**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

# IDENTITAS PENGUSUL

**NAMA : AHMAD SODIK**

**NRP : 5110100125**

**DOSEN WALI : Bilqis Amaliah, S.Kom, M.Kom.**

**DOSEN PEMBIMBING : 1. Dwi Sunaryono, S.Kom, M.Kom.  
 2. Adhatus Solichah, S.Kom, M.Sc.**

# JUDUL TUGAS AKHIR

“Aplikasi Mobile Pencarian Tempat Wisata Berdasarkan Daerah Menggunakan Metode Weighted Tree Berbasis Android ”

# LATAR BELAKANG

Android adalah sistem oprasi berbasis linux yang dirancang untuk perangkat seluler layar sentuh. Awalnya android dikembangkan oleh android.inc dan kemudian di beli oleh google pada tahun 2005. Dalam perkembangannya sampai tahun 2013, sistem android sangat besar. Pada tahun 2014, perkembangan android diperkirakan akan semakin menggurita. Sistem operasi android akan menguasai semua perangkat dengan prosentasi 40%. Pada tahun 2013 android sudah mendominasi pangsa pasar sekitar 38%. Tahun 2014 diperkirakan perkembangan android akan semakin besar. Pada tahun 2013 menurut catatan dari lembaga Riset Gatner bahwa sekitar 1,9 milliar unit perangkat android sudah dikapalkan masuk Indonesia. Sedangkan Apple dengan IOS dan MAC hanya mencapai sekitar 682 juta unit di tahun 2013. Menurut lembaga Riset Gatner pada tahun 2014 jumlah perangkat android akan meningkat naik sekitar 7.6% dibanding tahun 2013 **[5]**.

Objek wisata di Jawa Timur sangatlah banyak **[7]**, seperti Wisata Flora, Wisata Museum, Wisata Belanja, Wisata Religi, Wisata Sejarah dan wisata-wisata yang lain. Semua objek-objek wisata tersebut tersebar diseluruh daerah Jawa Timur. Banyaknya tempat wisata di Jawa Timur belum diketahui oleh masyarakat, mereka hanya mengetahui tentang tempat wisata yang terkenal saja seperti wisata di malang, padahal wisata didaerah yang lain tidak kalah bagus dengan yang sudah terkenal.

Algoritma *Weighted Tree* merupakan algoritma yang mempresentasikan sebuah sistem dalam sebuah *tree* yang memiliki node berlabel, cabang berlabel dan cabang berbobot. Algoritma *Weighted Tree* yang memiliki node berlabel, cabang berlabel dan cabang berbobot berfungsi untuk memberikan kecenderungan lebih pada cabang tertentu sehingga membuat pencarian lebih tepat (*precision*) **[4]**.

Dari paparan diatas, penulis membuat Aplikasi Mobile Pencarian Tempat Wisata Berdasarkan Daerah Dengan Menggunakan Metode *Weighted Tree* Berbasis Android. Dengan di buatnya aplikasi tersebut semoga dapat membantu masyarakat untuk mencari tempat-tempat wisata yang berada di daerah tertentu.

# RUMUSAN MASALAH

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut ini.

1. Bagaimana cara memodelkan kata kunci kedalam sebuah *tree* yang berbobot dan berlabel*.*
2. Bagaimana cara memodelkan kata kunci input dari pengguna kedalam sebuah *tree*.
3. Bagaimana cara menggunakan metode *weighted tree* untuk membandingkan tree *input* pengguna dengan tree sistem
4. Bagaimana cara melakukan manajemen ( menambah node (kata kunci pencarian), menghapus dan mengubah ) *tree* sistem.
5. Bagaimana cara mengimplementasikan pencarian tempat wisata berdasarkan daerah dengan menggunakan metode *weighted tree* ke dalam aplikasi mobile berbasis android.

# BATASAN MASALAH

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, di antaranya adalah sebagai berikut ini.

1. Bahasa yang digunakan adalah bahasa Java.
2. Aplikasi akan yang dibuat merupakan aplikasi mobile berbasis android.
3. Data tempat wisata yang digunakan berasal dari internet.
4. Server merupakan aplikasi web yang berfungsi untuk manajemen sistem.
5. Digunakan API Google Maps untuk membantu penunjuk arah pengguna.
6. Lingkup daerah provinsi Jawa Timur.

# TUJUAN PEMBUATAN TUGAS AKHIR

Adapaun tujuan dari Tugas Akhir ini adalah berikut ini.

1. Membuat aplikasi pencarian tempat wisata di daerah Jawa Timur.

2. Memberikan informasi kepada masyarakat tentang tempat wisata di suatu daerah

Jawa Timur.

# MANFAAT TUGAS AKHIR

Manfaat dari tugas akhir ini adalah membantu masyarakat dalam mencari informasi tempat wisata di suatu daerah dengan menggunakan aplikasi mobile pencarian tempat wisata berdasarkan daerah pada smartphone yang berbasis android.

