



## PENERAPAN ALGORITMA *RECURSIVE BEST FIRST SEARCH* (RBFS) DALAM PENYELESAIAN *TRAVELING SALESMAN PROBLEM* (TSP) DI PT. BINTANG SERVICE MANAGEMENT

Faozi<sup>✉</sup>, Amin Suyitno, Riza Arifudin

Jurusan Matematika, FMIPA, Universitas Negeri Semarang, Indonesia  
Gedung D7 Lt.1, Kampus Sekaran Gunungpati, Semarang 50299

### Info Artikel

*Sejarah Artikel:*

Diterima November 2016

Disetujui Desember 2016

Dipublikasikan Nopember 2017

*Keywords:*

*AngularJS, LeafletJS, PHP, RBFS, TSP.*

### Abstrak

*Rekursif Best First Search* (RBFS) adalah algoritma *linear space* yang memperluas titik pencarian dalam terbaik pertama bahkan dengan fungsi biaya *nonmonotonic*, dan menghasilkan lebih sedikit titik dari *Best first Search* dengan fungsi biaya monoton. Algoritma *Rekursif Best First Search* dapat digunakan untuk menyelesaikan *Traveling Salesman Problem*. Tujuan utama penelitian ini adalah memahami cara pembuatan program algoritma *Recursive Best First Search* dalam penyelesaian *Traveling Salesman Problem* di PT. Bintang Service Management menggunakan bahasa pemrograman *Hypertext Preprocessor*. Berdasarkan program tersebut diperoleh hasil peta rute terpendek yang merupakan siklus Hamilton dengan bobot terkecil yaitu 61,95 km, sementara siklus Hamilton lain dengan bobot terbesar yang mungkin dilewati yaitu 81,75 km, sehingga lebih efektif dalam jarak sejauh 19,8 km dan lebih efisien dalam waktu perjalanan. Hasil perhitungan program tersebut dapat digunakan untuk memberikan alternatif solusi pihak pengambil keputusan PT. Bintang Service Management untuk memperoleh rute terpendek yang akan dilalui sehingga lebih efektif dan efisien.

### Abstract

*Recursive Best First Search* (RBFS) is a linear space algorithm that expands the search point in the first even with the best cost function nonmonotonic, and produce fewer points than *Best first Search* with monotone cost function. *Recursive Best-First Search* algorithm can be used to resolve *Traveling Salesman Problem*. The main objective of this research is to understand how making the program *Recursive Best First Search* in solving *Traveling Salesman Problem* in PT. Bintang Service Management using a programming language *Hypertext Preprocessor*. Base on the program obtained results which is the shortest route map *Hamiltonian cycle* with the smallest weight 61.95 km, while the other *Hamiltonian cycle* with the greatest weight that may be passed that is 81.75 km, making it more effective at a distance of 19.8 km and more efficient the travel time. The calculation results of the program can be used to provide alternative solutions to the PT. Bintang Service Management decision maker's to obtain the shortest route that will be passed making it more effective and efficient.

### How to Cite

Faozi, Suyitno A. & Arifudin R. (2017). Penerapan Algoritma *Recursive Best First Search* (RBFS) dalam Penyelesaian *Traveling Salesman Problem* (TSP) di PT. Bintang Service Management. *Unnes Journal of Mathematics* 6(2): 211-220.

<sup>✉</sup>Alamat korespondensi:

E-mail: [faozimipa@gmail.com](mailto:faozimipa@gmail.com)

## PENDAHULUAN

Banyaknya perusahaan dalam industri, serta kondisi perekonomian saat ini telah menciptakan suatu persaingan yang ketat antar perusahaan. Persaingan dalam industri membuat setiap perusahaan semakin meningkatkan kinerja agar tujuan perusahaan dapat tetap tercapai. Menurut Hermuningsih (2013), tujuan utama perusahaan yang telah *go public* adalah meningkatkan kemakmuran pemilik atau para pemegang saham melalui peningkatan nilai perusahaan. Peningkatan nilai perusahaan dapat dilakukan dengan meningkatkan kinerja perusahaan, yaitu dengan mengalihdayakan atau menyerahkan proses-proses yang bukan merupakan kompetensi utama (*core competence*) perusahaan tersebut ke pihak lain agar perusahaan dapat memfokuskan pada kompetensi utama, aktivitas ini dikenal dengan istilah *outsourcing* (Iqbal & Dad, 2013).

*Outsourcing* dilakukan untuk memberikan respon atas perkembangan ekonomi secara global dan perkembangan teknologi yang begitu cepat serta berkembangnya persaingan yang bersifat global dan berlangsung sangat ketat. Menurut Çiçek & Özer (2011), *outsourcing* telah terbukti dapat meningkatkan daya saing usaha secara signifikan, karena dengan melakukan *outsourcing*, perusahaan dapat lebih fokus dalam menjalankan aktivitas kompetensi utamanya, sehingga dapat mendukung kecepatan perusahaan dalam merespon tuntutan pasar.

PT. Bintang Service Management sebagai salah satu perusahaan *outsourcing* telah menangani lebih dari tiga puluh perusahaan yang tersebar di wilayah Bandung, Indramayu, Cirebon dan sebagian besar area Jawa Tengah khususnya Semarang, Purwodadi, Pati, Pemasang, Slawi, Tegal, dan Brebes. Banyaknya perusahaan yang ditangani PT. Bintang Service Management mengakibatkan semakin banyak pilihan rute perjalanan yang harus dilalui pihak *outsourcing* untuk memenuhi beberapa kebutuhan perusahaan klien khususnya kegiatan pengadaan barang, dan pengecekan barang yang rutin dilakukan setiap akhir bulan, sehingga waktu yang ditempuh juga semakin lama dan cenderung tidak efektif.

