

LAPORAN TUGAS AKHIR

APLIKASI SPEEDTESTER BERBASIS ANDROID

**DIAJUKAN SEBAGAI SALAH SATU SYARAT KELULUSAN DI JURUSAN TEKNIK
INFORMATIKA**



Disusun Oleh :

Nama : Endry Haryono

NIM : 08523175

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

YOGYAKARTA

2014

LEMBAR PENGESAHAN PEMBIMBING

APLIKASI SPEEDTESTER BERBASIS ANDROID

TUGAS AKHIR

Oleh :

Nama : ENDRY HARYONO

NIM : 08523175

Yogyakarta, Juli 2014

Pembimbing Tunggal,

Ahmad M. Raf'ie Pratama, ST., MIT.

LEMBAR PERNYATAAN KEASLIAN

HASIL TUGAS AKHIR

Saya yang bertanda tangan dibawah ini :

Nama : Endry Haryono

NIM : 08 523 175

Menyatakan bahwa seluruh komponen dan isi dalam laporan tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti bahwa ada beberapa bagian dari karya ini adalah bukan karya saya sendiri, maka saya siap menanggung resiko dan konsekuensi apapun.

Demikian pernyataan ini kami buat tanpa ada paksaan dari pihak manapun, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, Juli 2014

(Endry Haryono)

LEMBAR PENGESAHAN PENGUJI
APLIKASI SPEEDTESTER BERBASIS ANDROID

TUGAS AKHIR

Oleh :

Nama : Endry Haryono

NIM : 08523175

Telah dipertahankan didepan Sidang Penguji sebagai Salah Satu Syarat Untuk Memeperoleh
Gelar Sarjana Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia.
Yogyakarta, Juli 2014

Tim Penguji

Ahmad M. Rafi'e Pratama, ST., MT. -----

Ketua

Affan Mahtarami S.Kom., MT. -----

Anggota 1

Lizda Iswari ST., M.Sc. -----

Anggota 2

Mengetahui,

**Ketua Program Studi Teknik Informatika Fakultas Teknologi Industri
Universitas Islam Indonesia**

Yudi Prayudi, S.Si., M.Kom

HALAMAN PERSEMBAHAN

1. Allah SWT. atas segala nikmat, hidayah, dan karunia-Nya.
2. Nabi Muhammad SAW. yang sebagai tauladan, teladan dan peyelamat umat manusia.
3. Ucapan spesial untuk orang-orang tercinta:
 - a) Bapak saya Sudimaryono.
 - b) Ibu saya Sri Endang Sugihartini.
 - c) Kakak kandung saya Eddy Handhoko.
 - d) Kakak kandung saya Enny Herawati.
 - e) Kakak Ipar saya Santi..
 - f) Kakak Ipar saya danu wicaksono.
 - g) Kakak sepupu saya Mas Nanang.yang terus menerus memberikan doa, dukungan, motivasi untuk keberhasilan Saya.
4. Keluarga besar yang juga terus memberikan semangat dan motivasi sehingga bisa terus maju melangkah lebih jauh.
5. Kawan-kawan Dwi Sigit, Dandy, Deddy, Ahmad, Dimas, Rhio, Dayyan, Wicak, Fuad, Ocky dan teman-teman yang tidak bisa disebutkan satu per satu, yang juga membantu memberikan dukungan, motivasi, dan doa.
6. Teman-teman dan keluarga besar KKN Unit 58 Angkatan 42, kalian benar-benar menginspirasi.
7. Daerah Istimewa Yogyakarta dengan segala hal kenangan yang akan saya kenang.

MOTTO

"Allah itu tunggal"

"Lihatlah mereka yang lebih tidak beruntung dari pada dirimu, sehingga kau tidak mungkin tidak berpuas diri atas keberuntungan yang diberikan Allah kepadamu "

(Nabi Muhammad SAW)

"Perubahan memang tidak menjamin kebahagiaan, tapi tidak ada

Kebahagiaan yang bias dicapai tanpa perubahan"

(Mario Teguh)

"Hidup adalah soal keberanian, menghadapi yang tanda tanya tanpa kita mengerti,

tanpa bisa kita menawar, terimalah dan hadapilah "

(Soe Hok Gie)

"Semua belum berakhir hingga benar-benar berakhir "

"Bila hal itu sebuah kebaikan maka jalankanlah, dan sebaliknya"

(Endry Haryono)

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Puji syukur penyusun panjatkan kehadiran Allah SWT, Tuhan Yang Maha Esa yang telah memberikan rahmat serta hidayah-Nya sehingga dapat menyelesaikan tugas akhir yang berjudul Aplikasi Speedtester Berbais Android.

Laporan tugas akhir ini dibuat sebagai salah satu syarat untuk memperoleh gelar sarjana Teknik Informatika Universitas Islam Indonesia dan juga sebagai salah satu sarana untuk mempraktekkan secara langsung ilmu yang telah diperoleh selama masa studi di Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.

Tugas akhir ini dapat terselesaikan dengan tepat waktu berkat bimbingan, dukungan, dan persetujuan dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada :

1. Allah SWT atas karunia dan rahmat-Nya, penulis dapat menyelesaikan tugas akhir ini dengan baik. Semoga bermanfaat bagi orang lain suatu hari nanti.
2. Orang Tua dan Keluarga tercinta, atas dukungan dan doanya.
3. Bapak Ahmad Munasir Raf'ie Pratama S.T., M.I.T. selaku dosen pembimbing tugas akhir, yang telah memberikan arahan, saran, dan dorongan semangat kepada penulis sehingga dapat secepatnya menyelesaikan kerja praktek ini
4. Sahabat-sahabat selama kuliah Dwi Sigit, Dandy, Deddy, Ahmad, Dimas, Rhio, Wicak, Ocky.
5. Rekan- rekan mahasiswa Teknik Informatika UII khususnya angkatan 2008.
6. Semua pihak yang telah berjasa yang telah membantu pelaksanaan tugas akhir ini sehingga tugas akhir ini dapat berjalan dengan baik.

Penulis menyadari masih banyak kekurangan dalam penyelesaian tugas akhir, baik dalam pelaksanaan maupun dalam penulisan laporan ini, oleh karena itu penulis juga mengharapkan kritik dan saran yang membangun sebagai bahan evaluasi dan pembelajaran agar dapat lebih baik dikemudian hari.

Wassalamu'alaikum Wr. Wb.

Yogyakarta, 1 Juli 2014

Endry Haryono

ABSTRAKSI

Beberapa tahun belakangan ini mobile atau *smartphone* berbasis android sedang marak-maraknya diperbincangkan karena banyak memiliki fitur-fitur pendukung seperti GPS, Map, internet, wi-fi yang bersifat *opensource* yang bisa dikembangkan di smarphone berbasis android. Pada makalah ini, disajikan sebuah aplikasi Speedtester Berbasis Android yang dibuat untuk memudahkan pengguna untuk mendapatkan informasi mengenai speedometer dimana aplikasi ini memberikan hasil informasi berupa mencatat kecepatan, jarak, dan waktu. Aplikasi ini bersifat *fleksible* dalam hal pemakaiannya tergantung kapan user ingin menggunakan aplikasi ini seperti saat anda berjalan, berlari, bersepeda, dan sedang memakai kendaraan bermotor atau melakukan hal lain di luar ruangan yang secara khusus dikembangkan pada smartphone berbasis android. Aplikasi ini memiliki 5 fitur utama, speedometer yang menampilkan informasi berupa kecepatan, jarak, dan waktu, statistika menampilkan informasi hasil akhir setelah aplikasi selesai di jalankan berupa kecepatan terakhir saat aplikasi ini di gunakan, jarak total, waktu tempuh, kecepatan rata-rata, dan kecepatan maksimal, grafik kecepatan menampilkan gambar grafik selama proses penghitungan kecepatan berlangsung, grafik jarak menampilkan gambar grafik selama proses penghitungan jarak berlangsung, history menampilkan data selama aplikasi ini berjalan baik data terkini maupun data yang lalu. Aplikasi ini telah diimplementasikan, diujikan, dan dapat diketahui dapat berjalan dengan baik pada berbagai smartphone android mulai dari versi 2.2 hingga versi 4.2

Kata kunci : Online, Speedometer, GPS, Android.

TAKARIR

<i>Interface</i>	Antarmuka
<i>Mobile</i>	Dapat dibawa kemana-mana
<i>Open Source</i>	Perangkat lunak yang kodenya dapat dimodifikasi dengan bebas
<i>Paper</i>	Sebuah jurnal ilmiah sederhana
<i>Input</i>	Masukan
<i>Output</i>	Keluaran
<i>Use Case</i>	pemodelan sistem
<i>Activity Diagram</i>	Penggambaran aktivitas dalam sistem
<i>HIPO</i>	Hierarchy plus Input-Proccess-Output
<i>VTOC</i>	Visual Table of Countents
<i>Up-to-date</i>	Pembaruan
<i>Overview Diagram</i>	Diagram Ringkasan
<i>Smartphone</i>	Ponsel pintar
<i>Device</i>	Perangkat
<i>User</i>	Pengguna
<i>User-friendly</i>	Mudah digunakan pengguna
<i>Image</i>	Gambar
<i>About</i>	Informasi aplikasi
<i>Online</i>	Terhubung jaringan internet
<i>Libraries</i>	Kumpulan komponen perangkat lunak
<i>Debugger</i>	Program computer untuk uji coba
<i>Emulator</i>	Perangkat virtual uji coba program

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN PEMBIMBING	ii
LEMBAR PERNYATAAN KEASLIAN	iii
HALAMAN PEENGESAHAN PENGUJI	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTTO	iv
KATA PENGANTAR.....	vii
ABSTRAKSI	ix
TAKARIR.....	x
DAFTAR ISI.....	xi
DAFTAR TABEL	xiv
DAFTAR GAMBAR.....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	3
1.6 Review Aplikasi Sejenis.....	3
1.7 Metodologi Penelitian.....	4
1.7.1 Pengumpulan Data.....	4
1.7.2 Pengembangan Sistem	4
1.7.3 Pengujian Sistem.....	5
1.8 Sistematika Penulisan	5
BAB II LANDASAN TEORI.....	7
2.1 Speedometer.....	7
2.2 Global Positioning System (GPS).....	8

2.3 Cara Kerja GPS.....	9
2.4 Android	11
2.4.1 Kelebihan Android.....	13
2.4.2 Kekurangan Android.....	14
2.5 Aplikasi Sejenis	14
 BAB III METODOLOGI PENELITIAN	 17
3.1 Analisis Kebutuhan.....	17
3.1.1 Analisis Kebutuhan Antar Muka	17
3.1.2 Analisis Kebutuhan Anatar Masukan	18
3.1.3 Analisis Kebutuhan Proses	18
3.1.4 Analisis Kebutuhan Output.....	18
3.2 Analisis Kebutuhan Perangkat Keras.....	19
3.3 Analisis Kebutuhan Lunak.....	19
3.4 Tahap Pengembangan Sistem	20
3.4.1 Gambaran Umum Sistem.....	20
3.4.2 Use Case Diagram	21
3.4.3 Diagram Aktivitas.....	22
3.4.4 Diagram HIPO	23
3.4.4.1 VTOC.....	23
3.4.4.2 Overview Diagram.....	24
3.4.4.3 Detail Diagram	25
3.5 Diagram Kelas	26
3.6 Perancangan Basis Data.....	26
3.7 Perancangan Antarmuka	28
3.7.1 Perancangan Halaman Speedometer.....	28
3.7.2 Perancangan Halaman Statistika.....	29
3.7.3 Perancangan Halaman Grafik Kecepatan dan Grafik Jarak.....	29
3.7.4 Perancangan Halaman History.....	30
3.8 Rencana Pengujian.....	30

BAB IV IMPLEMENTASI DAN ANALISIS	32
4.1 Batasan Implementasi	32
4.1.1 Perangkat Lunak	32
4.1.2 Perangkat Keras	33
4.2 Implementasi.....	34
4.2.1 Implementasi Pembuatan Aplikasi	34
4.2.2 Implementasi Tampilan Antarmuka	34
4.2.2.1 Halaman Splash	35
4.2.2.2 Menu Pilihan Speedometer	36
4.2.2.3 Menu Pilih Statistika.....	46
4.2.2.4 Menu Pilih Grafik Kecepatan	48
4.2.2.5 Menu Pilih Grafik Jarak.....	49
4.2.2.6 Menu Pilih History.....	49
4.3 Pengujian Pada <i>Smartphone</i> Berbasis Android	50
4.3.1 Perangkat Yang Diujikan.....	50
4.4 Hasil Pengujian	50
 BAB V KESIMPULAN DAN SARAN.....	 52
5.1 Kesimpulan	52
5.2 Saran	52
 DAFTAR PUSTAKA.....	 xvii

