

# Sistem Pendukung Keputusan Pemilihan Taksi Berbasis Web dengan Algoritma Pencarian Linear

Bimo Satriantoro, Ni Made Satvika Iswari

Program Studi Teknik Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia

bimo.satriantoro@student.umn.ac.id, satvika@umn.ac.id

Diterima 11 Desember 2015

Disetujui 30 Desember 2015

**Abstract**—Indonesian people have a high interest for public transportation. There are few kind of public transportation, and taxi is one of them. Taxi use taximeter to calculate passanger fares. Taximeters between taxi companies are different in term of results although using the same equation and principle. The total fare from the taximeter is based on equation that includes distance variable, speed of the vehicle, and time variable. People have a problem to choose taxi because of the differnces of taximeter fares and taxi pool. Based on the problem, the solution is an application to help customer to decide choosing taxi. The application will help customer by giving informations about taximeter and carpool form two different company. The informations are based on the distance of the start point and the finish point of the customer and based on the distances of the customer and the carpool of the taxi company that will processed trough the application to calculating the total fares for the customer between two taxi companies. By using algorithm for linear searching, customer will get the informations to help them to decide which taxi the customer will use.

**Keywords** : Linear Algorithm, Taximeter, Google Apps, Choice, Taxi.

## 1. PENDAHULUAN

Jakarta adalah kota ddengan tingkat kemacetan mencapai 9,5 atau sangat macet, hal ini diungkapkan oleh Mantan Menteri Perhubungan Indonesia Jusman Syafi`I Djamil [1]. Hal ini menyebabkan dampak negatif bagi para pekerja sehingga menjadi malas berkendara dan pada

akhirnya akan menggunakan kendaraan umum.

Taksi adalah salah satu dari sekian banyak kendaraan umum yang nyaman. Perbedaan taksi dan angkutan umum lainnya adalah pada metode pembayarannya. Taksi menggunakan argo sebagai acuan dalam pembayaran. Argometer adalah alat untuk penetap tarif pada taksi yang ditetapkan berdasarkan jarak dan waktu perjalanan.

Masalah yang dihadapi oleh calon penumpang taksi adalah dalam pemilihan armada taksi dikarenakan taksi mempunyai tarif yang berbeda-beda serta pool yang tersebar dimana-mana. Sehingga calon penumpang mengurungkan niatnya untuk menggunakan taksi.

Algoritma pencarian linear adalah algoritma yang digunakan untuk mencari satu nilai tertentu didalam sebuah array dengan cara memeriksa satu persatu dan melakukan perbandingan. Jika yang yang dibandingkan bernilai sama maka pencarian dihentikan, namun jika tidak pencarian akan terus dilakukan [2]. Algoritma ini dapat digunakan untuk mencari nilai maksimum dan minimum jarak antara posisi awal calon penumpang dengan posisi pool taksi.

Dari masalah diatas, dibutuhkan sebuah sebuah aplikasi yang dapat memberikan informasi mengenai tarif taksi dan jarak pool taksi terdekat.

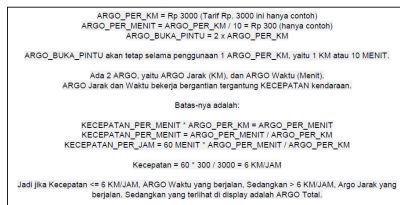
## 2. TINJAUAN PUSTAKA

### 1.1. Argometer

Argometer adalah alat untuk penetap tarif pada taksi yang ditetapkan berdasarkan jarak dan waktu perjalanan. Argometer dikalibrasi secara reguler dan disegel oleh instansi yang berwenang dalam hal ini Dinas Perdagangan khususnya yang menangani metrology [3].

Gambar 1 berikut menunjukkan cara

perhitungan argo taksi.



Gambar 1. Perhitungan Argo Taksi

## 1.2. Pencarian Linear

Algoritma Pencarian Linear (*Linear Search*) adalah algoritma yang digunakan untuk mencari nilai pada sebuah array atau draft nilai dengan cara memeriksa satu per satu. Pencarian Linear atau yang dikenal sebagai *sequential search* bekerja dengan memeriksa setiap elemen dari sebuah list sampai sebuah kecocokan ditemukan [4].

