

PENCARIAN HOTEL TERDEKAT DENGAN METODE KD-TREE DAN NEAREST NEIGHBOR

Mohtar Efendi¹, Andi Farmadi, S.Si, M.T²,
Radityo Adi Nugroho, S.T, M.Kom³

^{1,2,3}Prodi Ilmu Komputer FMIPA UNLAM

Jl. A. Yani Km 36 Kampus Unlam Banjarbaru, Kalimantan Selatan

Email: efendimohtar@gmail.com

Abstract

Hotels become the only right option for tourist to take a rest before the next day. The nearest hotel usually have been chosen before the others. Because of that, a system which supported the option is made to help accounting to get the nearest hotel. This system using KD-Tree method which holding in the system to makes a tree in use as a comparison in accounting, and Nearest Neighbor method to accounted the nearest object between user and hotel. With this system, the biggest hope is helping the user or tourist to get the nearest hotel easily and according with their wanted.

Key words : Hotel, KD-Tree, Nearest Neighbor

Abstrak

Hotel menjadi salah satu pilihan tepat bagi wisatawan yang sedang berlibur di suatu kota untuk menginap, beristirahat sebelum beraktivitas lagi. Hotel yang terdekat dengan lokasi wisatawan tersebut biasanya sering dipilih lebih dulu sebelum memilih hotel lainnya. Oleh karena itu dibuatlah suatu sistem pendukung keputusan yang dapat melakukan proses perhitungan untuk mendapatkan hotel terdekat. Sistem ini menggunakan metode KD-Tree yang peranya dalam sistem yaitu untuk membuat sebuah tree yang nantinya digunakan sebagai alur perbandingan perhitungan, dan metode Nearest Neighbor yakni untuk melakukan perhitungan jarak terdekat antara pengguna dengan hotel. Dengan adanya sistem ini diharapkan dapat membantu pengguna atau wisatawan sehingga dapat dengan mudah mendapatkan hotel terdekat dan sesuai dengan yang diinginkan.

Kata Kunci : Hotel, KD-Tree, Nearest Neighbor

1. PENDAHULUAN

1.1. Latar Belakang

Metode *Nearest Neighbor* merupakan salah satu teknik klasifikasi yang berdasarkan kedekatan objek, dengan membandingkan jarak setiap objek. Pendekatan yang digunakan pada *Nearest Neighbour* sendiri merupakan pendekatan klasifikasi yang mencari semua data latih yang relatif mirip dengan data uji. Teknik klasifikasi ini disebut *lazy learning* karena teknik ini tidak membangun model klasifikasi terlebih dahulu, seperti pohon keputusan (*decision tree*), klasifikasi berbasis aturan (*rule-based*), dan sebagainya[2].

Metode *K-Dimensional Tree* atau *KD-Tree* merupakan metode untuk mengelompokkan data. Disebut *k-dimensional tree* karena *k-dtree* adalah binary tree yang node-nya merupakan *k-dimensional point*. *K* dapat bernilai 2 atau lebih. Cara *k-d-tree* mengelompokkan data adalah sama dengan binary tree biasa. Suatu region dibagi menjadi dua, kemudian masing-masing dua region tadi dibagi lagi menjadi dua region, demikian seterusnya hingga tidak dapat dibagi lagi[3].

Hotel merupakan bentuk bangunan, lambang, perusahaan atau badan usaha akomodasi yang menyediakan pelayanan jasa penginapan, penyedia makanan dan minuman serta fasilitas jasa lainnya dimana semua pelayanan itu diperuntukkan bagi masyarakat umum, baik yang bermalam di hotel tersebut ataupun yang hanya menggunakan fasilitas tertentu yang dimiliki hotel itu[4]. Bagi masyarakat umum atau wisatawan yang baru datang dan belum pernah menjajakan kaki di suatu kota tertentu dan ingin segera mendapatkan tempat istirahat yang letaknya tidak jauh dari di mana dia berada, serta fasilitas dan jasa yang sesuai dengan yang diinginkan dari penyedia hotel yang ada, itu merupakan suatu hal yang menyulitkan. Untuk mendapatkan apa yang diinginkan oleh wisatawan maka dapat dibutuhkan suatu sistem yang dapat membantu dalam menentukan hotel terdekat. Pada Penelitian ini akan digunakan metode kombinasi *KD-Tree* dan *Nearest Neighbor* dalam penyelesaian pencarian hotel terdekat.