Permasalahan digambarkan di mana pihak perusahaan berangkat dari kantor PT. Bintang Service Management yaitu Jl. Kalipepe Baru No.9 Pudukpayung Semarang, harus mengunjungi setiap perusahaan klien di sekitar Semarang tepat satu kali dan kembali lagi ke kantor PT. Bintang Service Management. Permasalahan yang dihadapi tersebut dalam matematika disebut *Traveling Salesman Problem* (TSP). *Traveling Salesman Problem* merupakan salah satu masalah optimalisasi yaitu suatu permasalahan untuk menemukan siklus

Hamilton yang memiliki total bobot sisi minimum (Hlaing & Khine, 2011).

Menurut Kusriani & Istiyanto (2007), ada beberapa metode yang bisa menyelesaikan TSP, antara lain: *Linear Programming* (LP), Algoritma Genetik, *Nearest Neighbourhood Heuristic* (NNH) dan *Cheapest Insertion Heuristic* (CIH). Menurut Sutojo, *et al.* (2011), teknik pencarian heuristik merupakan suatu strategi untuk melakukan proses pencarian secara selektif dan dapat memandu proses pencarian yang memiliki kemungkinan sukses paling besar.

Algoritma Heuristik merupakan salah satu algoritma alternatif yang dapat digunakan untuk masalah tersebut, sebab prosesnya cepat dan memberikan hasil yang diinginkan. Algoritma yang umum digunakan dalam pencarian heuristik adalah *Generate and Test*, *Hill Climbing*, *A\**, *Best First Search*, dan *Recursive Best First Search*.

Menurut Korf (1993), *Rekursif Best-First Search* (RBFS) adalah algoritma linear space yang memperluas titik pencarian dalam terbaik-pertama bahkan dengan fungsi biaya nonmonotonik, dan menghasilkan titik yang lebih sedikit dari *Best first Search* dengan fungsi biaya monoton, sehingga menghasilkan *routing* dari graf yang memiliki nilai optimal.

Beberapa peneliti telah melakukan penelitian tentang *Traveling Salesman Problem* seperti penelitian TSP menggunakan algoritma semut yang dilakukan oleh Katiyar, *et al.* (2014). Penelitian TSP menggunakan algoritma genetika yang berjudul *Solving Travelling Salesman Problem Using Genetic Algorithm* oleh (Gupta & Panwar, 2013). Selain itu, penelitian serupa juga dilakukan di Indonesia seperti penelitian tentang pencarian rute terpendek pada PT. Jalur Nugraha Ekakurir (JNE) Semarang oleh Anitya, *et al.* (2013), penelitian algoritma *fuzzy evolusi* yang digunakan untuk memecahkan suatu pencarian nilai dalam sebuah masalah optimasi dengan bantuan perangkat lunak Matlab oleh Anggit, *et al.* (2014).

Permasalahan dalam artikel ini adalah bagaimana penerapan algoritma *Recursive Best First Search* dalam penyelesaian *Traveling Salesman Problem* di PT. Bintang Service Management serta bagaimana pembuatan program algoritma *Recursive Best First Search* dalam penyelesaian *Traveling Salesman Problem* di PT. Bintang Service Management dengan menggunakan bahasa pemrograman PHP? Dari permasalahan tersebut, maka tujuan dari penelitian ini adalah Mengetahui dan memahami penerapan algoritma *Recursive Best First Search* dalam penyelesaian *Traveling Salesman Problem* di PT. Bintang Service Management dan Mengetahui dan memahami cara pembuatan program algoritma *Recursive Best First Search* dalam

penyelesaian *Traveling Salesman Problem* di PT. Bintang Service Management menggunakan bahasa pemrograman *Hypertext Preprocessor* (PHP).

## METODE

Pada penelitian ini data yang digunakan sebagai studi kasus adalah data sekunder yang diunduh dari website PT. Bintang Service Management yaitu data perusahaan-perusahaan klien PT. Bintang Service Management di Wilayah Semarang tahun 2016. Data tersebut berisi nama perusahaan, alamat perusahaan dan lokasi perusahaan.

Pemodelan rute transportasi darat di suatu daerah dalam bentuk persamaan matematik yang menggambarkan sistem yang dimodelkan, misalnya tujuan sistem yang akan dicapai, parameter yang mempengaruhi baik yang sudah ada maupun yang ingin dicapai, batasan sistem yang ada maupun yang dikehendaki. Banyaknya pilihan rute perjalanan yang harus dilewati dapat diteliti sebelumnya untuk mengoptimalkan *Traveling Salesman Problem* yang dapat mempersingkat waktu serta jarak tempuh yang dilalui.

Semakin banyaknya perusahaan yang memilih untuk menggunakan jasa *outsourcing* mengakibatkan klien PT. Bintang Service Management semakin banyak, sehingga hal ini membuat pilihan rute dan waktu yang harus di tempuh pihak *outsourcing* untuk mengunjungi perusahaan klien semakin tidak efektif dan efisien.

