

**RANCANG BANGUN APLIKASI MOBILE PENYIMPAN
LOKASI PARKIR KENDARAAN MENGGUNAKAN APP
INVENTOR BERBASIS ANDROID**

SKRIPSI



**Disusun Oleh :
MOCHAMAD NUR ROFIK
09.18.012**

**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNIK INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013**

LEMBAR PERSETUJUAN DAN PENGESAHAN
RANCANG BANGUN APLIKASI MOBILE PENYIMPAN
LOKASI PARKIR KENDARAAN MENGGUNAKAN APP
INVENTOR BERBASIS ANDROID

SKRIPSI

Disusun dan Diajukan untuk melengkapi dan memenuhi persyaratan guna mencapai Gelar Sarjana Teknik Informatika Strata Satu (S-1)

Disusun Oleh :

Mochamad Nur Rofik

09.18.012

Diperiksa dan Disetujui,

Dosen Pembimbing I

Dosen Pembimbing II

Joseph Dedy Irawan, ST, MT

NIP. 197404162005031002

Survo Adi Wibowo, ST, MT

NIP. 1031000438

Ketua Jurusan Teknik Informatika S-1

Joseph Dedy Irawan, ST, MT

NIP. 197404162005031002

PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG



**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL
MALANG**

PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini:

Nama : Mochamad Nur Rofik
Nim : 09.18.012
Program Studi : Teknik Informatika S-1
Fakultas : Teknologi Industri

Menyatakan dengan sesungguhnya bahwa Skripsi saya yang berjudul:

**“RANCANG BANGUN APLIKASI MOBILE PENYIMPAN LOKASI
PARKIR KENDARAAN MENGGUNAKAN APP INVENTOR BERBASIS
ANDROID ”**

Adalah Skripsi saya sendiri bukan duplikat serta mengutip atau menyadur seluruhnya karya orang lain kecuali dari sumber aslinya.

Malang, 1 Agustus 2013

Yang membuat pernyataan



Mochamad Nur Rofik

RANCANG BANGUN APLIKASI MOBILE PENYIMPAN LOKASI PARKIR KENDARAAN MENGGUNAKAN APP INVENTOR BERBASIS ANDROID

Mochamad Nur Rofik

Program Studi Teknik Informatika S-1
Fakultas Teknologi Industri
Institut Teknologi Nasional Malang
Jl. Raya Karanglo Km 2 Malang
Email : rofiqkiditn@gmail.com

**Dosen Pembimbing: 1. Joseph Dedy Irawan, ST, MT
2. Suryo Adi Wibowo, ST, MT**

Abstrak

Penyimpan lokasi parkir kendaraan menggunakan app inventor berbasis android adalah aplikasi yang memanfaatkan teknologi (global positioning system) untuk menyimpan lokasi kendaraan. Selain itu juga sebagai pemunjuk arah dari lokasi dimana user berada ke lokasi parkir kendaraan user.

Pemulisan ini akan membahas tentang system yang digunakan untuk membangun aplikasi "Penyimpan Lokasi Parkir Kendaraan Menggunakan App Inventor Berbasis Android". Dalam proses pembuatannya menggunakan Aplikasi app inventor yang berbasis visual block programming. Metode yang digunakan dalam App Inventor ini menggunakan location sensor yang memiliki properti Latitude, Longitude sehingga bisa berkomunikasi dengan google maps, dan dapat mengakses informasi peta hingga ke nama jalan.

Dari hasil pengujian yang dilakukan diperoleh kesimpulan bahwa, Untuk hasil interface yang baik adalah pada dimensi layar 320x840 pixel, dan menggunakan operating sistem Android froyo 2.2 keatas. Dan rata-rata hasil rata-rata tingkat akurasi dengan pembanding pada smartphone blackberry 9000 diperoleh 0,00008 untuk latitude, dan 0,000003 untuk longitude.

Kata kunci: Android, App Inventor, Location sensor, Latitude, Longitude.



KATA PENGANTAR

Puji dan Syukur penyusun panjatkan kehadiran Tuhan Yang Maha Esa, karena telah memberikan rahmat-Nya sehingga penyusun dapat menyelesaikan skripsi dengan judul RANCANG BANGUN APLIKASI MOBILE PENYIMPAN LOKASI PARKIR KENDARAAN MENGGUNAKAN APP INVENTOR BERBASIS ANDROID dengan waktu yang ditentukan.

Penyusunan skripsi ini merupakan salah satu persyaratan untuk menyelesaikan program pendidikan Strata Satu (S-1) Teknik Informatika, Fakultas Teknologi Industri di Institut Teknologi Nasional Malang.

Pada kesempatan ini penyusun mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Ir. Soeparno, MT selaku Rektor ITN Malang.
2. Ir. Anang Subardi, MT selaku Dekan Fakultas Teknologi Industri ITN Malang dan.
3. Joseph Dedy Irawan, ST, MT selaku Ketua Jurusan Teknik Informatika ITN Malang.
4. Joseph Dedy Irawan, ST, MT selaku dosen pembimbing I
5. Suryo Adi Wibowo, ST, MT selaku dosen pembimbing II
6. Semua pihak yang telah membantu terselesaikannya skripsi ini.

Penyusun menyadari bahwa skripsi masih jauh dari sempurna, oleh karena itu penyusun mengharapkan kritik dan saran dari pembaca. Semoga skripsi ini bisa bermanfaat bagi para pembaca sekalian.

Malang, 1 Agustus 2013

Penyusun

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN	ii
ABSTRAK	iii
SURAT PERNYATAAN ORISINALITAS	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	viii
DAFTAR TABEL	ix
BAB I	1
PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4 Batasan Masalah	2
1.5 Metode Penelitian	2
1.6 Sistematika Penulisan	3
BAB II	4
LANDASAN TEORI	4
2.1 Parkir	4
2.2 Aplikasi Mobile	4
2.3 Android	5
2.4 Emulator	6
2.4.1 Alasan Dibuatnya Emulator	6
2.4.2 Kinerja Emulator.....	6
2.4.3 Memprogram Emulator	6
2.4.4 Jenis-jenis Emulator	7
2.4.5 Legalitas Penggunaan Emulator	7
2.5 App Inventor	9
2.5.1 Hal-hal yang bisa dilakukan dengan App Inventor	9
2.5.2 Database	10
2.6 Google Maps	12
2.6.1 Cara Kerja Google Maps	12
2.6.2 Google Maps API	13
2.7 Location Sensor	14
BAB III	16
PERANCANGAN APLIKASI	16
3.1 Pendahuluan	16
3.2 Identifikasi Masalah	16
3.3 Deskripsi Aplikasi	16
3.4 Sasaran Pengguna Aplikasi	17
3.5 Perancangan	17
3.6 Proses Pembuatan Aplikasi	17

3.6.1 Sistem Aplikasi	18
3.6.2 Kebutuhan Perangkat	18
3.7 Diagram Alir	19
3.7.1 Struktur Navigasi Aplikasi	21
3.7.2 Struktur Perancangan Aplikasi	23
BAB IV	33
IMPLEMENTASI DAN PENGUJIAN APLIKASI	33
4.1 Umum	33
4.2 Implementasi	33
4.3 Pengumpulan Data	34
4.3.1 Pembuatan Background Aplikasi	34
4.4 Pembuatan Halaman Screen1	35
4.4.1 Pembuatan Screen2	38
4.4.2 Pembuatan Screen3	39
4.4.3 Pembuatan Screen4	40
4.4.4 Pembuatan Screen6	41
4.5 Mendapatkan Lokasi Saat Ini	44
4.6 Menampilkan Data Lokasi Saat Ini	45
4.7 Menampilkan Arah Ke Lokasi Tersimpan	46
4.8 Menyimpan Data Lokasi Saat Ini pada TinyDB	49
4.9 Pengujian Aplikasi Pada Emulator	50
4.10 Transfer Aplikasi ke Handphone	53
4.11 Pengujian Aplikasi	54
4.11.1 Pengujian Pada handphone	54
4.11.2 Pengujian Waktu untuk transfer data Koordinat Lokasi	56
4.11.2. Pengujian Terhadap Akurasi data koordinat lokasi	58
BAB V	59
PENUTUP	59
5.1 Kesimpulan	59
5.2 Saran	59
DAFTAR PUSTAKA	61

DAFTAR GAMBAR

Gambar 2.1.	Bluestacks Produk Perusahaan Silicon Valley	8
Gambar 2.2.	Pembagian Gambar Peta Sebesar 256x256 Pixel	13
Gambar 3.1.	Diagram Alir Proses Pembuatan Aplikasi	17
Gambar 3.2.	Diagram Alir Aplikasi	20
Gambar 3.3.	Struktur Navigasi Aplikasi	22
Gambar 3.4.	Desain Halaman Pendahuluan	23
Gambar 3.5.	Desain Halaman Utama	24
Gambar 3.6.	Desain Halaman Aplikasi Penunjuk Koordinat Aplikasi	25
Gambar 3.7.	Desain Halaman Contact Person	25
Gambar 3.8.	Desain Halaman Help	26
Gambar 3.9.	Menyetujui Term of Service	27
Gambar 3.10.	Gambar Setting	28
Gambar 3.11.	USB Debugging	29
Gambar 3.12.	Jendela Developing	29
Gambar 3.13.	Membuat Projek Baru	30
Gambar 3.14.	Projek Baru	30
Gambar 3.15.	Komponen Desainer	31
Gambar 3.16.	Block Editor	32
Gambar 3.17.	Emulator	32
Gambar 4.1.	Lembar Kerja Baru	34
Gambar 4.2.	Pembuatan Background Pada Adobe Photoshop	35
Gambar 4.3.	Konfigurasi Screen pada App Inventor	36
Gambar 4.4.	Tombol Masuk	36
Gambar 4.5.	Pemberian Background Pada Screen1	37
Gambar 4.6.	Ukuran Tombol Masuk 150x150 pixel	37
Gambar 4.7.	Proses Pembuatan Screen2 Menu Utama.....	39
Gambar 4.8.	Proses Pembuatan Halaman Screen3	40
Gambar 4.9.	Tampilan Halaman Contact Person pada Screen4	41
Gambar 4.10.	Antar Muka Aplikasi pada Emulator	42
Gambar 4.11.	Komponen Desainer Antar Muka Aplikasi	43
Gambar 4.12.	Set Properti	43
Gambar 4.13.	Block Editor Untuk Mendapatkan Lokasi.....	45
Gambar 4.14.	Event Block Pada Tombol Simpan	46
Gambar 4.15.	Block Editor Untuk Tombol Arah Pada Screen6.....	48
Gambar 4.16.	Tampilan Pada Emulator	48
Gambar 4.17.	Block Event Menyimpan Address Lokasi.....	50
Gambar 4.18.	Tampilan Opening	50
Gambar 4.19.	Tampilan Menu Utama	51
Gambar 4.20.	Antar Muka Aplikasi Penyimpan Lokasi Kendaraan	52
Gambar 4.21.	Tampilan Contact person Pada Pengujian Emulator	52
Gambar 4.22.	Tampilan Halaman Help Pada Emulator	53

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dalam era *modern* seperti sekarang ini kebutuhan manusia akan teknologi semakin meningkat. Untuk membantu mempermudah aktifitas sehari – hari mereka dalam melakukan pekerjaan, sekedar hiburan pelepas penat, mencari informasi - informasi penting, atau bahkan teknologi menjadi gaya hidup masyarakat modern. *Android* adalah sistem operasi yang berbasis *linux* untuk telephone seluler. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri.

