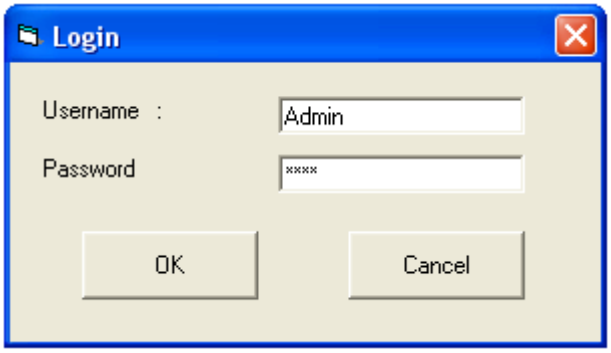
BAB IV
ANALISIS DAN PENGUJIAN

4.1 Implementasi Sistem

Implementasi sistem adalah penerapan hasil perancangan yang telah diuraikan sebelumnya pada bab perancangan sistem. Implementasi yang dilakukan antara lain adalah menerapkan perancangan antarmuka kedalam bentuk software. Tujauan implementasi adalah mengkonfirmasikan model program perancangan pada sistem sehingga user dapat memberikan masukan kepada pembangun sistem. Berikut ini merupakan gambaran tampilan antarmuka yang telah dibuat.

1) Tampilan halaman login

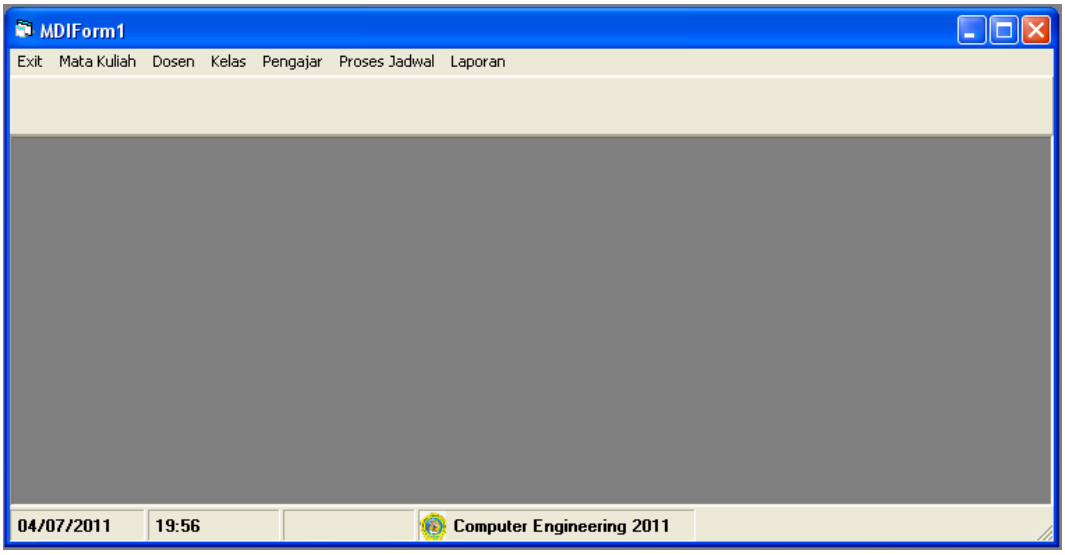
Pada proses ini, user yang akan menggunakan aplikasi akan diminta untuk memasukan username dan password. Sehingga apabila ada user tidak sesuai maka tidak akan melakukan proses selanjutnya, yaitu melakukan penyusunan jadwal menggunakan algoritma genetika.



