

KODE PROGRAM DAN MANUAL PERANGKAT LUNAK

**Perangkat Lunak Algoritma Genetika dan *Google Maps API* Untuk
Penjadwalan Rute Perjalanan Divisi Pemasaran STMIK El Rahma**

PENCIPTA

Herdiesel Santoso S.Kom.,S.T.,M.Cs.

Rachmad Sanuri, S.E., M.Eng

PEMEGANG HAK CIPTA

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN ILMU

KOMPUTER EL RAHMA

DAFTAR ISI

DAFTAR ISI.....	ii
DAFTAR GAMBAR.....	iii
1 BAB I KODE PROGRAM.....	4
2 BAB II MANUAL PENGGUNAAN PROGRAM.....	9

DAFTAR GAMBAR

Gambar 1.1 Kode program proses pengambilan jarak dan waktu tempuh berdasarkan lintang dan bujur menggunakan google maps api.....	5
Gambar 1.2 Kode program proses inisiasi populasi	5
Gambar 1.3 Kode program proses perhitungan nilai <i>fitness</i>	6
Gambar 1.4 Kode program proses <i>crossover</i> dengan <i>cycle crossover</i>	7
Gambar 1.5 Kode program proses <i>mutasi</i> dengan <i>swap mutation</i>	8
Gambar 2.1 Implementasi Halaman Depan Program Komputer.....	9
Gambar 2.2 Implementasi Menu Input Lokasi Awal	10
Gambar 2.3 Implementasi Menu Input Lokasi Tujuan.....	10
Gambar 2.4 Implementasi Menu Jarak dan Waktu Tempuh.....	11
Gambar 2.5 Implementasi Menu Hasil Perhitungan.....	12
Gambar 2.6 Implementasi Menu Pentunjuk Arah Semua Lintasan	12
Gambar 2.7 Implementasi Menu Petunjuk Arah Antara Lokasi	13

BAB I KODE PROGRAM

Proses pertama adalah melakukan inisiasi destinasi yang menjadi lokasi awal dan dan menjadi lokasi tujuan perjalanan, proses ini menggunakan bantuan *google maps api*.

```
[1]   $("#inisiasi").click(function()
[2]   {
[3]       var location = $('#location').val();
[4]       var start_time = $('#start-time').val();
[5]
[6]       if(!location && !start_time){
[7]           var chromosom = new Array();
[8]           var time_windows = new Array();
[9]           var event_data = new Array();
[10]          var latlng = $('#latitude').val() + ', ' + $('#longitude').val();
[11]
[12]
[13]          //Membuat JSON untuk data origin
[14]          var data = '{"location":"' + $('#location').val() + '","address":"' +
[15]              $('#address').val() + '","latlng":"' + latlng +
[16]              '","time_windows":["2:30","4:20"]}';
[17]          //Push Data Sebagai Array Pertama
[18]          event_data.push(JSON.parse(data));
[19]          //Push Data Koordinat
[20]          chromosom.push(latlng);
[21]
[22]          for(i=1; i<=number_destination; i++){
[23]              data = JSON.parse($('#body-dest #dest_'+i).val());
[24]
[25]              event_data.push(data);
[26]              chromosom.push(data.latlng);
[27]              time_windows.push(data.time_windows);
[28]          }
[29]
[30]          /* Begin Calculate Distance and duration with Lat and Lng*/
[31]          var service = new google.maps.DistanceMatrixService;
[32]          service.getDistanceMatrix({
[33]              origins: chromosom,
[34]              destinations: chromosom,
[35]              travelMode: google.maps.TravelMode.DRIVING,
[36]              unitSystem: google.maps.UnitSystem.METRIC,
[37]              avoidHighways: false,
[38]              avoidTolls: false
[39]          }, function(response, status) {
[40]              if (status !== google.maps.DistanceMatrixStatus.OK) {
[41]                  alert('Error was: ' + status);
[42]              } else {
[43]
[44]                  var originList = response.originAddresses;
[45]                  var destinationList = response.destinationAddresses;
[46]                  var distanceList = new Array();
[47]                  var durationList = new Array();
[48]
[49]                  /* Begin Capture Distance and duration */
[50]                  for (var i = 0; i < originList.length; i++) {
[51]                      var results = response.rows[i].elements;
[52]                      distanceList[i] = new Array();
```

```

[53]
[54]         for (var j = 0; j < results.length; j++) {
[55]             if(originList[i] != destinationList[j])
[56]             {
[57]                 distanceList[i][j] = results[j].distance.value;
[58]                 durationList[i][j] = results[j].duration.value;
[59]             }
[60]             else{
[61]                 distanceList[i][j] = null;
[62]                 durationList[i][j] = null;
[63]             }
[64]         }
[65]     }
[66]     $('#chromosom').val(JSON.stringify(event_data));
[67]     $('#time_windows').val(JSON.stringify(time_windows));
[68]     $('#distanceList').val(JSON.stringify(distanceList));
[69]     $('#durationList').val(JSON.stringify(durationList));
[70]     $('#form-event').submit();
[71] }
[72] });
[73] }
[74] else{
[75]     alert('Lokasi Asal Tidak Boleh Kosong dan Harus Memilih Lebih Dari 2
[76]         Destinasi Sekolah!!!');
[77] }
[78]
[79] });