Dalam memenuhi kebutuhan tersebut, permasalahan yang terjadi adalah masih belum

adanya perhitungan untuk menentukan solusi dari *Traveling Salesman Problem* baik secara manual maupun dengan menggunakan program.

Langkah-langkah untuk menyelesaikan masalah tersebut dapat dilakukan secara bertahap sebagai berikut.

- (1) Melakukan pengambilan data pada website PT. Bintang Service Management.
- (2) Menentukan posisi *latitude* dan *longitude* dari data penelitian menggunakan Google Maps.
- (3) Menyusun graf objek perusahaan klien dan rute transportasi darat dari data perusahaan klien PT. Bintang Service Management di wilayah Semarang menggunakan data jarak, posisi *latitude* dan *longitude* dari Google Maps dan Menampilkan keadaan dalam bentuk peta menggunakan *library* LeafletJS.
- (4) Mencari rute terpendek yaitu siklus Hamilton dengan bobot minimum dengan menggunakan algoritma *Recursive Best First Search* dengan menggunakan bahasa pemrograman *Hypertext Preprocessor* (PHP).
- (5) Menampilkan rute terpendek dalam bentuk peta menggunakan *library* LeafletJS.

## HASIL DAN PEMBAHASAN

Penelitian dimulai dengan Pengambilan data. Data penelitian berupa data posisi *latitude* dan *longitude* perusahaan yang diambil dari website resmi PT. Bintang Service Management, data tersebut dapat dilihat pada Tabel 1.

**Tabel 1** Daftar posisi *latitude* dan *longitude* PT. Bintang Service Management dan perusahaan klien di Semarang tahun 2016

No.	Kode	Perusahaan	Latitude	Longitude
1	$v_1$	Bintang Service Management	-7.098802	110.405891
2	$v_2$	Semesta Bilingual School	-7.053910	110.414352
3	$v_3$	My Kopi O	-7.015103	110.419632
4	$v_4$	Hotel Grand Edge	-7.009961	110.410255
5	$v_5$	City One Hotel	-7.006252	110.441757
6	$v_6$	RS. Panti Wilasa	-6.967353	110.433998
7	$v_7$	Leko Gajah Mada	-6.979157	110.417458
8	$v_8$	Dafam Hotel	-6.979400	110.409416
9	$v_9$	3 Durian	-6.973213	110.389771
10	$v_{10}$	Kantor Imigrasi	-6.985218	110.380806
11	$v_1$	Rumah Detensi Imigrasi	-6.987391	110.368996
12	$v_{12}$	Goory Swalayan	-6.999568	110.341370
13	$v_{13}$	Payon Amarta	-7.001265	110.333778

Data Posisi *latitude* dan *longitude* digunakan untuk Penentuan lokasi dan jarak dari PT. Bintang

Service Management dan perusahaan klien, maupun lokasi dan jarak antar perusahaan

perusahaan klien. Data tersebut diambil dari data penelitian jarak berdasarkan Google Maps tahun 2016. Dalam artikel ini data memperlihatkan keterhubungan titik-titik 'perusahaan-perusahaan pada jalur darat, misalnya bagaimana ketetanggaan titik dari klien My Kopi O dengan Hotel Grand

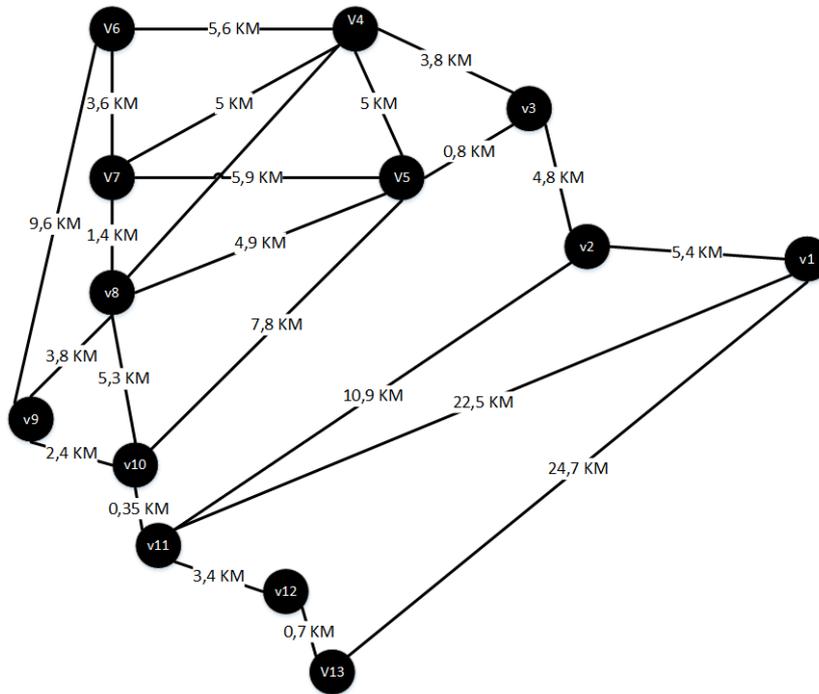
Edge, City One Hotel dengan Leko Gajah Mada dan lain sebagainya. Data tersebut disajikan dalam Tabel 2.

