

# Aplikasi Penjadwalan Asisten Penjaga Ujian dengan Metode Algoritma Genetika

Joseph Clinton Adi Gunawan Putra<sup>1</sup>, Kristo Radion Purba<sup>2</sup>, Henry Novianus Palit<sup>3</sup>

Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121 – 131 Surabaya 60236

Telp. (031) – 2983455, Fax. (031) – 8417658

Email :m26412001@john.petra.ac.id<sup>1</sup>, kristo@petra.ac.id<sup>2</sup>, hnpalit@petra.ac.id<sup>3</sup>

## ABSTRAK

Pada saat menjelang ujian tengah semester atau saat ujian akhir semester di Universitas Kristen Petra pasti dibutuhkan asisten yang akan menjaga pada setiap kelas pada ujian mata kuliah tertentu. Oleh karena itu diperlukan penjaga sebelum menentukan jadwal yang akan dijaga, maka diperlukan rapat supaya bisa menentukan jadwal tersebut. Jadi aplikasi ini dibuat untuk lebih memudahkan asisten dan koordinator ujian untuk menentukan jadwal tanpa membuang lebih banyak waktu antar koordinator. Koordinator dan asisten cukup membuka aplikasi ini. Asisten dapat memasukkan jadwal ujian mereka sesuai dengan tanggal dan jam mereka ujian, dan koordinator juga dapat menghasilkan program dengan menggunakan algoritma genetika sesuai batasan-batasan yang telah diberikan oleh koordinator ujian. Sehingga jadwal yang telah dihasilkan memiliki kecocokan dengan hasil akhirnya. Dan memiliki berbagai fitur tambahan seperti memberikan *notice* kepada asisten, dapat *export* jadwal jadi ke Microsoft Office Excel atau print langsung. Pada saat *survey* algoritma genetika, program ini memiliki jadwal sesuai dengan batasan batasan yang diberikan oleh koordinator ujian. Jadi program ini layak dipakai oleh koordinator ujian Universitas Kristen Petra dalam prodi Informatika.

**Kata Kunci:** Algoritma Genetika, Penjadwalan, Ujian

## ABSTRACT

*At the time ahead of midterm or in semester's final exam at Petra Christian University is definitely needed assistants that will keep on every class on the test subjects. Therefore we need a proctor before determining a schedule that will be covered, it would require a meeting in order to determine the schedule. So this application is made to be easier for assistants and coordinator of exam to determine the schedule and not waste more time between coordinator. The coordinator and assistant simply open the application. The assistant can put their schedule in accordance with the date and time of their examinations, and the coordinator can also produce the program by using genetic algorithm with the restrictions that have been given by the exam's coordinator. So the schedule that has been produced have a match with the end result. And has a variety of additional features such as providing notice to assistant, and export schedules make it to Microsoft Office Excel or print directly. At the time of the survey the genetic algorithm, this program has a schedule in accordance with the limit given by the coordinator of the exam. So this program should be used by the coordinator of the exam of Petra Christian University in the study program information technology.*

**Keywords:** Genetic Algorithm, Scheduling, Exam.

## 1. PENDAHULUAN

Penjadwalan yang ada di teknik Informatika masih tergolong dalam penjadwalan yang kurang efektif karena masih memiliki kendala waktu seperti memakan waktu yang banyak saat rapat, tentunya penjadwalan yang menggunakan teknik lama dapat digantikan juga dengan menggunakan teknik komputerisasi, penjadwalan yang dilakukan untuk para asisten dosen program studi Informatika masih menggunakan cara yang manual dengan mengadakan rapat, dan mencatat hasil dari penjadwalan dalam excel, oleh karena itu dengan cara yang manual ini pastinya kurang efisien dari pada menggunakan teknik komputerisasi, jadi untuk penjadwalan ujian cenderung lebih bagus jika dibuat secara komputerisasi karena diberikan rekomendasi jadwal sehingga lebih mudah dalam pembuatan jadwal asisten, salah satu algoritma yang dapat digunakan dalam membuat penjadwalan adalah algoritma genetika.