Pada saat ini ketersediaan tempat parkir merupakan salah satu kebutuhan dalam sistem transportasi, karena setiap perjalanan dengan kendaraan pribadi umumnya selalu dimulai dan diakhiri di tempat parkir. Kebutuhan ruang parkir cenderung meningkat dari tahun ke tahun. Seiring dengan bertambahnya jumlah kendaraan pribadi terutama mobil. Ketidak seimbangan ini sering dimanfaatkan untuk mendapatkan keuntungan lebih dari praktek parkir liar. Lokasi parkir seharusnya tidak terlalu jauh dari tempat yang dituju karena itu akan memberikan rasa tidak aman atau kecadaan lain yang membuat pemilik kendaraan khawatir. Tentunya parkir liar ini sering membuat kebingungan para pemilik kendaraan untuk mencari dimana lokasi parkir kendaraannya.

Dari permasalahan di atas maka muncul sebuah ide tentang bagaimana membuat “ RANCANG BANGUN APLIKASI MOBILE PENYIMPAN LOKASI PARKIR KENDARAAN MENGGUNAKAN APP INVENTOR BERBASIS ANDROID”, yang mana nantinya akan memudahkan pengguna untuk mengetahui lokasi memarkir kendaraannya.

App inventor adalah sebuah *tool* untuk membuat aplikasi *android*, secara sederhana kita bisa menyebutnya tanpa menuliskan kode program. *App inventor* bisa juga digunakan mengasah logika seperti halnya menyusun sebuah *puzzle*. Dengan bantuan peta dari google maps dan *Global Positioning System* pengguna aplikasi dapat mengetahui lokasi awal tempat memarkir kendaraan, karena tempat yang belum pernah mereka kunjungi atau belum mereka hafal.

1.2 Rumusan Masalah

Bagaimana membuat aplikasi *mobile* berbasis sistem operasi *android* yang bisa membantu pengguna untuk mengetahui lokasi dimana kendaraannya parkir.

1.3 Tujuan Penelitian

Tujuan perancangan ini adalah mampu menghasilkan aplikasi berbasis *android* yang akan membantu kita menemukan tempat parkir kendaraan yang bersifat *mobile*.

1.4 Batasan Masalah

Agar permasalahan mengarah sesuai dengan tujuan yang diharapkan, maka pembahasan dibatasi oleh hal-hal sebagai berikut:

1. Aplikasi parkir ini hanya berlaku untuk pengguna *android* froyo 2.2 ke atas.
2. Untuk saat ini aplikasi ini diterapkan untuk tempat parkir yang berada di ruang terbuka (*outdoor*).
3. Tidak dapat mendeteksi kendaraan dalam gedung bertingkat dan hanya dapat mendeteksi koordinat Latitude dan Longitude.
4. Pada Aplikasi ini memiliki tingkat ketelitian akses data hanya sampai ke nama jalan.
5. Informasi yang terdapat dalam aplikasi ini adalah informasi lokasi awal memarkir kendaraan, lokasi tersimpan dan menunjukkan arah dari lokasi terakhir berada sampai lokasi awal tempat memarkir kendaraan.

1.5 Metode Penelitian

Tahapan-tahapan pada metode penelitian yang digunakan dalam pelaksanaan penelitian ini, meliputi :

1. Pengumpulan Data
Data yang dikumpulkan, merupakan data yang diperoleh dari literatur-literatur, perpustakaan dan instansi lainnya yang terkait dengan penelitian ini.
2. Analisis Aplikasi
Populasi dari analisis data ini, adalah semua data yang berkaitan dengan lokasi parkir dan pengumpulan data pendukung pembuatan aplikasi yang akan digunakan untuk mendefinisikan aplikasi.
3. Perancangan dan Implementasi

Perancangan mekanisme aplikasi dilakukan berdasarkan analisis yang telah diperoleh untuk diimplementasikan ke dalam aplikasi.

4. Evaluasi dan Perbaikan

Implementasi ke dalam aplikasi yang dibuat, akan dievaluasi dalam bentuk pengujian aplikasi, serta perbaikan aplikasi apabila diperlukan.

1.6. Sistematika Penulisan

Untuk mempermudah dan memahami pembahasan pada penulisan skripsi ini, maka sistematika penulisan yang diperoleh sebagai berikut :

- BAB I : Pendahuluan
Berisi latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, metode penelitian dan sistematika penulisan.
- BAB II : Dasar Teori
Berisi tinjauan pustaka mengenai permasalahan yang berhubungan dengan penelitian ini.
- BAB III : Perancangan Aplikasi
Berisi mengenai perancangan aplikasi panduan wisata kabupaten malang berbasis *mobile phone* yang akan dibuat.
- BAB IV : Implementasi dan Pengujian
Berisi implementasi terhadap proses panduan wisata kabupaten malang berbasis aplikasi *mobile* mulai dari awal hingga akhir, serta melakukan pengujian terhadap aplikasi penyimpan lokasi kendaraan menggunakan *app inventor* berbasis *android*.
- BAB V : Kesimpulan dan Saran
Berisi kesimpulan dari hasil penelitian yang dilakukan dan saran yang dapat digunakan sebagai bahan pertimbangan untuk pengembangan penelitian selanjutnya.

BAB II LANDASAN TEORI

2.1 Parkir

Parkir adalah keadaan tidak bergerak suatu kendaraan yang bersifat sementara karena ditinggalkan oleh pengemudinya. Secara hukum dilarang untuk parkir di tengah jalan raya, namun parkir di sisi jalan umumnya diperbolehkan. Fasilitas parkir dibangun bersama-sama dengan kebanyakan gedung, untuk memfasilitasi kendaraan pemakai gedung. Termasuk dalam pengertian parkir adalah setiap kendaraan yang berhenti pada tempat-tempat tertentu baik yang dinyatakan dengan rambu lalu lintas ataupun tidak, serta tidak semata-mata untuk kepentingan menaikkan dan/atau menurunkan orang dan/atau barang. ¹⁶¹

2.2 Aplikasi *Mobile*

Aplikasi *mobile* adalah aplikasi yang melibatkan piranti bergerak dan melibatkan media komunikasi *nirkabel*. Aplikasi *mobile* dapat digunakan walaupun pengguna berpindah-pindah dari suatu tempat ke tempat lain. Aplikasi *mobile* digunakan pada perangkat seperti telepon seluler, pager, PDA (*Personal Digital Assistant*).

Mayoritas aplikasi *mobile* menggunakan *Java Micro Edition (JME) platform*, yang dikembangkan untuk perangkat kecil seperti ponsel dan set-top box. *Java Micro Edition (JME)* diperkecil menggunakan subset dari Java SE komponen, virtual *machine* dan API. Ini juga mendefinisikan API yang ditargetkan pada konsumen khusus *mobile* dan perangkat embedded. NetBeans *Java Micro Edition (JME)* mendukung dua konfigurasi dasar dari platform *Java Micro Edition (JME)*, *Connected Limited Device Configuration (CLDC)* dan *CDC*.

Karakteristik perangkat *mobile*:

1. Ukuran yang kecil

Aplikasi *mobile* memiliki ukuran yang kecil. Konsumen menginginkan ukuran kecil untuk kenyamanan mobilitas mereka

2. Memori yang terbatas

Perangkat *mobile* juga memiliki memori yang kecil, yaitu *primary* (RAM) dan *secondary* (disk)

3. Daya proses yang terbatas

Sistem *mobile* tidak setangguh desktop dalam masalah daya proses karena sistem *mobile* memiliki kemampuan *processor* yang lebih kecil.

4. Mengonsumsi daya yang rendah

Perangkat *mobile* menghabiskan daya lebih sedikit dibandingkan dengan perangkat desktop.

5. Kuat dan dapat diandalkan

Karena perangkat *mobile* selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapi benturan-benturan, gerakan, dan sesekali tetesan air.

6. Konektivitas yang terbatas

Perangkat *mobile* memiliki bandwidth rendah, beberapa dari mereka bahkan tidak terkoneksi.

7. Masa hidup yang pendek

Perangkat-perangkat konsumen ini menyala dalam hitungan detik kebanyakan dari mereka selalu menyala. ^[1]

2.3. Android

Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli *Android Inc.*, pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan *Android*, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Android juga merupakan OS *open source* maksudnya adalah OS yang membuka atau membebaskan source codenya untuk dilihat orang lain dan membiarkan orang lain untuk mengetahui cara kerja dari OS serta dapat membuat aplikasi baru atau menyempurnakan dari aplikasi yang sudah ada, dan satu keunggulan lagi adalah *open source* bisa didapatkan dan dipergunakan secara gratis atau *free*. ^{[1][12]}

2.4. Emulator

Emulator lebih tepatnya piranti lunak *emulator* memungkinkan suatu program atau piranti lunak yang dibuat pada awalnya oleh suatu *system computer* (arsitektur dan *system operasi*) dan untuk dijalankan dalam *system* itu (atau dijalankan dalam suatu sistem yang didedikasikan), dapat dijalankan dalam sistem komputer yang sama sekali berbeda. Sebagai contoh suatu aplikasi android yang sudah dibuat dapat dijalankan tanpa perlu memakai handphone atau *device* lainnya yang berbasis *android* hanya dengan menggunakan *emulator* yang sudah terinstall pada PC atau laptop. ^[13]

2.4.1. Alasan Dibuatnya Emulator

Terdapat beberapa alasan dibuatnya *emulator*, beberapa di antaranya adalah :

- a. Para pembuat aplikasi *android* jika tidak mempunyai *device* atau *handshet android* seperti *smartphone* bisa menjalankan aplikasi yang sudah dibuat melalui *emulator*, jadi bisa lebih praktis dan mudah.
- b. Menjalankan piranti lunak yang dibuat hanya untuk *platform-platform* tertentu.
- c. Memeriksa program-program yang dibuat oleh platform berbeda untuk dijalankan pada *platform* yang lain. Sebagai contoh misalnya pembuatan program atau piranti lunak pada *android* atau telepon genggam berbasis *android* yang terlebih dahulu diemulasikan di komputer personal (PC). ^[13]

2.4.2. Kinerja Emulator

Kebanyakan *emulator* tidak mampu untuk secara sempurna meng-emulate sistem yang hendak ditirunya. Contoh dari ketidaksempurnaan tersebut adalah *problem* pada *timing* (*framerate* yang lambat), gambar atau efek video yang tidak benar, suara yang kacau atau bahkan tidak bersuara sama sekali. ^[13]

2.4.3. Memprogram Emulator

Memprogram *emulator* adalah hal yang paling amat sulit dan dimulai dari mengumpulkan informasi tentang sistem yang hendak di-emulate dan mengakali komputer agar mampu meniru *hardware* dan instruksi-instruksi yang dijalankan oleh sistem tersebut.