# TINJAUAN PUSTAKA

## Rancang Bangun Perangkat Lunak

Rancang bangun merupakan tahapan untuk mendesain dan membangun suatu perangkat lunak tertentu berdasarkan kebutuhan-kebutuhan yang harus terpenuhi oleh sistem berdasarkan kebutuhan pengguna.

Rancang bangun perangkat lunak diperlukan untuk menentukan konsep, strategi, dan praktik yang baik untuk menciptakan perangkat lunak yang mudah digunakan oleh pengguna, berkualitas tinggi, sesuai anggaran biaya, mudah dalam pemeliharaannya, serta tidak membutuhkan waktu yang lama dalam pembangunannya **[8]**.

## Weighted Tree

Dalam penggunaan algoritma *Weighted Tree*, data disusun berdasarkan *tree* yang memiliki *node* berlabel, cabang berlabel dan cabang berbobot. Cabang yang berlabel memberikan pemahaman lebih kepada label *node*nya, begitu pula bobot cabang memungkinkan memberikan tingkat kecenderungan kepada cabang tertentu lebih dari yang lain. Suatu *tree* terdiri dari 1 atau lebih *parent* yang masing-masing dapat memiliki atribut berupa bobot serta nama *parent*. Setiap *parent* terdiri *identifier*, yang mempunyai atribut bobot **[4]**. Gambar 1 merupakan contoh representasi *tree* dari sistem tentang wisata surabaya.



Gambar 1. Contoh Representasi *Tree* dari Sistem

Sebuah *tree* Surabaya memiliki tiga *parent* yaitu Belanja, Religi dan Alam. Masing – masing *parent* memiliki bobot, yang jika di jumlahkan akan berniai 1. Bobot masing-masing *parent* diperoleh dari persamaan (1).

*Wi =* 1 / n (1)

Dimana

*Wi*  : Bobot *parent* ke-I

*n* : Total *parent* yang ada

Setiap parent memiliki anak yang disebut dengan *identifier*. Berdasarkan gambar 1 yang menjadi *identifier* dari *parent* Belanja adalah Modern dan Tradisional. *Identifier* juga memiliki bobot yang diperoleh dari persamaan (2).

*Wind = frek* / n (2)

Dimana

*Wi*  : Bobot *parent* ke-I

*frek* :Jumlah kemunculan *indentifier*

*n* : Total *parent* yang ada

Setelah menemukan bobot *parent* dan *identifier* dapat dilakukan langkah selanjutnya membentuk *tree* dari *input* yang dimasukkan oleh pengguna. Berikut Gambar 2 merupakan contoh representasi *tree* input dengan kata kunci “Surabaya – Belanja – Modern”.



Gambar 2. Contoh *Tree* dari *Input*.

Langkah selanjutnya, menghitung kemiripan antara *tree* sistem dan *tree* *input* dari pengguna. Nilai kemiripan tiap pasangan *subtree* berada diantara interval [0,1]. Nilai 0 bermakna berbeda sedangkan nilai 0 bermakna sama atau identik. Kedalam (*depth*) dan lebar (*breadth*) *tree* tidak dibatasi. Algoritma penghitung kemiripan *tree* secara *rekursif* menjelajahi tiap pasang *subtree* dari atas ke bawah mulai dari kiri ke kanan. Algoritma mulai menghitung kemiripan dari bawah ke atas ketika mencapai *leaf node*. Nilai kemiripan tiap pasang *subtree* dilevel atas dihitung berdasar kepada kemiripan *subtree* dilevel bawahnya. Kontribusi bobot cabang juga diperhitungkan. Bobot dirata-rata kemudian dikalikan nilai kemiripan. Nilai kemiripan diperoleh secara *rekursif* berdasarkan kemiripan *leaf node* **[1]**. Perhitungan kemiripan *tree* dapat dihitung dengan persamaan berikut ini.

∑*( A(Si)(wi + wi’) /* 2 *)*

*A(Si)* = Nilai kemiripan *leaf node*.

*wi* = Bobot cabang pada *tree* sistem

*wi’*= Bobot cabang pada *tree input*

Pada kondisi nyata, proses pencocok *input* dan sistem tidak akan selalu menemukan kesamaan. *Input* yang dimasukkan oleh pengguna bisa secara acak, seperti “modern belanja” yang seharusnya “belanja modern”. Maka dari itu diperlukan proses pengecekan ketika input dimasukkan. Untuk mengecek input dari pengguna, dilakukan dengan menyeplit input menjadi beberapa kata. Berikut representasi contoh *input* yang acak.