Pencarian dapat dilakukan pada barisan yang terurut secara menaik atau menurun ataupun tidak berurut. Pencarian linear dapat dilakukan dengan cara membandingkan data yang dicari (X) dengan data dalam barisan  $A[1] \dots A[n]$  dengan dimulai dari kata elemen pertama pada barisan A. Jika perbandingan bernilai sama, maka pencarian dihentikan dan dinyatakan sukses. Sedangkan apabila perbandingan tidak bernilai sama maka,

1. Jika tidak terurut, maka pencarian akan dilanjutkan ke data selanjutnya.
2. Jika data terurut secara menaik, maka pencarian akan hanya dilakukan ke data selanjutnya yang berbeda di sebelah kanan data yang sedang dibandingkan apabila data yang dicari (X) lebih besar dari pada data yang sedang dibandingkan sekarang.
3. Jika data berurut secara menurun, maka pencarian hanya akan dilakukan sampai data yang dicari (X) lebih kecil dari data yang dibandingkan sekarang.

Kelebihan dari algoritma pencarian linear antara lain :

1. Algoritma pencarian sekuensial ini cocok untuk pencarian nilai tertentu pada sekumpulan data terurut atau tidak.
2. Keunggulan algoritma ini adalah dalam mencari sebuah nilai dari sekumpulan kecil data.
3. Termasuk algoritma yang sederhana dan

cepat karena tidak memerlukan proses persiapan data.

Sedangkan kelemahan algoritma pencarian linear adalah bahwa dalam kasus terburuk (nilai tidak ditemukan), pembandingan nilai dilakukan sebanyak jumlah data dalam kumpulan nilai. Dengan demikian, proses pencarian akan bertambah lambat secara linear dengan bertambahnya banyaknya jumlah data [2].

Cara kerja algoritma pencarian linear dapat dituliskan sebagai berikut (Gambar 2):

```

1. i ← 0
2. Ketemu ← false
3. Selama (tidak ketemu) dan (i <= N) kerjakan baris 4
4. Jika (data[i] = x) maka ketemu ← true, jika tidak i ← i + 1
5. Jika (ketemu) maka i adalah indeks dari data yang dicari, jika tidak data tidak ditemukan.

```

Gambar 2. Algoritma Pencarian Linear

Pseudocode dari Algoritma linear maksimum/minimum adalah sebagai berikut :

```

function minimum(a[1..n])           function maximum(a[1..n])
minIndex := 1                       maxIndex := 1
minValue := a[1]                   maxValue := a[1]
for i from 2 to n                   for i from 2 to n
  if a[i] < minValue                 if a[i] > maxValue
    minIndex := i                   maxIndex := i
    minValue := a[i]                 maxValue := a[i]
return minValue                       return maxValue

```

Gambar 3. Algoritma Pencarian Linear Max Min

## 1.3. Uji Reliabilitas

Menurut Endang Kurniawan dan Endah Mutaqimah [5], reliabilitas adalah ketepatan atau keajegan suatu tes apabila diteskan kepada subyek yang sama. Tes dikatakan memiliki reliabilitas yang tinggi atau terdapat korelasi yang tinggi antara hasil tes pertama dengan hasil tes kedua apabila hasil skor tesnya sama. Kalau antara hasil tes pertama dengan hasil tes kedua tidak terdapat hubungan atau hubungannya rendah, maka tes tersebut dikatakan tidak reliabel.

Miller [6] menyebutkan bahwa reliabilitas (r) dinyatakan sebagai koefisien korelasi (ukuran suatu hubungan) dari skala 0,00 (tidak terdapat hubungan) sampai 1 (hubungan positif yang sempurna). Pada buku Depdikbud (1997), ada beberapa prosedur untuk menghitung indeks reliabilitas tes, diantaranya adalah koefisien Alpha atau Cronbach's Alpha, dengan rumus :

$$\alpha = \frac{k}{k-1} \left( 1 - \frac{\sum_{i=1}^k \sigma_{yi}^2}{\sigma_x^2} \right)$$

Keterangan :

$\alpha$  = koefisien Alpha atau Cronbach's Alpha,

K = jumlah butir soal dalam tes,

$\sigma_{yi}^2$  = varian butir soal i.

