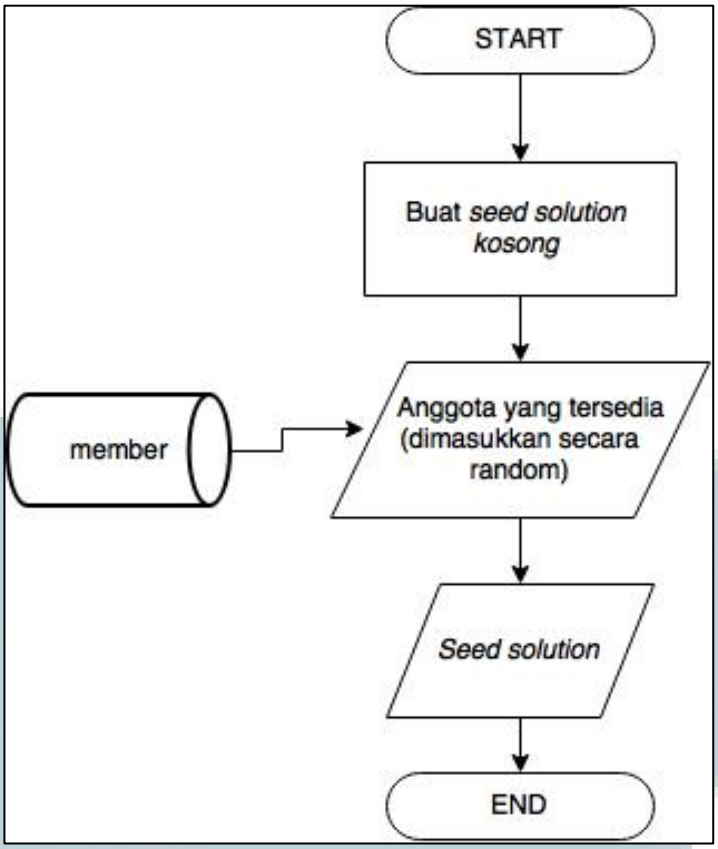
Pertama-tama, algoritma akan mengambil data dari *database* jadwal untuk membuat *seed solution* menggunakan *Diversification Generation Method*. *Seed solution* yang dibuat masih bersifat terdiversifikasi dan menyebar. *Improvement*

Method setelahnya akan memperbaiki *seed solution* tersebut dan membuat *improved set solution* (sekaligus bertindak sebagai *reference set*). Set solusi yang sudah 'diperbaiki' kemudian akan diperbarui dengan *Reference Set Update Method.* Kemudian, ia akan melalui *Subset Generation Method* untuk membentuk *reference set* baru. *Set* yang baru itu kemudian akan dikombinasikan dengan *Solution Combination Method.*

Apabila masih belum memenuhi jumlah iterasi yang diinginkan, *reference set* akan kembali ke proses *improvement*. Sedangkan bila sudah selesai, akan dipilih solusi terbaik berdasarkan bobot dan akan dimasukkan ke dalam *database.*

Proses untuk membuat *seed solution* dengan *Diversification Generation Method* dijelaskan melalui gambar 3.5. Pertama, akan dibuat 5 *seed solution* kosong. *Seed solution* ini kemudian diisi secara *random* oleh anggota yang tersedia, dengan ketentuan anggota yang sama tidak boleh dimasukkan dua kali.