2.4.4. Jenis-jenis emulator

Emulator terdiri dari 2 jenis: *single-system/dedicated emulator* dan *multi-system emulator*. *Single-system/dedicated emulator* hanya dapat meng-emulate 1 jenis sistem, sedangkan *multi-system emulator* dapat meng-emulate beberapa jenis sistem sekaligus. Pada dasarnya kualitas *single-system/dedicated emulator* lebih baik daripada *multi-system emulator* karena hanya didesain khusus untuk 1 jenis sistem sehingga kemungkinan untuk terjadi *compatibility problem* dapat diminimalisasi. Selain itu kebutuhan resources untuk *single-system/dedicated emulator* biasanya lebih kecil dibandingkan *multi-system emulator*. Mungkin Anda bertanya, apakah semua sistem sudah diemulate? Jawabannya, tergantung. Ada 2 faktor utama yang menentukan kemungkinan suatu sistem diemulate atau tidak:

- a. Kepopuleran: Semakin populer suatu sistem, maka semakin banyak orang yang berusaha untuk meng-emulate sistem tersebut.
- b. Tersedianya informasi: Semakin banyak informasi yang tersedia tentang suatu sistem (baik *hardware* maupun *software*) akan amat membantu proses penulisan *emulator*.^[13]

2.4.5. Legalitas Penggunaan Emulator

Sekarang, apakah *emulator* itu legal dan tidak melanggar hukum. Sejauh mata memandang sebenarnya *emulator* itu sendiri tidak melanggar hukum sejauh *emulator* tersebut tidak mengandung materi yang di-*copyright* seperti misalnya BIOS *image* sistem tertentu.

Terdapat berbagai macam *emulator* yang berkembang dan memiliki keunggulan masing-masing, tergantung kebutuhan pengguna. Antara lain :

1. Youwave Android

YouWave *Android* adalah *android* simulator yang bisa kita jalankan di PC atau Laptop . Kita juga bisa mengkonekkan langsung dengan internet. Youwave ini mempunyai beberapa fitur diantaranya :

- a. *Supports Android 2.3 Gingerbread*
- b. *Runs on Windows XP/Vista/7, 32/64 bit*

- c. *Simulated SD card functionality - enables game saving*
- d. *Saved State - enables fast restart*
- e. *Enables multi-player online games*
- f. *Dynamic rotating - phone-like instant response*
- g. *Volume control buttons*
- h. *Retractable control panel.* ^[13]

2. BlueStacks

Bluestacks adalah perusahaan perangkat lunak berbasis Silicon Valley yang menghasilkan Bluestacks App Player Bluestacks Cloud Connect dan kedua produk memungkinkan untuk dijalankan pada PC Windows, Komputer Macintos dan Windows tablet. Perusahaan ini didirikan pada tahun 2009 oleh Rosen Sharma, mantan CTO McAfee dan anggota dewan Cloud.com



Gambar 2.1 BlueStacks merupakan produk perusahaan Silicon Valley.

Bluestacks pada gambar diatas resmi diluncurkan pada 25 Mei 2011, pada konferensi Synergy Citrix di San Francisco. CEO Citrix Mark Templeton menunjukkan versi awal BlueStacks panggung dan mengumumkan bahwa perusahaan telah membentuk kemitraan. Web Player dan Cloud Connect resmi diluncurkan ke publik alpha pada 11 Oktober 2011. App Player adalah bagian *download* perangkat lunak Windows yang datang dengan beberapa aplikasi (termasuk Bloomberg News, Words With Friends dan Drag Racing).

Pada tanggal 27 Maret 2012, perusahaan merilis versi beta-1 perangkat lunak Pemain App nya. Program ini telah *download* lebih dari satu juta kali dalam 10 hari pertama. Hal ini diikuti oleh *investasi* yang tidak diungkapkan dari Qualcomm.

Pada 27 Desember 2012, perusahaan merilis versi beta dari software Pemain App untuk Mac OS. Perusahaan mengatakan Hal ini dapat menjalankan lebih dari 750.000 aplikasi *Android* di Mac. Pada tanggal 9 Mei 2013, perusahaan mengumumkan

produk terbaru mereka: GamePop. GamePop akan memungkinkan pengguna untuk bermain game *mobile* 500 top di TV. Mereka juga mengumumkan akan menjadi langganan model dan pengguna akan menerima lebih dari \$ 250 senilai *game* dibayar dengan langganan mereka. ^[13]

2.5 *App Inventor*

App Inventor adalah sebuah tool untuk membuat aplikasi *android*, yang menyenangkan dari tool ini adalah karena berbasis *visual block programming*, jadi kita bisa membuat aplikasi tanpa kode satupun. Disebut Visual Programming karena kita akan melihat, menggunakan, menyusun dan mendrag-drops blok yang merupakan *symbol – symbol* perintah dan fungsi event handler tertentu dalam membuat aplikasi, dan secara sederhana kita bisa menyebutnya tanpa menuliskan kode program atau *coding less*.

Jadi kita tidak harus memiliki *basic programming*, mengerti kode – kode atau berkecimpung dalam dunia TI untuk membuat aplikasi dengan *App Inventor*. Bahkan *App Inventor* tidak hanya untuk membuat aplikasi karena bisa digunakan untuk mengasah logika seperti halnya menyusun sebuah *puzzle* .

Framework visual programming ini terkait dengan bahasa pemrograman Scratch dari MIT yang secara *spesifik* merupakan dari Open Block yang didistribusikan oleh MIT Scheller Teacher Education Program yang diambil dari riset yang dilakukan oleh Ricarose Roque. *App Inventor* menggunakan Kawa Language Framework dan Kawa's dialect yang didevelop oleh Per Bothner dan didistribusikan sebagai bagian dari GNU OS oleh Free Software Foundation. ^{[4][5]}

2.5.1 Hal – hal yang bisa dilakukan dengan *App Inventor*

Terdapat beberapa hal yang bisa dilakukan dengan *App Inventor* yaitu:

1. Just for fun

Katakanlah anda memiliki Handshet *Android* atau baru memasang virtual handshet *Android* pada computer anda atau bahkan tidak sama sekali. *App Inventor* akan bisa menjadi sesuatu yang menyenangkan seperti halnya anda menyukai mengedit gambar, bermain *puzzle*, mengasah otak anda seperti mengisi TTS, atau belajar sebagai basis menguatkan kemampuan logika anda.

2. Learning tool

Entah anda seorang pelajar, mahasiswa, guru, dosen atau *hobbist* anda akan bisa menjadikan *App Inventor* sebagai tool belajar anda. Jika anda seorang guru atau dosen anda akan membuat *App Inventor* menjadi alat pengajar yang hebat karena visualisasi akan mempermudah siswa memahami dan menguasai apa yang anda ajarkan.

3. Membuat aplikasi

Mulai dari membuat *prototype*, aplikasi untuk kebutuhan pribadi dan organisasi atau aplikasi serius yang bisa anda jual. Apabila anda telah mencoba menggunakan *App Inventor*, anda akan merasakan betapa mudahnya, tentu disamping karena berbasis *visual drag drops* ini juga karena :

- a. Tidak perlu menghafal dan mengingat instruksi atau kode – kode program.
- b. Komponen & Blok event tersedia dengan lengkap anda tinggal menggunakannya. Layaknya sebuah objek anda tinggal merangkainya dengan komponen dan blok yang sesuai fungsinya tinggal meletakkannya seperti *puzzle*, merubah propertinya misal *member* nilai angka untuk mengeset *timer* dll.
- c. Event handler akan memudahkan anda dalam menangani setiap “kejadian” atau event yang anda ingin handle, misal membuat sebuah aplikasi untuk menangani sms, dalam *App Inventor* anda tinggal mengambil *drag drops* blok yang menangani *event* sms.

2.5.2. Database

Database bisa dikatakan sebagai sesuatu sistem yang menyimpan dan mengatur data sehingga data tersebut bisa diambil atau dicari dengan mudah dan efisien. Sebagai contoh sederhana dari database adalah penyimpanan sms yang sering kita lihat dan gunakan.

Kita telah mengenal komponen variable dan komponen properti, keduanya juga menyimpan data informasi. Akan tetapi begitu kita menutup aplikasi maka segera data itu akan hilang, ini berbeda dengan database. Karena data akan tetap tersimpan dan tidak hilang, seperti halnya sms kita pada handset yang tetap ada bahkan ketika handset kita matikan.

Appinventor menyediakan komponen yang mengguranagi kerumitan dalam menggunakan database, baik untuk menyimpan maupun mengambil kembali data dari database. Dan pada *appinventor* kita dapat membuat aplikasi yang menyimpan data langsung pada perangkat *android*, ataupun aplikasi yang membagi data dengan perangkat lain, dan menyimpan data kedalam database web terpusat.

App Inventor menyediakan dua komponen untuk memfasilitasi database :

1. TinyDB

App inventor menyediakan dua komponen untuk memfasilitasi aktifitas database : **TinyDB** dan **TinyWebDB**. **TinyDB** digunakan untuk menyimpan data secara langsung pada perangkat *android*, ini berguna untuk aplikasi yang tidak perlu membagi data informasinya dengan perangkat lain, seperti dalam aplikasi SuperSMS. Untuk memasukan data kedalam database kita memerlukan blok **TinyDB.StoreValue**.

2. TinyWebDB

TinyWebDB digunakan untuk menyimpan data dalam database web pada server online, dimana kemudian akan bisa diakses oleh perangkat lain. Database web ini misalnya digunakan permainan multi user dan aplikasi di mana pengguna dapat memasukan dan berbagi informasi, seperti aplikasi Quizkita kemudian bisa diakses bersama – sama.

Secara default oleh *app inventor* saat ini di set pada situs resmi *appinventor* <http://appinventortinywebdb.appspot.com> dan bisa diakses bersama – sama pengguna *app inventor* di seluruh dunia. **TinyWebDB** menggunakan **TinyWebDB.StoreValue**, untuk menyimpan data dan juga sama menggunakan pola **tag** dan **valueToStore**.

TinyWebDB menggunakan **TinyWebDB.StoreValue** untuk menyimpan data dan juga sama menggunakan pola **tag** dan **valueToStore**. Dan untuk mengambil data kembali menggunakan blok **TinyWebDB.GetValue** yang kemudian juga memerlukan blok event handler **TinyWebDB.GotValue** untuk menghandle data ketika data tersebut bisa diambil dari webdatabasc (string

disebut *callback procedure*). Kemudian terdapat event argument **tagFromWebDB** dan **valueFromWebDB**, ini bekerja mirip dengan yang terdapat pada TinyDB. ^{[4][5]}

2.6. Google Maps

Google Maps adalah layanan aplikasi peta *online* yang disediakan oleh *Google* secara gratis. Layanan peta *Google Maps* secara resmi dapat diakses melalui situs <http://maps.google.com>. Pada situs tersebut dapat dilihat informasi geografis pada hampir semua permukaan di bumi kecuali daerah kutub utara dan selatan. Layanan ini dibuat sangat interaktif, karena di dalamnya peta dapat digeser sesuai keinginan pengguna, mengubah level *zoom*, serta mengubah tampilan jenis peta.

Google Maps mempunyai banyak fasilitas yang dapat dipergunakan misalnya pencarian lokasi dengan memasukkan kata kunci, kata kunci yang dimaksud seperti nama tempat, kota, atau jalan, fasilitas lainnya yaitu perhitungan rute perjalanan dari satu tempat ke tempat lainnya. ^[3]

2.6.1. Cara Kerja Google Maps

Google Maps dibuat dengan menggunakan kombinasi dari gambar peta, database, serta obyek-obyek interaktif yang dibuat dengan bahasa pemrograman *HTML*, *Javascript* dan *AJAX*, serta beberapa bahasa pemrograman lainnya.