1.2. Perumusan Masalah

Berdasarkan latar belakang di atas maka perumusan masalah yang didapat adalah bagaimana merancang dan membangun sistem untuk menentukan hotel terdekat dengan menerapkan metode *KD-Tree* Dan *Nearest Neighbour* yang dapat membantu dalam pencarian oleh pengguna atau wisatawan.

1.3. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah :

- 1) Menerapkan metode *KD-Tree* dan *Nearest Neighbor* dalam sistem pendukung keputusan.
- 2) Menentukan hotel terdekat dengan menggunakan metode *KD-Tree* dan *Nearest Neighbor*.

1.4. Manfaat Penelitian

Manfaat dari penelitian ini adalah untuk mengetahui dan memahami bagaimana menemukan letak Hotel terdekat dengan menerapkan metode *KD-Tree* dan *Nearest Neighbor* dalam sistem.

1.5. Batasan Masalah

Batasan masalah pada penelitian ini yaitu data yang digunakan dalam penentuan hotel terdekat dengan menggunakan titik koordinat dan informasi tentang hotel tersebut berdasarkan data yang didapat dari konservasi langsung di mana hotel-hotel itu berada.

2. METODELOGI PENELITIAN

2.1. Identifikasi dan Analisa Masalah

Identifikasi masalah dilakukan dengan observasi secara langsung dengan masyarakat umum (wisatawan).

2.2. Metode Pengembangan Sistem

Penelitian ini dikembangkan dengan menerapkan model waterfall. Model ini mengusulkan sebuah pendekatan perkembangan perangkat lunak yang sistematis dan sekunsial.

- 1) Pengumpulan Data
Pengumpulan data dilakukan untuk mendeskripsikan sistem pencarian hotel terdekat, digunakan beberapa metode pengumpulan data antara lain :
 - a) Metode Wawancara (*Interview*)
Merupakan metode pengumpulan data secara langsung dengan mewawancarai pihak-pihak Hotel yang berhubungan terhadap data-data Hotel tersebut.
 - b) Metode Pustaka (*Library Research*)
Metode Pustaka dilakukan dengan cara mempelajari dan mengumpulkan beberapa literature dari media internet atau buku-buku yang berhubungan dalam proses penelitian ini.
 - c) Metode Observasi
Merupakan metode pengumpulan data secara langsung turun kelapangan untuk menentukan titik-titik koordinat hotel, jika titik koordinat Hotel tersebut belum didapatkan pada Metode Wawancara.
- 2) Analisis
Analisis merupakan tahapan untuk menspesifikasikan metode *KD-Tree* dan *Nearest Neighbor* yang akan dipakai. Di dalamnya termasuk proses dari awal pembentukan tree yang dilakukan oleh *KD-Tree* sampai perhitungan yang dilakukan oleh *Nearest Neighbor* sehingga menghasilkan jarak hotel terdekat.
- 3) Testing
Testing merupakan tahapan untuk mengetahui apakah program yang dibuat sudah sesuai dengan spesifikasi dari tahap design.

3. HASIL DAN PEMBAHASAN

3.1. Gambaran Umum Sistem

Sistem yang dibuat merupakan sistem pendukung keputusan yang menggunakan metode *KD-Tree* dan *Nearest Neighbor* berbasis *mobile android* yang dapat membantu pengguna atau wisatawan dalam pencarian hotel terdekat. Sistem ini dibuat dengan menggunakan bahasa pemrograman java dan database menggunakan MongoDB dan SQLite. Sistem kemudian akan menghasilkan informasi hotel terdekat berdasarkan jarak antara koordinat titik pengguna aplikasi dengan koordinat titik Hotel.

3.2. Metode KD-Tree dan Nearest Neighbor

Cara *k-d-tree* mengelompokkan data adalah sama dengan binary tree biasa. Suatu region dibagi menjadi dua, kemudian masing-masing dua region tadi dibagi lagi menjadi dua region, demikikian seterusnya hingga tidak dapat dibagi lagi[3].