Penerapan Algoritma ..., Putriasihta Baeha, FTI UMN, 2015

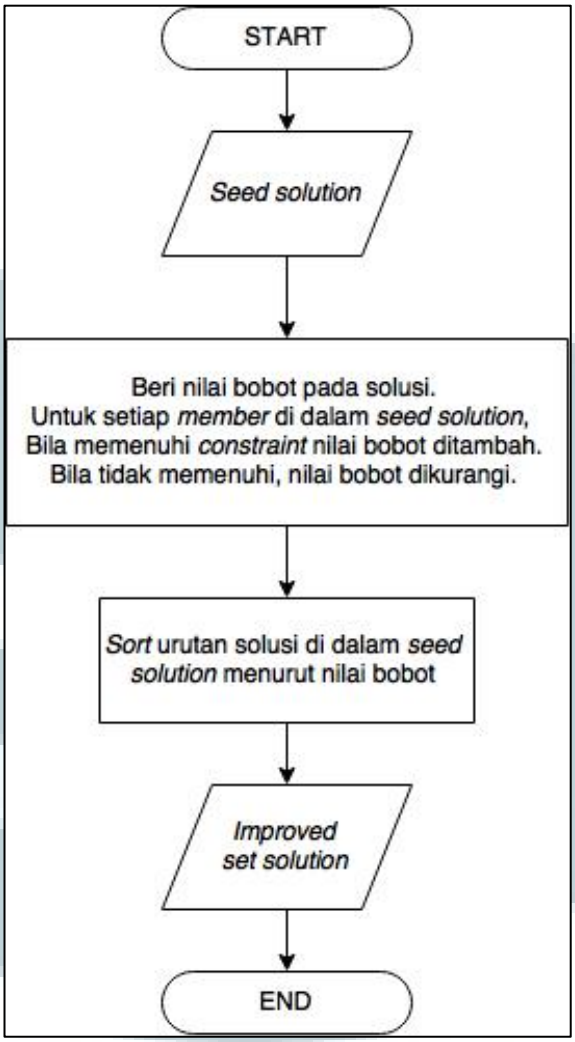


Gambar 3.5. *Flowchart* pembuatan *seed solution* dengan *Diversification Generation Method*

Proses berikutnya adalah *Improvement Method* yang akan memperbaiki *seed solution* yang ada dan menghasilkan *reference set*. Proses tersebut dapat dilihat melalui gambar 3.6. Pertama-tama, *seed solution* akan diberi bobot sesuai dengan *soft constraint* yang ada. Bila anggota yang ada di dalam *seed solution* tersebut memenuhi *constraint*, maka nilai bobot solusi akan ditambah. Apabila tidak, maka nilai bobot solusi akan dikurangi.

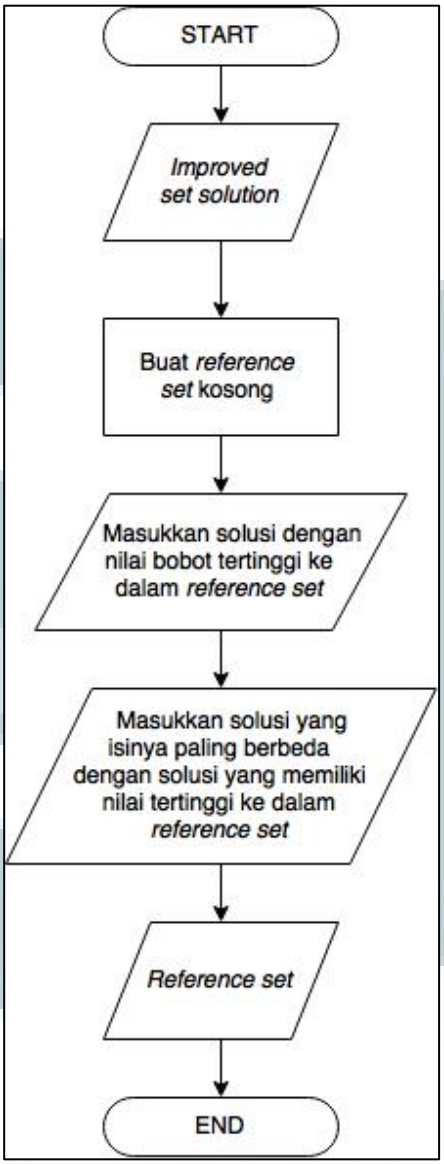
Setelah itu, solusi-solusi yang ada akan diurutkan berdasarkan nilai bobotnya.

Penerapan Algoritma ..., Putriasihta Baeha, FTI UMN, 2015



Gambar 3.6. *Flowchart* perbaiki *seed solution* dengan *Improvement Method*

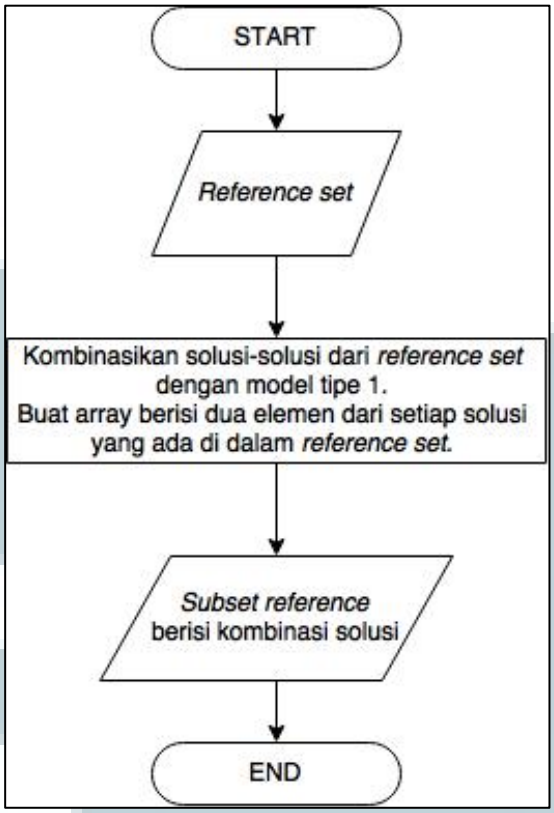
Hasil *improved set solution* kemudian akan mengalami proses *reference set update method* yang dapat dilihat di gambar 3.7. Sistem akan membuat *reference set* kosong. *Reference set* ini lalu diisi dengan solusi yang memiliki bobot tertinggi dan solusi yang paling berbeda dengan solusi yang yang memiliki bobot tertinggi.