## 2. LANDASAN TEORI

### 2.1 Algoritma Genetika

*Genetic Algorithm* atau algoritma genetika adalah algoritma yang mempunyai solusi dalam memecahkan masalah secara alamiah karena dalam algoritma genetika ini memiliki kesamaan dengan teori evolusi, algoritma ini banyak digunakan untuk mendapatkan suatu nilai atau solusi optimal terhadap suatu permasalahan, terdapat beberapa langkah-langkah pada algoritma genetika, pertama adalah inialisasi dilangkah ini membuat inialisasi populasi seperti menentukan kromosom yang akan dibentuk, langkah kedua adalah menentukan nilai *fitness* atau yang dapat disebut *fitness cost* yang dapat menentukan *scoring* untuk setiap kromosom dalam suatu populasi, ketiga adalah menentukan kedua *parent* yang akan dikawinkan banyak cara untuk salah satunya dengan metode *roulette wheels*, keempat setelah ketemu dua *parent* dengan menggunakan *roulette wheels* mengkawinkan atau meng-*crossover*-kan antara kedua *parent* tersebut dengan presentase *crossover* tertentu sehingga memiliki anak, langkah kelima adalah mutasi, mutasi tidak selalu terjadi dalam satu iterasi tetapi dengan presentase tertentu baru akan terjadi suatu mutasi, mutasi adalah penggantian secara acak pada salah satu gen yang akan digantikan dengan suatu *value* yang berbeda, setelah semua langkah-langkah itu terpenuhi maka akan dicek lagi apakah iterasi sudah mencapai batas maksimal jika belum maka langkah-langkah tadi akan diulang terus sampai iterasi selesai atau sampai *stopping criteria*, *stopping criteria* adalah bagian dari *genetic algorithm* untuk menghentikan iterasi jika dianggap hasil yang didapat sudah *valid*[1].

### 2.2 Phonegap

Untuk membangun sebuah aplikasi setiap perangkat *smartphone* seperti android, windows phone, iPhone membutuhkan sebuah *frameworks* dan bahasa pemrograman yang berbeda, jadi

phonegap memecahkan masalah ini dengan menjembatani antara *mobile device* dengan *web application*, phonegap adalah standar *compliant*. Phonegap adalah sebuah *framework open source* yang dipakai untuk membuat aplikasi *cross-platform mobile* dengan menggunakan bahasa HTML, CSS, Javascript. Phonegap telah di-*download* lebih dari 1 juta kali dan sedang digunakan lebih dari 400.000 *developers*, ribuan aplikasi dibuat menggunakan phonegap sudah ada di *app stores* dan *directories*, kita dapat mencarinya disana. Phonegap *code* telah diberikan ke *Apache Software Foundation*(ASF) dengan nama Apache Cordova, melalui ASF pengembangan phonegap akan memastikan bahwa ini sebagai *open project* dan bisa digunakan untuk siapapun dan *open source* dibawah lisensi Apache versi 2.0, apache cordova adalah sekumpulan API yang mengijinkan *developer mobile application* dapat mengakses semua fungsi native dari device seperti kamera, baterai status, dan lain sebagainya. Dengan API Cordova aplikasi dapat dibuat tanpa menggunakan kode-kode native seperti Java, C, dan bahasa pemrograman lainnya dari perkembangan aplikasi yang sudah ada[4].

### 2.3 Javascript

Javascript adalah bahasa pemrograman yang digunakan untuk membuat HTML, PHP ,dan *website* untuk *server*, laptop, PC, Android, dan lainnya. Kode javascript dapat diletakan pada halaman HTML, javascript digolongkan sebagai bahasa *scripting* sisi *client* artinya script javascript tersebut dapat dijalankan di computer dan dapat membantu sebuah halaman *website* menjadi lebih menarik *use / client* untuk memakai sebuah website tersebut. Kode javascript dimasukan kedalam halaman HTML dengan menggunakan kode seperti berikut `<script></script>`. Javascript juga ada di *file* dengan ekstensi .js seperti berikut `test.js`[5].