$\sigma_x^2$  = varian tes soal (skor total).

Tingkatan reabilitas dengan metode Cronbach's Alpha diukur berdasarkan skala Alpha 0 sampai dengan 1. Apabila skala tersebut dikelompokkan ke dalam 5 kelas dengan range yang sama, maka ukuran kemantapan Alpha dapat diinterpretasikan seperti tabel berikut [7].

Tabel 1. Tingkat Reliabilitas

Alpha	Tingkat Reliabilitas
0,00 – 0,20	Kurang Reliabilitas
0,20 – 0,40	Kurang Reliabilitas
0,40 – 0,60	Cukup Reliabilitas
0,60 – 0,80	Reliabilitas
0,80 – 1,00	Sangat Reliabilitas

### 3. METODOLOGI PENELITIAN

#### 1.4. Metode Penelitian

##### a. Studi Literatur

Melakukan studi mengenai teori – teori dan konsep yang berkaitan dengan pokok bahasan penelitian, seperti teori mengenai algoritma pencarian linear, dan berbagai konsep pendukung lainnya salah satunya adalah mengenai Google API. Referensi – referensi yang digunakan berupa buku, jurnal ilmiah, artikel, dan lain – lain.

##### b. Pengumpulan Data

Melakukan pengumpulan data tentang taksi yang ada di Jakarta dan di BSD, dan juga mengumpulkan data mengenai tarif argo yang ditetapkan dengan cara melakukan pencarian di website.

##### c. Rancangan Aplikasi

Pada rancangan aplikasi sistem pendukung keputusan, rancangan alur serta rancangan antarmuka dibuat. Rancangan ini digunakan

sebagai dasar dalam pembuatan aplikasi.

##### d. Pemrograman dan Implementasi

Melakukan pembangunan terhadap aplikasi dengan mengimplementasikan rancangan dan metode yang telah didefinisikan sebelumnya dengan menggunakan bahasa pemrograman yang telah ditentukan.

##### e. Uji Coba dan Evaluasi

Melakukan uji coba terhadap aplikasi disertai dengan evaluasi hasil yang didapat.

#### 1.5. Analisis Perhitungan Argo Taksi

Berikut adalah beberapa langkah dari perhitungan mengenai argo taksi serta memberikan informasi mengenai pool yang terdekat dengan *user*:

##### 1.5.1. Menentukan posisi awal dengan akhir

Dari hasil analisis yang dilakukan, untuk mendapatkan jarak harus dilakukan perhitungan diantara dua tempat. Google Maps API dapat memberikan informasi jarak diantara dua tempat dengan menginput posisi awal dan posisi akhir. Karena untuk menghitung argo taksi diperlukan variabel jarak.

##### 1.5.2. Menghitung argo taksi

Menurut Alam (2015) perhitungan argo taksi terdiri dari dua cara yaitu argo jarak dan argo waktu. Kondisi argo waktu terjadi apabila kecepatan mobil dibawah 10 km/jam (macet). Sebaliknya jika kecepatan di atas 10 km/jam maka yang dihitung adalah argo jarak. Setelah mendapatkan variabel jarak, dilakukan perhitungan. Penulis membuat tiga kriteria dalam perhitungan argo, yaitu cepat (lancar), sedang (padat merayap), dan lambat (macet). Kriteria tersebut dibuat agar *user* dapat mengetahui berapa argo yang harus dibayar jika kondisi jalan macet, padat, atau lancar. Jika dikonversi kedalam kecepatan, kriteria lambat berkecepatan 0 – 20 km/jam, padat mempunyai kecepatan 20 – 40 km/jam, dan pada kriteria cepat berkecepatan 40 – 60 km/jam.

#### 1.6. Analisis Pencarian Jarak dengan Pool Taksi

Fitur yang ada pada aplikasi ini adalah memberikan informasi mengenai pool terdekat untuk melakukan pemesanan taksi. Sehingga *user* dapat langsung mengetahui pool mana yang terdekat dengan lokasi awal sehingga taksi dapat langsung datang dengan waktu yang singkat setelah dipesan.