Gambar 3.7. *Flowchart* proses *Reference Set Update Method*

Reference set yang dihasilkan oleh *Reference Set Update Method* lalu akan dikombinasikan dengan *Subset Generation Method* seperti gambar 3.8. Metode kombinasi yang akan digunakan adalah tipe 1, di mana setiap *subset* memiliki kombinasi dari 2 elemen yang ada di *reference set*. Misalnya apabila ada 3 solusi, maka dibuat *array* yang berisi solusi 1, 2; solusi 1, 3; dan solusi 2, 3.

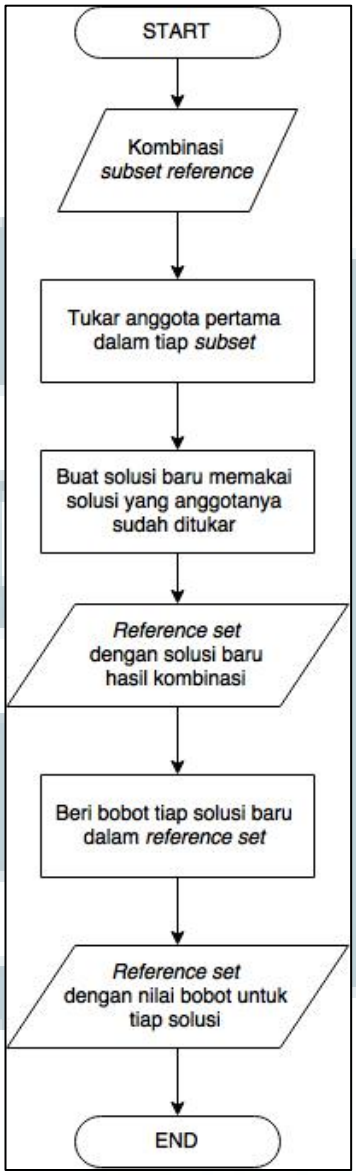
Penerapan Algoritma ..., Putriasihta Baeha, FTI UMN, 2015



Gambar 3.8. *Flowchart* pembuatan *subset* baru dengan *Subset Generation Method*

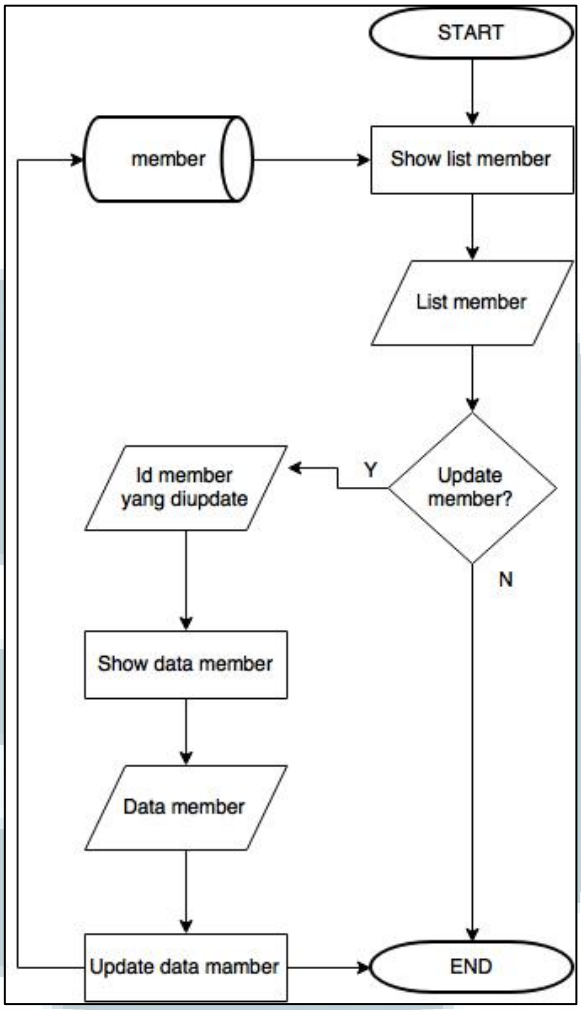
Gambar 3.9. menjelaskan proses berikutnya, yaitu *Subset Combination Method*. Setiap solusi yang ada di dalam array *subset reference* akan dikombinasikan dengan cara menukar elemen pertama yang ada di setiap solusi dengan elemen pertama yang ada di solusi pasangannya. Solusi yang mendapat elemen baru tersebut akan menjadi calon solusi terbaik yang baru. Setiap solusi yang mendapat elemen baru kemudian akan kembali dihitung bobotnya.