Proses rekonstruksi *KD-Tree* menggunakan metode rekursif dengan parameter pada setiap iterasi adalah arrays dari koordinat hotel dan depth. Nilai depth digunakan untuk menentukan nilai axis. Nilai awal untuk arrays adalah semua koordinat hotel dan nilai depth adalah 0 proses untuk setiap iterasi adalah sebagai berikut[1]:

- 1) Menghitung nilai axis dengan rumus: $\text{axis} = \text{depth} \bmod 2$
- 2) Melakukan penyortiran pada array koordinat hotel berdasarkan nilai axis. Jika axis = 0 maka sortir dilakukan berdasarkan longitude, dan jika axis = 1 berdasarkan latitude.
- 3) Mencari koordinat median, dengan cara, (i) $\text{index_median} = \text{jumlah_koordinat} \div 2$ dan (ii) $\text{koordinat_median} = \text{koordinat}[\text{index_median}]$.
- 4) Menentukan node: $\text{node} = \text{koordinat}[\text{index_median}]$
- 5) Menentukan left node dan right node menggunakan iterasi baru. Iterasi selanjutnya menggunakan parameter sebagai berikut: (i) array koordinat = sub array koordinat = $\text{koordinat}[0][\text{index_median} - 1]$ serta $\text{depth} = \text{depth} + 1$, dan (ii) array koordinat = sub array koordinat = $\text{koordinat}[\text{index_median} + 1][\text{jumlah_koordinat}]$

Koordinat hotel yang dipakai dalam perhitungan manual perbandingan tersebut ada 10 yang dijadikan contoh:

1. (-3.32381, 114.60398)
2. (-3.32399, 114.5914)
3. (-3.33472, 114.62178)
4. (-3.32377, 114.58736)
5. (-3.33675, 114.61646)
6. (-3.32646, 114.59583)
7. (-3.33977, 114.61875)
8. (-3.3239, 114.60139)
9. (-3.32866, 114.60192)
10. (-3.32769, 114.5897)

Iterasi 1

Data = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

n = 10

depth = 0

axis = depth mod 2 = 0 mod 2 = 0

sort by longitude

sorted data : 4, 10, 2, 6, 8, 9, 1, 5, 7, 3

median = $n \div 2 = 10 \div 2 = 5$

node = data [s] = (-3.3239, 114.60139)



Gambar 1. Tree Iterasi 1

Iterasi 1.1

data = 4, 10, 2, 6

n = 4

depth = 1

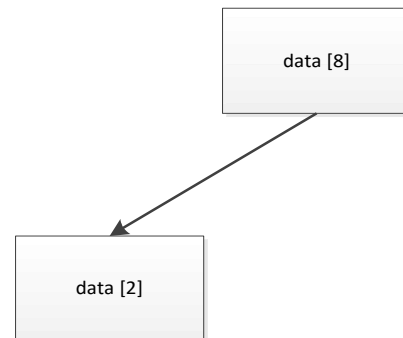
axis = 1 mod 2 = 1

sort by latitude

sorted data : 4, 2, 6, 10

median = $4 \div 2 = 2$

node = data[2]



Gambar 2. Tree Iterasi 1.1

Iterasi 1.2

Data = 9, 1, 5, 7, 3

n = 5

depth = 1

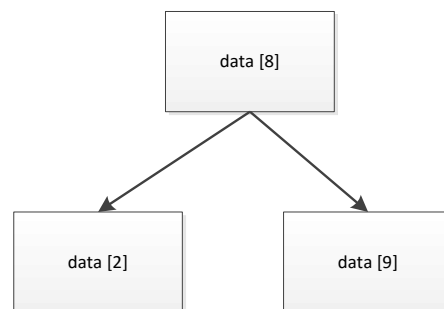
axis = 1 mod 2 = 1

sort by latitude

sorted data : 1, 9, 5, 7, 3

median = $5 \div 2 = 2$

node = data[9]