Gambar-gambar yang muncul pada peta merupakan hasil komunikasi dengan database pada *web server Google* untuk menampilkan gabungan dari potongan potongan gambar yang diminta. Keseluruhan citra yang ada diintegrasikan ke dalam database pada *Google Server*, yang nantinya akan dapat dipanggil sesuai kebutuhan permintaan. Bagian- bagian gambar map merupakan gabungan dari potongan gambar-gambar bertipe *PNG* yang disebut *tile* yang berukuran 256 x 256 pixel seperti yang terdapat pada Gambar berikut.



Gambar2.2 Pembagian gambar peta sebesar 256 x 256 pixel

Tiap-tiap potongan diatas, mewakili gambar tertentu dalam *longitude*, *latitude* dan *zoom level* tertentu. Kode *Javascript* yang digunakan untuk menampilkan peta *Google Maps* diambil dari link *URL*. Jadi untuk menampilkan peta suatu lokasi yang diinginkan, dapat dengan cara mengirimkan *URL* yang diinginkan, misalnya:

```
http://maps.google.com/?ie=UTF8&ll=-6.500899,106.918945  
spn=4.327078,4.938354&z=8
```

- ie=UTF8*, adalah karakter *encoding* untuk map.
- ll=-6.500899,106.918945*, adalah posisi titik tengah peta yaitu *latitude* (lintang) dan *longitude* (bujur) dari peta yang ditampilkan, pada *link* diatas posisi titik tengah peta pada *latitude*: -6.500899 dan *logitude*: 106.918945.
- spn=4.327078,4.938354*, adalah rentang dari *latitude* dan *longitude*-nya.
- z=8*, adalah tingkatan/level *zoom* peta.

2.6.2. Google Maps API (*Application Programming Interface*)

API atau *Application Programming Interface* merupakan suatu dokumentasi yang terdiri dari *interface*, fungsi, kelas, struktur dan sebagainya untuk membangun sebuah perangkat lunak. Dengan adanya *API* ini, maka memudahkan programmer untuk “membongkar” suatu *software* untuk kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak yang lain. *API* dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya yang memungkinkan programmer menggunakan sistem *function*. Proses ini dikelola melalui *operating system*. Keunggulan dari *API* ini adalah

memungkinkan suatu aplikasi dengan aplikasi lainnya dapat saling berhubungan dan berinteraksi. Bahasa pemrograman yang digunakan oleh *Google Maps* yang terdiri dari *HTML*, *Javascript* dan *AJAX* serta *XML*, memungkinkan untuk menampilkan peta *Google Maps* di *website* lain.

Google juga menyediakan layanan *Google Maps API* yang memungkinkan para pengembang untuk mengintegrasikan *Google Maps* ke dalam *website* masing-masing dengan menambahkan data *point* sendiri. Dengan menggunakan *Google Maps API*, *Google Maps* dapat ditampilkan pada *web site* eksternal. Agar aplikasi *Google Maps* dapat muncul di *website* tertentu, diperlukan adanya *API key*. *API key* merupakan kode unik yang digenerasikan oleh *google* untuk suatu *website* tertentu, agar *server Google Maps* dapat mengenali. [2][7]

2.7 Location Sensor

Ketika menggunakan komponen *Location Sensor* ini, maka salah satunya yang akan kita temui adalah (*Global Positioning System*). Teknologi ini pertama kali digunakan oleh Departemen Pertahanan Amerika Serikat -*United States Departement of Defense* (DOD) untuk kebutuhan militer. Sistem ini mulai digunakan sejak tahun 1980, namun pemakaian secara umum oleh *public* baru sekitar tahun 1990-an. Informasi data ditransmisikan oleh beberapa satelit yang umumnya untuk mendapatkan ketepatan dibutuhkan sekitar tiga buah satelit. Sehingga *global positioning system receiver* mampu mengkalukulasi dan menampilkan seakurat posisi, kecepatan dan informasi waktu kepada pengguna. Tetapi kadang kita akan terhalang oleh gedung, atau bangunan lain, atau juga cuaca yang tidak mendukung.

Sehingga dari karakteristik, tersebut yang harus kita pahami adalah kadang kita tidak cepat akan mendapatkan layanan data koordinat lokasi, keakuratan yang kurang atau bahkan tidak bisa mendapatkan layanan data koordinat lokasi sama sekali. Pada *App Inventor*, komponen *LocationSensor* memiliki properti *Latitude*, *Longitude* dan *Altitude*. Juga bisa berkomunikasi dengan *Google Maps*, sehingga bisa mengakses informasi peta hingga ke nama jalan.

Latitude adalah seberapa jauh posisi utara atau selatan kita, dibandingkan dengan garis *equator* dan garis khatulistiwa, nilainya untuk utara semain positif dan ke selatan semakin negatif dari -90 ke 90. *Longitude* adalah seberapa jauh posisi timur atau barat, dari garis *meridian* atau garis bujur, nilainya -180 hingga 180, koordinat timur

memiliki nilai positif, sedangkan posisi barat memiliki *negative*. Sedangkan kedua garis yang seolah-olah memotong bumi menjadi empat bagian itu sebenarnya adalah garis khayal atau garis *imajinasi*, dengan titik 0.0 ditetapkan pada sebuah kota di dekat London Inggris bernama Greenwich.

LocationSensor menggunakan blok *event handler* *LocationSensor*. *Locationchanged* untuk menghandle pembacaan data Koordinat Lokasi setiap kali posisi kita berubah. Perlu diingat mungkin karena kita dalam ruangan bisa jadi pembacaan data Koordinat tidak bisa dilakukan, itulah kenapa dibuat suatu variabel dengan nilai *default* "tidak diketahui" agar ketika data Koordinat tidak bisa diakses maka akan didapat informasi tidak diketahui tersebut. ^{[4][5]}

BAB III PERANCANGAN APLIKASI

3.1. Pendahuluan

Analisis dan perancangan aplikasi adalah merupakan tahapan perencanaan dari seluruh proses dalam membangun aplikasi *mobile* penyimpan lokasi parkir kendaraan menggunakan *app inventor* berbasis *android*. Analisis adalah merupakan suatu kegiatan untuk memaparkan suatu peristiwa yang dihadapi, sedangkan perancangan merupakan solusi yang dihasilkan untuk memenuhi dan memecahkan suatu masalah.

3.2. Identifikasi Masalah

Perencanaan membangun aplikasi *mobile* penyimpan lokasi parkir kendaraan menggunakan *app inventor* berbasis *android*, karena terdapat masalah-masalah yang terjadi pada pemilik kendaraan saat parkir.

Permasalahan yang terjadi pada pemilik kendaraan, antara lain :

1. Pemilik kendaraan belum terlalu mengetahui lokasi yang di datangnya atau *relatif* asing baginya.
2. Tidak efektifnya area parkir dan tukang parkir dalam membantu menemukan tempat parkir kendaraan.
3. Dibutuhkannya aplikasi alternatif sebagai media penunjuk arah yang mudah dan menarik.

3.3. Deskripsi Aplikasi

Penyimpan Lokasi Parkir Kendaraan Menggunakan *app inventor*, bertujuan untuk membantu mempermudah para pemilik kendaraan untuk menemukan dimana lokasi memarkir kendaraannya dan menunjukkan *route* menuju ke lokasi memarkir kendaraan jika pemilik kendaraan tidak tau sama sekali tempat tersebut. Aplikasi ini merupakan media penunjuk arah bagi para pemilik kendaraan dengan menggunakan *mobile phone* sebagai media penunjangnya.

Objek-objek yang terdapat pada aplikasi penyimpan lokasi parkir kendaraan, disajikan dalam berbagai bentuk *visual*, sehingga dapat terjadinya proses penjelasan dengan interaksi yang aktif dan saling berkaitan, sehingga aplikasi ini tidak terlihat monoton dan menarik bagi *user*.

3.4. Sasaran Pengguna Aplikasi

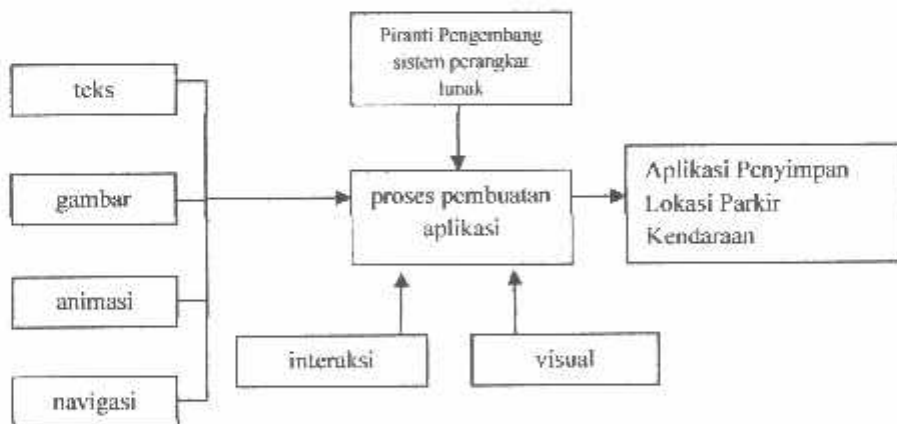
Sasaran dalam penggunaan Aplikasi *Mobile* Penyimpan Lokasi Parkir Kendaraan Menggunakan *App Inventor* Berbasis *Android* ditujukan sebagai sarana penunjuk arah bagi para pemilik kendaraan untuk menemukan lokasi parkir kendaraannya secara mudah dengan menggunakan media *digital*.

3.5. Perancangan

Perancangan aplikasi merupakan proses awal dari pembuatan aplikasi. Tahap perancangan aplikasi, dimulai dari tampilan awal dan tampilan utama yang menampilkan beberapa menu pilihan.

3.6. Proses Pembuatan Aplikasi

Pada Penyimpan Lokasi Parkir Kendaraan Menggunakan *App Inventor* Berbasis *Android* disajikan dalam berbagai bentuk *visual* yang interaktif dengan dilengkapi kontrol *navigasi*. Pada proses pembuatannya, elemen multimedia yang disusun akan digunakan sebagai bahan untuk proses produksi yang didukung oleh perangkat lunak, serta disisipkan unsur *interaksi*, dimana unsur-unsur tersebut akan dibuat oleh kreatifitas penulis agar dapat menghasilkan aplikasi penunjuk arah yang baik dan menarik bagi pengguna. Diagram alir dalam proses pembuatan Aplikasi Penyimpan Lokasi Parkir Kendaraan Menggunakan *App Inventor* Berbasis *Android*, dapat dilihat pada Gambar dibawah ini .



Gambar 3.1. Diagram Alir Proses Pembuatan Aplikasi.