```

Gambar 1.1 Kode program proses pengambilan jarak dan waktu tempuh berdasarkan lintang dan bujur menggunakan google maps api

Kode program Gambar 1.1 digunakan untuk mengambil jarak dan waktu tempuh berdasarkan koordinat lintang dan bujur yang dimasukan kedalam aplikasi.

```

[1]     private function pickRandom($dna)
[2]     {
[3]         $choices = $dna;
[4]
[5]         shuffle($choices);
[6]         $choices = implode('-', $choices);
[7]         return '1-' . $choices . '-1';
[8]     }
[9]
[10]    public function initialPopulation($dna, $numberPopulation){
[11]        $initialPopulation = array();
[12]        for ($i = 0; $i < $numberPopulation; $i++) {
[13]            $initialPopulation[$i] = $this->pickRandom($dna);
[14]            while (!in_array($initialPopulation[$i], $initialPopulation)) {
[15]                $initialPopulation[$i] = $this->pickRandom($dna);
[16]            }
[17]        }
[18]        return $initialPopulation;
[19]    }

```

Gambar 1.2 Kode program proses inisiasi populasi

Kode program Gambar 1.2 digunakan untuk melakukan pengkodean dan inisiasi populasi populasi. Satu skema pengkodean yang biasa digunakan adalah setiap individu berisi satu kromosom yang berisi gen-gen yang merepresentasikan nomor urut lokasi event. Skema pengkodean ini dikenal sebagai *permutation encoding* (pengkodean permutasi). Inisiasi adalah membangkitkan sejumlah individu secara acak atau melalui prosedur tertentu. Dengan menggunakan fungsi random akan dipilih suatu individu dengan memilih secara acak antara bilangan 1 sampai dengan jumlah gen hingga terbentuk suatu kromosom. Kromosom yang sudah dibangkitkan akan menjadi sebuah populasi.

```

[1] public function rate($dna)
[2]     {
[3]         $dna = explode('-', $dna);
[4]
[5]         $start_time = $this->start_time;
[6]         $penalti = 0;
[7]         $cost = 0;
[8]
[9]         for ($i = 0; $i < count($dna) - 1; $i++) {
[10]            //Google memberikan waktu tempuh dalam detik, di konversi dalam menit
[11]            dan dibulatkan
[12]            $duration = round($this->duration[$dna[$i] - 1][$dna[$i + 1] - 1] /
[13]            60);
[14]            $distance = round($this->distance[$dna[$i] - 1][$dna[$i + 1] - 1] /
[15]            1000);
[16]            $arrival = date('H:i', strtotime($start_time." +" . $duration.'
[17]            minutes'));
[18]            //Pulang ke tempat asal tidak ada waktu buka atau waktu tutup, tetapi
[19]            cuma travel time (waktu perjalanan saja) yg dicatat
[20]            if($i < (count($dna) - 2))
[21]            {
[22]                $open = $this->time_windows[$dna[$i+1]-2][0];
[23]                $close = $this->time_windows[$dna[$i+1]-2][1];
[24]
[25]                //Datang Keawalan
[26]                if(strtotime($arrival) < strtotime($open)) {