Gambar 4.1 Login

2) Tampilan menu utama

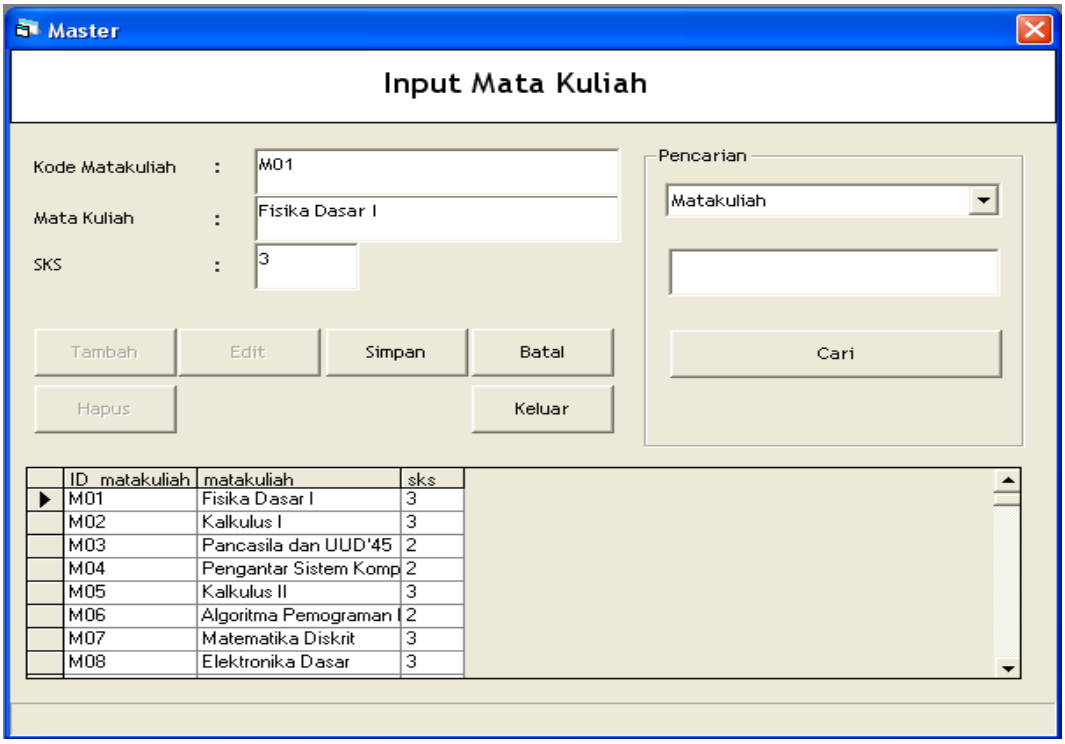
Pada tampilan ini halaman utama terdapat berberapa pilihan tombol diantaranya tobol keluar, untuk keluar dari aplikasi. Kemudian terdapat pula proses jadwal, untuk melakukan proses penjadwalan dengan menggunakan algoritma genetika.