##### 1.6.1. Menghitung jarak antara posisi awal dengan daftar pool

Perhitungan ini dilakukan dengan cara melakukan kalkulasi posisi awal terhadap posisi pool yang terdaftar. Setelah dilakukan perhitungan pada setiap pool, kemudian data tersebut disimpan ke dalam *database* untuk nantinya dilakukan pencarian.

#### 1.6.2. Mencari nilai minimum dengan algoritma pencarian linear

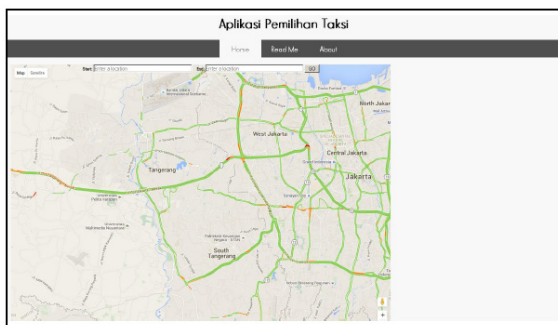
Setelah dilakukan perhitungan jarak antara posisi awal dengan pool, maka akan dilakukan pencarian nilai minimum untuk mendapatkan nilai terkecil yaitu jarak antara posisi awal dengan pool menggunakan algoritma pencarian linear. Pencarian tersebut mengacu pada tabel yang sudah isi sebelumnya yaitu pada proses perhitungan jarak antara posisi awal dengan pool. Setelah itu user akan mendapatkan informasi mengenai pool terdekat dari posisi awal serta jaraknya.

### 4. IMPLEMENTASI DAN UJI COBA

#### 4.1. Implementasi Sistem

Implementasi aplikasi Sistem Pendukung Keputusan Pemilihan Taksi adalah sebagai berikut:

#### 1.6.3. Tampilan Halaman Utama



Gambar 4. Tampilan Halaman Utama

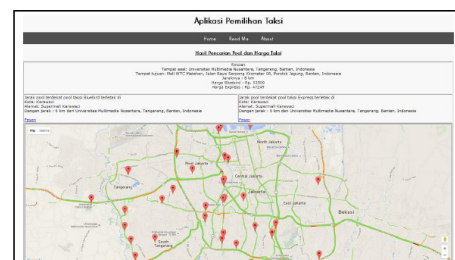
Halaman utama dari sistem ini adalah halaman awal. Halaman ini terdapat maps yang diambil dari google map service. Map tersebut memberikan informasi tentang nama tempat serta keadaan jalan. Jika jalanan berwarna hijau itu bertanda bahwa jalanan lancar, warna kuning menandakan jalanan padat, sedangkan warna merah menandakan bahwa jalanan macet. Pada bagian atas terdapat sebuah form untuk *user* menginputkan posisi awal dan posisi akhir. Pada bagian kanan terdapat kolom yang nantinya akan digunakan sebagai panel

informasi mengenai *route* yang akan dilewati. Pada bagian bawah terdapat *button* detail yang akan membara *user* ke halaman detail.

#### 1.6.4. Tampilan Halaman Detail

Pada halaman detail *user* akan mendapatkan informasi mengenai rincian dari *inputan* awal serta beberapa informasi yang lain seperti :

- Rincian tentang posisi awal dan akhir serta jaraknya, selain itu ada informasi mengenai argo dari dua armada taksi yaitu express dan bluebird. Informasi argo tersebut dihasilkan dengan mengambil variabel jarak yang telah di cari oleh proses direction route pada inputan sebelumnya.
- Setelah mendapatkan variabel jarak, sistem juga dapat menghitung jarak antara posisi awal dengan daftar pool yang sudah disimpan. Oleh karena itu *user* juga mendapatkan informasi mengenai pool taksi terdekat beserta informasinya.
- Yang terakhir adalah informasi map berupa marker dari daftar pool yang ada dalam daftar pool.