[27]                    $penalti += strtotime($open) - strtotime($arrival);
[28]                    $arrival = $open;
[29]                }
[30]
[31]                //Datang Terlambat
[32]                if(strtotime($arrival) > strtotime($close))
[33]                    $penalti += strtotime($arrival) - strtotime($close);
[34]                else
[35]                    $start_time = date('H:i', strtotime($arrival." + 30
[36]                    minutes"));
[37]
[38]                //Datang Tidak Terlambat Tapi Waktu Pelayanan Sudah Tutup
[39]                if(strtotime($start_time) > strtotime($close))
[40]                    $penalti += strtotime($start_time) - strtotime($close);
[41]
[42]            }
[43]            $cost += $duration * $distance;
[44]        }
[45]        //Penalti dalam menit
[46]        $this->penalti = $penalti / 60;
[47]        $this->cost = $cost;
[48]    }

```

Gambar 1.3 Kode program proses perhitungan nilai *fitness*

Setelah terbentuk satu populasi, tahap selanjutnya adalah menghitung nilai *fitness* setiap individu. Oleh karena masalah perjalanan atau kunjungan adalah permasalahan TSP-TW dan tujuannya adalah meminimalkan total biaya, dimana nilai pinalti didapatkan dari waktu berangkat dikurangi dengan jadwal tutup apabila waktu datang tidak dalam waktu antara jadwal buka dan jadwal tutup, namun apabila waktu datang masih dalam waktu antara jadwal buka dan jadwal tutup maka dianggap 0 untuk nilai pinaltinya. Sedangkan, nilai *fitness* didapatkan dari rumus $\frac{1}{(1+pinalti)/60}$. Total biaya adalah perkalian antara jarak tempuh dan waktu tempuh. Gambar 1.3 adalah kode program untuk perhitungan nilai *fitness*.

```
[1]     public function crossover($parent1, $parent2)
[2]     {
[3]         $child1  = array();
[4]         $child2  = array();
[5]         $return  = array();
[6]         $parent1 = explode('-', $parent1);
[7]         $parent2 = explode('-', $parent2);
[8]
[9]         $stop    = $parent1[1];
[10]        $child1[1] = $parent1[1];
[11]        $child2[1] = $parent2[1];
[12]        $x        = 1;
[13]        while ($parent2[$x] != $stop) {
[14]            $x      = array_search($parent2[$x], $parent1);
[15]            $child1[$x] = $parent1[$x];
[16]            $child2[$x] = $parent2[$x];
[17]        }
[18]
[19]        for ($i = 0; $i < count($parent1); $i++) {
[20]            if (empty($child1[$i])) {
[21]                $child1[$i] = $parent2[$i];
[22]            }
[23]
[24]            if (empty($child2[$i])) {
[25]                $child2[$i] = $parent1[$i];
[26]            }
[27]        }
[28]
[29]        ksort($child1, SORT_NUMERIC);
[30]        ksort($child2, SORT_NUMERIC);
[31]        $return[0] = $child1;
[32]        $return[1] = $child2;
[33]
[34]        return $return;
[35]    }
```