DAFTAR TABEL

Tabel 3.1 Tabel Kecepatan	27
Tabel 3.2 Tabel History	27
Tabel 3.3 Relasi Antar tabel.....	28
Tabel 3.4 Pengujian Sistem.....	31
Tabel 4.1 Perangkat Uji	49
Tabel 4.2 Hasil Pengujian	50

DAFTAR GAMBAR

Gambar 2.1 Gambar Digital Speedometer.....	15
Gambar 2.2 Gambar Simple Speedo.....	16
Gambar 2.3 Gambar Simple Speedo.....	16
Gambar 3.1 Gambaran Umum Sistem.....	20
Gambar 3.2 Use Case Diagram.....	22
Gambar 3.3 Diagram Aktivitas.....	23
Gambar 3.4 Visual Table of Content.....	24
Gambar 3.5 Overview Diagrams.....	25
Gambar 3.6 Detail Diagrams.....	25
Gambar 3.7 Diagram Kelas.....	26
Gambar 3.8 Perancangan Halaman Utama.....	29
Gambar 3.9 Rancangan Halaman Statistika.....	29
Gambar 3.10 Rancangan Halaman Grafik.....	30
Gambar 3.11 Rancangan Halaman History.....	30
Gambar 4.1 Halaman Splash.....	34
Gambar 4.2 Menu Speedometer.....	36
Gambar 4.3 Pilih Menu Statistik.....	46
Gambar 4.4 Tampil Halaman Grafik Kecepatan.....	48
Gambar 4.5 Tampil Halaman Grafik Jarak.....	49
Gambar 4.6 Tampil Halaman History.....	49

DAFTAR PUSTAKA

- Pramusinto, Wahyu. (2011). Perkembangan Sistem Operasi Android. <http://away.web.id/perkembangan-sistem-operasi-android/> (diaksesFebruari 2013)
- Wawan. (2011). Sejarah Android. <http://wawancoys.com/sejarah-android> (diaksesFebruari 2013)
- Wawan. (2011). Kelebihan Dan Kekurangan Android. <http://wawancoys.com/kelebihan-dan-kekurangan-android>(diaksesFebruari 2013)
- Fatimah, W. N. (2011). Pengenalan eclipse. wi01.files.wordpress.com/2011/02/pengenalan-eclipse.pdf(diaksesMaret 2013)
- Sumarna, Agus (2011) Pengenalansqlite<http://ri32.wordpress.com/2011/08/27/pengenalan-sqlite/> (diaksesMaret 2013)
- Masjito. (2009, 15 Mei), “*Cara Menggambar Diagram Use Case.*” <http://www.masjito.com/index.php?pilih=news&mod=yes&aksi=lihat&id=40> (diaksesFebruari 2013)
- Safaat H, Nazruddin. 2011. Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android, Penerbit Informatika.
- Akbarul Huda, Arif. 2012. *24 Jam!! Pintar Pemrograman Android*. Yogyakarta

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Kemajuan teknologi komputer sebagai pendukung pemrosesan data dan informasi menjadi hal yang wajib bagi suatu instansi maupun personal dalam manajemen keperluannya dalam kehidupan sehari-hari. Namun semakin berkembangnya teknologi, komputer sebagai media pendukung pemrosesan data dan informasi telah mengalami perubahan yang semula berukuran yang cukup praktis misalnya perangkat *mobile* maupun *gadget*. Meskipun ukuran yang cukup kecil *mobile* maupun *gadget* memiliki kemampuan yang tidak kalah dengan komputer pada umumnya dalam pemrosesan data dan informasi, hal ini menjadikan perangkat tersebut populer di era saat ini.

Android adalah sistem operasi untuk telepon seluler berbasis linux yang dikembangkan oleh Google Inc. Dibandingkan sistem operasi lain, android memiliki keunggulan sebagai software yang memakai basis kode komputer yang biasa didistribusikan secara terbuka, sehingga hal ini memungkinkan para pengembang perangkat lunak untuk mempelajarinya dan ikut mengembangkan aplikasi berbasis android atau bahkan mengembangkan sistem operasi itu sendiri dengan lebih mudah. Perangkat Smartphone yang memakai sistem operasi android biasanya dilengkapi fitur-fitur yang begitu canggih, seperti akses internet kecepatan tinggi, GPS.

Aplikasi ini dibuat untuk memudahkan pengguna untuk mendapatkan informasi mengenai speedometer dimana aplikasi ini memberikan hasil informasi berupa mencatat kecepatan, jarak, dan waktu. Aplikasi ini bersifat fleksibel dalam penggunaannya, user bisa menggunakan aplikasi ini saat berjalan, berlari, bersepeda, dan sedang memakai kendaraan bermotor atau melakukan hal lain di luar ruangan. Saat aplikasi ini selesai dijalankan, anda bisa melihat kecepatan

maksimal, kecepatan rata-rata, jarak, waktu. Aplikasi ini ditambah juga fitur berupa grafik sehingga user bisa melihat data progress aplikasi ini berjalan. Aplikasi ini juga dtambahkan fitur history sehingga masih bisa melihat data dari perjalanan sebelumnya.

1.2 RUMUSAN MASALAH

Rumusan masalah dari latar belakang di atas adalah bagaimana cara mengetahui kecepatan maksimum saat anda berjalan, berlari, bersepeda, dan kendaraan bermotor melalui fasilitas GPS. Selain itu bagaimana menampilkan jarak tempuh dari posisi awal ke posisi tujuan.

1.3 BATASAN MASALAH

Batasan masalah yang diperlukan dalam membangun aplikasi ini adalah:

1. Aplikasi ini hanya dapat berjalan pada smartphone berbasis android minimal froyo , untuk versi lebih tinggi kemungkinan aplikasi berjalan lancar.
2. Aplikasi ini menggunakan satuan kecepatan m/s dan KM/jam.
3. Informasi bisa disimpan secara permanen.
4. Informasi tentang kecepatan rata-rata dan kecepatan maksimal hanya bisa diketahui ketika fitur speedometer selesai dijalankan.
5. Data yang ditampilkan pada menu History merupakan 10 data terakhir pada saat aplikasi ini terakhir di jalankan.
6. Jika GPS tidak berjalan dengan baik atau mati maka data tidak akurat.
7. Setiap Handphone memiliki tingkat keakuratan yang berbeda dalam penggunaan GPS.

1.4 TUJUAN PENELITIAN

Tujuan dari penelitian ini adalah membangun suatu aplikasi yang diharapkan dapat membantu user untuk mengetahui kecepatan, jarak serta waktu tempuh ketika sedang melakukan perjalanan.

1.5 MANFAAT PENELITIAN

Adapun manfaat yang dapat diperoleh dari penelitian ini yaitu:

1. Melatih pola pikir mahasiswa dari mengumpulkan, menganalisa, menarik kesimpulan sampai mencari solusi dari suatu masalah.
2. Membantu masyarakat untuk mengetahui kecepatan, jarak, serta waktu yang ditempuh selama dalam perjalanan.
3. Aplikasi ini bersifat fleksibel bagi user karena bisa dipergunakan pada saat apapun yang berhubungan dengan melakukan perjalanan seperti jalan kaki, lari, bersepeda, ataupun saat memakai kendaraan bermotor seperti sepeda motor ataupun mobil.

1.6 REVIEW PENELITIAN SEJENIS

Aplikasi ini sejenis dengan aplikasi “Digital Speedometer” berbasis android dan merupakan acuan penelitian untuk membangun aplikasi ini. “Digital Speedometer” berbasis android ini terdapat beberapa fitur berupa penghitungan kecepatan, waktu tempuh, jarak, dan kecepatan rata-rata. Pada saat aplikasi ini baru dijalankan maka akan membutuhkan waktu sesaat untuk bisa menggunakan aplikasi ini di karenakan harus menunggu penyesuaian GPS.

Dari penelitian di atas, aplikasi digital speedometer saat ini sangat di bantu oleh perkembangan teknologi terutama dengan adanya GPS di mana sangat membantu dalam pengukuran kecepatan. Khususnya yang berbasis mobile sehingga sangat memudahkan user dalam mendapatkan informasi terkini hanya dengan menggunakan alat komunikasi atau alat elektronik yang terkoneksi dengan *internet* maka bisa dengan mudah dalam mendapatkan informasi yang dibutuhkan.

Dengan adanya aplikasi seperti di atas sangat membantu masyarakat saat ini apalagi dengan penggunaan *mobile* yang dapat dibawa kemana-mana sangat memudahkan masyarakat dalam menggunakan aplikasi ini. Ini menjadi salah satu hal yang mendorong peneliti membuat aplikasi sejenis dengan menambahkan beberapa kelebihan.

1.7 METODE PENELITIAN

Metode yang digunakan dalam penelitian ini meliputi tahap pengumpulan data dan pembuatan perangkat lunak.

1. 7. 1. Tahap Pengumpulan Data

Tahap pengumpulan data yang diperlukan adalah:

1. Observasi

Observasi dilakukan untuk mendapatkan informasi tentang bagaimana proses aplikasi ini berjalan serta informasi apa saja yang akan di dapat.

2. Studi Pustaka

Pengumpulan data dilakukan dengan cara mengumpulkan literatur, buku, jurnal, dan *paper* yang berkaitan dengan judul skripsi.

1. 7. 2. Tahap Pengembangan Sistem

1. *Requirements*

Merupakan bagian dari sistem yang terbesar dalam pengerjaan suatu proyek, dimulai dengan menetapkan berbagai kebutuhan dari semua elemen yang diperlukan sistem.

2. *Analisis*

Tahap menganalisis hal-hal yang diperlukan dalam pelaksanaan proyek pembuatan perangkat lunak.

3. *Design*

Implementasi data yang telah dianalisis ke bentuk yang *user-friendly*.

4. *Coding*

Tahap implementasi data atau pemecahan masalah yang telah dirancang menggunakan bahasa pemrograman tertentu.

5. *Testing*

Merupakan tahap untuk pengujian terhadap perangkat lunak yang dibangun.

6. *Maintenance*

Tahapan dimana suatu perangkat lunak yang telah dibangun mengalami perubahan dan penambahan.

1. 7. 3. Pengujian Sistem

Pada tahap pengujian sistem ini dilakukan pengujian terhadap aplikasi yang dibangun dengan cara mengimplementasikan ke dalam smartphone berbasis android. Hal ini dilakukan dengan tujuan mengetahui apakah sistem tersebut dapat berjalan dengan baik sesuai harapan.

1.8 SISTEMATIKA PENULISAN

Bab I Pendahuluan, berisikan pembahasan masalah umum yang meliputi latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

Bab II Landasan Teori, memuat konsep dasar dan teori-teori yang berkaitan dengan topik penelitian yang akan dilakukan dan hal-hal yang nantinya akan digunakan dalam proses analisis permasalahan yang berkaitan.

Bab III Metodologi, pada bab ini memberikan uraian tentang langkah-langkah penyelesaian masalah yang ada pada tugas akhir. Langkah-langkah tersebut berisi penjelasan tentang metode analisis kebutuhan perangkat lunak, metode analisis yang digunakan serta hasil analisis, masukan sistem (input), keluaran sistem (output), kebutuhan perangkat lunak, kebutuhan perangkat keras, antarmuka (interface) yang dibutuhkan dan fungsi yang diharapkan. Selain itu pada bagian ini memuat metode perancangan sistem yang terdiri dari *Use case*, *Activity Diagram*, *Class Diagram*, HIPO. Selain itu juga memuat , relasi tabel dan rancangan antarmuka.

Bab IV Implementasi Dan Analisis, bab ini menjelaskan implementasi perangkat lunak dan analisis kerja perangkat lunak. Implementasi perangkat lunak meliputi tampilan antarmuka, bahasa yang digunakan, prosedur bahasa pemrograman, basis data, pengujian dan penjelasan fungsi-fungsi dimana fungsi-fungsi tersebut merupakan tahap lanjutan dari proses menjadi suatu perangkat lunak dari sejumlah menu, proses, tampilan masukan (*input*) dan tampilan keluaran (*output*) yang mewakili dari pengolahan data. Analisis kinerja perangkat lunak, menjelaskan hasil analisa dan pengujian terhadap sistem yang dibangun.

Bab V Kesimpulan dan Saran, berisikan kesimpulan yang merupakan

rangkuman dari hasil analisis kinerja pada bagian sebelumnya dan saran yang perlu diperhatikan berdasarkan keterbatasan yang ditemukan dan asumsi-asumsi yang dibuat selama pembuatan aplikasi.