**Tabel 2** Daftar jarak PT. Bintang Service Management dan perusahaan klien di Semarang tahun 2016

No.	Asal	Tujuan	Jarak (km)
1	Bintang Service Management	Semesta Bilingual School	5.4
2	Bintang Service Management	Rumah Detensi Imigrasi	22.5
3	Bintang Service Management	Payon Amarththa	24.7
4	Semesta Bilingual School	My Kopi O	4.8
5	Semesta Bilingual School	Rumah Detensi Imigrasi	10.9
6	My Kopi O	Hotel Grand Edge	0.8
7	My Kopi O	City One Hotel	3.8
8	Hotel Grand Edge	City One Hotel	5
9	Hotel Grand Edge	Leko Gajah Mada	4.5
10	Hotel Grand Edge	Dafam Hotel	4.9
11	Hotel Grand Edge	3 Durian	7.2
12	Hotel Grand Edge	Kantor Imigrasi	10.9
13	City One Hotel	RS Panti Wilasa Citarum	5.6
14	City One Hotel	Leko Gajah Mada	5
15	City One Hotel	Dafam Hotel	5.9
16	Leko Gajah Mada	Dafam Hotel	7.4
17	RS Panti Wilasa Citarum	Leko Gajah Mada	3.6
18	RS Panti Wilasa Citarum	3 Durian	9.6
19	Dafam Hotel	3 Durian	3.8
20	Dafam Hotel	Kantor Imigrasi	5.3
21	3 Durian	Kantor Imigrasi	2.4
22	Kantor Imigrasi	Rumah Detensi Imigrasi	0.35
23	Rumah Detensi Imigrasi	Goori Swalayan	3.4
24	Goori Swalayan	Payon Amarththa	0.7

Penerapan algoritma *Recursive Best First Search* dalam penyelesaian *Traveling Salesman Problem* di PT. Bintang Service Management diawali dengan memodelkan lokasi perusahaan baik PT. Bintang Service Management dan klien PT. Bintang Service Management di Semarang yang akan dikunjungi sebagai himpunan titik, jarak antar perusahaan

tersebut sebagai himpunan sisi dan panjang jarak sisi tersebut kemudian disebut bobot sisi tersebut (Tanjung & Sopiah, 2010), sehingga keadaan yang terjadi dapat digambarkan dalam sebuah graf seperti Gambar 1.



**Gambar 1** Graf PT. Bintang Service Management dan perusahaan klien PT. Bintang Service Management

Graf pada Gambar 1 merupakan graf Hamilton dengan salah satu siklus Hamilton  $C_1 = (v_1, v_2, v_3, v_5, v_7, v_6, v_4, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13})$ . Selanjutnya graf tersebut akan dicari rute terpendek yaitu siklus Hamilton yang memiliki bobot minimum. Langkah-langkah mencari rute terpendek menggunakan algoritma *Recursive Best First Search* adalah sebagai berikut.

Langkah pertama yaitu membuka rute dari titik  $v_1$  dan mencari rute yang mungkin dilalui dari  $v_1$ , kemudian menentukan nilai minimum dan nilai minimum ke dua atau *threshold*. Nilai minimum akan digunakan sebagai rute yang akan di buka pada iterasi selanjutnya, sedangkan *threshold* akan digunakan sebagai pembanding dari rute yang mungkin berikutnya sehingga diperoleh nilai minimum berikutnya dan *threshold* berikutnya. Jika rute yang dibuka tidak memiliki rute yang mungkin dilalui maka diberikan bobot  $\infty$ .

Pada langkah pertama rute yang mungkin yaitu  $v_1, v_2$  dengan bobot 5,4,  $v_1, v_{11}$  dengan bobot 22,5 dan  $v_1, v_{13}$  dengan bobot 24,7, dari ketiga rute yang mungkin, diperoleh rute minimum  $v_1, v_2$  dengan bobot 5,4 dan *threshold*  $v_1, v_{11}$  dengan bobot 22,5. Rute  $v_1, v_{11}$  menjadi rute yang dibuka pada iterasi ke-dua.

Iterasi ke dua dengan rute yang dibuka  $v_1, v_2$  menghasilkan rute yang mungkin dilewati  $v_1, v_2, v_3$  dengan bobot 10,2 dan rute  $v_1, v_2, v_{11}$  dengan bobot 16,3. Nilai minimum diperoleh dengan mencari nilai minimum dari rute yang mungkin dilewati dan *threshold* pada iterasi sebelumnya, sehingga diperoleh  $v_1, v_2, v_3$  dengan bobot 10,2. *Threshold* diperoleh dengan mencari nilai minimum ke dua

dari rute yang mungkin dilewati dan semua rute sebelumnya, dalam iterasi ke dua ini diperoleh *threshold*  $v_1, v_2, v_{11}$  dengan bobot 16,3.

Iterasi ke tiga dengan rute yang dibuka  $v_1, v_2, v_3$  menghasilkan rute yang mungkin dilewati  $v_1, v_2, v_3, v_4$  dengan bobot 14 dan  $v_1, v_2, v_3, v_4$  dengan bobot 11. Rute minimum dan *threshold* diperoleh menggunakan cara yang identik dengan iterasi ke dua, sehingga diperoleh rute minimum  $v_1, v_2, v_3, v_4$  dengan bobot 11 dan *threshold*  $v_1, v_2, v_3, v_4$  dengan bobot 14.