Gambar 1.4 Kode program proses *crossover* dengan *cycle crossover*

Metode *crossover* yang digunakan adalah metode *cycle crossover*. Parameter yang dibutuhkan untuk melakukan *crossover* adalah gen dari orang tua 1 dan orang tua 2. Implementasi kode program proses *crossover* dapat dilihat pada Gambar 1.4

```

[1]     public function mutation($dna)
[2]     {
[3]         $dna    = explode('-', $dna);
[4]         $max    = count($dna) - 1;
[5]         $rand1  = 0;
[6]         $rand2  = 0;
[7]         while ($rand1 == $rand2) {
[8]             $rand1 = rand(2, $max);
[9]             $rand2 = rand(2, $max);
[10]        }
[11]        $temp          = $dna[$rand1 - 1];
[12]        $dna[$rand1 - 1] = $dna[$rand2 - 1];
[13]        $dna[$rand2 - 1] = $temp;
[14]
[15]        return $dna;
[16]    }

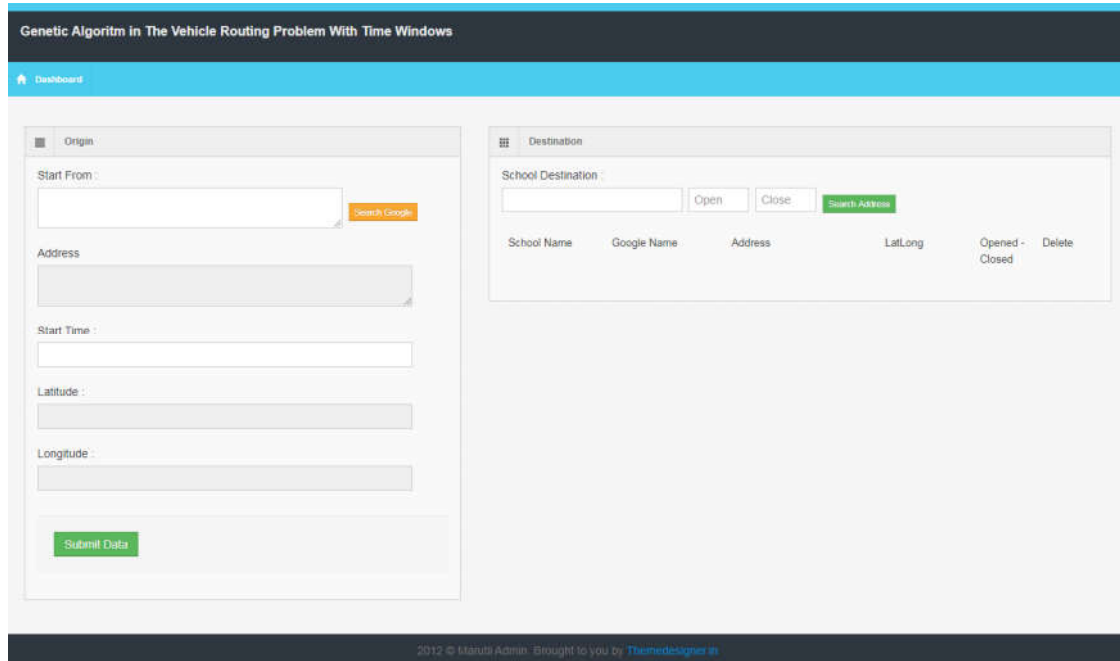
```

Gambar 1.5 Kode program proses *mutasi* dengan *swap mutation*

Teknik mutasi yang digunakan adalah menggunakan *Swap Mutation*. Proses mutasi dilakukan dengan cara membangkitkan bilangan random untuk setiap individu dalam *mating pool*. Jika $\text{random} < P_m$, maka individu tersebut mengalami mutasi. Jika $\text{random} > P_m$, individu tidak mengalami mutasi. Gambar 1.5 adalah implementasi kode program *swap mutation*. Proses *crossover* dan *mutasi* untuk kasus TSP adalah gen pertama dan terakhir tidak mengalami *crossover* dan *mutasi* karena kedua gen tersebut merupakan lokasi awal dan lokasi akhir perjalanan, sehingga jika ada sejumlah n gen dalam satu populasi, gen yang mengalami *crossover* atau mutasi adalah gen kedua hingga gen $n-1$. Sementara dari proses *crossover* dan mutasi dihasilkan DNA anak baru.