### 2.4 HTML5

HTML5 adalah versi yang baru dari HTML dan XHTML. HTML yang standar adalah sebuah *single language* yang dapat ditulis pada HTML dan XML, HTML5 mencoba untuk memecahkan masalah yang ada dalam HTML[3]. Perbedaan antara HTML4 dengan HTML5, *syntax* dari penulisan sebuah *coding, doctype, MathML* dan SVG, elemen-elemen baru di HTML5, atribut baru, perubahan elemen sebelumnya contohnya *syntax strong* , pergantian atribut seperti dahulu `alt` tidak bisa disambungkan dengan `img` tetapi di HTML5 sekarang `alt` bisa disambungkan dengan `img`, HTML5 dilengkapi dengan API's, ekstensi ke document, ekstensi ke HTML element, opsi ke *interface* yang lain.

### 2.5 GCM

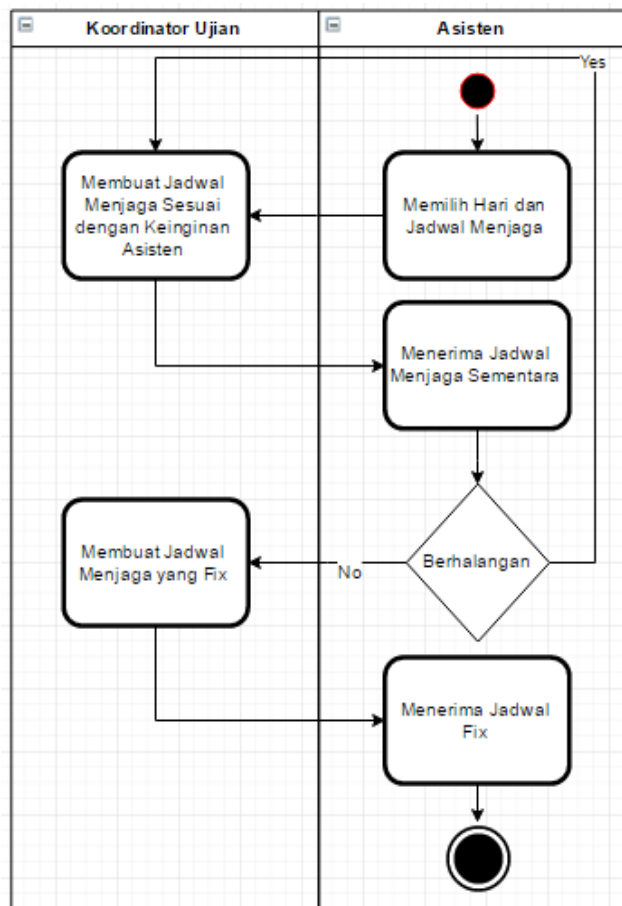
GCM (Google Cloud Messaging) merupakan layanan gratis yang membantu pengembang mengirim pesan ke berbagai *platform* : Android, iOS, dan Chrome. Misalnya, *server* dapat mengirim pesan ke satu perangkat, sekumpulan perangkat, atau perangkat yang berlangganan ke topik. Selain itu, aplikasi di perangkat dapat langsung mengirim pesan ke *server* dan ke perangkat yang ada dalam grup yang sama[2].

## 3. ANALISIS DAN DESAIN SISTEM

### 3.1 Analisis Sistem Manual

Proses penjadwalan asisten di program studi informatika Universitas Kristen Petra masih menggunakan sistem kumpul untuk menentukan jadwal yang pas untuk para asisten dan rapat kurang lebih 2 minggu sebelum ujian UTS atau UAS dimulai, koordinator ujian memberikan jadwal ujian, kemudian asisten memilih jadwal menjaga mereka masing-masing, kemudian

koordinator cek apakah bentrok dengan ujian asisten juga, terlalu banyak menjaga pada hari itu atau tidak, angkatan bawah tidak boleh menjaga angkatan atas, kelas mengulang harap dijaga oleh laki-laki, mengecek ketegasan asisten, kemudian asisten menerima hasil menjaga mereka. Untuk *activity diagram* sistem *manual* dapat dilihat pada Gambar 1.