3.6.1. Sistem Aplikasi

Penyimpan Lokasi Parkir Kendaraan Menggunakan *App Inventor* Berbasis *Android* ini memiliki beberapa fungsi sistem, antara lain :

1. Menyajikan penunjuk arah *visual* yang dikonfigurasi dengan tombol yang ada pada aplikasi, dimana pada saat pemilik kendaraan melakukan interaksi dengan tombol tersebut, maka pemilik kendaraan dapat mengenali, mengetahui dan memahami maksud dari setiap tombol yang ada karena dalam aplikasi ini dibuat dengan bahasa Indonesia, sedangkan *visual* dibuat agar *user* lebih mengetahui gambaran daerah tersebut secara nyata.
2. Terdapat menu peta *google maps* yang akan menunjukkan jalan dan rute yang harus dilewati pemilik kendaraan untuk sampai ke lokasi parkir tersimpan.
3. Setiap rute yang akan dilalui disertai dengan penjelasan dan gambar mengenai penunjuk arah dari *google maps*, serta menu *update* secara otomatis untuk mengetahui lokasi keberadaan pemilik saat itu.

3.6.2. Kebutuhan Perangkat

Kebutuhan perangkat yang digunakan untuk membuat Aplikasi Penyimpan Lokasi Parkir Kendaraan Menggunakan *App Inventor* Berbasis *Android*, meliputi perangkat lunak dan perangkat keras yang dihubungkan dengan sistem operasi. Perangkat lunak yang digunakan untuk membuat Aplikasi Penyimpan Lokasi Parkir Kendaraan Menggunakan *App Inventor* Berbasis *Android*, meliputi :

1. Perangkat lunak *App Inventor*, berfungsi sebagai perangkat untuk *interface* dan memberikan *code* untuk menjalankannya aplikasi android dan interaksi *visual* berbasis grafis.
2. Perangkat lunak *Youwave Android 2_3_4*, berfungsi emulator untuk menampilkan hasil jadi dari aplikasi yang sudah dibuat.
3. Perangkat lunak *blueStacks HD AppPlayerPro*, berfungsi emulator untuk menampilkan hasil jadi dari aplikasi yang sudah dibuat.
4. Perangkat lunak *Adobe Photoshop CS3* berfungsi untuk membuat desain background pada aplikasi.
5. Perangkat lunak *Microsoft Visio 2007* berfungsi untuk membuat diagram alir aplikasi.

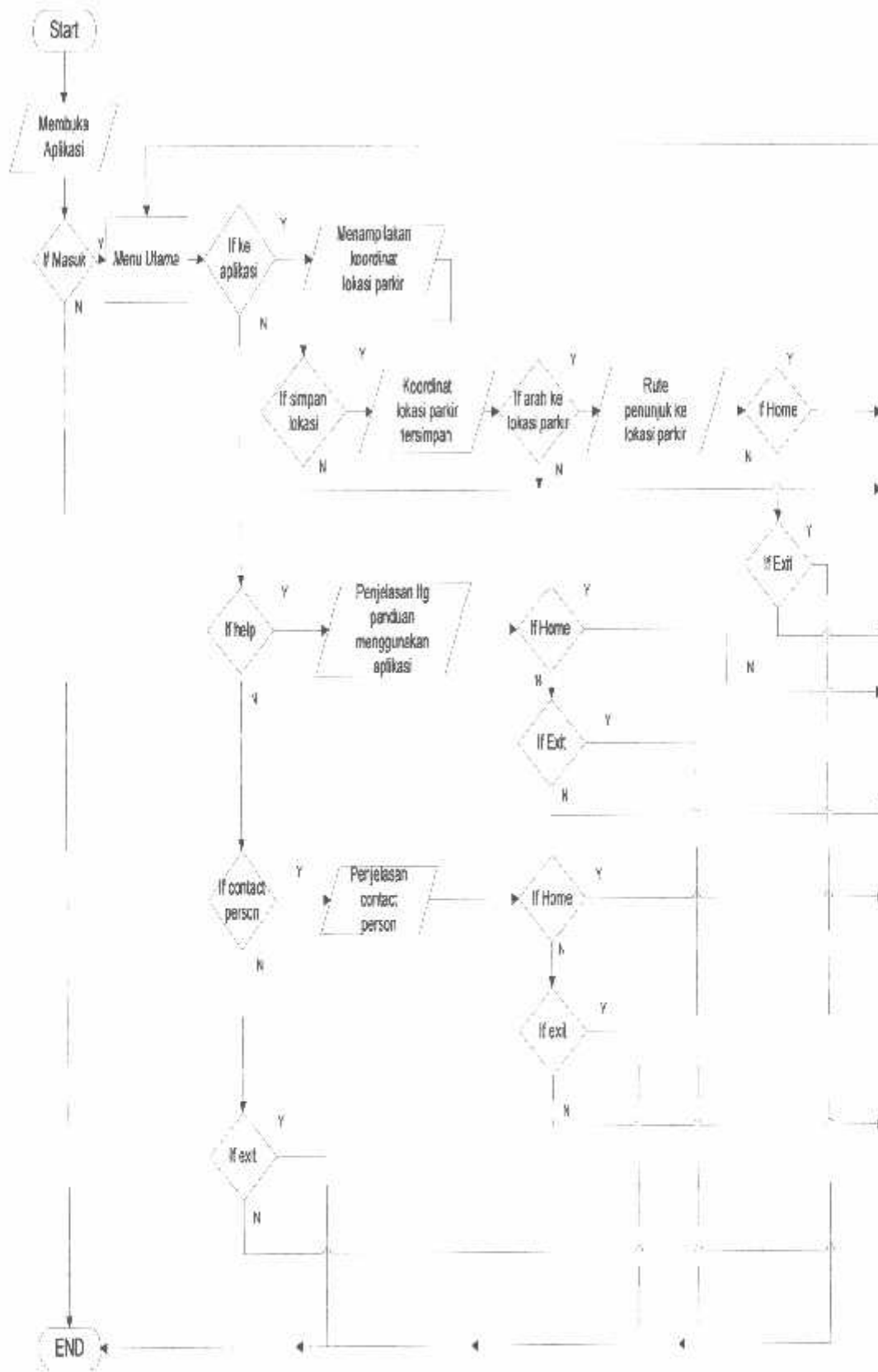
Sedangkan perangkat keras yang digunakan dalam membuat Aplikasi Penyimpan Lokasi Parkir Kendaraan, meliputi :

1. *Mainboard Intel Core i3.*
2. *Processor Intel Core i3 2.27 GHz.*
3. *Memory 2GB.*
4. *Harddisk ST920325AS ATA Device 320GB.*
5. *Monitor Generic PnP Monitor 14" 1280x800 Pixel.*
6. *Sound ATI High Definition Audio Device.*

Sistem operasi sebagai penghubung perangkat lunak dan perangkat keras dalam membuat Aplikasi Penyimpan Lokasi Parkir Kendaraan Menggunakan *App Inventor* Berbasis *Android*, menggunakan sistem operasi *Microsoft Windows Ultimate*.

3.7. Diagram Alir

Pada tahap desain Aplikasi Penyimpan Parkir Lokasi Kendaraan Menggunakan *App Inventor*, membutuhkan penjelasan pada setiap halaman aplikasi secara bertahap, mulai dari proses awal ketika memulai aplikasi, memasuki menu dan sub menu pada aplikasi sampai dengan mengakhiri aplikasi. *Flowchart* aplikasi pencari lokasi parkir kendaraan ini, dapat dilihat dalam Gambar 3.2.



Gambar 3.2. Diagram Alir Aplikasi.

3.7.1. Struktur Navigasi Aplikasi

Struktur navigasi pada aplikasi penyimpanan lokasi parkir kendaraan ini menggunakan struktur *navigasi* kombinasi, dimana setiap struktur yang ada, dikombinasikan dengan *navigasi* yang dapat dikontrol dengan menggunakan *mouse* komputer atau *touchpad* pada *mobile*. Struktur *navigasi* Aplikasi Penyimpan Lokasi Parkir Kendaraan Menggunakan *App Inventor*, meliputi :

1. Menu Pendahuluan

Pada menu pendahuluan, terdiri dari satu menu yang akan tampil di awal aplikasi, dimana menu tersebut berisi *navigasi* “masuk” untuk melanjutkan ke menu utama dari aplikasi.

2. Menu Utama

Terdiri dari lima navigasi atau tombol utama, yaitu tombol *home*, ke aplikasi, *help*, *contact person*, dan *exit* dengan isi pada masing-masing tombol sesuai dengan konten yang tertera pada nama tombol.

3. *Navigasi home*

Dalam navigasi *home* ini semua tombol *navigasi* untuk menghubungkan ke form lainnya yang berhubungan dengan aplikasi.

4. *Navigasi ke Aplikasi*

Tombol ini menghubungkan ke aplikasi penyimpanan lokasi parkir kendaraan, yang akan menunjukkan koordinat lokasi parkir dan memberikan rute untuk sampai ke tempat parkir kendaraan pemilik.

5. *Navigasi Contact Person*

Navigasi ini menjelaskan tentang *contact* yang bisa dihubungi apabila pengguna aplikasi ingin menyampaikan kritik atau saran kepada pembuat aplikasi.

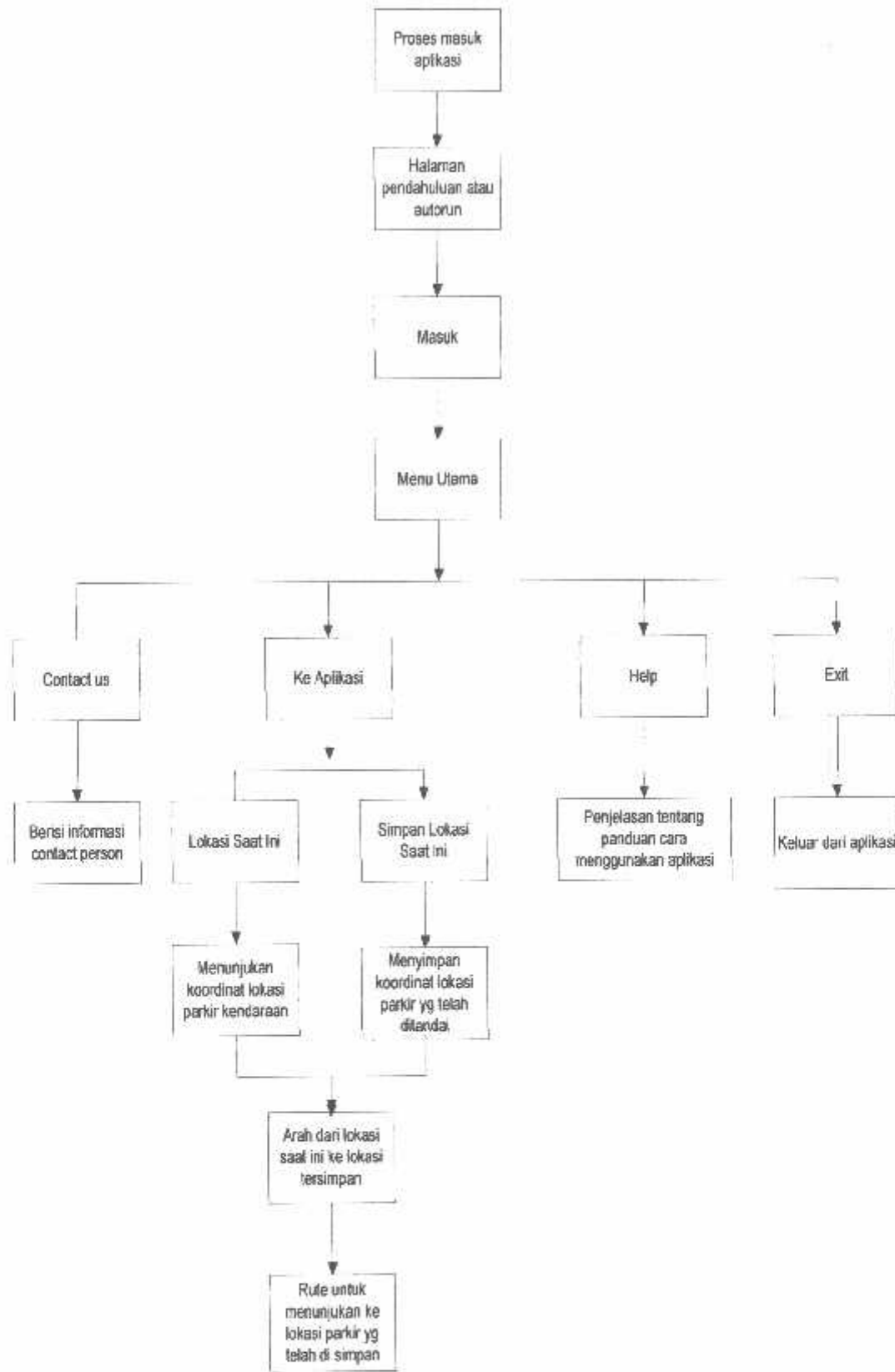
6. *Navigasi Help*

Navigasi ini berisi tentang cara menggunakan aplikasi atau semacam petunjuk untuk menolong kepada pengguna jika kebingungan menggunakan aplikasi.