Hasil solusi dari metode ini akan kembali ke *improvement method* apabila iterasi belum selesai. Bila iterasi sudah selesai, maka dipilih solusi dengan bobot tertinggi dari *Subset Combination Method* ini.



Gambar 3.9. *Flowchart* kombinasi solusi dengan *Subset Combination Method*

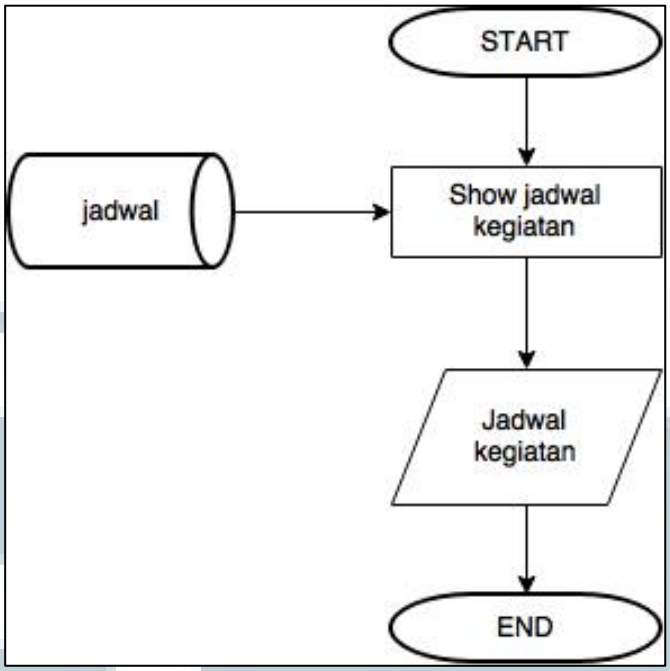
Setelah proses pembuatan jadwal, berikutnya adalah fitur entri *member* untuk memperbaharui data anggota yang ada. Prosesnya dapat dilihat di gambar 3.10.



Gambar 3.10. Proses entri *member*

Saat pengguna memilih *menu* entri member, maka akan muncul daftar *member* yang sudah ada dari database. Dari daftar tersebut pengguna dapat memilih untuk memperbaharui data anggota yang ada. Yang dapat diperbaharui adalah data nama, tim, *ranking*, dan ketersediaan anggota.

Menu berikutnya yaitu *Show* jadwal berfungsi untuk menunjukkan jadwal yang sudah ada di dalam *database*. *Flowchart* untuk proses tersebut ada di gambar 3.11.



Gambar 3.11. Proses *show* jadwal kegiatan.

Sistem akan mengambil data dari *database* jadwal dan menunjukkannya ke pengguna dalam bentuk kalender jadwal.

3.2.3 Constraint

Dalam sistem ini, ada beberapa *constraint* yaitu persyaratan yang harus dipenuhi oleh aplikasi dalam membuat jadwal.

a. *Hard constraint*

Adalah persyaratan yang mutlak harus dipenuhi. Bila tidak dipenuhi, maka jadwal tidak dapat dibuat. Persyaratan tersebut adalah:

- Dalam satu tanggal dan *shift* yang sama, tidak boleh ada dua kegiatan sekaligus.