Proses iterasi dilakukan terus menerus sampai selesai sehingga memuat semua siklus Hamilton yang mungkin dilewati, Kemudian dari semua siklus yang mungkin dilewati dicari siklus dengan bobot minimum. Siklus dengan bobot minimum yang dihasilkan dari algoritma *Recursive Best First Search* yaitu  $C = (v_1, v_2, v_3, v_5, v_4, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13}, v_1)$  dengan bobot 61,95.

Hasil perhitungan menggunakan algoritma *Recursive Best First Search* dengan data graf Hamilton tersebut diperoleh hasil siklus Hamilton yaitu Bintang Service Management – Semesta Bilingual School – My Kopi O – Hotel Grand Edge – City One Hotel – RS Panti Wilasa Citarum – Leko Gajah Mada – Dafam Hotel – 3 Durian – Kantor Imigrasi – Rumdenim – Goori Swalayan – Payon Amarta – Bintang Service Management dengan bobot minimum 61,95 km.

**Penerapan algoritma (RBFS) dalam penyelesaian (TSP) menggunakan Hypertext Preprocessor (PHP)**

Pembuatan program penerapan algoritma *Recursive Best First Search* dalam penyelesaian *Traveling Salesman Problem* di PT. Bintang Service Management dengan menggunakan bahasa pemrograman *Hypertext Preprocessor* (PHP) dimulai dengan membuat *Backend* dan *Frontend*. *Backend* (aplikasi sisi server) berisi konfigurasi dasar aplikasi seperti koneksi, algoritma RBFs, operasi *create, read, update, delete* (crud) dan sebagainya. *Frontend* (aplikasi sisi client) berisi konfigurasi sisi client *Controller* dan tampilan menu-menu yang akan di tampilkan ke user seperti menu *home*, menu keadaan, menu solusi dan menu *about*.

Menu keadaan merupakan menu yang menampilkan keadaan graf atau peta dan penginputan keadaan (titik dan sisi). Peta menggambarkan wilayah Semarang beserta Lokasi PT. Bintang Service Management dan lokasi

perusahaan klien PT. Bintang Service Management beserta jarak antar perusahaan tersebut.

Setelah membuat menu keadaan, tahap selanjutnya yaitu membuat menu operasi, menu operasi tersebut terbagi menjadi dua yaitu operasi titik dan operasi sisi. Operasi titik digunakan untuk menginput data perusahaan, edit data perusahaan dan menghapus data perusahaan. Operasi sisi digunakan untuk menambahkan jarak antar perusahaan (bobot), mengedit bobot, dan menghapus bobot.

Setelah semua data keadaan perusahaan yang akan dicari rutenya sudah diinputkan, data diproses menggunakan bahasa pemrograman PHP dimulai dengan memetakan perusahaan (titik) dan bobot dari jarak antar perusahaan (sisi) dalam bentuk array objek dengan nama variable \$peta, di mana masing-masing perusahaan memuat id, nama, bobot dan path (relasi dengan perusahaan lain). Secara blok kode dapat dilihat pada Gambar 2.

```

$peta = array();
while($row = $result->fetch_assoc()) {
    $querytwo = "SELECT c.bobot ,a.code
                FROM relasi c, markers a
                WHERE c.path=a.id
                AND c.origin='$row[id]';";
    $resulttwo = $konek->query ($querytwo);
    $relasi = array();
    while ($rowtwo = $resulttwo->fetch_assoc ()) {
        $relasi[$rowtwo["code"]] = floatval(
            $rowtwo["bobot"]
        );
    }

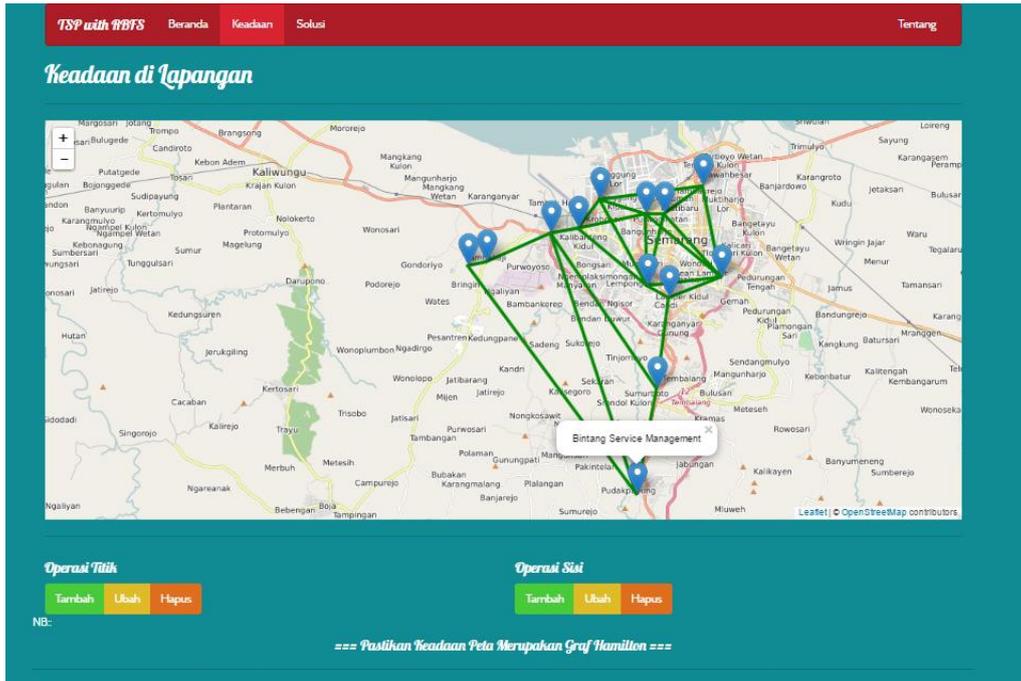
    $peta[$row["code"]] = (object) array (
        "id" => $row["code"],
        "name" => $row["name"],
        "path" => $relasi
    );
    $relasi = array();
}

```

**Gambar 2** Blok kode Pemetaan titik dan sisi

Blok kode pada Gambar 2 kemudian menghasilkan data posisi dan nama perusahaan dalam bentuk JSON untuk diparsing ke LeafletJS pada AngularJS dalam bentuk peta atau graf.