BAB II MANUAL PENGGUNAAN PROGRAM

1. Buka alamat <http://tsptw.stmikelahma.ac.id/> menggunakan browser mozilla firefox, google chrome atau opera. Maka akan tampil aplikasi Penjadwalan Rute Perjalanan Divisi Pemasaran STMIK El Rahma.



Gambar 2.1 Implementasi Halaman Depan Program Komputer

2. Halaman depan aplikasi terdiri dari menu *input origin* yaitu menu untuk memasukan lokasi awal perjalanan dan menu *input destination* yaitu menu untuk memasukan lokasi yang menjadi tujuan perjalanan dari divisi pemasaran STMIK El Rahma. Pada menu *input origin* pengguna memasukan objek yang menjadi lokasi awal perjalanan dan waktu mulai perjalanan (*start time*). Jika objek tersebut dikenali oleh aplikasi, maka akan muncul nama lokasi, alamat beserta koordinatnya. Selanjutnya pengguna memasukan waktu memulai perjalanan.

Gambar 2.2 Implementasi Menu Input Lokasi Awal

3. Pada menu *input destination* pengguna perangkat lunak memasukan objek yang menjadi tujuan perjalanan promosi serta waktu ketersediaan kunjungan. Jika objek tersebut dikenali oleh aplikasi, maka akan muncul nama lokasi, alamat beserta koordinat lintang dan bujur. Gambar 1.3 merupakan implementasi menu *input* lokasi tujuan.

School Name	Google Name	Address	LatLong	Opened - Closed	Delete
smk muhammadiyah 3 yogyakarta	62 Yogyakarta	Jl. Pramuka No.62, Giwangan, Kec. Umbulharjo, Kota Yogyakarta, Daerah Istimewa Yogyakarta 55163, Indonesia	-7.822085200000001, 110.3891582	08.00 - 14.00	Delete
smk pin 1 yogyakarta	14 Yogyakarta	Jl. Kemuning No.14, Baciro, Kec. Gondokusuman, Kota Yogyakarta, Daerah Istimewa Yogyakarta 55225, Indonesia	-7.794251200000001, 110.38304540000001	09.00 - 13.00	Delete
ma muhammadiyah 1 kota yogyakarta	87 Yogyakarta	Jalan Wahid Hasyim No.87, Notoprajan, Ngampilan, Kota Yogyakarta, Daerah Istimewa Yogyakarta 55262, Indonesia	-7.806697999999999, 110.35618169999998	07.00 - 14.30	Delete
smk negeri 2 yogyakarta	47 Yogyakarta	Jl. AM. Sangaji No.47, Cokrodiningratan, Kec. Jetis, Kota Yogyakarta, Daerah Istimewa Yogyakarta 55233, Indonesia	-7.7771363, 110.36735950000002	08.00 - 14.00	Delete

Gambar 2.3 Implementasi Menu Input Lokasi Tujuan

- Selanjutnya pengguna tinggal menekan tombol submit data. Selanjutnya palikasi dengan bantuan *Google maps API* menghitung jarak dan waktu tempuh dari lokasi awal ke masing-masing lokasi tujuan.