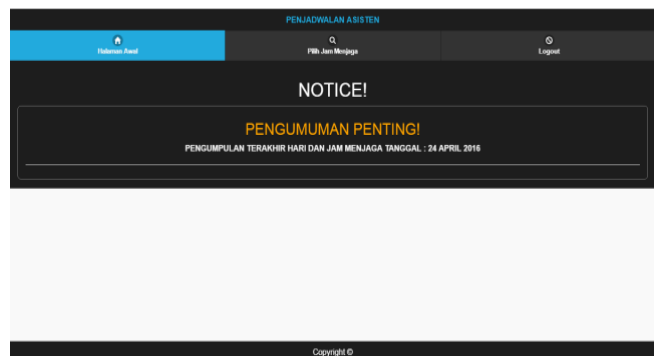


Gambar 1. Activity Sistem Manual.

## 4. IMPLEMENTASI

### 4.1 Implementasi Aplikasi Mobile

Pada aplikasi *mobile* ini memiliki beberapa menu seperti pada halaman awal asisten memiliki notice untuk asisten, pemilihan jam menjaga yang diinginkan, *logout*. Halaman aplikasi asisten dapat dilihat pada Gambar 2.



Gambar 2. Halaman Aplikasi Asisten.

## 4.2 Implementasi Aplikasi Website

Aplikasi ini dikhususkan untuk admin dapat mengakses berbagai macam fitur yang digunakan untuk *me-manage* asisten, jadwal, dan *notice* asisten. Untuk *me-manage* asisten dapat dilihat pada Gambar 3 dan pada Gambar 4.

Gambar 3. Tampilan Add Asisten.

List Asisten				
Nama Asisten	NRP	Angkatan	Jenis Kelamin	No Hp
tester	26412001	12	laki-laki	08121222282
tester4	26412172	12	laki-laki	82137172312
gary	26412070	12	laki-laki	81751400000
tester3	26412059	12	laki-laki	8811123122
jeffrey	26412038	12	laki-laki	87851818104
ronald	26412027	12	laki-laki	81703642414
tester2	26412008	12	laki-laki	889123122
michael	26413112	13	laki-laki	87810930049
wonges	26413051	13	laki-laki	83830030079

Gambar 4. Tampilan List Asisten.

Untuk menu *settings* pembobotan ditujukan ke admin supaya admin dapat menambahkan bobot yang sesuai dan mendapatkan *fitness value* yang seperti admin inginkan pada algoritma genetika, *settings* pembobotan dapat dilihat pada Gambar 5 dan Gambar 6.

Gambar 5. Tampilan Setting Pembobotan.

Gambar 6. Tampilan Setting Pembobotan 2.

Untuk menu menambahkan *notice* asisten ditujukan pada *admin* supaya *admin* dapat menambahkan *notice* asisten dan *admin* dapat *edit* dan *delete* juga *notice* yang telah dibuat, untuk tampilannya dapat dilihat pada Gambar 7 dan pada Gambar 8.

### Add Notice Asisten

Judul Notice:

Deskripsi Notice:

Expired Notice:

**Gambar 7. Tampilan Add Notice Asisten.**

### NOTICE ASISTEN

ID Notice :

Isi : 

UNTUK PARA BRO  
DAN SIS ASISTEN  
TOLONG DIISI  
JADWAL UJIAN KALIAN

Expired Date :

**Gambar 8. Tampilan Edit Notice Asisten.**

Untuk *menu* yang terakhir adalah *menu* membuat jadwal asisten dimana *menu* ini memiliki beberapa fungsi lagi di dalamnya seperti *import excel*, *generate jadwal*, *export excel*, *print jadwal*, *edit jadwal generate*, *view komentar asisten*, untuk tampilan dapat dilihat pada Gambar 9 dan pada Gambar 10.