Gambar 4.2 Menu utama

3) Tampilan data mata kuliah

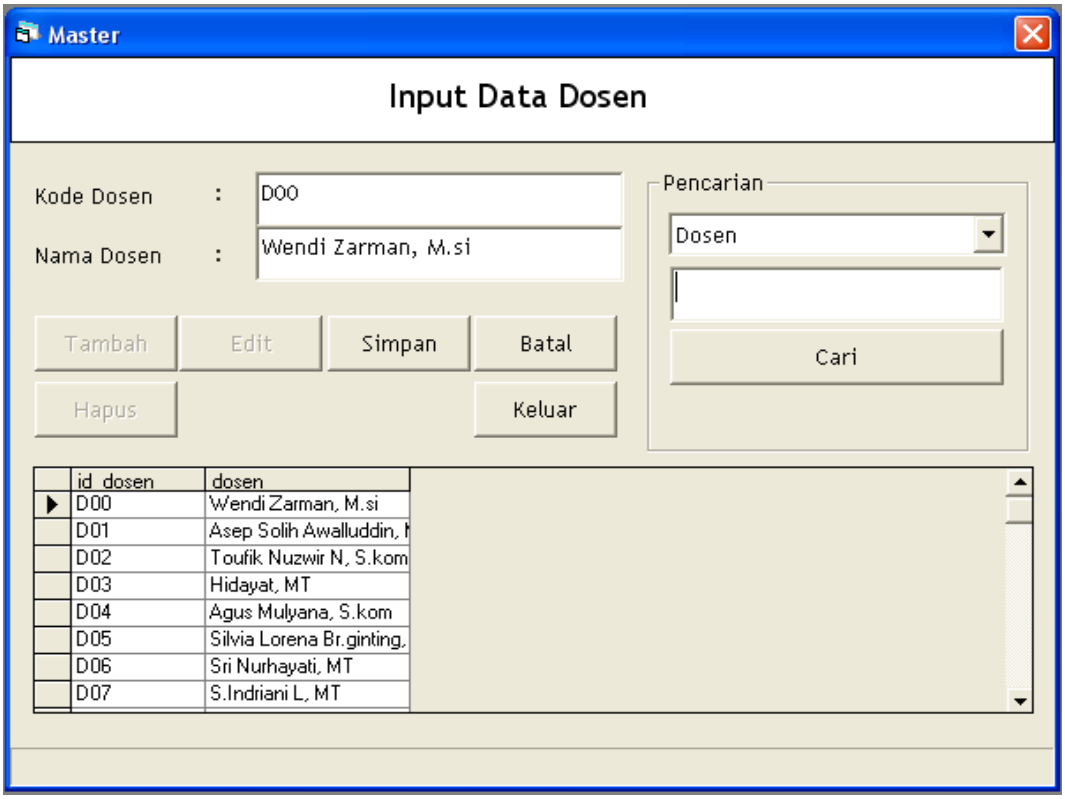
Dengan mengklik data mata kuliah pada menu, maka akan tampil tampilan tabel mata kuliah yang didalamnya terdapat tombol-tombol pengolahan data mata kuliah yang disediakan, yaitu proses pencarian, tambah data, edit data, simpan data, batal, menghapus data dan keluar dari aplikasi.



Gambar 4.3 Data mata kuliah

4) Tampilan data dosen

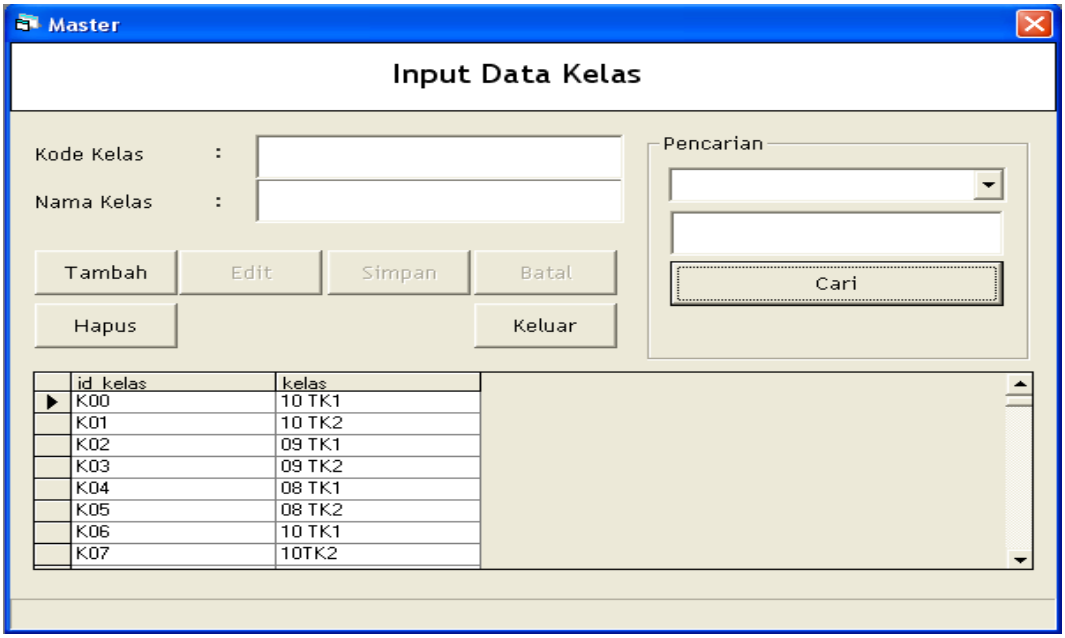
Dengan mengklik data dosen pada menu, maka akan tampil tampilan tabel dosen yang didalamnya terdapat tombol-tombol pengolahan data dosen yang disediakan, yaitu proses pencarian, tambah data, edit data, simpan data, batal, menghapus data dan keluar dari aplikasi.