Gambar 5. Tampilan Halaman Detail

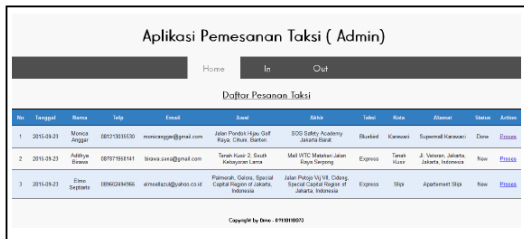
#### 1.6.5. Tampilan Halaman Pesan

Pada halaman pemesanan *user* akan memasukan informasi mengenai nama, telp, serta alamat email yang nantinya akan digunakan admin untuk memverifikasi pemesanan taksi, seperti terlihat dari Gambar 6.

Gambar 6. Tampilan Halaman Pesan

1.6.6. Tampilan Halaman Admin

Pada halaman admin (Gambar 7), admin akan mendapatkan informasi mengenai daftar dari pesanan yang sudah dilakukan oleh *user*. Admin dapat melakukan proses pemesanan dengan mengklik tulisan proses yang ada pada setiap pesanan.



Gambar 7. Tampilan Halaman Admin

4.2. Uji Coba Sistem

Uji coba sistem dibagi menjadi dua, yaitu :

1. Yang pertama adalah uji coba mengenai penggunaan sistem dengan menggunakan algoritma pencarian linear. Dari 56 kuesioner yang dikembalikan, terdapat 45 pegawai swasta dan 11 pengusaha yang telah menguji aplikasi ini. Kuesioner menunjukkan hasil 70% menyatakan pencarian nilai jarak dengan algoritma linear sudah sesuai dengan harapan, sedangkan 30% menyatakan kurang sesuai dengan harapan.
2. Yang kedua adalah untuk menguji keseluruhan aplikasi. Berikut ini adalah hasil uji coba sistem :

a. Usefulness

Tingkat Kegunaan



■ Sangat Membantu ■ Cukup Membantu ■ Netral  
■ Kurang Membantu ■ Tidak Membantu

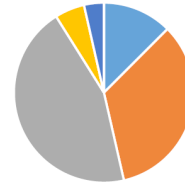
Gambar 8. Tingkat Kegunaan Sistem

Gambar 8 menunjukkan tingkat kegunaan dari sistem pendukung keputusan pemilihan taksi. Sebanyak 11 responden menyatakan sangat membantu, 22 responden menyatakan cukup membantu, 14 responden menyatakan

netral, dan 9 responden menyatakan kurang membantu.

b. Ease of Use

Tingkat Kemudahan untuk digunakan



■ Sangat Mudah ■ Cukup Mudah ■ Netral ■ Sulit ■ Sangat Sulit

Gambar 9. Tingkat Kemudahan Sistem

Gambar 9 menunjukkan tingkat kemudahan untuk digunakan dari sistem pendukung keputusan pemilihan taksi. Sebanyak 7 responden menyatakan sangat mudah, 19 responden menyatakan cukup mudah, 25 responden menyatakan netral, 3 responden menyatakan sulit, dan 2 responden menyatakan sangat sulit.

c. Ease of Learning

Gambar 10 menunjukkan tingkat kemudahan untuk dipelajari dari sistem pendukung keputusan pemilihan taksi. Sebanyak 7 responden menyatakan sangat mudah, 24 responden menyatakan cukup mudah, 19 responden menyatakan netral, 3 responden menyatakan sulit, dan 2 responden menyatakan sangat sulit.

Tingkat Kemudahan untuk dipelajari



■ Sangat Mudah ■ Cukup Mudah ■ Netral ■ Sulit ■ Sangat Sulit

Gambar 10. Tingkat Kemudahan Sistem

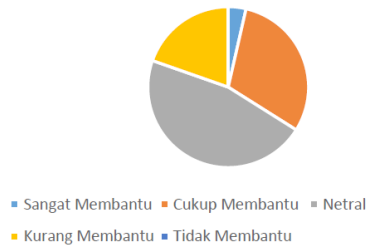
d. Help Facility

Gambar 11 menunjukkan tingkat bantuan dari sistem pendukung keputusan pemilihan taksi. Sebanyak 2 responden menyatakan sangat membantu, 17 responden menyatakan cukup membantu, 26 responden menyatakan netral, dan 11 responden menyatakan kurang



membantu.