## 5. PENGUJIAN SISTEM

Berdasarkan pengujian algoritma genetika pada program ini didapatkan beberapa macam percobaan dengan menggantikan presentase *crossover*, presentase mutasi, banyak populasi, banyak iterasi dalam algoritma genetika pada beberapa gambar grafik ini sudah memiliki presentase mutasi, presentase crossover, banyak populasi yang sudah ditetapkan, pada grafik ini yang akan diujikan adalah banyak iterasi yang akan diganti untuk mengetahui pada iterasi yang mana mendapatkan hasil *top fitness* yang paling baik.

### Membuat Jadwal Asisten

#### Upload Jadwal UTS / UAS

File:

---

#### Generate Jadwal UTS / UAS

---

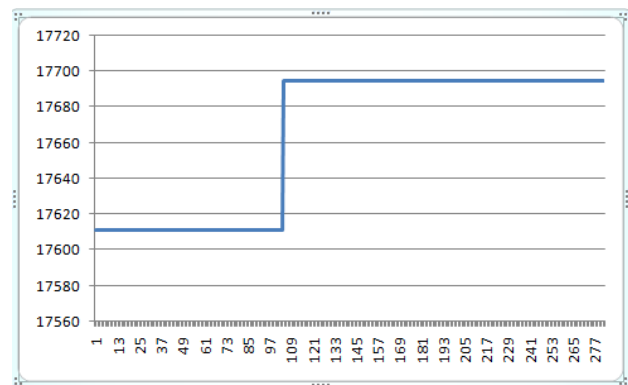
#### Export Excel

**Gambar 9. Tampilan Membuat Jadwal Asisten.**

JADWAL REKOMENDED ASISTEN						
	Dosen	Ruang	Peserta	Penjaga	Koordinator	Edit
k	Prof. Ir. ROLLY INTAN, M.A.Sc., Dr.Eng	P622	33	-	Gunadi	<input type="button" value="Edit"/>
k	ADI WIBOWO, ST., M.T.	P505	28	-	Gunadi	<input type="button" value="Edit"/>
	Ir. KARTIKA GUNADI, M.T.	1208	26	jonatan	Gunadi	<input type="button" value="Edit"/>
	Ir. KARTIKA GUNADI, M.T.	1407	55	adam	Gunadi	<input type="button" value="Edit"/>

© 2015 Easy Admin Panel. All Rights Reserved | Design by w3layouts.

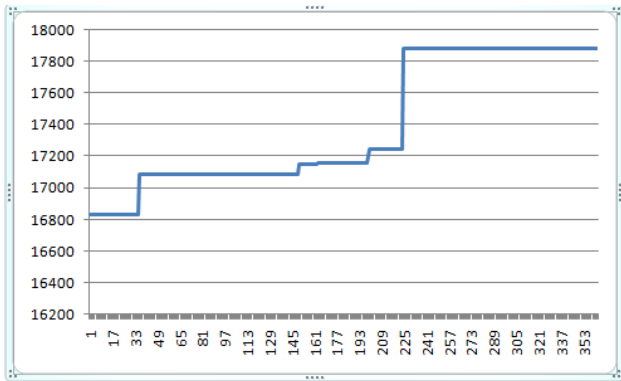
**Gambar 10. Tampilan Membuat Jadwal Asisten 2.**



**Gambar 11. Perhitungan Fitness dengan Parameter Presentase Crossover 60%, Presentase Mutasi 0.007, Populasi 30, Iterasi 500 kali.**

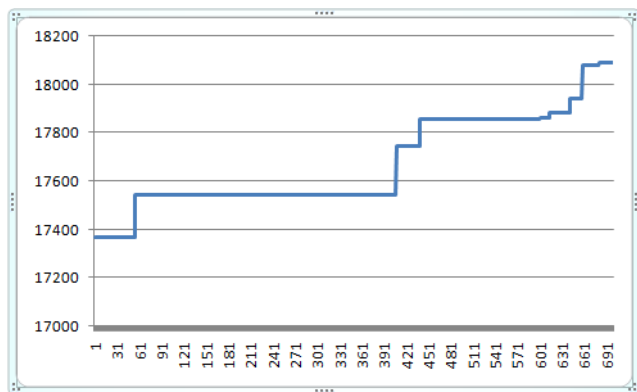
Pada Gambar 11 adalah contoh pengujian algoritma dengan melihat grafik kenaikan pada setiap iterasi dengan memiliki *fitness cost* dalam range 17.680 sampai 17.700, dengan parameter presentase mutasi sebesar 0.007 atau 7/1000, presentase *crossover*

sebesar 0.6 atau 60%, banyak populasi sebanyak 30, dan banyak iterasi sebanyak 500 kali.