Gambar 4.4 Data dosen

5) Tampilan data kelas

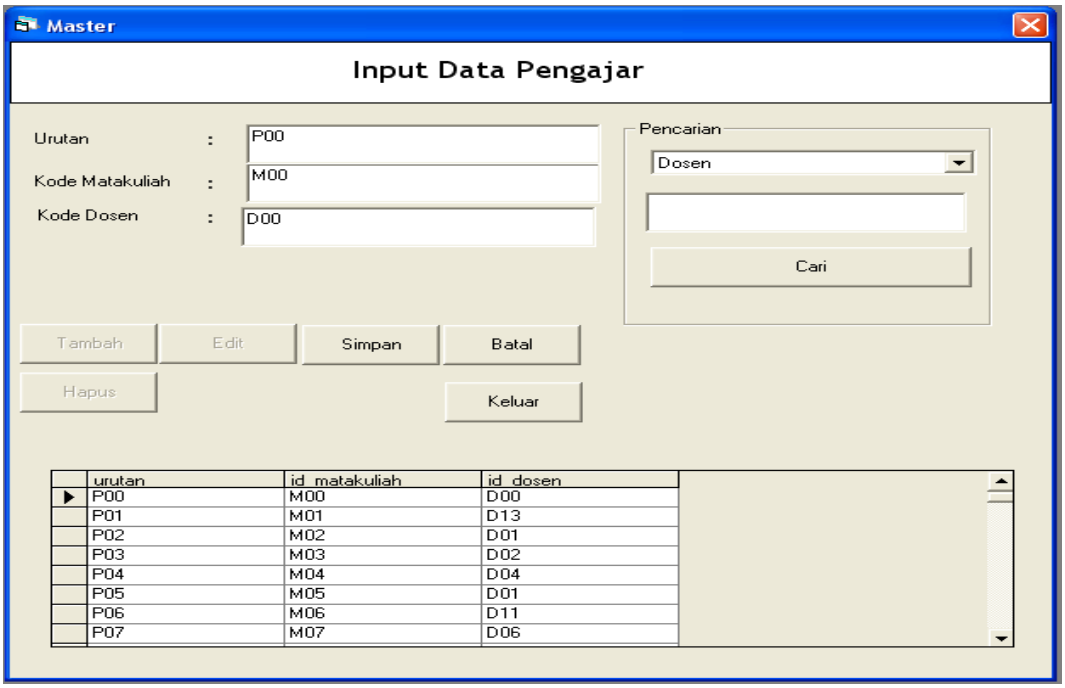
Dengan mengklik data kelas pada menu, maka akan tampil tampilan tabel kelas yang didalamnya terdapat tombol-tombol pengolahan data kelas yang disediakan, yaitu proses pencarian, tambah data, edit data, simpan data, batal, menghapus data dan keluar dari aplikasi.