BAB II

LANDASAN TEORI

2.1 Speedometer

Speedometer adalah alat pengukur kecepatan kendaraan darat, yang merupakan perlengkapan standar setiap kendaraan yang beroperasi di jalan. Speedometer diciptakan oleh Josip Belušić dari Kroasia pada tahun 1888, dan pada awalnya disebut *velocimeter*. Sekarang universal dipasang untuk kendaraan bermotor, mereka mulai akan tersedia sebagai pilihan pada 1900-an, dan sebagai perlengkapan standar dari sekitar 1910 dan seterusnya speedometer untuk kendaraan lainnya. Memiliki nama khusus dan menggunakan cara lain penginderaan kecepatan. Untuk perahu, ini adalah *log pit*. Untuk pesawat terbang, ini merupakan *airspeed indicator*.

Speedometer berfungsi agar pengemudi mengetahui kecepatan kendaraan yang dijalkannya dan dijadikan informasi utama untuk mengendalikan kecepatan dikawasan/jalan agar tidak terlalu lambat atau terlalu cepat, bisa mengatur waktu perjalanan dan mengendalikan kecepatan di jalan yang kecepatannya dibatasi.

Ada beberapa jenis sensor pengukuran kecepatan :

1. **Mekanis**, adalah perangkat pengukur kecepatan yang dihubungkan langsung dengan roda depan ataupun transmisi dengan menggunakan suatu kabel yang ikut berputar saat kendaraan bergerak, gerakan berputar ini kemudian diubah untuk menggerakkan jaruk kecepatan.
2. **Elektronik**, adalah pengukur kecepatan yang bekerja atas dasar sensor yang ditempatkan di poros penggerak kendaraan yang mendeteksi jumlah putaran poros untuk selanjutnya data dikirim ke speedometer dengan prinsip *arus*

Eddy yang menggerakkan jarum kecepatan ataupun menunjukkan kecepatan secara digital.

3. **GPS**, adalah perangkat pengukur kecepatan yang menggunakan perubahan data posisi koordinat bumi yang diperoleh dari satelit GPS yang diolah oleh *processor* menjadi informasi kecepatan.

2.2 *Global Positioning System (GPS)*

Global Positioning System atau dalam bahasa indonesianya Sistem Pemosisi Global (GPS)) adalah sistem untuk menentukan letak di permukaan bumi dengan bantuan penyelarasan (*synchronization*) sinyal satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke Bumi. Sinyal ini diterima oleh alat penerima di permukaan, dan digunakan untuk menentukan letak, kecepatan, arah, dan waktu. Sistem yang serupa dengan GPS antara lain GLONASS (Rusia), Galileo (Uni Eropa), IRNSS (India).

Sistem ini dikembangkan oleh Departemen Pertahanan Amerika Serikat, dengan nama lengkapnya adalah **NAVSTAR GPS** (kesalahan umum adalah bahwa NAVSTAR adalah sebuah singkatan, ini adalah salah, NAVSTAR adalah nama yang diberikan oleh John Walsh, seorang penentu kebijakan penting dalam program GPS). Kumpulan satelit ini diurus oleh 50th Space Wing Angkatan Udara Amerika Serikat. Biaya perawatan sistem ini sekitar US\$750 juta per tahun, termasuk penggantian satelit lama, serta riset dan pengembangan.

GPS Tracker atau sering disebut dengan **GPS Tracking** adalah teknologi AVL (Automated Vehicle Locator) yang memungkinkan pengguna untuk melacak posisi kendaraan, armada ataupun mobil dalam keadaan Real-Time. GPS Tracking memanfaatkan kombinasi teknologi GSM dan GPS untuk menentukan koordinat sebuah obyek, lalu menerjemahkannya dalam bentuk peta digital.

2.3 Cara Kerja GPS

Sistem ini menggunakan sejumlah satelit yang berada di orbit bumi, yang memancarkan sinyalnya ke bumi dan ditangkap oleh sebuah alat penerima. Ada tiga bagian penting dari sistem ini, yaitu bagian kontrol, bagian angkasa, dan bagian pengguna.

1. Bagian Kontrol

Seperti namanya, bagian ini untuk mengontrol. Setiap satelit dapat berada sedikit diluar orbit, sehingga bagian ini melacak orbit satelit, lokasi, ketinggian, dan kecepatan. Sinyal-sinyal dari satelit diterima oleh bagian kontrol, dikoreksi, dan dikirimkan kembali ke satelit. Koreksi data lokasi yang tepat dari satelit ini disebut dengan data ephemeris, yang nantinya akan di kirimkan kepada alat navigasi kita.

2. Bagian Angkasa

Bagian ini terdiri dari kumpulan satelit-satelit yang berada di orbit bumi, sekitar 12.000 mil diatas permukaan bumi. Kumpulan satelit-satelit ini diatur sedemikian rupa sehingga alat navigasi setiap saat dapat menerima paling sedikit sinyal dari empat buah satelit. Sinyal satelit ini dapat melewati awan, kaca, atau plastik, tetapi tidak dapat melewati gedung atau gunung. Satelit mempunyai jam atom, dan juga akan memancarkan informasi 'waktu/jam' ini. Data ini dipancarkan dengan kode 'pseudo-random'. Masing-masing satelit memiliki kodenya sendiri-sendiri. Nomor kode ini biasanya akan ditampilkan di alat navigasi, maka kita bisa melakukan identifikasi sinyal satelit yang sedang diterima alat tersebut. Data ini berguna bagi alat navigasi untuk mengukur jarak antara alat navigasi dengan satelit, yang akan digunakan untuk mengukur koordinat lokasi. Kekuatan sinyal satelit juga akan membantu alat dalam penghitungan. Kekuatan sinyal ini lebih dipengaruhi oleh lokasi satelit, sebuah alat akan menerima sinyal lebih kuat dari satelit yang berada tepat diatasnya (bayangkan lokasi satelit seperti posisi matahari ketika jam 12 siang) dibandingkan dengan satelit yang berada di garis cakrawala (bayangkan lokasi

satelit seperti posisi matahari terbenam/terbit).

Ada dua jenis gelombang yang saat ini dipakai untuk alat navigasi berbasis satelit pada umumnya, yang pertama lebih dikenal dengan sebutan L1 pada 1575.42 MHz. Sinyal L1 ini yang akan diterima oleh alat navigasi. Satelit juga mengeluarkan gelombang L2 pada frekuensi 1227.6 Mhz. Gelombang L2 ini digunakan untuk tujuan militer dan bukan untuk umum.

3. Bagian Pengguna

Bagian ini terdiri dari alat navigasi yang digunakan. Satelit akan memancarkan data almanak dan ephemeris yang akan diterima oleh alat navigasi secara teratur. Data almanak berisikan perkiraan lokasi (approximate location) satelit yang dipancarkan terus menerus oleh satelit. Data ephemeris dipancarkan oleh satelit, dan valid untuk sekitar 4-6 jam. Untuk menunjukkan koordinat sebuah titik (dua dimensi), alat navigasi memerlukan paling sedikit sinyal dari 3 buah satelit. Untuk menunjukkan data ketinggian sebuah titik (tiga dimensi), diperlukan tambahan sinyal dari 1 buah satelit lagi.

Dari sinyal-sinyal yang dipancarkan oleh kumpulan satelit tersebut, alat navigasi akan melakukan perhitungan-perhitungan, dan hasil akhirnya adalah koordinat posisi alat tersebut. Makin banyak jumlah sinyal satelit yang diterima oleh sebuah alat, akan membuat alat tersebut menghitung koordinat posisinya dengan lebih tepat.

Akurasi atau ketepatan perlu mendapat perhatian bagi penentuan koordinat sebuah titik/lokasi. Koordinat posisi ini akan selalu mempunyai 'faktor kesalahan', yang lebih dikenal dengan 'tingkat akurasi'. Misalnya, alat tersebut menunjukkan sebuah titik koordinat dengan akurasi 3 meter, artinya posisi sebenarnya bisa berada dimana saja dalam radius 3 meter dari titik koordinat (lokasi) tersebut. Makin kecil angka akurasi (artinya akurasi makin tinggi), maka posisi alat akan menjadi semakin tepat. Harga alat juga akan meningkat seiring dengan kenaikan tingkat akurasi yang bisa dicapainya.

Pada pemakaian sehari-hari, tingkat akurasi ini lebih sering dipengaruhi oleh faktor sekeliling yang mengurangi kekuatan sinyal satelit. Karena sinyal satelit tidak dapat menembus benda padat dengan baik, maka ketika menggunakan alat, penting sekali untuk memperhatikan luas langit yang dapat dilihat.

Ketika alat berada disebuah lembah yang dalam (misal, akurasi 15 meter), maka tingkat akurasinya akan jauh lebih rendah daripada di padang rumput (misal, akurasi 3 meter). Di padang rumput atau puncak gunung, jumlah satelit yang dapat dijangkau oleh alat akan jauh lebih banyak daripada dari sebuah lembah gunung. Jadi, jangan berharap dapat menggunakan alat navigasi ini di dalam sebuah gua.

Karena alat navigasi ini bergantung penuh pada satelit, maka sinyal satelit menjadi sangat penting. Alat navigasi berbasis satelit ini tidak dapat bekerja maksimal ketika ada gangguan pada sinyal satelit. Ada banyak hal yang dapat mengurangi kekuatan sinyal satelit:

1. Kondisi geografis, seperti yang diterangkan diatas. Selama kita masih dapat melihat langit yang cukup luas, alat ini masih dapat berfungsi.
2. Hutan. Makin lebat hutannya, maka makin berkurang sinyal yang dapat diterima.
3. Air. Jangan berharap dapat menggunakan alat ini ketika menyelam.
4. Kaca film mobil, terutama yang mengandung metal.
5. Alat-alat elektronik yang dapat mengeluarkan gelombang elektromagnetik.
6. Gedung-gedung. Tidak hanya ketika di dalam gedung, berada di antara 2 buah gedung tinggi juga akan menyebabkan efek seperti berada di dalam lembah.
7. Sinyal yang memantul, misal bila berada di antara gedung-gedung tinggi, dapat mengacaukan perhitungan alat navigasi sehingga alat navigasi dapat menunjukkan posisi yang salah atau tidak akurat.

2.4 Android

Android adalah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi. Sistem operasi ini sama

fungsinya dengan sistem operasi *smartphone* lainnya.

Sistem operasi Android ini menjadi pilihan yang baik bagi vendor-vendor *smartphone* karena memiliki biaya lisensi yang cukup murah dan sifatnya yang *opensource*. Selain itu android juga mendukung berbagai layanan dari Google misalnya Google maps dan lain lain. Untuk sekarang ini vendor-vendor *smartphone* yang menggunakan sistem operasi Android diantaranya adalah Samsung, Sony Ericson, Sony, HTC, Nexian, Smartfren, Lenovo, Aha, Mito dan lain lain.

Pada Juli 2005, Google bekerjasama dengan Android, Inc, dan pada tahun ini lah sistem operasi android ini dikembangkan. Android dikembangkan dengan menggunakan kernel linux. Walau sempat terganjal oleh masalah trademark Android, Google akan tetap merilis sistem operasi Android ke pasaran. Sebelumnya Google terganjal masalah trademark nama Android yang sebelumnya sudah dipatenkan oleh sebuah perusahaan software dan aplikasi internet di *US patent* dan *Trademark Office* (USPTO) dengan nama Android Data.

Akhirnya pada bulan September 2007, Google mematenkan sistem operasi ini dan pada awal tahun 2010 Google mengenalkan Nexus One salah satu jenis *smartphone* bersistem operasi android produksi HTC dan kemudian disusul oleh produk-produk *smartphone* lainnya dengan fitur yang canggih.

Keunggulan yang paling menonjol dari sistem operasi Android adalah open source dimana sistem operasi ini dapat dikembangkan oleh siapa saja. Oleh karena itu *smartphone* yang menggunakan sistem operasi Android ini dijual lebih murah daripada *smartphone* yang lainnya. Walaupun harga *smartphone* dengan sistem operasi Android ini lebih murah, akan tetapi minat dan kebutuhan akan *smartphone* ini semakin banyak dari tahun ke tahun dikarenakan fitur-fitur dari *smartphone* ini yang canggih. Beberapa fitur dari Android adalah multi-touch, Wifi hotspot, Multitasking, GPS, support dengan java, mendukung banyak jaringan (GSM/EDGE, CDMA, EV-DO, dan lain lain) serta memiliki kemampuan dasar telepon seluler pada umumnya yaitu telepon dan sms.