Gambar 3. Tree Iterasi 1.2

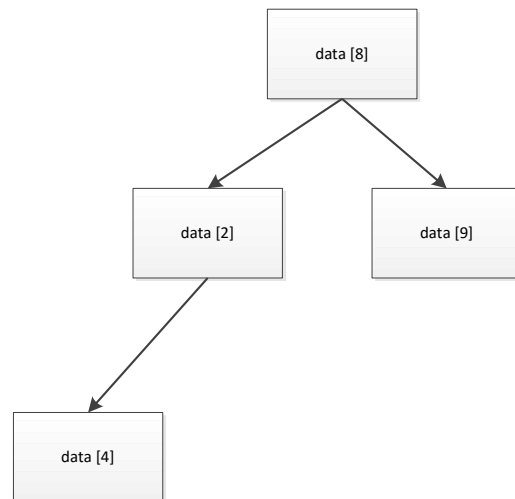
Iterasi 1.1.1

Data = 4

n = 1

depth = 2

karena n = 1 maka node = data[4]



Gamba 4. Tree Iterasi 1.1.1

Iterasi 1.1.2

Data = 6, 10

n = 2

depth = 2

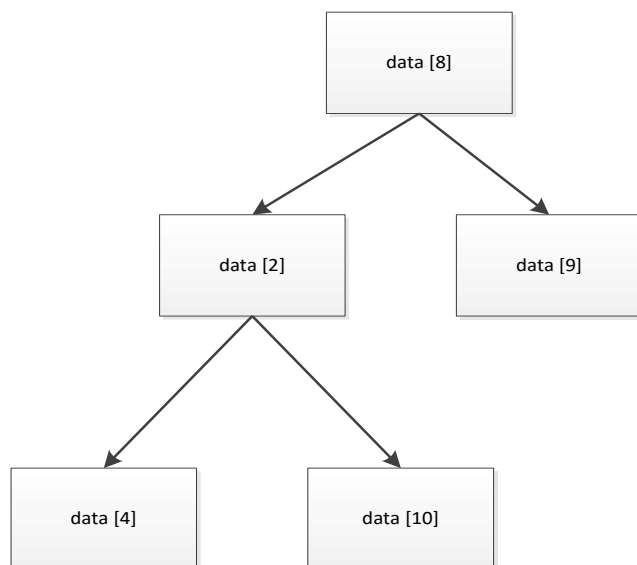
axis = 2 mod 2 = 0

sort by longitude

sorted data : 10, 6

median = 2 div 2 = 1

node = data[10]



Gambar 5. Tree Iterasi 1.1.2

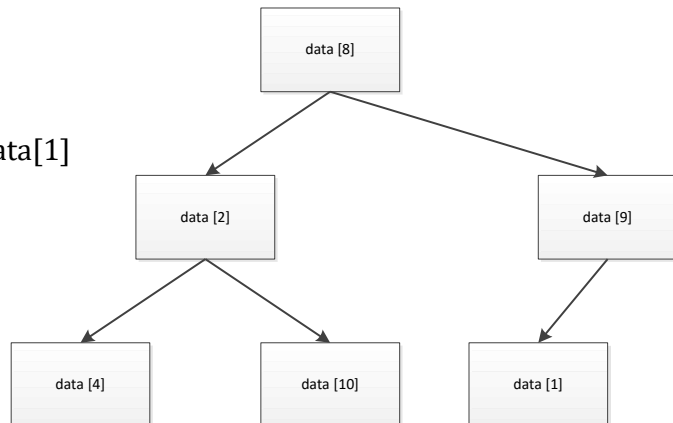
Iterasi 1.2.1

Data = 1

n = 1

depth = 2

karena n = 1, maka node = data[1]



Gambar 6. Tree Iterasi 1.2.1

Iterasi 1.2.2

Data = 5, 7, 3

n = 3

depth = 2

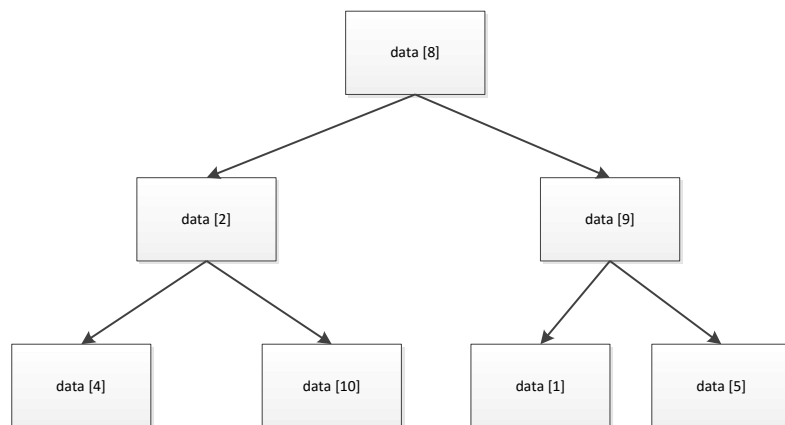
axis = $2 \bmod 2 = 0$

sort by longitude

sorted data : 5, 7, 3

median = $3 \div 2 = 1$

node = data[5]