Tingkat Bantuan

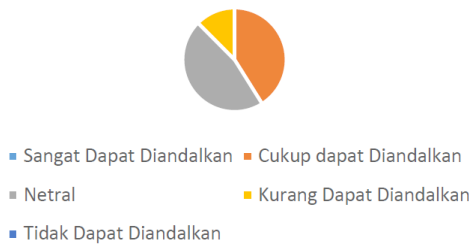


Gambar 11. Tingkat Bantuan Sistem

e. Reliability

Gambar 12 menunjukkan tingkat keandalan dari sistem pendukung keputusan pemilihan taksi. Sebanyak 23 responden menyatakan cukup dapat diandalkan, 26 responden menyatakan netral, dan 7 responden menyatakan kurang dapat diandalkan.

Tingkat Keandalan

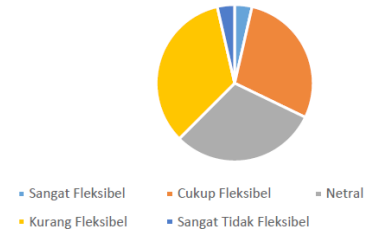


Gambar 12. Tingkat Keandalan Sistem

f. Flexibility

Gambar 13 menunjukkan tingkat fleksibilitas dari sistem pendukung keputusan pemilihan taksi. Sebanyak 2 responden menyatakan sangat fleksibel, 16 responden menyatakan cukup fleksibel, 17 responden menyatakan netral, 19 responden menyatakan kurang fleksibel, dan 2 responden menyatakan sangat tidak fleksibel.

Tingkat Fleksibilitas



Gambar 13. Tingkat Fleksibilitas Sistem

Untuk menguji realibilitas dari kuesioner, maka dapat digunakan rumus *Cronbach Alpha* sebagai berikut :

$$\alpha = \left(\frac{6}{5}\right) \left(1 - \left(\frac{2.096619896}{8.631377551}\right)\right) = 0.908511896$$

Berdasarkan hasil tersebut, dapat ditemukan bahwa nilai *Cronbach Alpha* berada diantara 0,80 dan 1,00 yakni  $0,80 < 0.908511896 < 1,00$ , sehingga dinyatakan sangat *reliable*.

4.2.1. Uji Validasi Sistem

Uji validasi sistem digunakan untuk menemukan kekuatan dan keakuratan dari aplikasi ini. Validasi sistem yang diberikan berupa survey mengenai informasi mengenai hasil perhitungan argo serta jarak pool taksi mempermudah dalam memilih taksi atau tidak.

Tabel 2. Tingkat Bantuan dari Sistem

No	Keterangan	Jumlah Responden
1	Ya	39
2	Tidak	17

Dari 56 responden yang diuji dapat dilihat bahwa 39 responden menyatakan bahwa informasi mengenai hasil perhitungan argo serta jarak pool taksi dapat membantu dalam pemilihan taksi.

V. SIMPULAN

Berdasarkan penelitian yang telah dilakukan mengenai sistem pendukung keputusan dengan menggunakan algoritma pencarian linear, maka dapat ditarik kesimpulan sebagai berikut.

Penelitian tentang sistem pendukung keputusan untuk memilih taksi berdasarkan harga termurah serta jarak pool taksi terdekat

menggunakan algoritma pencarian linear berhasil dibangun dengan bahasa pemrograman HTML dan PHP. Berdasarkan hasil uji coba sistem, ditemukan nilai *Cronbach Alpha* berada diantara 0,80 dan 1,00 yakni  $0,80 < 0.908511896 < 1,00$ , sehingga dinyatakan sangat *reliable*.

#### DAFTAR PUSTAKA

- [1] Sulistyawati L. Berapa 'Rating' Tingkat Kemacetan Kota Jakarta?. 2013
- [2] Manurung H. Algoritma, Kompleksitas dan Teori Bilangan. 2013 Feb 28
- [3] Alam S. Ternyata Begini Toh Perhitungan Argo Taxi. 2015
- [4] Manurung H. Perancangan Perangkat Lunak Pencarian Data. 2013.
- [5] Kurniawan E, Mutaqimah. Penilaian. Jakarta: Depdiknas. 2009
- [6] Miller. MAT Reliability and Validity. 2008.
- [7] Sugiyono. Statistika Untuk Penelitian Alfabeta. 2006