**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 operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh. Awalnya Android dikembangkan oleh Android.inc dan kemudian dibeli oleh Google pada tahun 2005. Dalam perkembangannya sampai tahun 2013, sistem operasi Android sangat besar. Pada tahun 2014, perkembangan Android diperkirakan akan semakin menggurita. Sistem operasi Android akan menguasai semua perangkat dengan persentase 40%. Pada tahun 2013 sistem operasi Android sudah mendominasi pangsa pasar sekitar 38%. Tahun 2014 diperkirakan perkembangan sistem operasi 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 [4].

Objek wisata di Jawa Timur sangatlah banyak [5], seperti Wisata Flora, Wisata Museum, Wisata Belanja, Wisata Religi, Wisata Sejarah dan wisata-wisata yang lain. Semua objek wisata tersebut tersebar diseluruh daerah Jawa Timur. Banyaknya tempat wisata di Jawa Timur belum banyak diketahui oleh masyarakat, mereka hanya mengetahui tentang tempat wisata yang terkenal saja seperti wisata di Malang, padahal wisata di daerah 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*) [1].

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

# 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 dari *input* pengguna kedalam sebuah *tree*?
3. Bagaimana cara menggunakan metode *weighted tree* untuk membandingkan *tree* dari *input* pengguna dengan *tree* dari sistem?
4. Bagaimana cara manajemen (menambah *node* (kata kunci pencarian), menghapus dan mengubah) *tree* pada 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, diantaranya 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

Adapun 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 daerah

Jawa Timur.

# MANFAAT TUGAS AKHIR

Manfaat dari Tugas Akhir ini adalah membantu masyarakat dalam mencari informasi tempat wisata di 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 [3].

## 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 satu atau lebih *parent* yang masing-masing dapat memiliki atribut berupa bobot serta nama *parent*. Setiap *parent* terdiri *identifier*, yang mempunyai atribut bobot [1]. 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 dijumlahkan akan berniai 1. Bobot masing-masing *parent* diperoleh dari Persamaan 1.

*Wi =* 1 / n (1)

Keterangan:

*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)

Keterangan:

*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] [2]. Nilai 0 bermakna berbeda sedangkan nilai 1 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* di *level* atas dihitung berdasarkan kemiripan *subtree* di *level* bawahnya. Kontribusi bobot cabang juga diperhitungkan. Bobot dirata-rata kemudian dikalikan nilai kemiripan. Nilai kemiripan diperoleh secara *rekursif* berdasarkan kemiripan *leaf node* [2]. Perhitungan kemiripan *tree* dapat dihitung dengan Persamaan 3.

∑*( A(Si)(wi + wi’) /* 2 *)* (3)

Keterangan:

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

*wi* = Bobot cabang pada *tree* sistem

*wi’*= Bobot cabang pada *tree input*

Pada kondisi nyata, proses pencocokan *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 membagi *input* menjadi beberapa kata. Berikut representasi contoh *input* yang acak.

*Input* : Modern Belanja

Seharusnya : Belanja Modern

Proses:

1. Membagi *input* menjadi beberapa kata.

“Modern”(1), “Belanja”(2)

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

menentukan posisi *level* dari kata kunci.

3. Setelah setiap kata diketahui setiap *level*, maka *tree* dari *input* dapat dibuat.

Gambar 3 menunjukkan proses pencarian struktur *tree* dari *input* melalui *tree* dari sistem.



1. Pengecekan kata kunci “Modern”, ternyata

bukan merupakan *tree* di *level* 2, maka

lanjut kata kunci selanjutnya.

2. Pengecekan kata kunci “Belanja”.

merupakan *tree* di *level* 2, maka kata kunci

“Belanja” merupakan sebuah *parent*.

3. Pengecekan kata kunci “Modern” pada

*subtree* Belanja. Merupakan *identifier*

dari *parent* Belanja.

4. Struktur *tree* dari *input* sudah diketahui.

5. Memodelkan *input* ke dalam *tree* dan

menghitung pembobotannya.

Gambar 3. Contoh Proses Pencarian Struktur *Tree* dari *Input*.

Gambar 4 menunjukkan struktur *tree* dari *input* yang telah terbuat.



Gambar 4. Contoh Struktur *Tree* dari *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 di setiap cabangnya. Untuk perhitungan proses manajemen *tree* pada sistem dapat digunakan persamaan 1 dan 2. Gambar 5 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* dibagi 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 arsitektursistem yang akan dibangun.



Gambar 6. Arsitektur Sistem

Keterangan:

Kalimat : *Input* dari pengguna

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

*Tree Sistem* : *Tree* dari sistem

Untuk membantu manajemen sistem, *server* dibuat dengan berbasis *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 berbasis *mobile* yang dapat membantu masyarakat dalam mencari informasi tentang tempat wisata di daerah.

Pada bab 3 telah dijelaskan tentang latar belakang dibuat aplikasi pencarian tempat wisata berdasarkan daerah. Masyarakat Jawa Timur belum mengetahui tempat wisata yang ada di Jawa Timur. 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 dibutuhkan 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 hasilnya ke pengguna. Selain itu aplikasi bisa menunjukkan arah dari tempat pengguna ke tempat wisata melalui 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 manajemen sistem, *server* dibuat *web* yang hanya bisa digunakan oleh *admin*. *Admin* bisa menambah, merubah, 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

Tabel 1 merupakan penjelasan waktu yang dibutuhkan untuk mengerjakan Tugas Akhir.

Tabel 1. Waktu yang Dibutuhkan untuk Mengerjakan Tugas Akhir

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

# DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [1] | Y. Halim, R. Somya and C. Fibriani, "Algoritma Extended Weighted Tree Similarity untuk Memberikan Solusi Memasak pada J2ME," *JdC, Vol. 1, No 1,,* pp. 1-11, 2012. |
| [2] | R. Sarno and F. Rahutomo, "PENERAPAN ALGORITMA WEIGHTED TREE SIMILARITY," *JUTI,* pp. 35-42, Volume 7, Nomor 1, Januari 2008. |
| [3] | I. Sommerville, Software Engineering, 9th Edition, United States of America: Addison-Wesley, 2010. |
| [4] | K. Dwi, "Tahun 2014 Android Kian Mendominasi," 2013. [Online]. Available: http://bungkusdah.com/tahun-2014-android-kian-mendominasi/. [Accessed 3 Maret 2014]. |
| [5] | TokoBunga, "JAWA TIMUR," [Online]. Available: http://wisata.tokobunganusantara.com/jawa-timur/html. [Accessed 3 Maret 2014]. |