- Setiap anggota memiliki jadwal kegiatan masing-masing seperti sekolah atau kuliah. Kegiatan JKT48 yang dijadwalkan tidak bisa masuk ke dalam waktu di mana anggota tidak tersedia.

b. *Soft constraint*

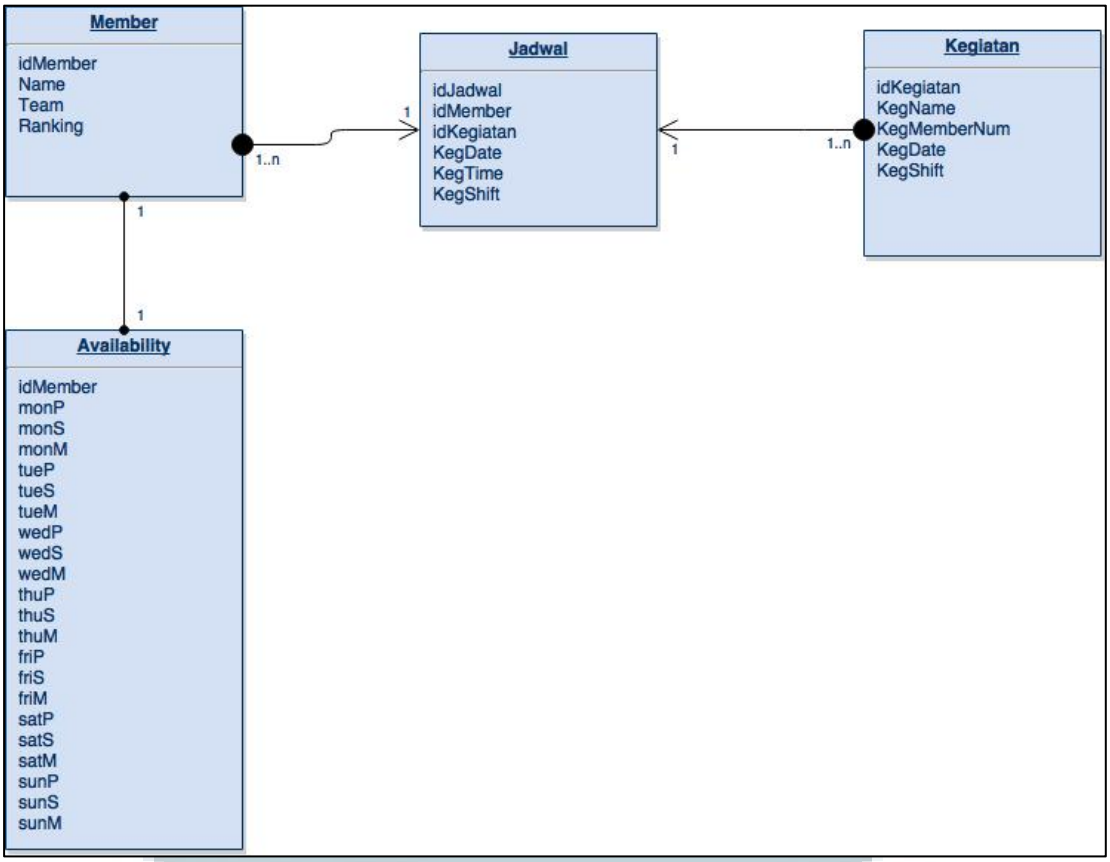
Adalah persyaratan yang tidak mutlak harus dipenuhi, tetapi sebaiknya dipenuhi untuk mendapatkan solusi optimal yang paling memenuhi kebutuhan pengguna. Persyaratan tersebut adalah:

- Terdapat 5 kegiatan, yaitu *TV show*, teater, latihan, rekaman, dan *event*.
- Kegiatan *TV show* dan *event* lebih baik bila dilakukan pada akhir minggu (hari Sabtu dan Minggu). Bila dilakukan pada hari biasa (Senin-Jumat), maka lebih baik bila dilakukan dalam *shift* malam.
- Anggota dengan *ranking* 'Senbatsu' (member top) diprioritaskan untuk mengikuti kegiatan *TV show* dan *event*. Sedangkan anggota dengan ranking 'Trainee' tidak diprioritaskan.
- Untuk kegiatan latihan dan rekaman, anggota 'Trainee' lebih diprioritaskan.
- Untuk kegiatan teater, semua anggota mendapat bobot yang sama.

3.2.4. Entity Relationship Diagram

Entity Relationship Diagram dari *database* yang berhubungan dengan pembuatan aplikasi penjadwalan digambarkan dalam gambar 3.12.

Penerapan Algoritma ..., Putriasihta Baeha, FTI UMN, 2015



Gambar 3.12. *Entity Relationship Diagram* aplikasi penjadwalan

Jadwal hanya bisa dibuat bila paling tidak ada satu *member* dan satu kegiatan. Setiap data *member* dan kegiatan akan disatukan dalam sebuah tabel jadwal.

Berikut adalah struktur dari tabel-tabel yang digunakan.