Gambar 7. Tree Iterasi 1.2.2

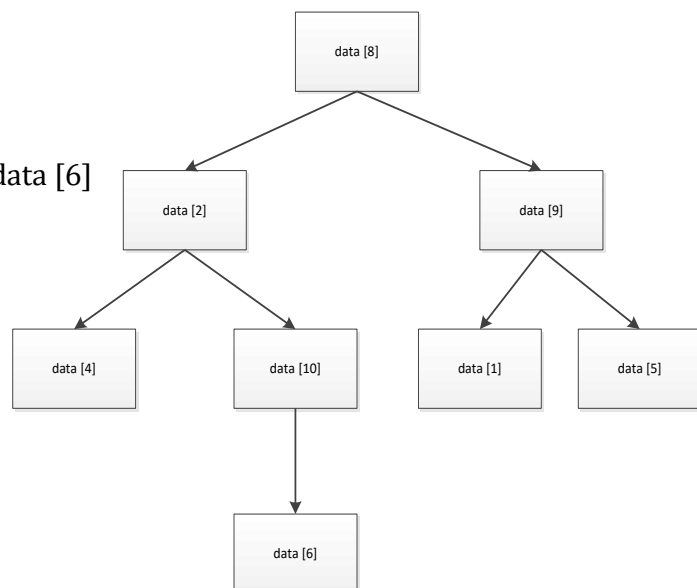
Iterasi 1.1.2.1

Data = 6

n = 1

depth = 3

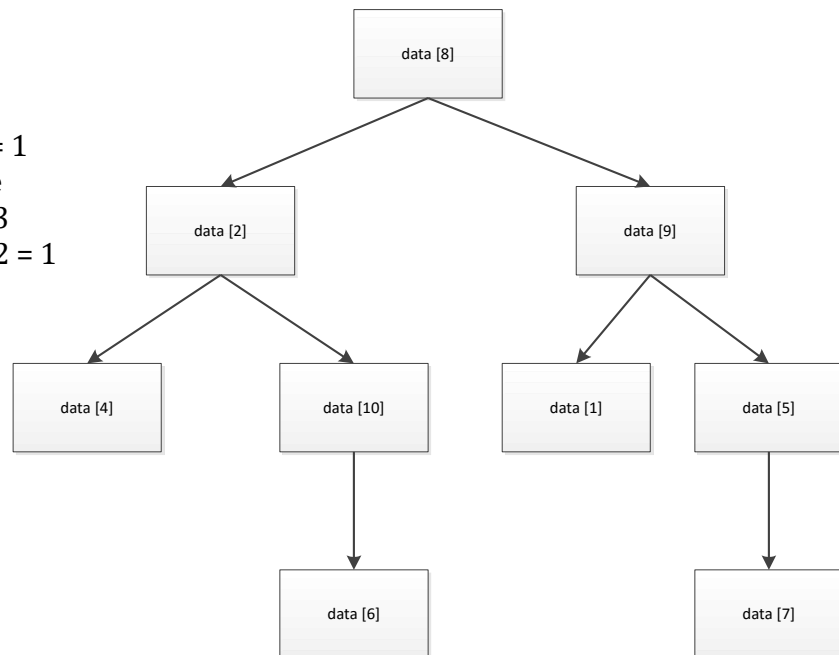
karena n = 1, maka node : data [6]