7. *Navigasi exit*

Pada *Navigasi* ini digunakan untuk keluar dari aplikasi.

Struktur pada *navigasi* aplikasi pencari lokasi kendaraan pada area parkir menggunakan *app inventor*, dapat dilihat pada Gambar 3.3.

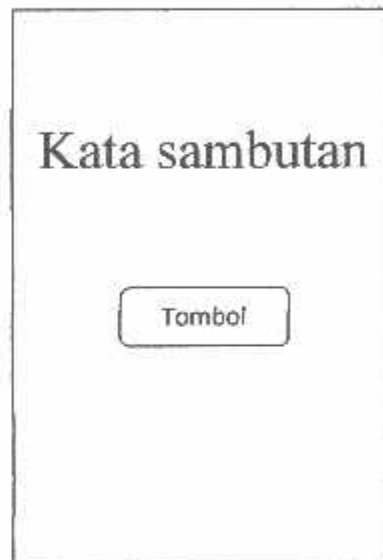


Gambar 3.3 Struktur *Navigasi* Aplikasi.

3.7.2. Struktur Perancangan Aplikasi

1. Halaman Pendahuluan

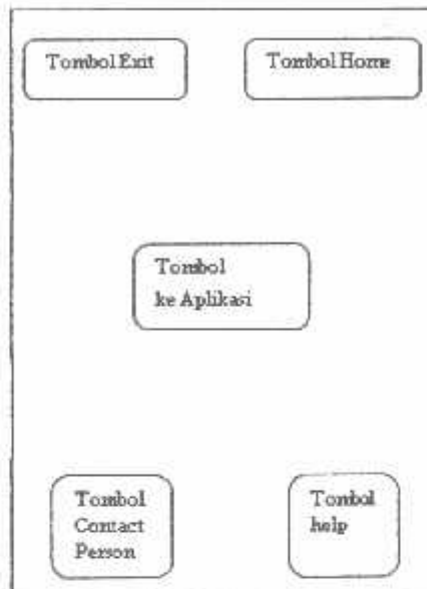
Halaman pendahuluan yang berada pada awal tampilan aplikasi ini merupakan tampilan pembuka sebelum memasuki halaman utama. Pada Aplikasi Penyimpan Lokasi Parkir Kendaraan Menggunakan *App Inventor* Berbasis *Android*, halaman pendahuluan menampilkan judul dan juga tombol *navigasi*, tombol *navigasi* yang dimaksud yaitu masuk. Desain pendahuluan dapat dilihat pada Gambar 3.4.



Gambar 3.4. Desain Halaman Pendahuluan.

2. Halaman Utama

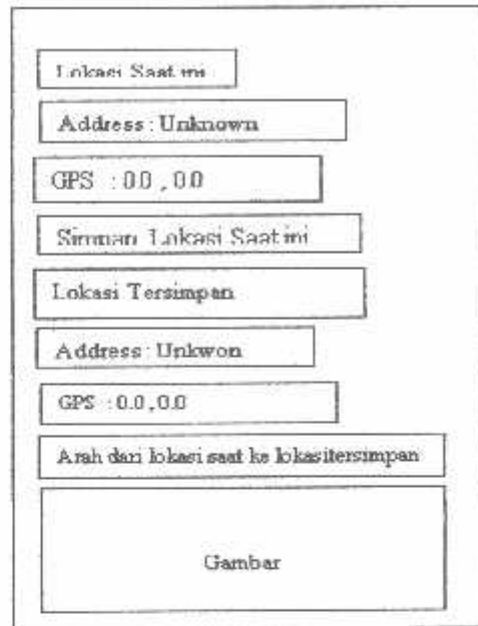
Halaman utama merupakan halaman yang menampilkan seluruh menu dari Aplikasi Penyimpan Lokasi Parkir Kendaraan Menggunakan *App Inventor* Berbasis *Android*. *Navigasi* yang terdapat pada halaman utama antara lain *home*, ke aplikasi, *help*, *contact person* dan *exit*. Semua tombol *navigasi* pada halaman utama tersebut ditampilkan dengan tambahan beberapa animasi teks maupun gambar. Desain halaman utama dapat dilihat pada Gambar 3.5.



Gambar 3.5. Desain Halaman Utama.

3. Halaman Menu Ke Aplikasi

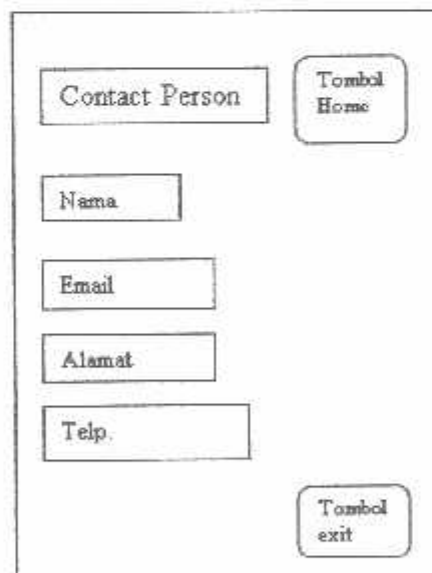
Halaman menu keAplikasi adalah merupakan halaman yang menampilkan informasi utama dari aplikasi penyimpanan lokasi parkir ini. dimana informasi yang dijelaskan merupakan informasi tentang koordinat lokasi parkir kendaraan pemilik kendaraan, data koordinat yang telah disimpan untuk menandai tempat parkir kendaraan saat di lokasi atau area parkir, menunjukkan arah atau rute yang harus dilalui pemilik kendaraan untuk sampai ke lokasi dimana pemilik kendaraan memarkir kendaraannya. Setelah masuk pada halaman yang dipilih selanjutnya terdapat beberapa navigasi yaitu untuk terus ke halaman selanjutnya, halaman *google maps*, *home* dan halaman *exit*. Tombol *navigasi* pada tiap halaman ditampilkan dengan beberapa tambahan *icon* yang menarik dan tidak monoton. Desain pada halaman menu ke aplikasi dapat dilihat pada Gambar 3.6.



Gambar 3.6. Desain Halaman Aplikasi Penunjuk Koordinat Lokasi.

4. Halaman *Contact Person*

Halaman *contact person* yaitu halaman yang tampil pada saat navigasi tombol *contact person* dipilih/ diklik oleh pengguna. Dalam halaman *contact person* ini akan menampilkan data mengenai *contact* yang bisa dihubungi oleh pengguna jika ingin memberikan kritik dan saran tentang aplikasi ini. Pada halaman ini. Desain pada halaman *contact person* dapat dilihat pada Gambar 3.7.



Gambar 3.7 Desain Halaman *Contact Person*.

5. Halaman *Help*

Halaman menu *help* merupakan halaman yang menampilkan petunjuk tentang tata cara pemakaian aplikasi penyimpan lokasi parkir kendaraan menggunakan *app inventor* berbasis *android*. Menu ini dimaksudkan agar pengguna yang kebingungan menggunakan bisa membaca menu *help* ini. Tujuan menu ini agar semua *user* atau pemilik kendaraan bisa menggunakan aplikasi ini dengan mudah atau *userfriendly*. Desain pada halaman *help* dapat dilihat pada Gambar 3.8.



Gambar 3.8. Desain Halaman *Help*.

Setelah kita selesai membuat rancangan tentang tampilan dari menu dan salah satu *screen* akan dijelaskan langkah – langkah penggunaan *App Inventor*. Berikut ini langkah – langkah penggunaan *App Inventor* dan dimulai dari cara penginstalan sampai penggunaannya. Langkah penggunaannya adalah sebagai berikut :

- a. Memiliki *Account Gmail* terlebih dahulu, dan masuk *–Log In* ke <http://beta.appinventor.mit.edu/>, apabila anda belum memiliki *Account Gmail* maka terlebih dahulu mendaftar pada <http://www.gmail.com>.
- b. Setelah masuk anda akan diminta untuk membaca & menyetujui *term of service* dari google, klik pada tombol *I accept the terms of service*, Dapat dilihat pada Gambar 3.9.

To use App Inventor for Android, you must accept the following terms of service

I accept the terms of service

Terms of Service

Gambar 3.9 Menyetujui *term of service*.

Dan langkah selanjutnya adalah langkah untuk memulai *App Inventor* :

- a. Miliki *Account gmail*, <http://www.gmail.com>
- b. Masuk ke <http://beta.appinventor.mit.edu/>
- c. *Download & Install AppInventor_Setup_Installer_v_1_2.exe.*
- d. <http://beta.appinventor.mit.edu/learn/setup/setupwindows.html>
- e. *Have fun with App Inventor*

Setelah itu anda akan dibawa ke halaman *App Inventor Setup*, dimana terdapat dua tahapan dasar pada setup yaitu :

1. Pada platform windows, - apabila anda menggunakan *Linux* anda bisa *download App Inventor setup* untuk *Linux* :
 - a. **Setup Komputer**, yaitu memiliki komputer PC dgn syarat untuk *Windows*. Sistem Operasi *Windows* yang didukung *Windows XP, Windows Vista, Windows 7.*
 - b. Telah menginstall Browser minimal *Google Chrome 4.0 – rekomendasi Apple Safari 5.0 , Microsoft Internet Explorer 7, Mozilla Firefox 3.6*
 - c. Menginstall Java 6– *java 1.6, bisa juga java 1.7.*
 - d. Kemudian install *AppInventor_Setup_Installer_v_1_2.exe.*
2. **Setup lingkungan *Developing***, yaitu :
 - a. Menggunakan *Emulator*,: pada penggunaan *emulator* tidak memerlukan setup di awal, karena tinggal diaktifkan dari *blok editor.*
 - b. Menggunakan *Handset Android.* - Kalau tidak ingin mengesetnya kita bisa langsung melakukan *developing.*

Setup Handset Android

Tahap ini dilakukan apabila anda ingin langsung mencoba hasil *develop* anda ke *handset android*. Pertama anda harus memiliki kabel data USB, dan **menginstall driver handset Android anda**. Hampir semua *handset android* di dukung oleh *App Inventor*, dan pastikan juga anda telah memiliki memory *SD Card* yang terpasang.