Android berkembang dengan cepat dan menjadi salah satu sistem operasi yang populer bagi pasar *smartphone* dan *tablet*, karena merupakan *platform*

dengan fitur yang lengkap dan memiliki dukungan dari komunitas *open source* dunia sehingga Android berkembang pesat dari segi teknologi maupun dari segi jumlah *perangkat*.

2.4.1 Kelebihan Android

Sistem operasi Android memiliki beberapa kelebihan dan keunggulan dibandingkan dengan sistem operasi *smartphone* lainnya. Beberapa kelebihan tersebut adalah :

1. Open source

Android merupakan sistem operasi yang berbasis kernel Linux. Karena sifat opensource ini, Android bebas dan memudahkan untuk mengembangkan aplikasi dan sistem operasi ini.

2. Ketersediaan aplikasi yang berkualitas

Karena bersifat opensource, pengembang tidak perlu untuk berhadapan dengan platform OS berbayar, karena aplikasi ini bebas didistribusikan dengan lisensi open source, shareware bahkan freeware. Aplikasi yang berkualitas ini dapat diunduh di Google *Android Market*.

3. Dapat menginstall ROM yang dimodifikasi

Terdapat banyak custom *ROM* yang dapat dipakai di ponsel bersistem operasi Android. Pengguna dapat menggunakan custom *ROM* yang diinginkan.

4. Widget

Adanya fasilitas widget di *homescreen* ponsel Android membuat pengguna dengan mudah mengakses berbagai setting dan aplikasi dengan cepat dan mudah.

5. Terkoneksi dengan seluruh layanan Google

Sistem operasi ini mendukung layanan Google maps, Gmail, Google Docs, Youtube, Google Talk, Google+, dan mesin pencari Google.

6. Kemudahan dalam notifikasi

Setiap ada *SMS*, *Email*, datau bahkan artikel terbaru dari *RSS* reader akan selalu ada notifikasi di *homescreen* ponsel Android sehingga tidak akan terlewatkan notifikasi yang dianggap penting.

2.4.2 Kekurangan Android

1. Mode jaringan data yang selalu aktif

Sistem operasi Android merupakan sistem operasi yang selalu berkoneksi dengan internet. Oleh karena itu bagi pengguna *smartphone* Android ini disarankan berlangganan paket internet.

2. Boros daya

Hal ini merupakan masalah yang paling dikeluhkan oleh sebagian besar pemakai *smartphone* Android. Penyebab Android boros daya karena di Android banyak terdapat proses *background* atau aplikasi yang berjalan di belakang layar sehingga akan memakan banyak daya.

3. Ancaman *Malware* yang semakin banyak

Sistem operasi Android kini menjadi *platform* yang paling banyak diserang oleh program jahat (*malware*). Para pembuat *malware* memanfaatkan popularitas Android untuk menyebarkan program jahat buatannya.

Menurut studi yang dilakukan McAfee, terjadi peningkatan yang cukup signifikan dalam hal serangan. Android kini menjadi target utama *malware mobile*. Jumlah *malware* yang menyerang perangkat Android melonjak hampir 37 persen. Hampir semua *malware* mobile baru membidik Android pada kuartal tiga di tahun 2011.

d. Banyak iklan

Karena gratis, sering pada aplikasi android masih banyak dijumpai iklan. Akan tetapi iklan ini tidak mengganggu karena biasanya terletak di bagian atas atau bawah dari aplikasi.

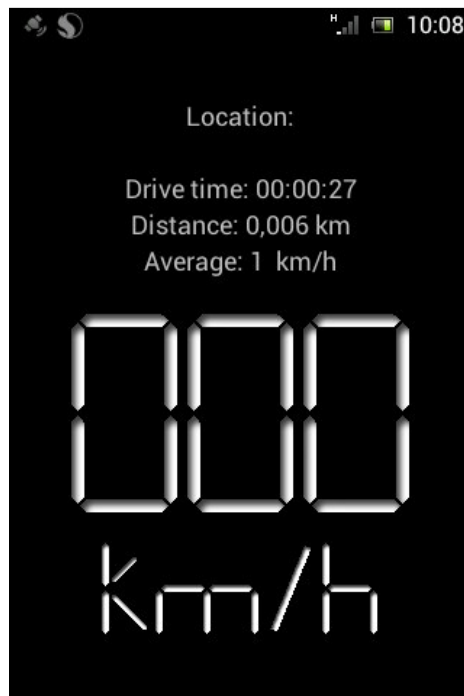
2.5 Aplikasi Sejenis

Dari penelitian-penelitian sebelumnya telah ditemukan aplikasi dengan aplikasi sejenis dengan aplikasi yang akan dibangun. Aplikasi tersebut diantaranya:

1. Aplikasi Digital Speedometer

Aplikasi ini hampir sama dengan yang saya buat di bagian speedometranya,

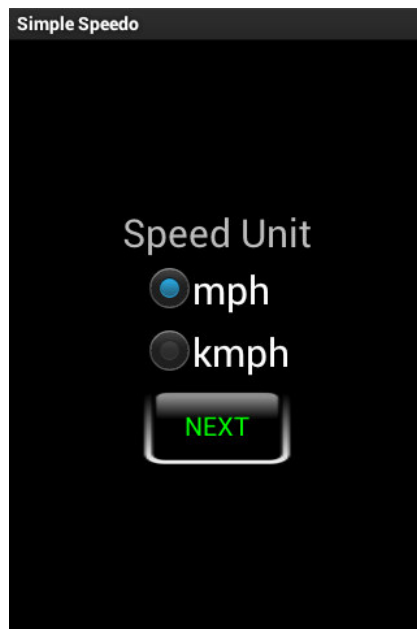
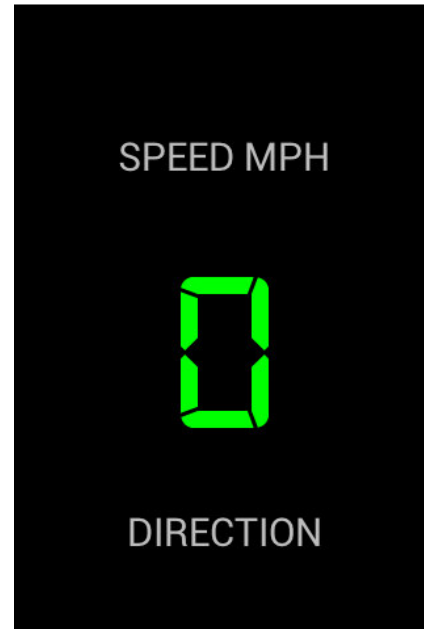
pada aplikasi digital speedometer berbasis android ini terdapat beberapa fitur berupa penghitungan kecepatan, waktu tempuh, jarak, dan kecepatan rata-rata. Pada saat aplikasi ini baru dijalankan maka akan membutuhkan waktu sesaat untuk bisa menggunakan aplikasi ini di karenakan harus menunggu penyesuaian GPS. Menu utama pada Digital Speedometer berbasis android terlihat pada gambar 2.1



Gambar 2.1 Digital Speedometer

2. Aplikasi Simple Speedo

Pada aplikasi simple speedo menampilkan fitur berupa speed unit, yaitu pilihan untuk memilih menggunakan satuan kecepatan mph atau kmph, kemudian dilanjutkan dengan penyesuaian GPS baru masuk ke pase penghitungan kecepatan. Contoh tampilan simple speedo terlihat pada gambar 2.2 dan 2.3

**Gambar 2.2****Gambar 2.3**

Pada gambar 2.2 merupakan tampilan awal pada Simple Speedo dimana user akan di berikan pilihan untuk menggunakan satuan kecepatan yang di inginkan user. Pada aplikasi ini memberikan satuan kecepatan berupa mph dan kmph. Kemudian pada gambar 2.3 merupakan tampilan isi dari Simple Speedo dimana saat user masuk pada bagian aplikasi ini maka aplikasi ini akan secara otomatis menghitung kecepatan.

BAB III

METODOLOGI PENELITIAN

3.1 Analisis Kebutuhan

Metode analisis digunakan untuk mengembangkan sistem aplikasi ‘Digital Speedometer’ berbasis android menjadi komponen-komponen untuk diidentifikasi dan dievaluasi permasalahannya. Sistem yang dianalisis adalah sistem yang berisi informasi tentang segala sesuatu yang berkaitan dengan pengeksplorasian desain *Graphic User Interface* (GUI) yang diaplikasikan pada sistem operasi Android menggunakan interaksi *touchscreen*.

Sistem yang dianalisis adalah bagaimana cara kerja aplikasi Digital Speedometer ini pada sistem android. Dalam tahap ini dilakukan perbandingan antara aplikasi yang sejenis untuk mengetahui bagaimana kekurangan dan kelebihan aplikasi tersebut agar dapat dijadikan acuan untuk membentuk suatu aplikasi digital speedometer berbasis android yang lebih baik untuk memudahkan *user* dalam mendapatkan suatu informasi yang diinginkan. Tahapan analisis ini merupakan tahapan yang paling penting dalam program yang akan dirancang, karena jika terjadi kesalahan dalam tahap ini akan menyebabkan terjadinya kesalahan pada tahap selanjutnya.

3.1.1. Analisis Kebutuhan Antarmuka

Kebutuhan antarmuka yang diperlukan dalam Aplikasi Speedtester berbasis android ini adalah aplikasi ini dibuat untuk memudahkan pengguna untuk mendapatkan informasi mengenai speedometer dimana aplikasi ini memberikan hasil informasi berupa mencatat kecepatan, jarak, dan waktu. Aplikasi ini bersifat fleksibel dalam penggunaannya, user bisa menggunakan aplikasi ini saat berjalan, berlari, bersepeda, dan sedang memakai kendaraan bermotor atau melakukan hal lain di luar ruangan. Saat aplikasi ini selesai dijalankan, anda bisa melihat kecepatan maksimal, kecepatan rata-rata, jarak, waktu. Aplikasi ini ditambah juga fitur berupa grafik sehingga user bisa melihat data progress aplikasi ini berjalan.

Aplikasi ini juga dtambahkan fitur history sehingga masih bisa melihat data dari perjalanan sebelumnya.

3.1.2. Analisis Kebutuhan Masukan

Kebutuhan masukan yang diperlukan dalam aplikasi Speedtester ini merupakan input data yang sebelumnya sudah di tetapkan oleh pembuat aplikasi sehingga pengguna tidak dapat memasukan dan merubah data selain data yg sudah ditetapkan pada aplikasi ini. Untuk aplikasi ini kebutuhan masukan yang dibutuhkan yaitu:

1. Pilihan Speedometer dimana pada menu ini terdapat tombol start untuk memulai aplikasi ini.

3.1.3 Analisis Kebutuhan Proses

Setelah menentukan analisis kebutuhan masukan, langkah selanjutnya yaitu melakukan analisa kebutuhan proses. Kebutuhan proses yang dilakukan oleh aplikasi Speedtester ini adalah:

1. Proses mendapatkan informasi kecepatan, jarak, waktu.
2. Proses mendapatkan informasi kecepatan rata-rata dan kecepatan maksimal.
3. Proses mendapatkan informasi dari grafik yang ditampilkan aplikasi.
4. Proses mendapatkan informasi dari history yang ditampilkan aplikasi.

3.1.4 Analisis Kebutuhan Output

Hasil *Output* dalam aplikasi ini merupakan bentuk dari fungsi yang ada dalam aplikasi untuk menampilkan informasi yang dibutuhkan oleh pengguna. Kebutuhan keluaran yang diperlukan dalam aplikasi ini adalah:

1. Informasi tentang kecepatan, jarak serta waktu.
2. Informasi grafik serta progress pada saat aplikasi berjalan.
3. Informasi statistik yang berupa hasil informasi keseluruhan dari aplikasi.

3.2 Analisis Kebutuhan Perangkat Keras

Perangkat keras merupakan bagian fisik komputer yang berfungsi untuk memberikan masukan, mengolah perintah dan menampilkan keluaran. Spesifikasi perangkat keras yang dibutuhkan dibagi menjadi dua sisi, yaitu sisi pembangunan aplikasi dan sisi penggunaan aplikasi. Adapun perangkat keras yang digunakan dalam pembuatan aplikasi ini adalah laptop yang memiliki komponen-komponen sebagai berikut:

1. Piranti masukan seperti *keyboard* dan *mouse*.
2. Piranti keluaran seperti monitor.
3. Wifi untuk koneksi internet.
4. *Processor Core i3*.
5. Kartu grafis 512 MB.
6. RAM ukuran 4 GB.