Genetic Algorithm in The Vehicle Routing Problem With Time Windows

Dashboard

Genetic Algorithm in The Vehicle Routing Problem With Time Windows

Distance From/To	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
C1	0	2.05	2.012	3.472	7.386	6.707	4.238	4.669	5.95	2.972
C2	2.052	0	4.888	3.177	7.429	7.207	6.242	6.672	9.616	5.431
C3	1.763	3.57	0	4.094	7.416	6.962	4.192	4.549	5.831	2.002
C4	3.597	3.199	4.944	0	4.252	4.03	4.452	5.514	6.706	4.763
C5	6.179	7.047	6.54	4.225	0	1.482	3.791	5.583	7.634	7.027
C6	6.244	6.281	6.609	3.082	2.074	0	4.334	6.126	8.176	7.096
C7	5.097	6.261	3.646	4.401	4.306	3.852	0	1.792	3.842	4.134
C8	4.812	6.684	3.043	5.138	6.116	5.661	1.809	0	2.051	2.972
C9	6.317	8.169	4.548	6.644	6.63	8.305	3.798	1.988	0	4.477
C10	2.639	4.966	1.325	4.731	7.536	7.081	4.192	2.972	4.253	0

Duration From/To	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
C1	0	7.3	7.017	11.067	22.217	20	12.55	12.817	16.233	8.517
C2	7.267	0	12.217	10.217	23.467	22.317	18.867	19.133	17.333	11.867
C3	5.583	11.733	0	13.167	22.217	19.883	12.267	12.65	16.033	5.783
C4	12.4	10.35	15.883	0	13.25	12.1	14.367	17.617	23.333	14.8
C5	20.383	21.883	19.017	13.85	0	5.5	11.217	16.183	22.633	19.85
C6	20.317	20.433	19.717	10.083	7.217	0	12.183	17.15	23.6	20.55
C7	15.35	19	9.467	14.2	12.75	10.433	0	4.967	11.4	10.283
C8	13.983	19.4	8.383	16.9	18.05	15.717	5.283	0	6.45	7.783
C9	19.483	24.9	13.883	22.4	22.05	21.4	12.517	7.233	0	13.283
C10	8.083	14.433	3.417	14.35	21.683	19.367	10.817	7.8	11.183	0

DisxDur From/To	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
C1	0	14.965	14.118	38.423	164.092	134.14	53.187	59.841	96.588	25.312
C2	14.911	0	59.715	32.458	174.334	160.836	117.766	127.658	166.677	64.448
C3	9.843	41.888	0	53.904	164.759	138.428	51.422	57.545	93.49	11.578
C4	44.603	33.11	78.527	0	56.339	48.763	63.96	97.138	156.473	70.492
C5	125.949	154.212	124.369	58.516	0	8.151	42.522	90.352	172.783	139.486
C6	126.857	128.342	130.307	31.077	14.967	0	52.803	105.061	192.954	145.823
C7	78.239	118.969	34.515	62.494	54.902	40.189	0	8.9	43.799	42.511
C8	67.288	129.67	25.51	86.832	110.394	88.972	9.558	0	13.225	23.132
C9	123.076	203.906	63.141	148.826	190.292	177.727	47.538	14.38	0	59.469
C10	21.332	71.676	4.527	67.89	163.406	137.135	45.343	23.182	47.563	0

Calculate Shortest Route

- C1 : STMIK El Rahma - Jl. Singamangaraja No. 76, Karangajen, Brontokusuman, Kec. Mergansan, Kota Yogyakarta, Daerah Istimewa Yogyakarta 55153, Indonesia
- C2 : madrasah aliyah ali maksum - Jl. Kh. Ali Maksum, Krapyak Kulon, Panggunharjo, Sewon, Bantul, Daerah Istimewa Yogyakarta 55188, Indonesia
- C3 : smk muhammadiyah 1 yogyakarta - Jl. Nitiikan Baru No.48, Sorosutan, Kec. Umbulharjo, Kota Yogyakarta, Daerah Istimewa Yogyakarta 55162, Indonesia
- C4 : ma muhammadiyah 1 kota yogyakarta - Jalan Wahid Hasyim No 87, Notoprajan, Ngampilan, Kota Yogyakarta, Daerah Istimewa Yogyakarta 55262, Indonesia
- C5 : smk negeri 2 yogyakarta - Jl. AM Sangaji No.47, Cokrodiningratan, Kec. Jetis, Kota Yogyakarta, Daerah Istimewa Yogyakarta 55233, Indonesia
- C6 : smk negeri 7 yogyakarta - Jalan Gowongan Kidul Blok J13 No.416, Gowongan, Kec. Jetis, Kota Yogyakarta, Daerah Istimewa Yogyakarta 55232, Indonesia
- C7 : smk piri 1 yogyakarta - Jl. Kemuning No.14, Baciro, Kec. Gondokusuman, Kota Yogyakarta, Daerah Istimewa Yogyakarta 55225, Indonesia
- C8 : smk negeri 5 yogyakarta - Jl. Kenari No.71, Muja Muju, Kec. Umbulharjo, Kota Yogyakarta, Daerah Istimewa Yogyakarta 55165, Indonesia
- C9 : ma negeri 4 bantul - Jl. Majapahit, Pringgolayan, Pranti, Kec. Banguntapan, Bantul, Daerah Istimewa Yogyakarta 55198, Indonesia
- C10 : smk muhammadiyah 3 yogyakarta - Jl. Pramuka No.62, Giwangan, Kec. Umbulharjo, Kota Yogyakarta, Daerah Istimewa Yogyakarta 55163, Indonesia