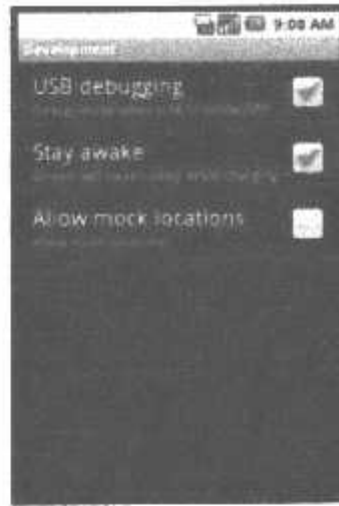
Setelah itu setting pada handset kita :

- a. Masuk pada *home screen*.
- b. Pilih *Setting > applications*.
- c. Pada *Unkown sources* di ceklist, dapat dilihat pada Gambar 3.10.



Gambar 3.10. Gambar setting

- d. Pilih *Development*, ceklist pada *USB Debugging* dan *Stay Awake*, dapat dilihat pada Gambar 3.11.



Gambar 3.11 *USB Debugging*.

3. Setelah itu koneksikan *Handset* anda dengan kabel USB Data ke komputer – *dalam keadaan screen unlock*, hingga akan muncul dua pesan notifikasi pada atas layar yaitu :
 - a. *USB Connected*, yang berarti handset telah terhubung ke komputer.
 - b. *USB Debugging Connected*, yang membuat *App Inventor* di komputer mengontrol handset.

Sampai disini berarti *handset android* kita telah siap untuk digunakan untuk mencoba aplikasi yang kita buat dengan *App Inventor*. Setelah semua konfigurasi dan setting selesai, anda bisa langsung masuk ke lingkungan *developing* dengan melalui <http://beta.appinventor.mit.edu/> apabila tadi masih *login* anda langsung akan dibawa ke jendela *developing* dapat dilihat pada Gambar 3.12.



Gambar 3.12 Jendela *developing*.

Lalu klik pada *New*, dan mulai projek baru misal dengan nama *Helloworld*, kemudian klik pada tombol OK, dapat dilihat pada Gambar 3.13.



Gambar 3.13 Membuat projek baru.

maka akan muncul lembaran projek kita, dapat dilihat pada Gambar 3.14.



Gambar 3.14 projek baru.

4. Lingkungan Kerja *App Inventor*, yaitu :

Sebelum kita melanjutkan untuk memulai *developing*, kita mengenal terlebih dahulu lingkungan kerja pada *App Inventor* yang terdiri dari :

- A. **Komponen Desainer** – *Component Designer* yang berjalan pada Browser, digunakan untuk memilih komponen dan mengatur *property*. Komponen *desainer* dapat dilihat pada Gambar 3.15 di bawah ini :



Gambar 3.15 Komponen *desainer*.

Komponen desainer , terdiri dari :

- a. *Viewer* : untuk menempatkan komponen dan mengaturnya sesuai tampilan yang diinginkan.
- b. *Pallette* : adalah *list* dari komponen yang bisa dipakai
- c. *Component list* : merupakan tempat *list* komponen dari proyek yang kita pakai
- d. *Media* : mengambil media audio dan gambar untuk proyek kita
- e. *Properties* : ketika anda klik komponen pada *viewer* maka *properties*nya akan terlihat pada panel ini

B. **Blok Editor** – *Blocks Editor* berjalan diluar *browser*, dimana digunakan untuk membuat dan mengatur behaviour dari komponen yang kita pilih pada komponen desainer. *Block editor* yang sedang aktif bisa dilihat pada Gambar 3.16 di bawah ini.



Gambar 3.16 *Block editor*.

C. **Emulator** digunakan untuk menjalankan dan mengetest aplikasi yang kita bangun. Ini sangat bermanfaat apabila kita belum menggunakan *handset* langsung, karena *Emulator Android* ini telah terintegrasi dengan baik. Salah satu contoh *emulator* yang digunakan bisa dilihat pada Gambar 3.17.



Gambar 3.17 *Emulator*.

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1. Umum

Tahapan implementasi dan pengujian aplikasi *mobile* penyimpan lokasi parkir kendaraan menggunakan *app inventor* berbasis *android*, dilakukan setelah seluruh proses pengumpulan data dianggap telah mencukupi untuk dilanjutkan ke dalam proses pembuatan aplikasi dengan mengkombinasikan data-data pendukung aplikasi atau elemen-elemen pendukung aplikasi *android* menggunakan perangkat lunak *App Inventor* agar menghasilkan halaman-halaman yang saling terintegrasi antar satu halaman dengan halaman yang lainnya. Apabila pada saat proses pembuatan aplikasi *mobile* penyimpan lokasi parkir kendaraan menggunakan *app inventor* berbasis *android* membutuhkan beberapa data pendukung, maka akan dilakukan proses penambahan pengumpulan data.

4.2 Implementasi

Pada proses implementasi ke dalam aplikasi *mobile* penyimpan lokasi parkir kendaraan menggunakan *app inventor* berbasis *android*, yang meliputi :

1. Memasukkan Data *Visual*

Hasil dari proses pengumpulan data yang berupa *visual*, dimasukkan ke dalam *screen* atau halaman pada perangkat lunak *App Inventor* yang disesuaikan dengan setiap halaman yang berisi mengenai tata letak dan fungsi halaman itu sendiri.

2. Membuat dan mengatur *behavior* dari komponen yang sudah kita pilih pada komponen desainer.

Membuat dan mengatur *behavior* pada komponen yang sudah dipilih pada *App Inventor* dilakukan pada *block editor*, dan *block editor* ini berjalan diluar *browser*. Proses tersebut dilakukan agar setiap komponen yang ada pada komponen desainer bisa dijalankan sesuai dengan apa yang kita mau. *Block editor* ini bisa dikatakan seperti *script*, tapi bedanya *script* ini tidak dalam bentuk tulisan akan tetapi seperti *puzzle* yang harus dirangkaikan.

4.3. Pengumpulan Data

Sebelum dilakukan pembuatan aplikasi diperlukan pengumpulan data yang nantinya akan di gunakan untuk mengisi aplikasi sebagai media/sarana informasi untuk menunjang *system*.

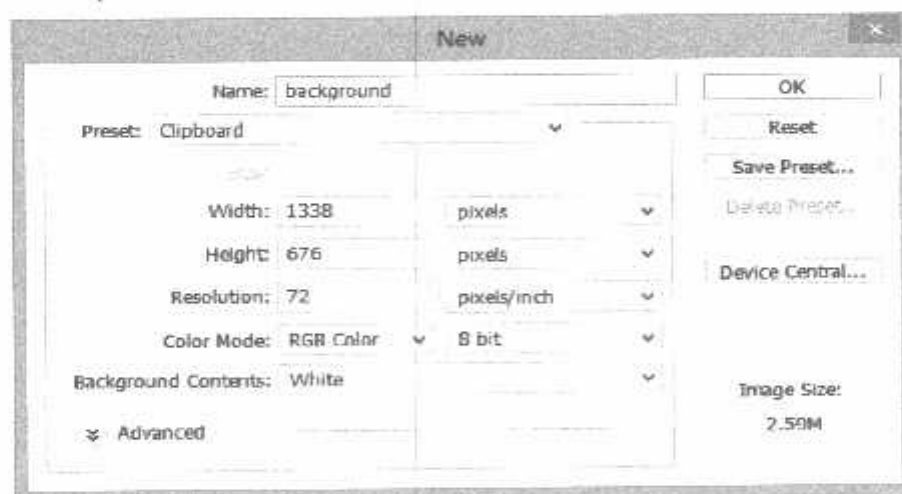
4.3.1. Penambahan *Background* Pada Aplikasi

Penambahan *background* pada aplikasi digunakan sebagai identitas dari aplikasi yang dibuat, selain itu penambahan animasi pada materi ini juga bertujuan agar aplikasi yang dibuat mampu menghasilkan suatu media penunjuk arah yang menarik yang dibuat melalui perangkat lunak *Adobe PhotoShop CS3*. Dalam penambahan *background* pada aplikasi, memerlukan beberapa teknik dan cara untuk menghasilkan tampilan *background* yang baik, tahapan tersebut meliputi :

1. Membuka perangkat lunak *Adobe MaterCollection CS3*.
2. Setelah *Adobe MasterCollection CS3* klik pada tombol *Adobe PhotoShop CS3* untuk memulai membuat lembar kerja baru.

Setelah melakukan proses pembuatan lembar kerja kosong pada *Adobe Photoshop CS3*, diperlukan tahapan-tahapan untuk melakukan pembuatan *background* pada lembar kerja atau *stage*, antara lain :

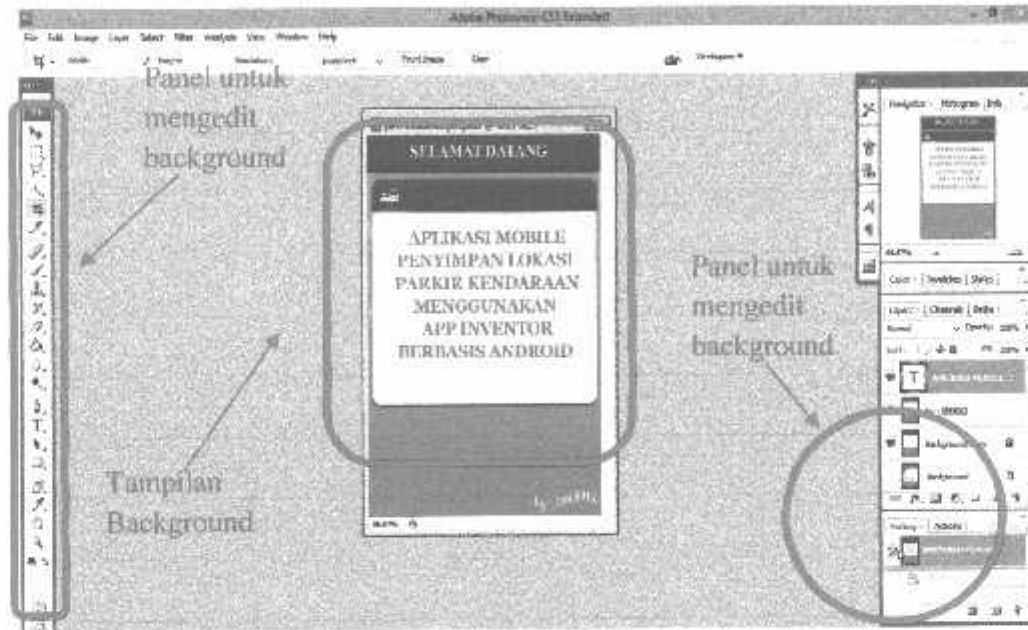
1. Langkah pertama adalah klik *File – New*, kemudian akan muncul lembar kerja baru seperti Gambar 4.1.