Sedangkan spesifikasi perangkat keras yang harus dimiliki oleh pengguna untuk menjalankan aplikasi speedtester adalah sebagai berikut:

1. Smartphone atau *tablet* bersistem operasi android minimal 2.2 (Froyo).
2. Piranti keluaran ponsel berupa layar dengan resolusi minimal 320x240 pixel.
3. Piranti masukan smartphone atau tablet dengan layar sentuh atau tombol fisik
4. *Internal* memori untuk penyimpanan aplikasi minimal 10 mb.

3.3 Analisis Kebutuhan Perangkat Lunak

Untuk mendukung perangkat keras agar dapat berjalan dengan semestinya dibutuhkan suatu perangkat lunak agar tercipta aplikasi yang diinginkan. Kebutuhan perangkat lunak memiliki dua kategori, yaitu perangkat lunak yang digunakan untuk pembuatan aplikasi dan perangkat lunak yang digunakan untuk menjalankan aplikasi

Kebutuhan perangkat lunak untuk pengimplementasian aplikasi ini adalah sebagai berikut :

1. Sistem operasi yang akan digunakan dalam pembuatan aplikasi ini yaitu Windows 7 64 bit.
2. Eclipse indigo.
3. Android Development Tools (ADT) 21.0.0.
4. SDK Android minimal versi 2.0.

Sedangkan perangkat lunak yang digunakan untuk menjalankan aplikasi ini adalah smartphone android dengan spesifikasi sistem operasi yang digunakan mulai dari android versi 2.2 (Froyo) keatas, yang terbaru yaitu android versi 4.2 (Jelly Bean).s

3.4 Tahap Pengembangan Sistem

Berdasarkan analisis yang telah dilakukan maka dapat diketahui apa saja yang menjadi masukan system, keluaran system, metode yang digunakan system, serta antar muka system yang dibuat sehingga system yang dibuat tersebut sesuai dengan apa yang diharapkan. Adapun perancangan system perangkat lunak ini adalah sebagai berikut Tahap pembuatan perangkat lunak merupakan tahapan pengembangan sistem agar sistem tersebut dapat memenuhi kebutuhan. Tahap pembuatan perangkat lunak yang digunakan yaitu metode *waterfall* yang terdiri dari proses *information engineering*, *analysis*, *design*, *coding*, *testing*, dan *maintenance*.

3.4.1 Gambaran Umum Sistem



Gambar 3.1 Gambaran Umum Sistem

Gambaran umum system dapat dilihat pada gambar 3.1 dimana aplikasi Speedtester ini akan menerima perintah dari user berupa dijalankannya aplikasi ini dengan menekan tombol *Start*. Jika user sudah menekan tombol start maka aplikasi ini akan berjalan sampai user menekan tombol stop di tekan. Setelah tombol stop ini ditekan maka aplikasi ini akan secara otomatis masuk ke menu statistik dimana user akan mengetahui hasil informasi dari dijalankannya aplikasi ini.

3.4.2 Use Case

Use Case adalah rangkaian/uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. Use case digunakan untuk membentuk tingkah laku benda/things dalam sebuah model serta di Realisasikan oleh sebuah *collaboration*. Use case menggambarkan proses system (kebutuhan system dari sudut pandang user).

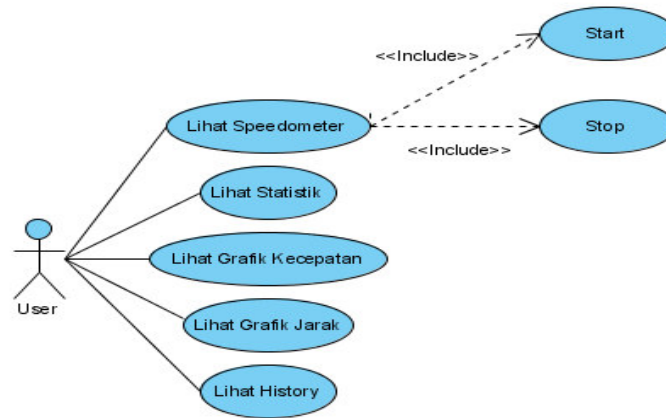
Secara umum Use Case adalah :

1. Pola perilaku sistem
2. Urutan transaksi yang berhubungan yang dilakukan oleh satu aktor (*user*).

Use case terdiri dari:

3. Use case
4. Actors
5. Relationship
6. System boundary boxes (optional)
7. Packages (optional)

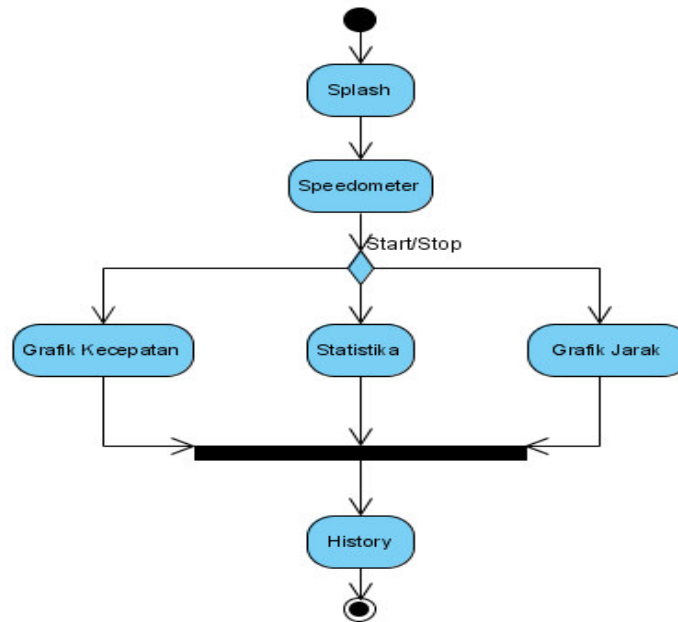
Pada diagram Use case ini dijabarkan 4 proses *user* untuk menampilkan Speedometer, Grafik kecepatan, Grafik Jarak, Statistik, history. Seperti yang ditunjukkan pada gambar 3.2.



Gambar 3.2 Use Case

3.4.3 Diagram Activity

Activity Diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Activity Diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Activity Diagram merupakan state khusus, dimana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (*internal processing*). Activity Diagram aplikasi Speedtester berbasis android terlihat seperti gambar 3.3



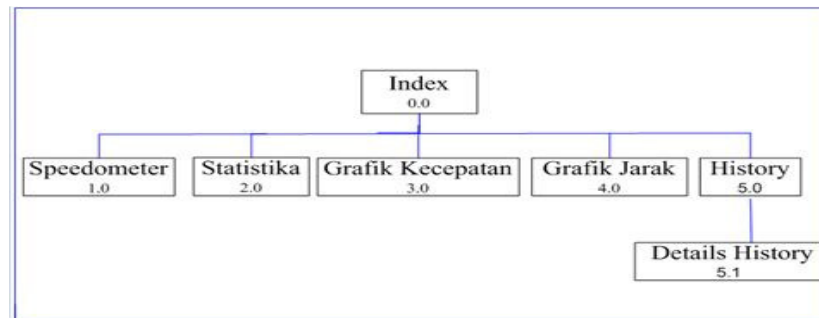
Gambar 3.3 Diagram Activity

3.4.4 Diagram HIPO

HIPO (*Hierarchy plus Input-Process-Output*) merupakan paket yang berisikan suatu set diagram yang secara grafis menjelaskan fungsi suatu sistem dari tingkat yang paling umum ke tingkat yang lebih khusus. Paket HIPO terdiri dari tiga jenis diagram, yaitu sebagai berikut :

3.4.4.1 Visual Table of Countents (VTOC)

Visual Tabel of contents atau daftar isi visual terdiri dari satu diagram hirarki atau lebih. Diagram ini menggambarkan hubungan dari fungsi-fungsi secara berjenjang. Daftar isi visual menggambarkan seluruh program HIPO baik rinci maupun ringkasan yang terstruktur. Pada kasus aplikasi yang akan dibuat, rancangan VTOC ditunjukkan pada gambar 3.4



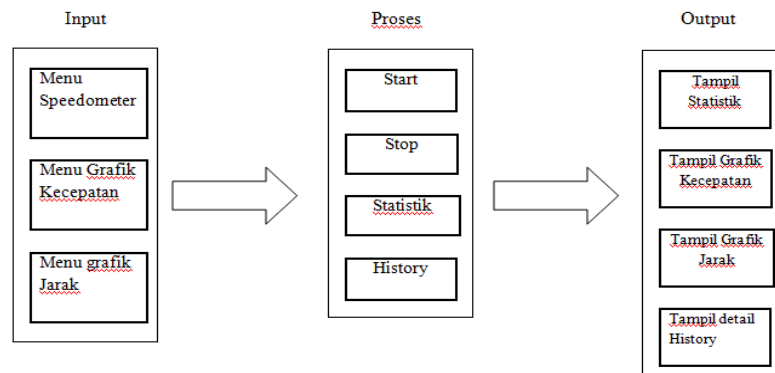
Gambar 3.4 *Visual Table of Content*

Bagian penjelasan :

- 0.0 Menu Utama
- 1.0 Menu Speedometer
- 2.0 Menu Statistika
- 3.0 Menu Grafik Kecepatan
- 4.0 Menu Grafik Jarak
- 5.0 Menu History
- 5.1 Tampil informasi lebih rinci dari pilihan history yang dipilih

3.4.4.2 Overview Diagram

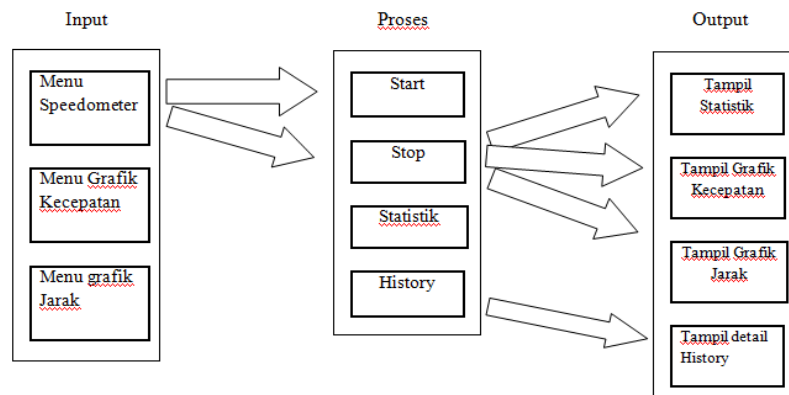
Diagram Overview atau diagram ringkasan yaitu suatu seri diagram fungsional yang menggambarkan fungsi dan referensi utama dari suatu system yang menunjukan secara garis besar hubungan dari input, proses dan *output*. Fungsi dan referensi ini diperlukan oleh program untuk memperluas fungsi sampai uraian yang terkecil. Overview diagram dapat dilihat pada gambar 3.5



Gambar 3.5 Overview Diagram

3.4.4.3 Detail Diagram

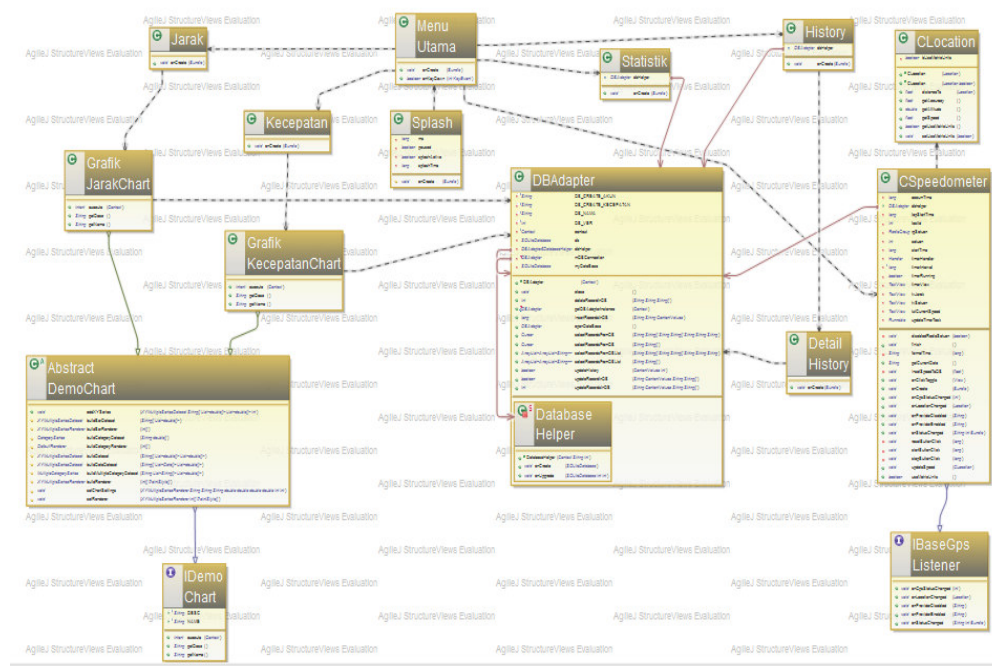
Detail Diagram atau diagram rinci yaitu suatu seri diagram fungsional dan masing-masing diagram dihubungkan dengan sebuah sub-fungsi system. Diagram rinci merupakan diagram yang paling rendah dalam diagram yang terdapat dalam HIPO. Diagram rinci berisi unsur-unsur dasar. Fungsi dari diagram ini adalah menjelaskan fungsi-fungsi khusus, menunjukkan item-item output dan input yang khusus dan menunjukkan diagram rinci lainnya. Detail diagram dapat dilihat pada gambar 3.6



Gambar 3.6 Detail Diagram

3.5 Diagram Kelas

Diagram Kelas adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem. Diagram kelas memberikan gambaran tentang system dan relasi-relasi yang ada di dalamnya. Diagram kelas aplikasi ini ditunjukkan pada gambar 3.7



Gambar 3.7 Diagram Kelas

3.6 Perancangan Basis Data

Basis data merupakan satu satu komponen yang penting pada sebuah system informasi karena berfungsi sebagai penyedia informasi kepada penggunaannya. Berikut ini adalah desain basis data dari system yang akan dibuat.

a) Tabel Kecepatan

Tabel Kecepatan dibuat untuk penyimpanan proses dari setiap kali aplikasi berjalan.