Keadaan yang terjadi dapat dilihat seperti Gambar 3, Gambar tersebut menggambarkan graf yang akan dicari rute terpendeknya.



Gambar 3 Menu keadaan setelah data diinput

Menu selanjutnya yaitu menu solusi. Menu solusi merupakan menu yang menampilkan rute terpendek beserta bobot minimum dari graf atau keadaan yang dicari. Dalam proses pencarian, dibuat sebuah klas TSP selanjutnya peta tersebut 'dibaca' dengan fungsi possible() untuk mengetahui kemungkinan rute yang akan dilalui dari suatu

perusahaan. Fungsi possible() dapat dilihat pada Gambar 4, masukannya adalah id perusahaan, dan keluarannya adalah kode daftar perusahaan yang mungkin dilalui. Pada kode tersebut dilakukan iterasi pada peta dan mencari rute yang mungkin untuk perusahaan dengan id yang cocok.

```
function possible($data){
    $ret = array();
    foreach($this->peta as $next){
        foreach($next->path as $key=>$target){
            $tmp = array();
            if($next->id == $data){
                $tmp['id'] = $key;
                $tmp['len'] = $target;
                array_push($ret,$tmp);
            }else{
                if($key == $data){
                    $tmp['id'] = $next->id;
                    $tmp['len'] = $next->path[$data];
                    array_push($ret,$tmp);
                }
            }
        }
    }
    return $ret;
}
```

Gambar 4 Blok kode fungsi possible

Fungsi utama untuk algoritma *Recursive Best First Search* (RBFS) ada pada fungsi `route()`. Fungsi `route()` membutuhkan masukan perusahaan yang sedang dibuka (`open`) dan menghasilkan rute dengan bobot (`cost`) yang minimal dan rute yang akan dijadikan `backup / threshold`. Blok kode fungsi `route()` dapat dilihat pada Gambar 5. Pertama akan dicari rute yang mungkin dilewati dengan pemanggilan fungsi `possible()`. Setelah itu daftar rute yang akan dilewati akan dihitung satu-persatu dengan iterasi. Masing-masing rute akan dihitung besar bobot dari rute awal yaitu pada variabel `$tmp[lenmin]`.

Setelah masing-masing bobot dihitung, maka semua kemungkinan rute yang ada akan diurutkan berdasarkan bobot yang terkecil dengan fungsi `usort()`. Setelah diurutkan, makan rute dengan bobot terkecil akan dibuka (variabel `$open`) dan rute dengan bobot terkecil kedua akan dijadikan `threshold` untuk perbandingan (variabel `$back`). Selanjutnya pembukaan rute baru dilakukan pada bagian `$this->route($open)`. Pada fungsi `KOTA()` dengan masukan rute yang terbuka fungsi `KOTA()` melakukan penerjemahan dari kode menjadi nama perusahaan, sehingga keluaran program akhir berupa urutan nama perusahaan yang dilewati (sikel Hamilton).