A. Tabel *Member*

Nama tabel: Member

Fungsi: Menyimpan data anggota grup JKT48

Primary key: idMember

Foreign key: -

Tabel 3.1. Struktur tabel *member*

Nama *field*	Tipe	*Length*	Keterangan
idMember	int	3	Nomor ID dari anggota grup.
Name	varchar	150	Nama dari anggota grup.
Team	varchar	3	Tim dari anggota grup. Dipakai karena ada beberapa kegiatan yang hanya bisa dilakukan oleh anggota dari tim yang sama.
Ranking	int	2	Kategori peringkat anggota grup. Dipakai karena ada beberapa kegiatan yang hanya bisa memakai anggota dengan peringkat tertentu.

B. Tabel Kegiatan

Nama tabel: Kegiatan

Fungsi: Menyimpan jenis kategori kegiatan grup JKT48

Primary key: idKegiatan

Foreign key: -

Tabel 3.2. Struktur tabel kegiatan

Nama *field*	Tipe	*Length*	Keterangan
idKegiatan	int	3	Nomor ID dari kegiatan.
KegName	varchar	150	Nama kegiatan.
KegMemberNum	int	3	Jumlah minimum anggota yang dibutuhkan untuk melakukan sebuah kegiatan. Jumlah minimum ini harus dipenuhi saat membuat jadwal.

C. Tabel Jadwal

Nama tabel: Jadwal

Fungsi: Menyimpan jadwal grup yang telah dibuat dengan menggunakan nama anggota, jenis kegiatan, dan waktu kegiatan.

Primary key: idJadwal

Foreign key: idKegiatan, idMember

Tabel 3.3. Struktur tabel jadwal

Nama *field*	Tipe	*Length*	Keterangan
idJadwal	int	3	Nomor ID dari kegiatan.
idKegiatan	int	3	Nomor ID dari kegiatan.
KegDate	date		Tanggal kegiatan tersebut.
KegShift	char	1	Menyimpan waktu *shift* dari kegiatan.
idMember	string	150	Menyimpan id dari *member* yang akan ikut serta dalam kegiatan, digabung dalam 1 string panjang.

D. Tabel Ketersediaan Anggota

Nama tabel: *Availability*

Fungsi: Menyimpan waktu di mana anggota tidak tersedia.

Foreign key: idMember

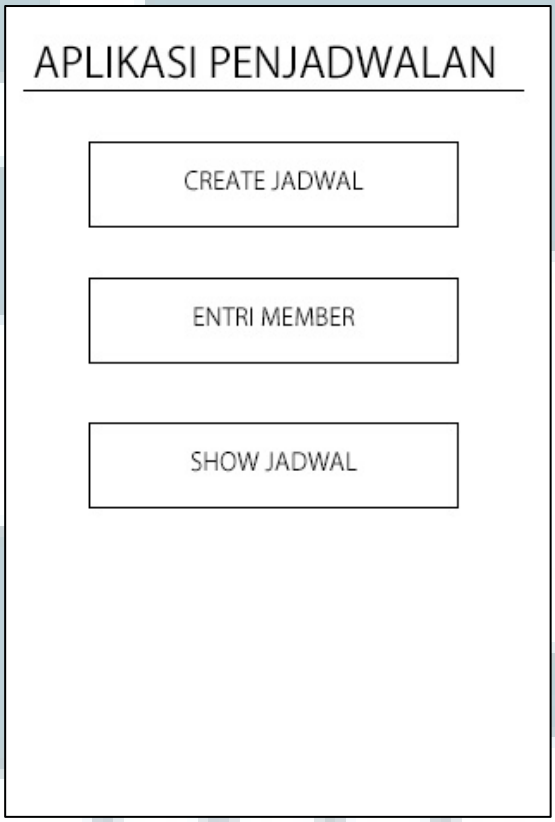
Tabel 3.4. Struktur tabel *availability*

Nama *field*	Tipe	*Length*	Keterangan
idMember	int	3	Nomor ID dari kegiatan.
monP sampai dengan sunM	int	1	Menunjukkan kode apakah anggota tersedia di hari Senin – Minggu. Apabila berisi 0, maka anggota tersedia. Bila berisi 1, maka anggota tidak tersedia. Nama kolomnya adalah kode tiga huruf pertama nama hari digabung dengan *shift*. Contohnya monP = 'Monday'+'Pagi'.