Gambar 4.5 Data kelas

6) Tampilan data pengajar

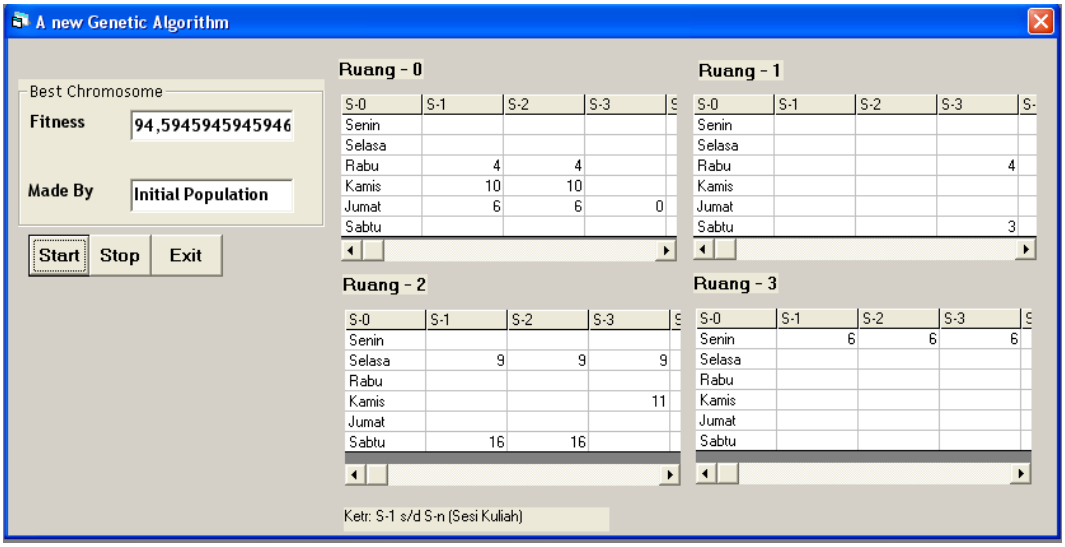
Dengan mengklik data pengajar pada menu, maka akan tampil tampilan tabel pengajar yang didalamnya terdapat tombol-tombol pengolahan data pengajar yang disediakan, yaitu proses pencarian, tambah data, edit data, simpan data, batal, menghapus data dan keluar dari aplikasi.



Gambar 4.6 Data pengajar

7) Tampilan pembentukan populasi awal

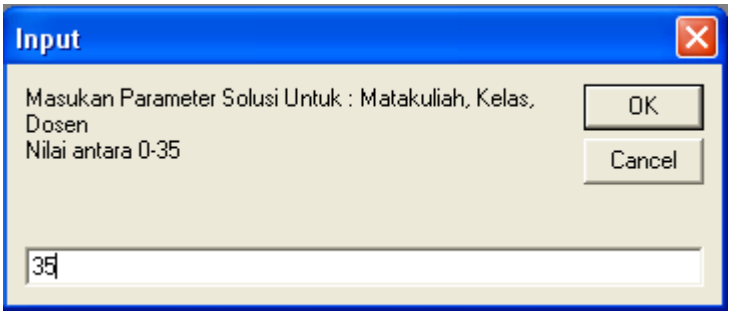
Untuk melakukan proses pembentukan populasi awal dimulai dengan memasukan parameter-parameter solusi yang udah tersusun.



Gambar 4.7 Data Pembentukan populasi awal

8) Tampilan parameter solusi

Dengan memasukan parameter solusi maka penjadwalan mata kuliah akan dibentuk secara acak untuk menempati timeslot yang udah tersusun seperti terlihat digambar 4.7.



Gambar 4.8 Parameter solusi

4.2 Pengujian Sistem

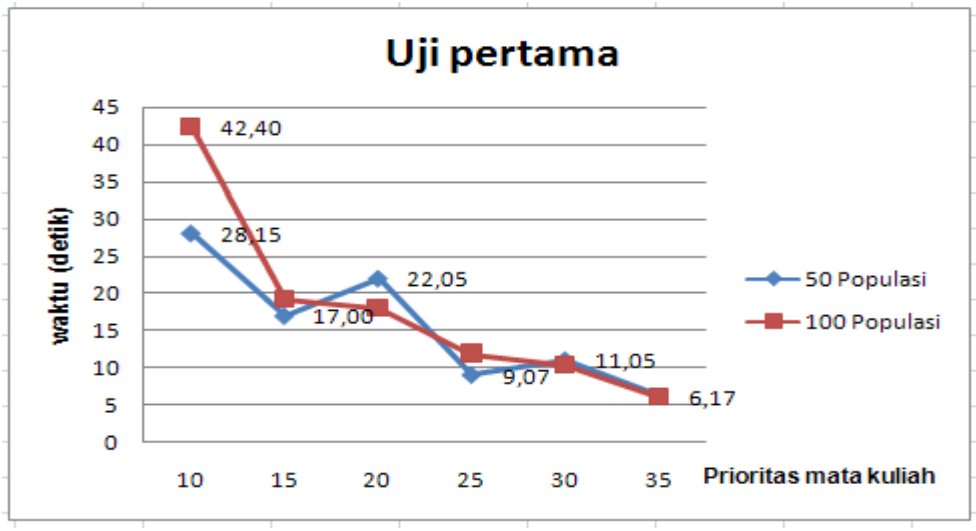
Pengujian parameter algoritma genetika ini bertujuan untuk menganalisa kinerja algoritma genetika pada proses penyusunan jadwal mata kuliah sesuai dengan ketentuan yang telah ditetapkan pengujian nantinya dilakukan dengan cara mengubah-ubah masukan parameter pada proses penjadwalan sehingga nanti diharapkan bisa didapat rekomendasi atau patokan parameter algoritma genetika yang menghasilakan proses penyusunan jadwal mata kuliah yang baik.