*Input* : Modern Belanja

Seharusnya : Belanja Modern

Proses :

1. Membagi input menjadi beberapa kata kunci.

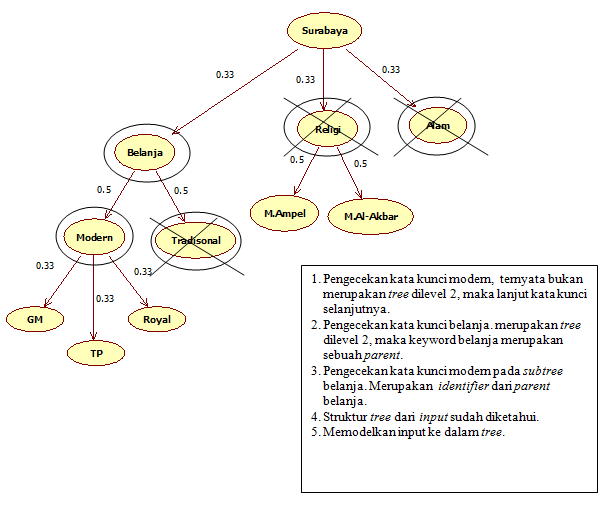
“Modern”(1), “Belanja”(2)

2. Melakukan pengecekan hasil split 1-2 dari *input* dengan *tree* sistem untuk

menentukan posisi *level* dari kata kunci.

3. Setelah setiap kata kunci di ketahui setiap *level*, maka *tree* *input* dapat di buat.

Berikut representasi pencarian struktur *tree input* melalui tree sistem.



Gambar 3. Contoh Representasi pencarian struktur *tree input*.

4. Sehingga *tree input* dapat di gambarkan sebagai berikut ini.



Gambar 4. Contoh Representasi struktur *tree input*.

Pada saat ini, banyak tempat wisata buatan baru yang dibuat oleh pemerintah daerah, sehingga belum terdaftar pada sistem. Dalam kondisi tersebut perlu dilakukan manajemen sistem agar wisata baru tersebut masuk kedalam sistem. Proses penambahan tempat wisata mempengaruhi struktur *tree* pada sistem, sehingga perlu dilakukan proses perhitungan ulang. Proses perhitungan ulang dilakukan setiap kali *tree* mengalami perubahan struktur dan bobot setiap cabangnya. Untuk perhitungan proses manajemen *tree* pada sistem dapat digunakan persamaan 1 dan 2. Gambar 3 merupakan contoh representasi *tree* pada sistem yang kategorinya bertambah dan tempat wisata “belanja modern” bertambah.



Gambar 5. Contoh Representasi *Tree* Sistem yang telah bertambah

Kategori dan Tempat Wisata Modern

# RINGKASAN ISI TUGAS AKHIR

Aplikasi yang dibagun merupakan aplikasi mobile berbasis android. Aplikasi akan menerima *input* kata kunci dari pengguna. Jika *input* terdiri dari 2 kata atau lebih maka *input* dipisah (split) menjadi beberapa kata, setiap kata merupakan kata kunci yang nantinya digunakan untuk membentuk node dari *tree input*. Setiap kata kunci dicari kedalaman (*depth*) *level* dengan cara mencocokkan dengan *tree* pada sistem (proses ini akan menentukan kata kunci merupakan *root, parent* atau *identifier*). Setelah struktur *tree* dari input terbuat, kemudian dilakukan pembobotan *tree*. Selanjutnya dilakukan perhitungan kemiripan antara *tree* pada sistem dengan *tree* dari *input*. Setelah menemukan hasil, aplikasi kemudian menampilkan hasilnya ke pengguna berupa daftar tempat wisata yang sesuai. Gambar 6 menunjukkan *arsitektur* sistem yang akan dibangun.



Gambar 5. *Arsitektur Sistem*

Keterangan :

Kalimat : Input dari pengguna

*Tree Input* : *Tree* dari *input* pengguna

*Tree Sistem* : *Tree* dari sistem

Untuk membantu memanjemen sistem, *server* dibuat dengan *versi* web. *Admin* bisa menambah, menghapus ataupun mengubah data pada sistem. Menambah jika ada kategori atau tempat wisata baru, menghapus jika menurut *admin* tempat wisata tersebut tidak sesuai dan mengubah jika *admin* salah melakukan penambahan atau tempat wisata ada informasi tambahan.

# METODOLOGI

## Penyusunan proposal tugas akhir

Proposal tugas akhir ini berisi rencana pembangunan sebuah perangkat lunak yang dapat membantu masyarakat dalam mencari informasi tentang tempat wisata didaerah.