**Gambar 12. Perhitungan Fitness dengan Parameter Presentase Crossover 60%, Presentase Mutasi 0.007, Populasi 30, Iterasi 600 kali.**

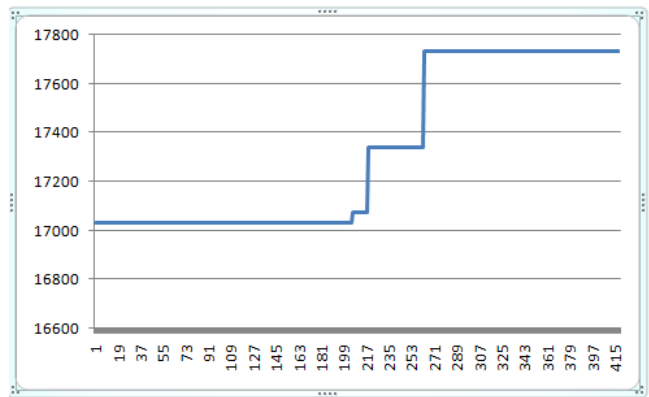
Pada Gambar 12 adalah contoh pengujian algoritma dengan melihat grafik kenaikan pada setiap iterasi dengan memiliki *fitness cost* dalam *range* 17.480 sampai 18.000, dengan parameter presentase mutasi sebesar 0.007 atau 7/1000, presentase *crossover* sebesar 0.6 atau 60%, banyak populasi sebanyak 30, dan banyak iterasi sebanyak 600 kali.



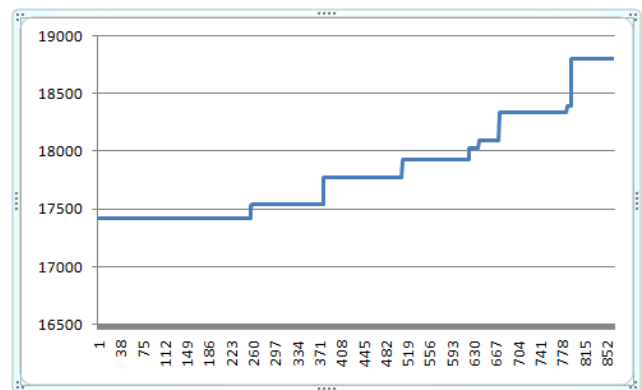
**Gambar 13. Perhitungan Fitness dengan Parameter Presentase Crossover 60%, Presentase Mutasi 0.007, Populasi 30, Iterasi 700 kali.**

Pada Gambar 13 adalah contoh pengujian algoritma dengan melihat grafik kenaikan pada setiap iterasi dengan memiliki *fitness cost* dalam *range* 17.200 juta sampai 18.200, dengan parameter presentase mutasi sebesar 0.007 atau 7/1000, presentase *crossover* sebesar 0.6 atau 60%, banyak populasi sebanyak 30, dan banyak iterasi sebanyak 700 kali.

Pada Gambar 14 adalah contoh pengujian algoritma dengan melihat grafik kenaikan pada setiap iterasi dengan memiliki *fitness cost* dalam *range* 17.000 sampai 17.800, dengan parameter presentase mutasi sebesar 0.007 atau 7/1000, presentase *crossover* sebesar 0.6 atau 60%, banyak populasi sebanyak 30, dan banyak iterasi sebanyak 800 kali.