Gambar 8. Tree Iterasi 1.1.2.1

Iterasi 1.2.2.1

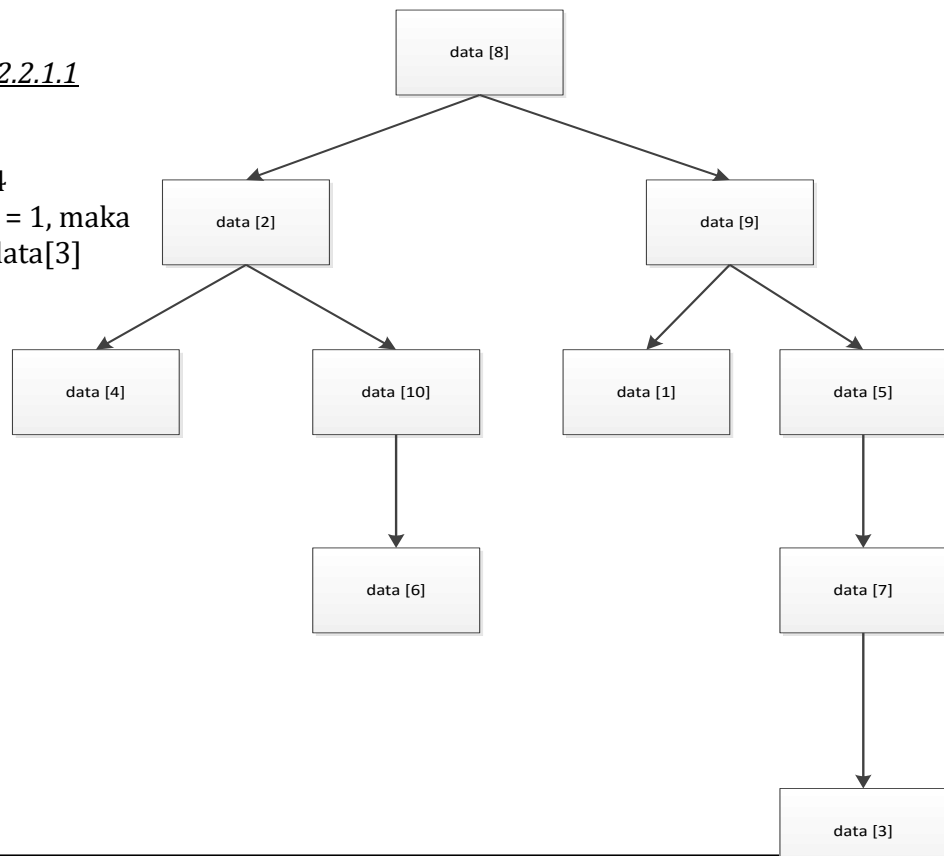
Data = 7, 3
n = 2
depth = 3
axis = $3 \bmod 2 = 1$
sort by latitude
sorted data : 7, 3
median = $2 \div 2 = 1$
node = data[7]



Gambar 9. Tree Iterasi 1.2.2.1

Iterasi 1.2.2.1.1

Data = 3
n = 1
depth = 4
karena n = 1, maka
node = data[3]



Gambar 10. Tree Iterasi 1.2.2.1.1 Perbandingan Hasil

Nearest Neighbor merupakan teknik klasifikasi yang berdasarkan kedekatan objek. Kedekatan disini didefinisikan dengan ukuran jarak, misalnya *Euclidean*. Jarak *Euclidean* antar dua titik, misal Titik1=(x1, y1) dan Titik2=(x2, y2) adalah : $\text{Dist}(\text{Titik1}, \text{Titik2}) = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$ [2].

Prinsip kerja *Nearest Neighbor* yang dimasukkan pada aplikasi adalah mencari jarak terdekat antara data yang akan dicari yaitu hotel terdekat dengan user aplikasi tersebut.

Tabel 1. Perhitungan Manual *Nearest Neighbor*

Posisi User

X1	-3.54439
Y1	114.84220

No	Nama Hotel	X2	Y2	Jarak
1	Hotel A	-3.32769	114.58970	0.332740149
2	Hotel B	-3.32058	114.58225	0.343024509
3	Hotel C	-3.32377	114.58736	0.33707194
4	Hotel D	-3.32417	114.59019	0.334674124
5	Hotel E	-3.32399	114.59140	0.333882736
6	Hotel F	-3.32640	114.59051	0.33296902
7	Hotel G	-3.32724	114.59662	0.327817827
8	Hotel H	-3.32859	114.59687	0.326737375
9	Hotel I	-3.32866	114.60192	0.322916048
10	Hotel J	-3.32646	114.59583	0.328926334
11	Hotel K	-3.32390	114.60139	0.326506016
12	Hotel L	-3.32381	114.60398	0.32466165
13	Hotel M	-3.32982	114.61583	0.311904665
14	Hotel N	-3.33675	114.61646	0.30671444
15	Hotel O	-3.33977	114.61875	0.302985256
16	Hotel P	-3.35119	114.62845	0.288125104
17	Hotel Q	-3.35421	114.62962	0.285235779
18	Hotel R	-3.34720	114.62178	0.295752703
19	Hotel S	-3.30733	114.58822	0.347425146
20	Hotel T	-3.29292	114.58958	0.356447796
21	Hotel U	-3.44395	114.83307	0.100856097
22	Hotel V	-3.44299	114.82483	0.10287897
23	Hotel W	-3.44280	114.82181	0.103617984
24	Hotel Z	-3.55302	114.84571	0.009314638