Pada bab 3 telah jelaskan tentang latar belakang dibuat aplikasi pencarian tempat wisata berdasarkan daerah. Indonesia terbagi menjadi daerah-daerah. Setiap daerah memiliki banyak tempat wisata. Masyarakat belum mengetahui tentang tempat wisata didaerah. Mereka hanya tahu tempat-tempat wisata yang sudah terkenal. Pada bab 4 membahas tentang rumusan masalah yang harus dipecahkan. Pada bab 5 menjelaskan tentang batasan masalah. Pada bab 6 menjelaskan tentang tujuan pembuatan tugas akhir. Pada bab 7 menjelaskan manfaat tugas akhir. Pada bab 8 menjelaskan tentang tinjauan pustaka yang berisi metode yang digunakan dan perhitungan yang dilakukan. Pada bab 9 menjelaskan tentang ringkasan tugas akhir yang berisi arsitektur sistem. Pada bab 10 menjelaskan tentang metodologi yang berisi tahap-tahap pengerjaan tugas akhir. Pada bab 11 menjelaskan tentang waktu yang dibutuh dalam proses pembuatan aplikasi. Pada bab 12 menjelaskan tentang daftar pustaka.

## Studi Literatur

Studi literatur yang digunakan untuk menunjang pengerjaan tugas akhir ini adalah sebagai berikut ini.

1. Memodelkan data ke dalam *tree*.
2. Memodelkan *input* ke dalam *tree*.
3. Penggunaan metode *weighted tree* untuk menghitung kemiripan *tree.*
4. Penggunaan API google maps untuk petunjuk arah.
5. Pembuatan aplikasi android.

## Analisis dan Desain Perangkat Lunak

Analisis dan desain perangkat lunak yang dilakukan adalah berdasarkan rumusan masalah yang telah dijelaskan sebelumnya. Kebutuhan fungsional dari perangkat lunak adalah melakukan pencarian tempat wisata yang sesuai dengan *input* yang dimasukkan oleh pengguna dan menampilkan hasil ke pengguna. Selain itu aplikasi bisa menunjukkan arah dari tempat pengguna ke tempat wisata dengan google maps,.

## Implementasi Perangkat Lunak

Aplikasi yang dibangun merupakan aplikasi mobile yang berbasis android. Aplikasi akan menerima input berupa kata kunci dari pengguna dan hasilnya merupakan daftar tempat wisata yang sesuai dengan kata kunci yang dimasukkan. Adapun hal-hal yang diperlukan dalam implementasi sebagai berikut ini.

1. *NetBeans/Eclipse*

2. Java Development Kit (JDK)

3. Android SDK

4. Android Development Tool (ADT)

5. Android Virtual Device (AVD)

Untuk memudahkan memanajemen sistem, *server* di buat web yang hanya bisa di akses oleh *admin*. Admin bisa menambah, merubah dan menghapus tempat wisata ataupun kategori.

## Pengujian dan Evaluasi

Pengujian dan evaluasi perangkat lunak ini dilakukan secara langsung pada perangkat android. Pengguna mencari tempat wisata yang diinginkan dengan memasukkan kata kunci pada aplikasi dan kemudian aplikasi menampilkan daftar tempat wisata yang sesuai dengan kata kunci. Tujuan dari tahap ini untuk mendapatkan kesesuaian atara perancangan awal dengan aplikasi yang dibuat .

## Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan
   1. Latar Belakang
   2. Rumusan Masalah
   3. Batasan Tugas Akhir
   4. Tujuan
   5. Metodologi
   6. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

# JADWAL KEGIATAN

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tahapan |  | | | | Bulan (Tahun 2014) | | | | | | | | | | | | | |
| Maret | | | | | April | | | | Mei | | | | Juni | | | |
| Analisa kebutuhan dan studi literature |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |
| Perancangan system |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementasi |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |
| Uji coba dan evaluasi |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |
| Penyusunan buku |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |

# DAFTAR PUSTAKA

x

|  |  |
| --- | --- |
| [1] | Riyanarto Sarno, and Faisal Rahutomo, "Penerapan Algoritma Weighted Tree Similarity untuk Pencarian Semantik," Januari 2008 . |
| [2] | Lu Yang, Biplab K. Sarker, and Harold Boley., "A Weighted Tree Simplicity Algorithm for Similarity Matching of Partial Product Descriptions". |
| [3] | Virendrakumar C. Bhavsar, Harold Boley and Lu Yang, "A Weigthed Tree Similarity Algorithm for Multi Agent System in E-Business Environments ,". |
| [4] | Yosephine Halim, Ramos Somya and Charitas Fibriani, "Algoritma Extended Weighted Tree Similarity untuk memberikan Solusi Memasak pada J2ME," 2012. |
| [5] | http://bungkusdah.com/tahun-2014-android-kian-mendominasi/ |
| [6] | http://id.wikipedia.org/wiki/Daftar\_tempat\_wisata\_di\_Jawa |
| [7] | http://wisata.tokobunganusantara.com/jawa-timur/html |
| [8] | I. Sommerville, Software Engineering Ninth Edition, United States of America: Pearson Education Inc., 2011 |

x