Tabel 3.2 Tabel Kecepatan

Nama Field	Type	Keterangan
Id_kecepatan	Integer	Primary Key
Id_history	Integer	Null
Speed	Text	Not null

b) Tabel History

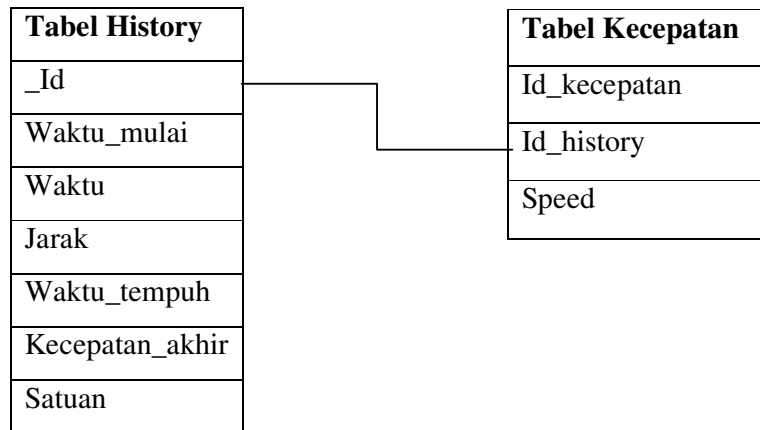
Tabel History dibuat untuk penyimpanan hasil dari semua data setiap aplikasi dijalankan sehingga bisa dilihat kembali oleh user pada menu history.

Tabel 3.1 Tabel History

Nama Field	Type	Keterangan
_Id	Integer	Primary Key
Waktu_mulai	Text	Not null
Waktu	Text	Not null
Jarak	Text	Not null
Waktu_tempuh	Text	Not null
Kecepatan_akhir	Text	Not null
Satuan	Text	Not null

c) Relasi Antar table

Dari dua tabel diatas maka relasi antar tabel dari aplikasi ini seperti dibawah ini.

Tabel 3.3 Relasi Antar Tabel

3.7 Tahap Perancangan Antarmuka (*Interface*)

Perancangan antarmuka adalah bagian dimana terjadi komunikasi antara pengguna dengan sistem. Antarmuka dirancang semudah mungkin agar memudahkan *user* untuk menggunakan sistem. Antarmuka yang sulit dipahami akan membingungkan penggunanya dan menyebabkan sistem tidak dapat digunakan dengan sempurna. Kemudahan pengguna dalam menggunakan sistem dapat dikatakan sebagai keberhasilan antarmuka dalam melakukan komunikasi dengan pengguna.

3.7.1 Tampilan Antarmuka Halaman Menu Speedometer

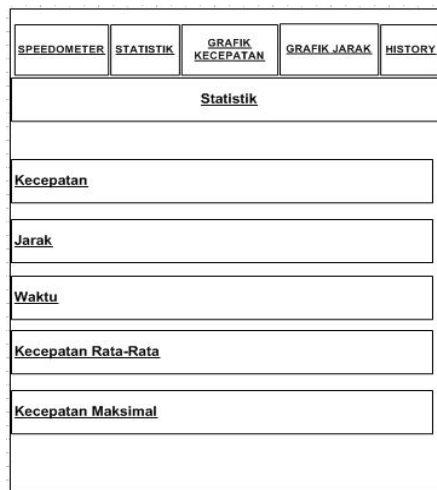
Saat pertama kali aplikasi dijalankan, akan muncul sebuah halaman Splash kemudian akan menampilkan barisan menu utama dari aplikasi (Speedometer, Statistika, Grafik Kecepatan, Grafik Jarak, History). Rancangan antarmuka pada halaman menu ditunjukkan pada gambar 3.8



Gambar 3.8 Perancangan Halaman Utama

3.7.2 Tampilan Antarmuka Halaman Statistika

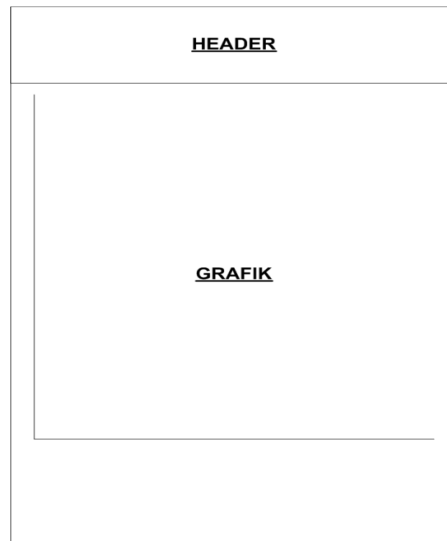
Pada halaman ini akan menampilkan isi dari list informasi terakhir saat aplikasi selesai dijalankan yang seperti terlihat pada gambar 3.9



Gambar 3.9 Perancangan Halaman Statistika

3.7.3 Tampilan Antarmuka Halaman Grafik Kecepatan dan Grafik Jarak

Pada halaman ini *user* akan diberikan informasi mengenai status tingkat ke stabilan kecepatan serta berapa jauh jarak yang ditempuh selama dalam proses perjalanan, seperti terlihat pada gambar 3.10



Gambar 3.10 Perancangan Halaman Grafik

3.7.4 Tampilan Antarmuka Halaman History

Pada halaman ini menampilkan History dari data aplikasi yang dulu sampai yang terkini, seperti terlihat pada gambar 3.11

SPEEDOMETER	STATISTIK	GRAFIK KECEPATAN	GRAFIK JARAK	HISTORY
Data 1				
Data 2				
Data 3				
Data 4				

Gambar 3.11 Perancangan Halaman History

3.8 Rencana Pengujian

Rencana pengujian ditujukan untuk mengetahui apakah sistem yang dibuat telah layak digunakan sebagai alternatif manajemen skripsi online.

Pengujian Kompabilitas

Perancangan pengujian aplikasi Speedtester berbasis android ini dilakukan untuk mengetahui apakah sistem yang dibuat berjalan sesuai dengan rancangan pembuatan aplikasi atau tidak, performa serta kompatibilitas aplikasi di berbagai macam *device smartphone* dan versi android. Pengujian akan dilakukan pada *smartphone* yang bersistem operasi android dengan versi system operasi yang berbeda atau sama dan *smartphone* yang berbeda merk. Perangkat berbasis android yang digunakan untuk melakukan pengujian memiliki spesifikasi *hardware* yang berbeda-beda. Perangkat berbasis android yang akan digunakan adalah Sony Ericsson Live With Walkman/WT19i, Sony Xperia P, Xperia S, Xperia SL, Samsung Galaxy Ace, Samsung Galaxy Young.

Berikut keterangan beberapa sistem android yang digunakan dalam pengujian aplikasi berita sepak bola berbasis android ini.

Tabel .3.1 Pengujian Sistem

No	Jenis Smartphone	Merk	Processor	RAM	Versi Android
1	Sony Ericsson Live With Walkman	Sony Ericsson	1 GHz	512MB	Ice Cream Sandwich 4.0
2	Sony Xperia P	Sony	Dualcore 1GHz	1GB	Ice Cream Sandwich 4.0
3	Sony Xperia S	Sony	Dualcore 1,5 GHz	1GB	Jeally Bean 4.1
4	Sony Xperia SL	Sony	Dualcore 1,7 GHz	1GB	Ice Cream Sandwich 4.0
5	Samsung Galaxy Ace	Samsung	800 MHz	278MB	Froyo 2.2
6	Samsung Galaxy Young	Samsung	1 GHz	768MB	Gingerbread 2.3

Perangkat yang digunakan dalam mengujikan aplikasi Speedtester berbasis android ini berbeda-beda mulai dari jenis, merk, processor, RAM , maupun versi android. Merk *smartphone* yang digunakan berupa Sony Ericsson, Sony, dan Samsung. Sedangkan perangkat yang paling sering digunakan adalah Sony karena merk ini mudah didapat selain itu perangkat dari Sony terkenal handal. Processor yang digunakan berkisar antara 800 Mhz hingga 1.7 Ghz.

BAB IV

IMPLEMENTASI DAN ANALISIS

4.1 Batasan Implementasi

Implementasi Aplikasi Speedtester ini bertujuan agar memudahkan masyarakat lebih mudah dalam mendapatkan informasi mendapatkan informasi tentang speedometer saat melakukan perjalanan.

Aplikasi Speedtester ini hanya terbatas untuk melihat informasi yang sudah tersedia dalam aplikasi ini yaitu penghitungan kecepatan, jarak serta waktu. Batasan implementasi hasil perancangan perangkat lunak dan perancangan antarmuka dibagi menjadi dua bagian, yaitu perangkat lunak dan perangkat keras. Batasan implementasi perangkat lunak antara lain bahasa pemrograman dan alasan pemilihannya, sedangkan perangkat keras adalah perangkat keras yang digunakan dalam implementasi yang telah dibuat.

4.1.1. Perangkat Lunak

Untuk mendukung perangkat keras agar dapat berjalan dengan semestinya dibutuhkan suatu perangkat lunak agar tercipta aplikasi yang diinginkan. Kebutuhan perangkat lunak memiliki dua kategori, yaitu perangkat lunak yang digunakan untuk pembuatan aplikasi dan perangkat lunak yang digunakan untuk menjalankan aplikasi,

Kebutuhan akan perangkat lunak yang dibutuhkan dalam pengimplementasian aplikasi ini adalah sebagai berikut :

1. Sistem operasi yang digunakan dalam pembuatan aplikasi ini yaitu Windows 7 64 bit.
2. Eclipse indigo.
3. Android Development Tools (ADT) 21.0.0.
4. SDK Android minimal versi 2.3.
5. Sublime text 2 dan notepad ++ yang digunakan untuk proses coding (pengkodean) dalam pembangunan *web services*.

6. Mozilla Firefox dan SQLite manager sebagai *interface* dari sistem informasi web service yang berfungsi sebagai tempat untuk penginputan data.

Sedangkan perangkat lunak yang digunakan untuk menjalankan aplikasi ini adalah *smartphone* android dengan spesifikasi sistem operasi yang digunakan mulai dari android versi 2.2 (Froyo) keatas, yang terbaru yaitu android versi 4.2 (Jelly Bean).

4.1.2. Perangkat Keras

Perangkat keras merupakan bagian fisik komputer yang berfungsi untuk memberikan masukan, mengolah perintah dan menampilkan keluaran. Spesifikasi perangkat keras yang dibutuhkan dibagi menjadi dua sisi, yaitu sisi pembangunan aplikasi dan sisi penggunaan aplikasi. Adapun perangkat keras yang digunakan dalam pembuatan aplikasi ini adalah laptop yang memiliki komponen-komponen sebagai berikut:

1. Piranti masukan seperti *keyboard* dan *mouse*.
2. Piranti keluaran seperti monitor.
3. Wifi untuk koneksi internet.
4. *Processor Core i3*.
5. Kartu grafis 512 MB.
6. RAM ukuran 4 GB.