Pada tabel 4.1 dibawah ini menunjukan beberapa hasil pengujian sistem algoritma genetika dengan mengubah-ubah parameter masukan dan probabilitas *crossover* Pc dengan nilai 75 dan probabilitas mutasi Pm dengan nilai 3. Apabila jumlah populasi diset 50 dan 100, jumlah generasi diset 100.

Tabel 4.1 Hasil pengujian pertama:

Populasi	Pc	Pm	*Fitness*	Aktivitas Perkuliahan			Waktu (detik)
				Jum Mtkul	Frekuensi Dosen	Frekuensi Kelas	
50	75	3	90,909	10	1	3	28,15
50	75	3	93,75	15	2	1	17,00
50	75	3	90	20	1	2	22,05
50	75	3	96	25	1	2	9,07
50	75	3	96,774	30	0	3	11,05
50	75	3	**97,222**	**35**	**0**	**0**	**6,17**
100	75	3	90,909	10	0	3	42,40
100	75	3	93,75	15	0	2	19,21
100	75	3	95,238	20	1	2	17,99
100	75	3	**96,153**	**25**	**0**	**2**	**11,89**
100	75	3	96,774	30	2	3	10,33
100	75	3	97,222	35	1	2	6,06

Dari data tabel 4.1 terlihat bahwa jadwal terbaik diproleh pada masukan 35 mata kuliah dan jumlah populasi 50 dimana hanya satu frekuensi dosen yang tinggi dalam sehari dan tidak terdapat frekuensi kelas yang tinggi dalam sehari dengan waktu pengeksekusian 6,17 detik, sedangkan pada populasi 100 jadwal yang terbaik pada masukan 25 mata kuliah dimana tidak terdapat frekuensi dosen yang tinggi tetapi terdapat dua frekuensi kelas yang tinggi dan waktu pengeksekusian 11,89 detik.



Gambar 4.9 Grafik Uji pertama.

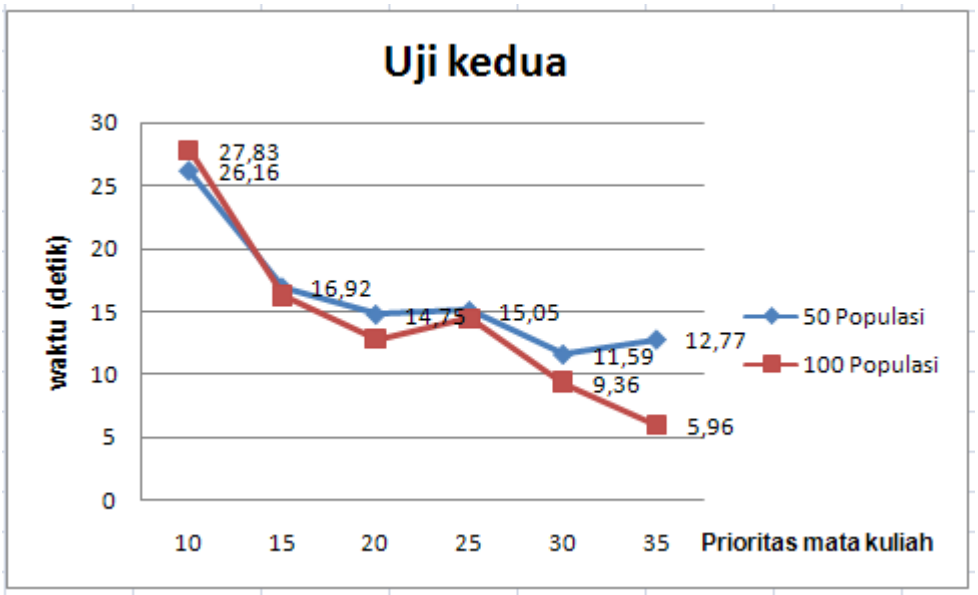
Pada gambar 4.9 menunjukan laju kecepatan algoritma genetik dalam mengeksekusi, bisa telihat bahwa semakin besar ukuran populasi kemungkinan lebih besar untuk menemukan solusi yang terbaik, namun membutuhkan waktu pengeksekusian yang lebih lama.