2012 © Maria Admi. Brought to you by [Themadesigner.in](#)

Gambar 2.4 Implementasi Menu Jarak dan Waktu Tempuh

11

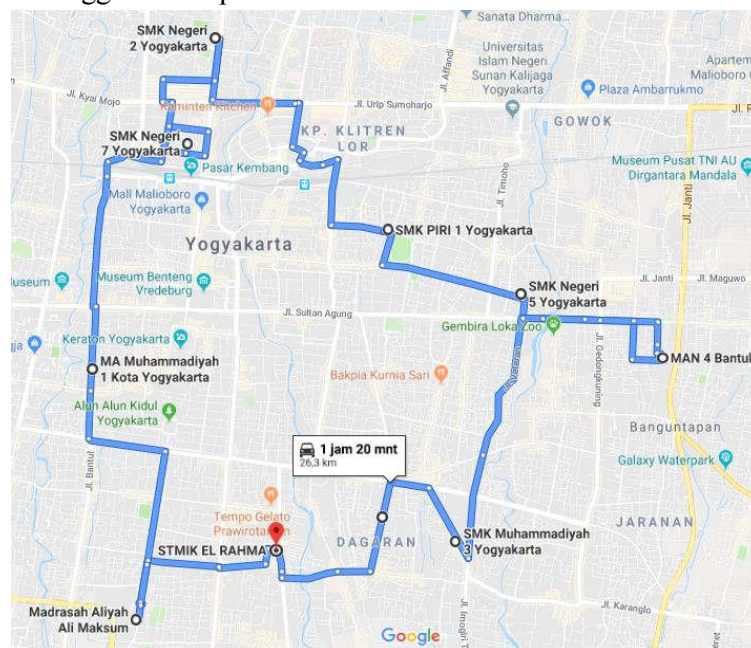
5. Selanjutnya aplikasi melakukan perhitungan menggunakan Algoritma Genetika dengan parameter waktu kedatangan, waktu ketersediaan, jarak, waktu tempuh, jumlah populasi, jumlah generasi, nilai probabilitas *crossover* (Pc) dan probabilitas mutasi (Pm) yang sudah ditentukan. Hasil perhitungan adalah solusi rute terbaik, nilai *fitness*, rekomendasi rute perjalanan lengkap dengan estimasi jarak dan waktu tempuhnya. Implementasi menu hasil perhitungan dengan Algoritma Genetika ditampilkan pada Gambar 1.5.

Solusi Terbaik Yang Ditemukan Adalah 1-8-4-10-5-3-6-9-2-7-1 dengan fitness 0.0040983606557377 dengan 165 generations.