Sedangkan spesifikasi perangkat keras yang harus dimiliki oleh pengguna untuk menjalankan aplikasi virtual tour guide Gembira Loka sebagai berikut:

1. *Smartphone* atau *tablet* bersistem operasi android minimal 2.2 (Froyo).
2. Piranti keluaran ponsel berupa layar dengan resolusi minimal 320x240 pixel.
3. Piranti masukan *smartphone* atau *tablet* dengan layar sentuh atau tombol fisik.
4. *Internal* memori untuk penyimpanan aplikasi minimal 10 mb.

4.2 Implementasi

Perangkat lunak yang dibutuhkan untuk mengimplementasikan Aplikasi Speedtester ini adalah Android 2.2. Untuk Android versi 2.3, dan pada versi 4.2 aplikasi dapat dijalankan dengan lancar, namun untuk versi lainnya ada kemungkinan aplikasi tidak berjalan dengan maksimal

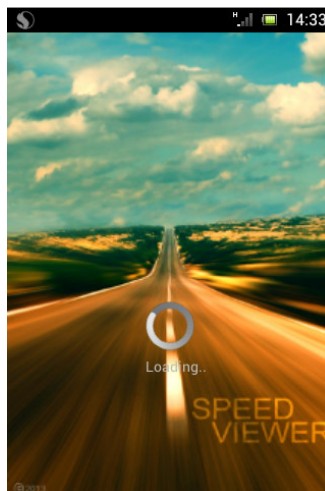
4.2.1 Implementasi Pembuatan Aplikasi

Tahap implementasi merupakan tahap dimana sistem mampu diaplikasikan dalam keadaan sesungguhnya. Pada tahap ini akan diketahui apakah sistem yang dibuat dapat berjalan dengan baik atau tidak dan apakah sistem dapat menghasilkan *output* sesuai dengan perancangan yang telah dilakukan sebelumnya.

4.2.2 Implementasi Tampilan Antarmuka

4.2.2.1 Halaman Pembuka (*Splash Screen*)

Splash screen merupakan sebuah halaman pembuka atau halaman yang akan pertama kali keluar ketika aplikasi dijalankan. Tampilan halaman *splash screen* dapat dilihat pada gambar berikut



Gambar 4.1 *Halaman Splash*

Saat pertama kali aplikasi dibuka, akan muncul sebuah halaman pembuka atau *splash screen* yang akan menampilkan sebuah *image* dan menampilkan

sebuah pesan yang memberitahukan status dari aplikasi. Tampilan *splash screen* ditunjukkan pada gambar di atas 4.1.

Class : Splash.java

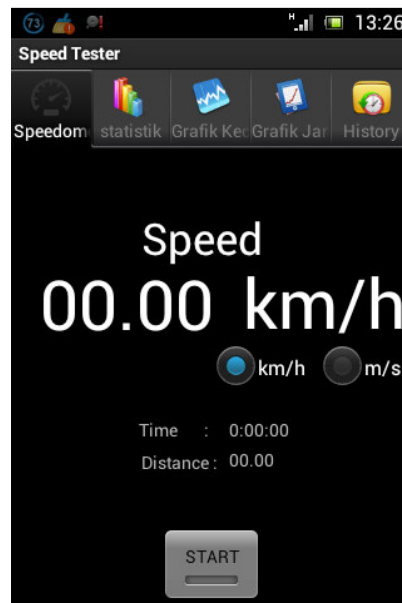
```
this.requestWindowFeature(Window.FEATURE_NO_TITLE);
setContentView(R.layout.splash);
    Thread mythread = new Thread() {
        @Override
        public void run() {
            try {
                while (splashActive && ms <
splashTime) {
                    if(!paused)
                        ms=ms+100;
                    sleep(100);
                }
            } catch (Exception e) {}
            finally {
                Intent intent = new
Intent (Splash.this, MenuUtama.class);
                startActivity(intent);
            }
        }
    };
    mythread.start();
}
```

Keterangan :

Pada *class* ini hanya terdapat dua fungsi `run()` yang digunakan untuk menampilkan gambar pada tampilan awal selama satu detik dan setelah itu kemudian layar secara langsung berpindah ke *class* `MenuUtama.java`. fungsi yang kedua yaitu `requestWindowFeature(Window.FEATURE_NO_TITLE)` yang berfungsi memanggil halaman layout dari `splash.xml`.

4.2.2.2 Menu Pilihan Speedometer

Pada menu pilihan aplikasi ini, *user* akan diberikan menu speedometer yang di dalamnya terdapat fitur untuk penghitungan kecepatan, jarak serta waktu. Setelah *user* menekan tombol start, maka akan otomatis aplikasi ini akan berjalan dan user pun akan mendapatkan informasi dari speedometer. Halaman menu pilihan speedometer seperti yang ditunjukkan pada gambar 4.2.



Gambar 4.2 Menu Speedometer

Kelas: MenuUtama.java

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    TabHost tabHost = getTabHost();

    tabHost.addTab(tabHost.newTabSpec("Speedometer")
        .setIndicator("Speedometer",
            getResources().getDrawable(R.drawable.icon_speedometer_tab))
        .setContent(new Intent(this, CSpeedometer.class)
            .addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP)));

    tabHost.addTab(tabHost.newTabSpec("statistik")
```

```

.setIndicator("statistik",getResources().getDrawable(R.drawable.icon_statistik_tab))

        .setContent(new Intent(this, Statistik.class)
            .addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP));

        tabHost.addTab(tabHost.newTabSpec("Grafik Kecepatan")
            .setIndicator("Grafik Kecepatan",
getResources().getDrawable(R.drawable.icon_grafik_kecepatan_tab))
            .setContent(new Intent(this, Kecepatan.class)
                .addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP)));

        tabHost.addTab(tabHost.newTabSpec("Grafik Jarak")
            .setIndicator("Grafik Jarak",
getResources().getDrawable(R.drawable.icon_grafik_jarak_tab))
            .setContent(new Intent(this, Jarak.class)
                .addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP)));

        tabHost.addTab(tabHost.newTabSpec("History")
            .setIndicator("History",
getResources().getDrawable(R.drawable.icon_history_tab))
            .setContent(new Intent(this, History.class)
                .addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP)));
        getTabHost().invalidate();
    }

```

Pada source code di atas berfungsi untuk menampilkan lima menu utama diantaranya yaitu speedometer, statistika, grafik kecepatan, grafik jarak, dan history. Pada menu speedometer akan memanggil kelas CSpeedometer.java, dan pada menu statistik akan memanggil kelas Statistik.java, pada menu Grafik kecepatan akan memanggil kelas Kecepatan.java kemudian akan memanggil kelas GrafikKecepatanChart.java yang merupakan bagian libraries dari grafik yang dipakai pada aplikasi ini, pada menu Grafik jarak akan memanggil kelas Jarak.java kemudian akan memanggil kelas GrafikJarakChart.java yang juga merupakan bagian libraries dari grafik yang dipakai pada aplikasi ini yang mana berfungsi untuk menampilkan grafik jarak pada aplikasi Speedtester ini, pada

menu History akan memanggil History.java yang kemudian akan memanggil kelas DetailHistory.java yang menampilkan data dari suatu perjalanan yang telah dipilih pada saat masuk ke menu history.

Berikut adalah *source code* dari menu Speedometer:

Class CSpeedometer.java

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    {
        this.setContentView(R.layout.speedometer_main);
    }
    satuan = 1;
    tvSatuan = (TextView) findViewById(R.id.tvSatuan);
    if(satuan == 1){
        tvSatuan.setText("km/h");
    }else{
        tvSatuan.setText("m/s");
    }
    rgSatuan = (RadioGroup) findViewById(R.id.rgTipe);
    rgSatuan.setOnCheckedChangeListener(new
OnCheckedChangeListener() {
    public void onCheckedChanged(RadioGroup group, int
checkedId) {

        switch (checkedId) {
            case R.id.rbKm:

                satuan = 1;
                tvSatuan.setText("km/h");
                break;

            case R.id.rbMeter:

                satuan = 2;
                tvSatuan.setText("m/s");
                break;
```

```

        }
    }
});

txtCurrentSpeed = (TextView)
this.findViewById(R.id.txtCurrentSpeed);
tvJarak = (TextView) this.findViewById(R.id.tvJarak);

txtCurrentSpeed.setText("00.00");
tvJarak.setText("00.00");

timerHandler = new Handler();

timerView = (TextView) findViewById(R.id.tvWaktu);
timerView.setText(formatTime(0));

LocationManager locationManager = (LocationManager) this
    .getSystemService(Context.LOCATION_SERVICE);

locationManager.requestLocationUpdates(LocationManager.GPS_PROVID
ER, 0,
    0, this);

this.updateSpeed(null);
}

public void onClickToggle(View view) {

    boolean on = ((ToggleButton) view).isChecked();

    if (on) {

        long now = System.currentTimeMillis();
        startButtonClick(now);
    } else {

```

```

        if (timerRunning == true) {

            String tanggal_sekarang = getCurrentDate();
            ContentValues initialValues = new
ContentValues();

            initialValues.put("waktu", tanggal_sekarang);

            initialValues.put("kecepatan",
txtCurrentSpeed.getText()
                                .toString());
            initialValues.put("jarak",
tvJarak.getText().toString());

            initialValues.put("kecepatan_view",
txtCurrentSpeed.getText()
                                .toString().replace(".00", ""));
            initialValues.put("jarak_view",
tvJarak.getText().toString()
                                .replace(".00", ""));

            String t[] =
timerView.getText().toString().split(":");
            int jam = Integer.parseInt(t[0]);
            int menit = Integer.parseInt(t[1]);
            int detik = Integer.parseInt(t[2]);

            int menit_total = (jam * 60) + menit + (detik
/ 60);

            initialValues.put("waktu_tempuh_view",
menit_total + "");

            initialValues.put("waktu_tempuh",
timerView.getText()
                                .toString());

            DBAdapter db1 = DBAdapter

```

```

        .getDBAdapterInstance(CSpeedometer.this);
        db1.openDataBase();
        db1.insertRecordsInDB("history", null,
initialValues);

        db1.close();

        long now = System.currentTimeMillis();
        stopButtonClick(now);

        final String myTabTag = "Statistik";
        final int myTabIndex = 1;

        TabActivity parent = (TabActivity)
getParent();

        TabHost tabhost = parent.getTabHost();
        tabhost.setCurrentTab(1);
    }
}

private void startButtonClick(long now) {

    timerHandler.removeCallbacks(updateTimerTask);

    startTime = now;
    lapStartTime = now;

    timerRunning = true;
    timerHandler.postDelayed(updateTimerTask, timerInterval);
}

private void stopButtonClick(long now) {

    if (timerRunning == false) {
        return;
    }
}

```

```

    }

    timerHandler.removeCallbacks(updateTimerTask);

    accumTime = accumTime + (now - startTime);

    timerView.setText(formatTime(accumTime));

    timerRunning = false;
}
private void resetButtonClick(long now) {

    accumTime = 0;

    timerView.setText(formatTime(0));

    if (timerRunning) {
        startTime = now;
        lapStartTime = now;
    }
}
private Runnable updateTimerTask = new Runnable() {

    public void run() {

        long now = System.currentTimeMillis();
        accumTime = accumTime + (now - startTime);

        startTime = now;

        timerView.setText(formatTime(accumTime));

        timerHandler.postDelayed(updateTimerTask,
timerInterval);

    }
}

```

```

    };

    private String formatTime(long timeMs) {
        String retValue;

        timeMs = timeMs + 5;

        int seconds = (int) (timeMs / 1000);

        int minutes = seconds / 60;

        int hours = minutes / 60;

        minutes = minutes % 60;

        seconds = seconds % 60;

        int hundreds = (int) (timeMs % 1000) / 10;

        retValue = String.format("%d:%02d:%02d", hours, minutes,
seconds);

        return retValue;
    }

    public void finish() {
        super.finish();
        System.exit(0);
    }

    public String getCurrentDate() {
        final Calendar c = Calendar.getInstance();
        int year, month, day;
        year = c.get(Calendar.YEAR);
        month = c.get(Calendar.MONTH);
        day = c.get(Calendar.DATE);
        return day + "/" + month + "/" + year + " " +
c.get(Calendar.HOUR)