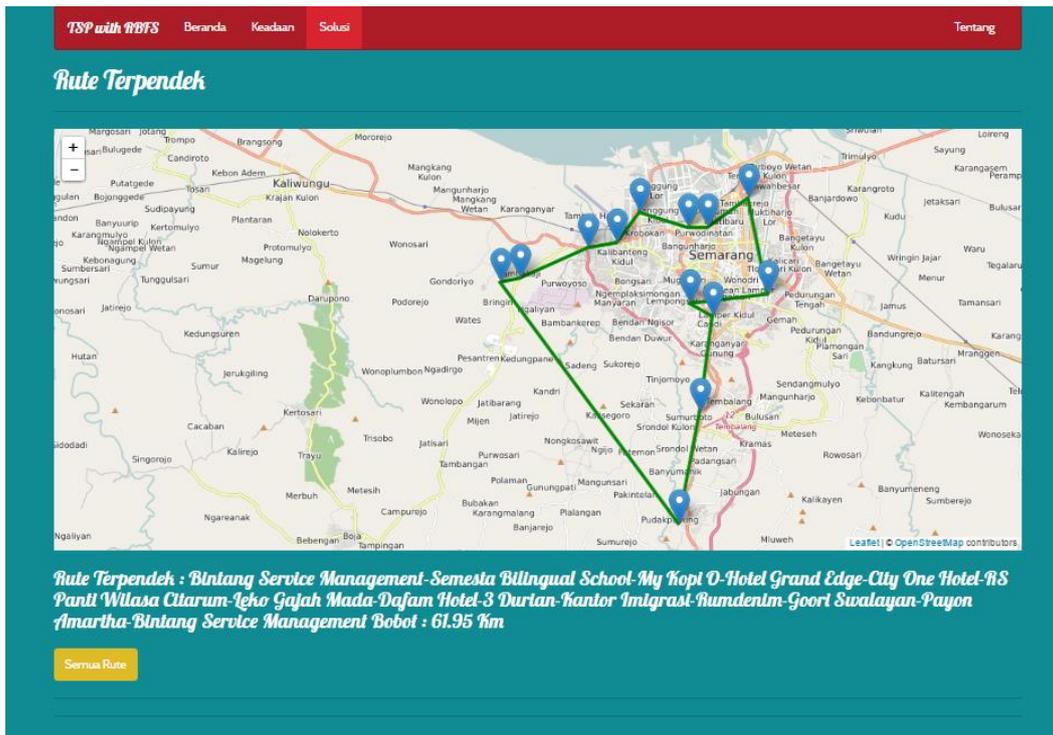
```
function route($start){
    $next = $this->possible($start['id']);
    $lastone = true;
    foreach($next as $nxt){
        $chkpath = $start['path'].'-'. $nxt['id'];
        $chkp = explode('-', $chkpath);
        if(!strpos('x'. $start['path'], $nxt['id'])
            || ($chkp[0]==$nxt['id']
                && sizeof($chkp)==sizeof($this->peta)+1)){
            $tmp = $nxt; if($start['path']==''){
                $tmp['lenmin'] = $nxt['len'];
                $tmp['path'] = $start['id'].'-'. $tmp['id'];
            }else{$tmp['lenmin'] = $start['lenmin'] + $nxt['len'];
                $tmp['path'] = $start['path'].'-'. $tmp['id'];
            } array_push($this->routing, $tmp);
            $lastone = false;} }
    usort($this->routing, function($a,$b){
        return $a['lenmin'] - $b['lenmin'];
    });$open = $this->routing[0]; $back = $this->routing[1];
    if($open['path']){$tmp = array();
        $tmp['path'] = $this->KOTA($open['path']);
        $tmp['cost'] = $open['lenmin'];
        $rray_push($this->result, $tmp);
    }
    if($open['path']!=''){ unset($this->routing[0]);
        $this->route($open);
    }
}
```

**Gambar 5** Blok kode fungsi `route()`

Hasil dari perhitungan kode tersebut data di *convert* ke bentuk JSON untuk ditampilkan dalam bentuk peta lengkap dengan *marker* dan rute yang dilewati. Pembuatan peta tersebut memanfaatkan *library* LeafletJS dan API Google Maps.

Penerapan program algoritma *Recursive Best First Search* dalam penyelesaian *Traveling Salesman Problem* di PT. Bintang Service Management dengan titik titik penelitian  $V(G) =$

$\{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13}\}$  dan  $E(G) = \{e_1, e_2, e_3, \dots, e_{24}\}$  dapat digambarkan dalam bentuk graf seperti Gambar 4.5. Hasil perhitungan *Traveling Salesman Problem* tersebut menggunakan program diperoleh rute terpendek  $C_1 = (v_1, v_2, v_3, v_5, v_4, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13})$  dengan bobot 61,95 km. Rute tersebut selengkapnya dapat dilihat pada Gambar 6.



Gambar 6 Hasil perhitungan program

Pada Gambar 6 diperoleh panjang rute minimum dengan bobot 61,96 km . Rute yang dimaksud yaitu siklus Hamilton yang dimulai dari Bintang Service Management – Semesta Bilingual School – My Kopi O – Hotel Grand Edge – City One Hotel – RS Panti Wilasa Citarum – Leko Gajah Mada – Dafam Hotel – 3 Durian – Kantor Imigrasi – Rumdenim – Goori Swalayan – Payon Amarta – Bintang Service Management.

Penerapan program algoritma *Recursive Best First Search* dalam penyelesaian *Traveling Salesman Problem* di PT. Bintang Service Management dari graf Hamilton  $G$  di mana  $V(G) = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13}\}$  dan  $E(G) = \{e_1, e_2, e_3, \dots, e_{24}\}$  menggunakan bahasa pemrograman PHP diperoleh rute terpendek  $C_1 = (v_1, v_2, v_3, v_5, v_4, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13})$  dengan bobot 61,95 km. Total rute yang mungkin dilalui ada 48 rute dengan rute terpanjang yaitu  $C_{47} = (v_1, v_2, v_3, v_4, v_8, v_7, v_6, v_9, v_5, v_{10}, v_{11}, v_{12}, v_{13})$  dengan bobot 81,75 km. Selisih dari rute terpanjang dan rute terpendeknya yaitu bobot  $C_1$  dikurangi bobot  $C_{47}$  sehingga  $81,75 - 61,95 = 19,8$ , artinya kita lebih efektif dalam jarak sejauh 19,8 km dan lebih efisien dalam waktu perjalanan sehingga program yang telah dibuat dapat mempermudah proses untuk mendapatkan urutan lokasi pendistribusian barang maupun pengecekan barang secara rutin setiap akhir bulan.

Hasil dari program berupa siklus Hamilton yang digambarkan dalam bentuk Peta

menggunakan *library* Google Maps versi gratis, sehingga keadaan rute dapat digambarkan seperti keadaan aslinya. Kendala yang muncul adalah ketika titik yang digunakan dalam penelitian ini berjumlah lebih dari sepuluh titik, maka keadaan sisi (jarak antar perusahaan) tersebut tidak dapat digambarkan dalam bentuk peta. Solusi dari masalah tersebut dapat digunakan Google Maps versi berbayar atau menggunakan *library* yang gratis atau open source.