Perjalan Ke -	Rekomendasi Tujuan	Estimasi Jarak	Estimasi Waktu	Rute
1	STMIK El Rahma ke madrasah aliyah ali maksum	2.05 Km	7.3 menit	Rute
2	madrasah aliyah ali maksum ke ma muhammadiyah 1 kota yogyakarta	3.183 Km	10.18 menit	Rute
3	ma muhammadiyah 1 kota yogyakarta ke smk negeri 7 yogyakarta	4.03 Km	12.1 menit	Rute
4	smk negeri 7 yogyakarta ke smk negeri 2 yogyakarta	2.074 Km	7.22 menit	Rute
5	smk negeri 2 yogyakarta ke smk piri 1 yogyakarta	3.791 Km	11.22 menit	Rute
6	smk piri 1 yogyakarta ke smk negeri 5 yogyakarta	1.792 Km	4.97 menit	Rute
7	smk negeri 5 yogyakarta ke man 4 bantul	2.051 Km	6.45 menit	Rute
8	man 4 bantul ke smk muhammadiyah 3 yogyakarta	4.477 Km	13.28 menit	Rute
9	smk muhammadiyah 3 yogyakarta ke smk muhammadiyah 1 yogyakarta	1.322 Km	3.42 menit	Rute
10	smk muhammadiyah 1 yogyakarta ke STMIK El Rahma	1.763 Km	5.58 menit	Rute
Total		26.533 Km	81.72 menit	

Gambar 2.5 Implementasi Menu Hasil Perhitungan

6. Pengguna juga dapat melihat petunjuk arah menuju lokasi tujuan baik rute seluruh lokasi yang akan menjadi tujuan perjalanan seperti Gambar 1.6 atau secara lokasi ke lokasi seperti Gambar 1.6. Rute yang ditampilkan adalah rute tercepat saat pengguna menggunakan aplikasi berdasarkan kondisi *real time* lalu lintas yang direkomendasikan. Rute tersebut dapat berubah sesuai dengan kondisi lalulintas saat pengguna menggunakan aplikasi.



Gambar 2.6 Implementasi Menu Pentunjuk Arah Semua Lintasan

Genetic Algorithm in The Vehicle Routing Problem With Time Windows

Dashboard

Direction

Jl. Sisingamangaraja Jl. Karangkajen No.76, Brontokusuman, Kec. Mergangsan, Kota Yogyakarta, Daerah Istimewa Yogyakarta 55153, Indonesia

4,2 km. Sekitar 12 mins

1. Ambil arah timur di Jl. Karangkajen menuju Jl. Sisingamangaraja 10 m
2. Belok kiri ke Jl. Sisingamangaraja
Lewati Alfamart Sisingamangaraja (di kiri 400 m lagi) 0,8 km
3. Belok kanan ke Jl. Kolonel Sugiyono
Lewati Commonwealth Life (di kiri) 0,3 km
4. Belok kiri ke Jl. Taman Siswa
Lewati Fakultas Hukum - Universitas Islam Indonesia (di kiri) 1,6 km
5. Belok kiri ke Jl. Sultan Agung 68 m
6. Belok kanan ke Jl. Letkol Subadri/Jl. Suryopranoto 0,4 km
7. Belok sedikit ke kanan menuju Jl. Ki Mangunsarkoro 0,3 km
8. Terus lurus ke Jl. Sukonandi I 0,2 km
9. Belok kiri ke Jl. Sukonandi 0,1 km
10. Belok kanan ke Jl. Bung Tarjo/Jl. Gayam 60 m
11. Belok kiri ke Jl. Andong/Jl. M. Djamil Dalhar 0,3 km
12. Belok kiri ke Jl. Tunjung
Tujuan ada di sebelah kanan. 38 m

Jl. Kemuning No.14, Baciro, Kec. Gondokusuman, Kota Yogyakarta, Daerah Istimewa Yogyakarta 55225, Indonesia

Map data ©2020

Map

2012 © Manula Admin. Brought to you by [Themedesigner.in](#)

Gambar 2.7 Implementasi Menu Petunjuk Arah Antara Lokasi