```

```

        + ":" + c.get(Calendar.MINUTE);
    }

    public void updateSpeed(CLocation location) {
        // String t = "";
        if (timerRunning == true) {
            float nCurrentSpeed = 0;
            float jarak = 0;
            if (location != null) {

location.setUseMetricUnits(this.useMetricUnits());
                nCurrentSpeed = location.getSpeed();

                if(satuan == 1){
nCurrentSpeed = (float) (nCurrentSpeed * 1.609);
                }else{
nCurrentSpeed = (float) (nCurrentSpeed * 1.609 * 1000);
                }
                jarak = nCurrentSpeed * (accumTime / 3600);
            }

            Formatter fmt = new Formatter(new StringBuilder());

            fmt.format(Locale.US, "%.2f", nCurrentSpeed);

            String strCurrentSpeed = fmt.toString();
            strCurrentSpeed = strCurrentSpeed.replace(' ', '0');

            Formatter fmt1 = new Formatter(new StringBuilder());
            fmt1.format(Locale.US, "%.2f", jarak);
            String strjarak = fmt1.toString();
            strjarak = strjarak.replace(' ', '0');

            String strUnits = "km/jam";
            if (satuan == 2) {
                strUnits = "m/s";
            }
        }
    }

```

```

        txtCurrentSpeed.setText(strCurrentSpeed);

        Float total_jarak = Float.parseFloat(strjarak)
            +
Float.parseFloat(tvJarak.getText().toString());

        Formatter fmt2 = new Formatter(new StringBuilder());

        fmt2.format(Locale.US, "%.2f", total_jarak);

        tvJarak.setText(fmt2.toString());

        String mydate =
java.text.DateFormat.getDateInstance().format(
            Calendar.getInstance().getTime());
    } else {

    }
}

public boolean useMetricUnits() {

    return false;
}

public void onLocationChanged(Location location) {
    if (location != null) {
        CLocation myLocation = new CLocation(location,
            this.useMetricUnits());
        this.updateSpeed(myLocation);
    }
}

public void onProviderDisabled(String provider) {
    // TODO: do something one day?
}

public void onProviderEnabled(String provider) {
    // TODO: do something one day?
}
}

```

```

    public void onStatusChanged(String provider, int status, Bundle
extras) {
        // TODO: do something one day?
    }
    public void onGpsStatusChanged(int event) {
        // TODO: do something one day?
    }
}

```

4.2.2.3 Menu Pilih Statistika

Halaman menu Pilih Statistika ini menampilkan informasi setelah aplikasi speedometer selesai dijalankan yang mana setelah user menekan tombol stop pada menu speedometer maka akan secara otomatis masuk ke menu statistika. Pada menu statistika ini terdapat informasi berupa kecepatan terakhir saat aplikasi ini di hentikan, jarak tempuh total, waktu tempuh, kecepatan rata-rata serta kecepatan maksimal saat aplikasi ini dijalankan. Menu halaman Statistika seperti yang ditunjukkan pada gambar 4.3



Gambar 4.3 Pilih Menu Statistik

Class Statistik.java

```

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.statistik);

        DBAdapter db1 = DBAdapter.getDBAdapterInstance(this);

        db1.openDataBase();
        Cursor c= db1.selectRecordsFromDB("select
waktu,kecepatan,jarak,waktu_tempuh, (select avg(kecepatan) from history) as
kecepatan_rata, (select max(kecepatan) from history) as kecepatan_max from
history order by _id desc limit 1", null);
        Cursor c1= db1.selectRecordsFromDB("select * from history",
null);

        TextView kc = (TextView)findViewById(R.id.edKecepatan);
        TextView jr = (TextView)findViewById(R.id.edJarak);
        TextView wk = (TextView)findViewById(R.id.edWaktu);
        TextView kr =
(TextView)findViewById(R.id.edKecepatanRata_rata);
        TextView km =
(TextView)findViewById(R.id.KecepatanMaksimal);

        String[] nama_list = new String[c.getCount()];
        int i= 0;
        if(c1.getCount()>0){
            if(c.moveToFirst())
            {
                kc.setText(c.getString(c.getColumnIndex("kecepatan")).toString());

                jr.setText(c.getString(c.getColumnIndex("jarak")).toString());

                wk.setText(c.getString(c.getColumnIndex("waktu_tempuh")).toString())

                kr.setText(c.getString(c.getColumnIndex("kecepatan_rata")).toString());
            }
        }
    }

```

```

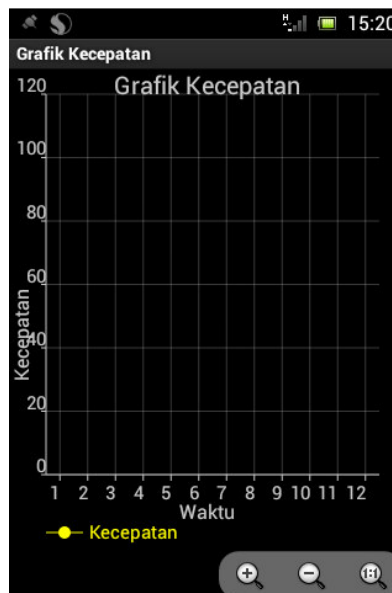
        km.setText(c.getString(c.getColumnIndex("kecepatan_max")).toString());
    }
}
if(c!=null){
    c.close();
}
if(c1!=null){
    c1.close();
}
if(db1!=null) db1.close();
}

```

Pada *source code* diatas menampilkan hasil penghitungan dari kecepatan, jarak, waktu tempuh, kecepatan rata-rata dan kecepatan maksimal. Selain itu ada pula source code untuk database dimana nanti hasil data statistik ini akan tersimpan pada database agar bisa di lihat kembali pada saat membuka menu history

4.2.3 Menu Pilih Grafik Kecepatan

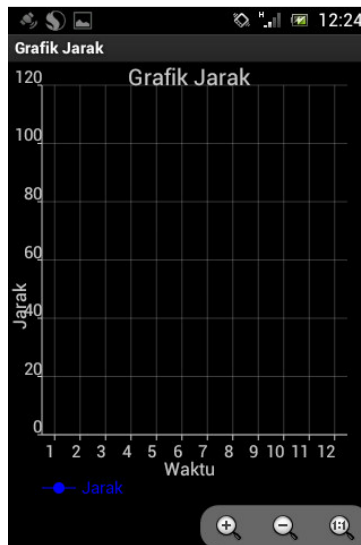
Halaman menu Grafik kecepatan ini menampilkan progress dari kecepatan pada saat aplikasi dijalankan. Menu halaman Grafik kecepatan seperti yang ditunjukkan pada gambar 4.4



Gambar 4.4 Tampil halaman Grafik kecepatan

4.2.2.5 Menu Pilih Grafik Jarak

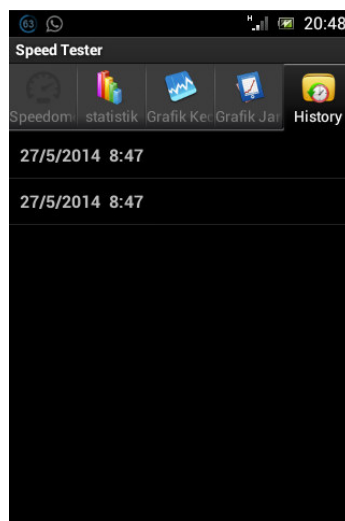
Halaman menu Grafik jarak ini menampilkan progress dari jarak pada saat aplikasi dijalankan. Menu halaman Grafik Jarak seperti yang ditunjukkan pada gambar 4.7



Gambar 4.5 Tampil halaman Grafik Jarak

4.2.2.6 Menu Pilih History

Halaman menu History ini menampilkan data yang tersimpan pada saat penggunaan aplikasi. Menu halaman History seperti yang ditunjukkan pada gambar 4.6



Gambar 4.6 Tampil halaman History

4.3 Pengujian pada *Smartphone* lain Berbasis Operasi android

Setelah melakukan pengujian pada *smartphone* android Sony Ericsson Live With Walkman/WT19i dengan hasil semua aplikasi Speedtester mampu berjalan, pengujian dilanjutkan dengan menggunakan *smartphone* berbasis android yang lain. Pengujian ini dilakukan agar dapat mengetahui pada perangkat mana saja aplikasi Speedtester ini dapat berjalan.

4.3.1 Perangkat yang di Ujikan

Perangkat berbasis android yang digunakan untuk melakukan pengujian memiliki spesifikasi *hardware* yang berbeda-beda. Perangkat berbasis android yang akan digunakan adalah Sony Ericsson Live With Walkman, Sony Xperia P, Xperia S, Xperia SL, Samsung Galaxy Ace, Samsung Galaxy Young.

Tabel .4.1.Perangkat Uji

No	Jenis Smartphone	Merk	Processor	RAM	Versi Android
1	Sony Ericsson Live With Walkman	Sony Ericsson	1 GHz	512MB	Ice Cream Sandwich 4.0
2	Sony <u>Xperia P</u>	Sony	<u>Dualcore 1GHz</u>	1GB	Ice Cream Sandwich 4.0
3	Sony <u>Xperia S</u>	Sony	<u>Dualcore 1,5 GHz</u>	1GB	<u>Jelly Bean 4.1</u>
4	Sony <u>Xperia SL</u>	Sony	<u>Dualcore 1,7 GHz</u>	1GB	Ice Cream Sandwich 4.0
5	Samsung Galaxy Ace	Samsung	800 MHz	278MB	<u>Froyo 2.2</u>
6	Samsung Galaxy Young	Samsung	1 GHz	768MB	Gingerbread 2.3

4.4 Hasil Pengujian

Dari hasil pengujian untuk aplikasi Speedtester berjalan dengan baik pada seluruh perangkat uji yang bervariasi dari sisi spesifikasi, *hardware*, maupun versi android yang digunakan. Berikut hasil pengujian aplikasi ini.

1. Aplikasi Speedtester ini dapat dijalankan pada android dengan versi yang berbeda, mulai dari versi 2.2 (Froyo), versi 2.3 (Gingerbread), versi 4.0 (Ice Cream Sandwich), dan versi 4.1 (Jelly Bean).
2. Aplikasi Speedtester ini dijalankan pada android dengan *smartphone* yang berbeda, mulai dari Samsung Galaxy Ace (Froyo), Samsung Galaxy Young

(Gingerbread), Sony Ericsson Live With Walkman, Sony Xperia P dan Sony Xperia SL (Ice Cream Sandwich), dan Sony Xperia S dan (Jelly Bean).

3. Aplikasi Speedtester ini cukup kompatibel dijalankan pada *smartphone* di atas dengan berbagai versi android yang berbeda, baik pada android versi 2.2 hingga android versi 4.1.
4. Antarmuka dari Aplikasi Speedtester ini sama baik ketika dijalankan pada android versi 2.2 maupun pada android versi 4.1.
5. Performa yang ditunjukkan oleh Aplikasi Speedtester ketika dijalankan pada android versi 2.2 maupun pada android versi 4.1 cukup baik dan dapat di eksekusi tanpa ada masalah.

Hasil pengujian aplikasi ini dapat dilihat pada tabel dibawah.

Tabel 4.2. Hasil Pengujian

No	Jenis Smartphone	Speedometer	Statistik	Grafik Kecepatan	Grafik Jarak	History
1	Sony Ericsson Live With Walkman	berjalan	berjalan	berjalan	berjalan	berjalan
2	Sony Xperia P	berjalan	berjalan	berjalan	berjalan	berjalan
3	Sony Xperia S	berjalan	berjalan	berjalan	berjalan	berjalan
4	Sony Xperia SL	berjalan	berjalan	berjalan	berjalan	berjalan
5	Samsung Galaxy Ace	berjalan	berjalan	berjalan	berjalan	berjalan
6	Samsung Galaxy Young	berjalan	berjalan	berjalan	berjalan	berjalan

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan serta pengujian yang dilakukan pada berbagai macam *smartphone* dengan versi Android dan spesifikasi hardware yang berbeda-beda. Dapat disimpulkan bahwa Aplikasi Speedtester Berbasis Android ini dapat memberikan hasil informasi berupa mencatat kecepatan, jarak, dan waktu, kecepatan maksimal, kecepatan rata-rata serta beberapa tambahan fitur berupa grafik sehingga user bisa melihat data progress aplikasi ini berjalan. Aplikasi ini juga dtambahkan fitur history sehingga masih bisa melihat data dari perjalanan sebelumnya. Meskipun aplikasi yang telah dikembangkan ini dapat berjalan dengan baik di seluruh perangkat uji, masih terdapat beberapa ruang untuk pengembangan ke depan. Penambahan fitur-fitur baru, seperti penambahan fitur maps pada aplikasi, dan sebagainya.

5.2 Saran

Mengingat banyak keterbatasan yang dialami penulis, maka penulis menyarankan untuk mengembangkan penelitian ini dimasa mendatang, seperti :

1. Menambahkan fitur lain sehingga menambah kelengkapan dari aplikasi ini.
2. Menyempurnakan tampilan serta isi dari aplikasi ini.