Pada tabel 4.2 dibawah ini menunjukan beberapa hasil pengujian sistem algoritma genetika dengan mengubah-ubah parameter masukan dan probabilitas *crossover* Pc dengan nilai 75 dan probabilitas mutasi Pm dengan nilai 4. Apabila jumlah populasi diset 50 dan 100, jumlah generasi diset 100.

Tabel 4.2 Hasil pengujian kedua:

Populasi	Pc	Pm	*Fitness*	Aktivitas Perkuliahan			Waktu (detik)
				Jm Mtkul	Frekuensi Dosen	Frekuensi Kelas	
50	75	4	83,333	10	0	3	26,16
50	75	4	79,947	15	2	2	16,92
50	75	4	**90,909**	**20**	**0**	**2**	**14,75**
50	75	4	96,153	25	0	3	15,05
50	75	4	96,774	30	1	2	11,59
50	75	4	97,142	35	1	2	12,84
100	75	4	90,909	10	2	3	27,83
100	75	4	88,235	15	0	0	16,25
100	75	4	90	20	0	2	12,84
100	75	4	**96,153**	**25**	**0**	**1**	**14,47**
100	75	4	96,774	30	1	1	9,36
100	75	4	97,222	35	0	2	5,96

Dari data tabel 4.2 terlihat bahwa jadwal terbaik diproleh pada masukan 20 mata kuliah dan jumlah populasi 50 dimana tidak terdapat frekuensi dosen yang tinggi dalam sehari dan terdapat dua frekuensi kelas yang tinggi dalam sehari dengan waktu pengeksekusian 14,75 detik, sedangkan pada populasi 100 jadwal yang terbaik pada masukan 25 mata kuliah dimana tidak terdapat frekuensi dosen yang tinggi tetapi terdapat satu frekuensi kelas yang tinggi dan waktu pengeksekusian 14,75 detik.



Gambar 4.10 Grafik Uji kedua.

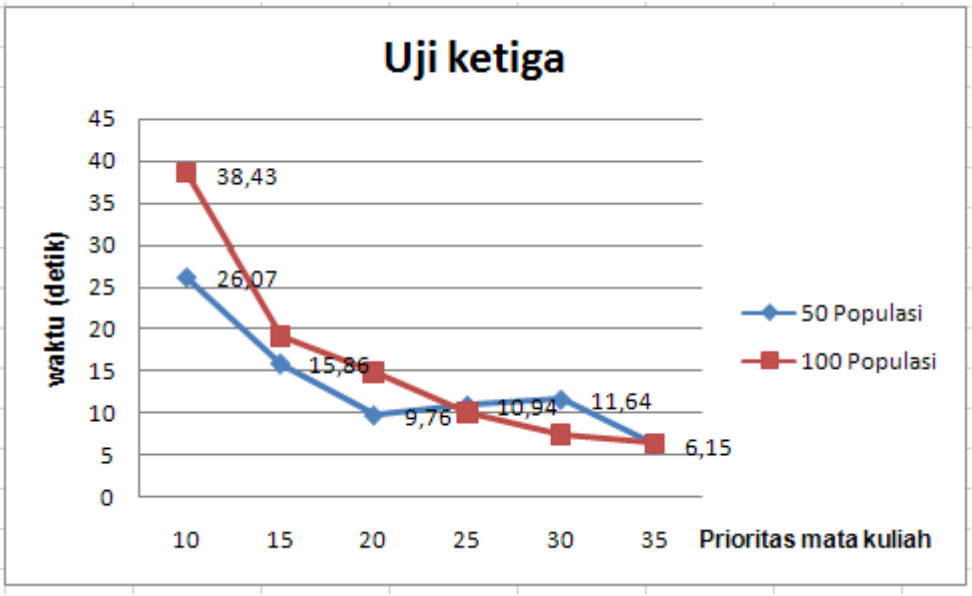
Pada gambar 4.10 menunjukan laju kecepatan algoritma genetik dalam mengeksekusi data, bisa telihat bahwa semakin besar ukuran populasi kemungkinan lebih besar untuk menemukan solusi yang terbaik, namun membutuhkan waktu pengeksekusian yang lebih lama.