Jarak Terkecil

0.009314638

Tabel di atas merupakan tabel tentang data titik koordinat hotel beserta koordinat *user*. Dari koordinat tersebut bisa dihitung dan diperoleh hasil terkecil yang nantinya dimaksudkan sebagai titik terdekat. Dimisalkan posisi hotel (*latitude* (x2) ,*longitude* (y2)) dan posisi *user* (*latitude* (x1) dan *longitude* (y1)), kemudian data satu persatu dihitung dengan menggunakan *Nearest Neighbor*. Setelah dilakukan semua perhitungan maka didapatkan hasil terkecil yaitu (0.009314638) Hotel Z, hasil tersebut dimaksudkan menjadi titik terdekat dari *user*.

Begitu pula dengan hasil yang ditampilkan pada aplikasi pencarian hotel terdekat. Aplikasi dijalankan pada posisi *user* yang sama pada koordinat (*latitude* (-3.54439) dan *longitude* (114.84220) ditampilkan dengan titik berwarna biru pada peta google. Setelah proses dilakukan maka didapat hotel terdekat yaitu Hotel Z seperti yang ditampilkan pada gambar 11 di bawah ini.



Gambar 11. Hotel Terdekat

Dari hasil di atas dapat diambil bahwa *Nearest Neighbor* pada aplikasi memang berjalan seperti semestinya dan berjalan dengan baik dalam perhitungan pencarian titik koordinat terdekat.

Di dalam aplikasi ini *KD-Tree* mempunyai peran dalam membantu mempercepat perhitungan yang dilakukan oleh *Nearest Neighbor* yaitu dengan membuat suatu pohon (tree) dari semua titik-titik koordinat yang

ada, yang sebenarnya pohon (tree) itu adalah sebuah jalur yang menyambungkan antara titik-titik yang nantinya digunakan sebagai alur perbandingan bagi *Nearest Neighbor*. *Nearest Neighbor* tidak perlu membandingkan semua titik koordinat yang ada, akan tetapi hanya membandingkan sesuai dengan jalur yang telah *KD-Tree* buat. Ketidakperluan membandingkan semua data koordinat itu lah yang membuat *Nearest Neighbor* lebih *efisien* dengan bantuan *KD-Tree* di bandingkan tidak menggunakannya[1].

4. SIMPULAN

Kesimpulan yang diperoleh dari penelitian ini adalah :

1. Mampu mengimplementasikan metode *KD-Tree* dan *Nearest Neighbor* dalam sistem ini.
2. Metode *KD-Tree* dan *Nearest Neighbor* di dalam sistem ini telah mampu untuk menentukan hotel terdekat dari titik pengguna ke titik hotel terdekat dengan tepat.

5. DAFTARPUSTAKA

- [1]. Efendi, Mohtar, **“Rancang Bangun Aplikasi Mobile Phone Untuk Mencari Hotel Terdekat Dengan Metode KD-Tree dan Nearest Neighbor Berbasis Android”**, Program Studi Ilmu Komputer FMIPA UNLAM, Banjarmasin, May 2014
- [2]. Han, Jiawei, and Micheline Kamber, **“Data Mining: Concepts and Techniques”**, Elsevier Inc, United States of America, 2006.
- [3]. Rudi, Rudolf Hermanto, **“Pemanfaatan Tree untuk Indexing Basis Data Spasial”**, Program Studi Teknik Informatika, Bandung, 2012.
- [4]. Tarmoezi, Trizno, **“Hotel Front Office”**, Kesaint Blanc, Jakarta, 2000.