**Gambar 14. Perhitungan Fitness dengan Parameter Presentase Crossover 60%, Presentase Mutasi 0.007, Populasi 30, Iterasi 800 kali.**



**Gambar 15. Perhitungan Fitness dengan Parameter Presentase Crossover 60%, Presentase Mutasi 0.007, Populasi 30, Iterasi 900 kali.**

Pada Gambar 15 adalah contoh pengujian algoritma dengan melihat grafik kenaikan pada setiap iterasi dengan memiliki *fitness cost* dalam *range* 17.000 sampai 19.000, dengan parameter presentase mutasi sebesar 0.007 atau 7/1000, presentase *crossover* sebesar 0.6 atau 60%, banyak populasi sebanyak 30, dan banyak iterasi sebanyak 900 kali.

Jadi dari kelima percobaan dengan menggunakan grafik dapat disimpulkan bahwa pada grafik Gambar 14 memiliki jumlah nilai *fitness cost* yang tertinggi, Hasil percobaan dapat dilihat pada Tabel 1 berikut ini.

**Tabel 1. Tabel Perbandingan Antara 5 Grafik.**

	Prese ntase Mutasi	Prese ntase Cross over	Banyak Populasi	Banyak Iterasi	Top Fitness
Grafik 1	0. 007	0.6	30	500	17695
Grafik 2	0. 007	0.6	30	600	17883

**Tabel 1. Tabel Perbandingan Antara 5 Grafik.**

	Prese ntase Mutasi	Prese ntase Cross over	Banyak Populasi	Banyak Iterasi	Top Fitness
Grafik 3	0.007	0.6	30	700	18087
Grafik 4	0.007	0.6	30	800	17733
<b>Grafik 5</b>	<b>0.007</b>	<b>0.6</b>	<b>30</b>	<b>900</b>	<b>18802</b>

## 6. KESIMPULAN & SARAN

Berdasarkan pengujian yang sudah dilakukan dapat disimpulkan bahwa aplikasi *mobile* yang ditujukan untuk asisten dapat digunakan dengan mudah oleh asisten karena desain dibuat secara simple dan juga mudah di pahami oleh asisten, dan juga dengan aplikasi *website* untuk *admin* juga mudah dipahami oleh *admin* dan memiliki beberapa fungsi-fungsi dalam *website* yang mudah dipahami oleh *admin*, pada algoritma genetika juga didesain sedemikian rupa sehingga sesuai dengan batasan-batasan yang diberikan oleh *admin*, sehingga jadwal *recommended* yang di-*generate*-kan menjadi sesuai dengan keinginan *admin* dan juga dengan keinginan asisten, Program ini dapat digunakan untuk pihak koordinator ujian sebagai adminnya, selain dapat

mempermudah dalam pembuatan jadwal rekomendasi program ini juga dapat membantu dalam *import* excel supaya koordinator ujian atau *admin* dapat dengan mudah memasukkan sebuah *data* tidak dengan cara *manual* seperti mengetik data satu persatu, dan juga memiliki fitur *export* excel dan *print* jadwal yang sudah *fix*..

Saran yang didapatkan adalah menambahkan fitur untuk membaca excel lebih baik lagi sehingga dengan berbagai indikasi seperti warna *cell* atau warna tulisan menjadi suatu indikasi tertentu. Memberikan batasan-batasan lagi sesuai dengan *request admin*. Pembuatan aplikasi *mobile* pada apple/ iOS bukan hanya pada android. Meng-*edit* tampilan *website* ataupun *mobile* agar lebih menarik.

## 7. REFERENCES

- [1] Arie B. 2014. *Struktur Umum Algoritma Genetika*. Retrieved May 25, 2016 from <http://informatika.web.id/struktur-umum-algoritma-genetika.html>
- [2] GCM. 2015. *GCM introduction*. Retrieved May 25, 2016 from <https://developers.google.com/cloud-messaging/>
- [3] Pieters S. 13 April 2015. *Difference from html4*. Retrieved October 24, 2015 from <https://html-differences.whatwg.org/>
- [4] Phonegap. 2015. *Phonegap introduction*. Retrieved May 25, 2016 from <http://phonegap.com/about/>
- [5] Yogesh P. 2012. *Beginning Phonegap : Mobile Web Framework for Javascript and Html5*. New York: Apress.