Dalam penelitian ini penulis memilih menggunakan LeafletJS untuk mengatasi keterbatasan pada *library* Google Maps, LeafletJS mampu menggambarkan peta layaknya Google Maps dan mampu menggambarkan keadaan (graf) dalam bentuk peta, namun demikian LeafletJS juga memiliki sedikit kekurangan yaitu jarak antar perusahaan pada peta tersebut digambarkan dalam bentuk ruas garis lurus, keadaan ini masih belum sesuai dengan keadaan sesungguhnya di lapangan mengingat keadaan jalan sesungguhnya memungkinkan adanya belokan-belokan.

## PENUTUP

Algoritma *Recursive Best First Search* dapat diterapkan untuk menyelesaikan *Traveling Salesman Problem* di PT. Bintang Service Management. Hasil perhitungan algoritma berupa siklus Hamilton dengan bobot terkecil yang merupakan solusi dari *Traveling Salesman Problem*.

Pembuatan program algoritma *Recursive Best First Search* untuk penyelesaian *Traveling Salesman Problem* di PT. Bintang Service Management dengan bahasa *Hypertext Preprocessor* (PHP) dengan memvisualisasikan keadaan data di lapangan (lokasi perusahaan) dalam bentuk peta (Graf), kemudian data diolah dengan algoritma RBFS dalam fungsi TSP, sehingga menghasilkan data dalam bentuk JSON, data JSON selanjutnya diparsing ke AngularJS menggunakan `LeafletJs` sehingga keluaran program berupa siklus Hamilton yang divisualisasikan dalam bentuk peta di lapangan dan keterangan tentang siklus tersebut.

Bobot yang digunakan dalam penelitian ini berupa jarak (dalam satuan km) dan mengabaikan halangan seperti kepadatan lalu lintas, lampu lalu lintas, portal jalan, pengalihan jalan dan halangan sejenisnya. Diharapkan penelitian selanjutnya dapat menggunakan kepadatan lalu lintas, lampu lalu lintas, portal jalan, pengalihan jalan dan halangan sejenisnya.

Dalam penelitian ini penulis menggunakan bahasa pemrograman *Hypertext Preprocessor* (PHP) untuk menyelesaikan *Traveling Salesman Problem*, selain itu program tersebut dapat memvisualisasikan dalam bentuk peta. Peta yang di sajikan dihubungkan dalam bentuk ruas garis lurus, sehingga dalam penelitian ini saran yang dapat disampaikan adalah untuk penelitian-penelitian selanjutnya yang mengkaji masalah *Traveling Salesman Problem* dapat mencoba untuk menampilkan rute sesuai dengan keadaan yang sebenarnya.

#### DAFTAR PUSTAKA

- Anggit, W. D., Alamsyah & Abidin, Z. 2014. Solusi Travelling Salesman Problem Menggunakan Algoritma Fuzzy Evolusi. *Unnes Journal of Mathematics*, 3(1):39-43.
- Anitya, S. F., Sugiharti, E. & Dwijanto. 2013. Implementasi Algoritma Genetika untuk Menyelesaikan Travelling Salesman Problem. *UNNES Journal of Mathematics*, 2(2):118-120.
- Çiçek, I. & Özer, B. 2011. The Effect of Outsourcing Human Resource on Organizational Performance: The Role of Organizational Culture. *International Journal of Business and Management Studies*, 3(2):131-144.
- Gupta, S. & Panwar, P. 2013. Solving Travelling Salesman Problem Using Genetic Algorithm. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(6):376-380.

- Hermuningsih, S. 2013. Buletin Ekonomi Moneter dan Perbankan. *Pengaruh Profitabilitas, Growth Opportunity, Struktur Modal terhadap Nilai Perusahaan pada Perusahaan Publik di Indonesia*, Oktober:127-144.
- Hlaing, Z. C. S. S. & Khine, M. A. 2011. Solving Traveling Salesman Problem by Using Improved Ant Colony Optimization Algorithm. *International Journal of Information and Education Technology*, 1(5): 404-409.
- Iqbal, Z. & Dad, A. M. 2013. Outsourcing: A Review of Trends, Winners & Losers and Future Directions. *International Journal of Business and Social Science*, IV(8):91-107.
- Katiyar, S., Sonigoswami, Mehta, R. & Gaurav Singh. 2014. Implementation of Travelling Salesman Problem Using ant Colony Optimization. *Journal of Engineering Research and Applications*, 4(6): 63-67.
- Korf, R. E. 1993. Linear-space best-first search. *Artificial Intelligence*, 1(62):41-78.
- Kusrini & Istiyanto, J. E. 2007. Penyelesaian Travelling Salesman Problem dengan Algoritma Cheapest Insertion Heuristics dan Basis Data. *Jurnal Informatika*, 8(2):109-114.
- Sutojo, T., Mulyanto, E. & Suhartono, V. 2011. *Kecerdasan Buatan*. I ed. Yogyakarta: ANDI.
- Tanjung, W. N. & Sopiah. 2010. *Implementing Ant Colony Optimization (Aco) In Traveling Salesman Problem (Case Study at Distribution Store PB)*. Jakarta, Proceeding 7th International Seminar on Industrial Engineering and Management.