3.2.5. User Interface

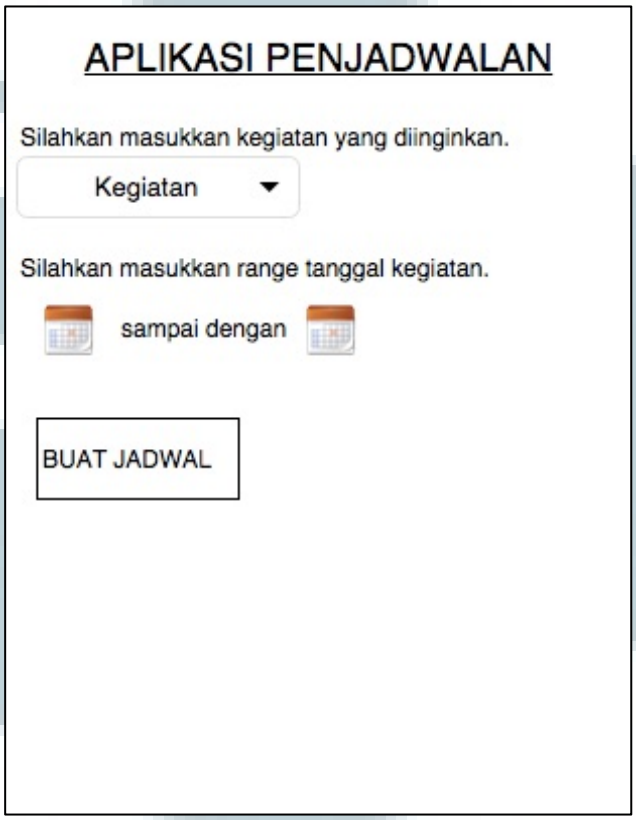
Berikut adalah rancangan *user interface* dari aplikasi penjadwalan. Karena basis aplikasi ini adalah *mobile web,* maka *user interface*-nya juga dirancang agar menyesuaikan perangkat *mobile* yang sering digunakan untuk mengakses *mobile web.* Dalam perancangan berikut, yang digunakan sebagai basis adalah tampilan layar *smartphone*.

Pertama, gambar 3.13. menggambarkan rancangan antar muka awal dari aplikasi penjadwalan. Terdapat tiga *menu* untuk memilih proses yang ditawarkan oleh aplikasi, yaitu "Create Jadwal", "Entri Member", dan "Show Jadwal".



Gambar 3.13. Halaman awal aplikasi penjadwalan

Apabila pengguna memilih untuk membuat jadwal terlebih dahulu, akan muncul halaman di mana pengguna diminta untuk memasukkan informasi yang dibutuhkan, seperti terlihat dalam gambar 3.14.

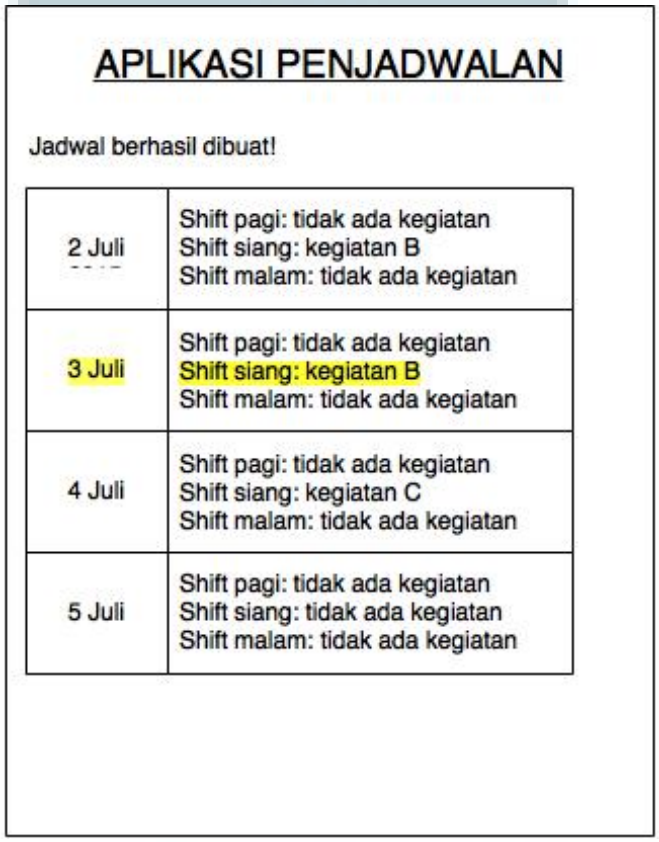


Gambar 3.14. Halaman saat pengguna menekan "CREATE JADWAL".

Pengguna akan diminta untuk memilih kegiatan yang diinginkan dan *range* tanggal di mana kegiatan tersebut ingin dilaksanakan. Kegiatan akan menggunakan *dropdown menu*, dan *range* tanggal kegiatan akan menggunakan *date picker*. Setelah semua informasi yang dibutuhkan dimasukkan, pengguna dapat menekan tombol Buat Jadwal untuk membuat jadwal yang diinginkan.