Pada tabel 4.3 dibawah ini menunjukan beberapa hasil pengujian sistem algoritma genetika dengan mengubah-ubah parameter masukan dan probabilitas *crossover* Pc dengan nilai 75 dan probabilitas mutasi Pm dengan nilai 5. Apabila jumlah populasi diset 50 dan 100, jumlah generasi diset 100.

Tabel 4.3 Hasil pengujian ketiga:

Populasi	Pc	Pm	*Fitness*	Aktivitas Perkuliahan			Waktu (detik)
				Jm Mtkul	Frekuensi Dosen	Frekuensi Kelas	
50	75	5	90,909	10	2	3	26.07
50	75	5	93,75	15	1	4	15.85
50	75	5	95	20	0	2	9.76
50	75	5	96,153	25	0	3	10.94
50	75	5	96,774	30	1	2	11.64
50	75	5	**97,222**	**35**	**0**	**1**	**6.15**
100	75	5	90,909	10	1	3	38.43
100	75	5	**93,75**	**15**	**0**	**1**	**19.21**
100	75	5	95	20	0	2	14.87
100	75	5	96,153	25	1	1	10.04
100	75	5	96,774	30	0	2	7.41
100	75	5	97,222	35	0	2	6.39

Dari data tabel 4.3 terlihat bahwa jadwal terbaik diproleh pada masukan 35 mata kuliah dan jumlah populasi 50 dimana tidak terdapat frekuensi dosen yang tinggi dalam sehari dan terdapat satu frekuensi kelas yang tinggi dalam sehari dengan waktu pengeksekusian 6,15 detik, sedangkan pada populasi 100 jadwal yang terbaik pada masukan 15 mata kuliah dimana tidak terdapat frekuensi dosen yang tinggi tetapi terdapat satu frekuensi kelas yang tinggi dan waktu pengeksekusian 19,21 detik.



Gambar 4.11 Grafik Uji ketiga.

Pada gambar 4.11 menunjukan laju kecepatan algoritma genetik dalam mengeksekusi data, bisa telihat bahwa semakin besar ukuran populasi kemungkinan lebih besar untuk menemukan solusi yang terbaik, namun membutuhkan waktu pengeksekusian yang lebih lama.

Pada cobaan diatas bisa dilhat bahwa dengan merubah para meter masukan dan para meter algoritma genetika seperti ukuran populasi, probabilitas crossover dan probabilitas mutasi menghasilkan data yang bervariasi baik dalam waktu pengeksekusian maupun nilai fitnessnya. Jadi ukuran parameter lebih besar belum tentu menghasilkan jadwal terbaik, hasil penjadwalan bisa dilihat pada tabel lampiran.

Ukuran populasi dan generasi mempengaruhi untuk kerja yang baik dan keefektifan algoritma genetika. Algoritma genetika dengan populasi dan generasi yang kecil biasanya unjuk kerjanya buruk karena populasi tidak menyediakan cukup materi untuk mencakup ruang persoalan. Populasi yang lebih besar dibutuhkan untuk mempresentasikan keseluruhan ruang persoalan. Namun, semakin besar ukuran populasi dan generasi maka kecepatan eksekusi program algoritma genetika akan semakin lambat.