Informasi yang akan ditampilkan setelah jadwal berhasil dibuat adalah nama kegiatan, tanggal, *shift* kegiatan, serta daftar anggota yang akan berpartisipasi dalam

kegiatan tersebut. Jadwal akan ditampilkan dalam jangka waktu seminggu dari tanggal kegiatan yang disediakan oleh sistem. Rancangan tampilan untuk ini dapat dilihat di gambar 3.15. Jadwal yang baru saja berhasil dibuat akan ditandai dengan warna kuning.



Gambar 3.15. Antar muka pemberitahuan bila sebuah jadwal telah berhasil dibuat.

Berikutnya adalah rancangan antar muka untuk fitur entri *member*. Rancangan ini dapat dilihat di gambar 3.16. Pertama-tama, akan muncul daftar *member*. Pengguna dapat memilih ingin memperbaharui data *member* yang disediakan.

APLIKASI PENJADWALAN

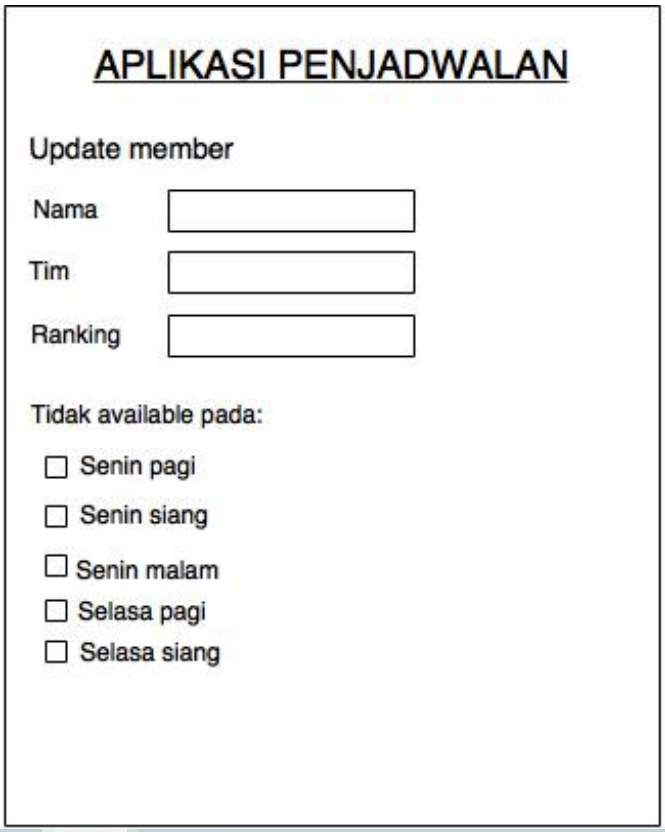
Entri member

ID Member	Nama	Team	Ranking	Update
1	A	J	Senbatsu	Update
2	B	J	Tim	Update
3	C	J	Trainee	Update
4	D	K	Trainee	Update
5	E	K	Senbatsu	Update

Gambar 3.16. Rancangan antar muka halaman awal entri *member*.

Apabila pengguna menekan tombol *update*, maka akan tampil layar berikutnya yang berisi informasi detil dari anggota yang dipilih. Rancangan tampilan ini dapat dilihat di gambar 3.17.

Informasi yang sudah ada dari anggota tersebut akan ditampilkan dan dapat di-*edit* oleh pengguna. Informasi tersebut mencakup nama, tim, *ranking*, dan ketersediaan anggota.



Gambar 3.17. Rancangan antar muka untuk entri *member*.

Terakhir adalah rancangan *user interface* untuk *menu show jadwal*. Di sini, jadwal yang udah ada di dalam *database* akan ditampilkan dalam bentuk kalender per bulan. Rancangan tampilannya dapat dilihat di gambar 3.18.

Di setiap kotak tanggal, akan tampil kegiatan yang sudah ada di dalam *database* sebelumnya.

Gambar 3.18. Rancangan antar muka untuk menu *Show* Jadwal.

3.1. Metode Pengujian

Pengujian dilakukan dengan menghitung kecepatan komputasi dan aksesibilitas dari *mobile device* dan *mobile browser* yang berbeda. Kecepatan komputasi diukur dengan menghitung berapa lama waktu yang dibutuhkan untuk membuat sebuah jadwal. Sedangkan aksesibilitas diukur dengan menguji apakah aplikasi dapat tampil dan bekerja sesuai dengan yang diinginkan di beberapa tipe *mobile browser* yang berbeda.