Gambar 4.1 lembar kerja baru.

2. Setelah lembar kerja baru terbuka maka siap untuk melakukan proses pembuatan *background* gambar.

3. Setelah gambar berhasil dimasukkan ke dalam lembar kerja maka proses pembuatan *background* bisa dimulai dengan cara memanfaatkan panel pada sisi bawah lembar kerja. *background* pada *stage* atau lembar kerja dapat dilihat pada Gambar 4.2.



4.2. Pembuatan *background* pada lembar kerja *Adobe Photoshop*.

4.4. Pembuatan Halaman *Screen1*

Pada proses aplikasi *mobile* penyimpan lokasi parkir kendaraan menggunakan *app inventor* berbasis *android*. Tahapan-tahapan yang diperlukan dalam pembuatan *screen1*, antara lain :

1. Menjalankan perangkat lunak *App Inventor*.
2. Membuat *project* baru dengan nama "Rancang_Bangun" untuk bisa diaktifkan oleh halaman lain.
3. Melakukan konfigurasi pada *screen* yang masih kosong, dengan membuat tampilan tombol masuk agar bisa masuk ke dalam menu utama aplikasi.
4. Melakukan konfigurasi dengan pemberian *background* berupa gambar yang telah dibuat pada *Adobe PhotoShop*.

Proses pemberian warna pada latar *screen* serta konfigurasi *background color*, *background image*, *icon* dll dapat dilakukan di *properties*. Proses tersebut dapat dilihat pada Gambar 4.3.



Gambar 4.3 Konfigurasi *screen* pada *App Inventor*.

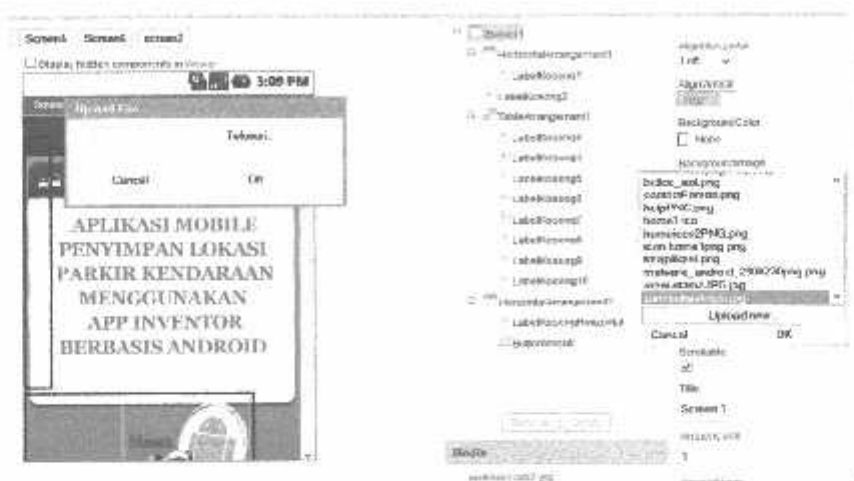
Setelah melakukan konfigurasi pada *screen1*, maka proses pembuatan *screen1* sudah dapat dilakukan. Proses dalam pembuatan *screen1*, meliputi :

1. Proses pembuatan *screen1*, dimana tombol ini menghubungkan ke *screen2* yang merupakan menu utama pada aplikasi ini. Dalam *screen2* tersebut terdapat 5 tombol menu yaitu *home*, ke-aplikasi, *help*, *contact person*, dan tombol *exit*. Tombol masuk dapat dilihat pada Gambar 4.4.



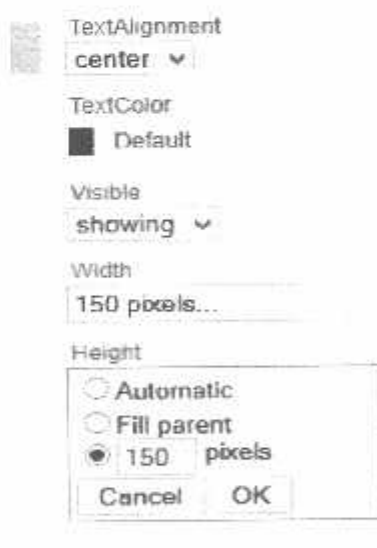
Gambar 4.4. Tombol Masuk.

2. Proses pemberian background pada *screen1*, dengan cara meng*import* gambar *background* yang telah dibuat pada *Adobe PhotoShop* Sebelumnya. Pilih *screen1* => *Upload new* => pilih file *background* yang diinginkan => *OK*. Gambar pemberian *background* pada *screen1* dapat dilihat pada Gambar 4.5.



Gambar 4.5 Pemberian *background* pada *Screen1*.

3. Proses pengaturan ukuran tombol, dengan gambar yang ukurannya tidak sama satu sama lainnya tentunya merepotkan dalam pengaturan tata letak aplikasi. Untuk mengatur agar tampilan pada aplikasi semakin menarik maka tombol dibuat dengan ukura *150x150 pixel*. Ukuran *pixel* tombol dapat dilihat pada Gambar 4.6.



Gambar 4.6 Ukuran tombol masuk *150x150 pixel*.

Secara keseluruhan pada proses pembuatan *screen2* yang merupakan menu utama pada aplikasi *mobile* penyimpan lokasi parkir kendaraan menggunakan *app inventor* berbasis *android* dapat dilihat pada Gambar 4.7 dibawah ini.



Gambar 4.7 Proses pembuatan *screen2* menu utama.

4.4.2. Pembuatan *Screen3*

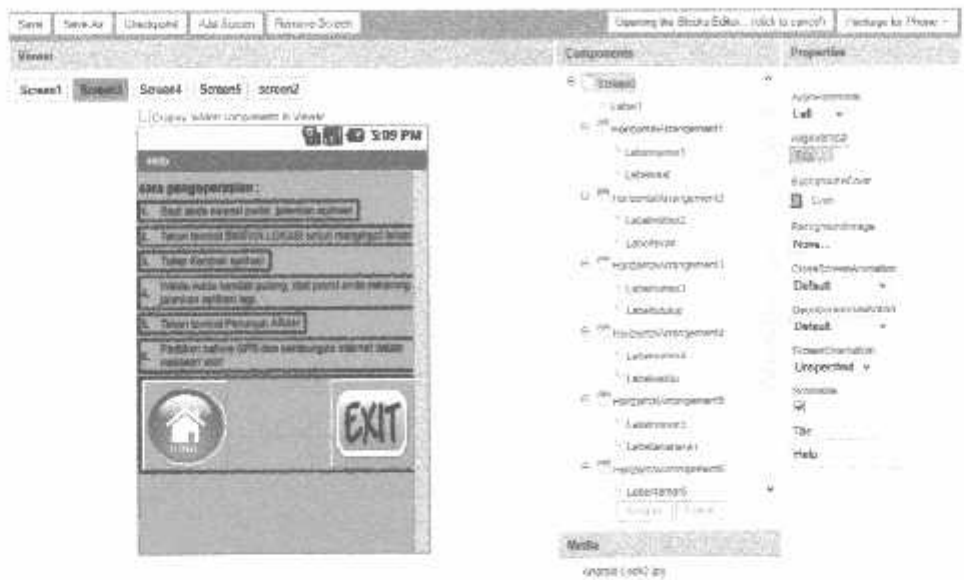
Tahapan dalam pembuatan *screen3* halaman menu *help* , meliputi :

1. Membuat *screen* baru dengan cara *Add Screen "Screen3"*,
2. Terdapat 2 tombol pada halaman menu *help* ini.
3. Pada halaman ini berisi tentang informasi cara pengoperasian aplikasi penyimpan lokasi parkir kendaraan. Pada halaman ini terdapat tombol *home* dan *exit*, tombol *home* menghubungkan *screen3* dengan menu utama yaitu *screen2*. *Block Editor* pada setiap tombol dapat dilihat pada Tabel 4.2.

Tabel 4.2 *Block editor* pada *screen3*

Tombol	Block Editor
Home pada <i>Screen3</i>	
Exit pada <i>Screen3</i>	

Proses keseluruhan pembuatan *screen3* yang merupakan halaman menu *help* dapat dilihat pada Gambar 4.8.



Gambar 4.8 Proses pembuatan halaman *screen3*

4.4.3. Pembuatan *Screen4*

Tahapan yang diperlukan dalam pembuatan halaman *contact person* antara lain :

1. Membuat *screen* baru dengan cara *Add Screen* nama "*Screen4*" yaitu halaman menu *contact person*.
2. Pada *screen contact person* terdapat informasi mengenai *contact* yang bisa dihubungi jika user mengalami kesulitan atau masalah tentang aplikasi. User bisa mengirimkan kritik atau saran melalui email atau bisa juga langsung menghubungi pembuat aplikasi dengan nomer yang telah tersedia.
3. Dalam halaman ini terdapat tombol *home* dan tombol *exit*. Tombol *home* pada halaman ini untuk kembali ke menu utama dan *exit* untuk keluar langsung dari aplikasi. *Block editor* tombol pada *screen4* dapat dilihat pada Tabel 4.3.

Tabel 4.3 *Block editor* pada *sreen4*.

Tombol	Block Editor
Home pada <i>screen4</i>	
Exit pada <i>screen4</i>	

Tampilan *screen4* atau halaman *contact person* secara utuh dapat dilihat pada Gambar 4.9 di bawah ini.



Gambar 4.9. Tampilan halaman *contact person* pada *screen4*.

4.4.4. Pembuatan *Screen6*

Antar muka *visual* aplikasi penyimpanan lokasi parkir terdiri dari beberapa komponen label dan dua buah *button*. Sekumpulan label pertama menunjukkan lokasi aktual dari ponsel/pengguna, berupa alamat dan koordinat lokasi (*Global Positioning System*). Tepat di bawahnya, terdapat sebuah tombol untuk merekam posisi aktual menjadi posisi yang di ingat (*remembered*), yang artinya atributnya diberikan oleh grup Label kedua. Bila tombol ditekan aplikasi *mobile* penyimpanan lokasi parkir kendaraan menggunakan *app inventor* berbasis *android* akan menyalin alamat dan data koordinat lokasi (*Global Positioning System*) aktual dari grup label kedua, serta menyimpan data ini ke dalam basis data perangkat. Tombol terakhir merupakan tombol mencari rute/arah posisi sekarang ke posisi terekam dengan menggunakan *GoogleMaps*. Antar muka aplikasi ini pada *emulator* dapat dilihat pada Gambar 4.10.



Gambar 4.10 Antar muka aplikasi pada emulator.

Tahapan yang diperlukan dalam pembuatan halaman KeAplikasi antara lain :

1. Membuat *screen* baru dengan cara *Add Screen* nama "*Screen6*" yaitu halaman menu KeAplikasi.
2. Buatlah antar muka pada komponen desainer *app inventor*, sesuai dengan rancangan antarmuka yang kita buat.
3. Selanjutnya tambahkan komponen *non visible* yaitu : *Location sensor* untuk memanfaatkan *Global Positioning System*, *TinyDB* untuk penyimpanan dalam database.
4. Koordinat lokasi diambil dari *Global Positioning System*, dan *Activity Starter* untuk meluncurkan *GoogleMaps*.
5. Set test properti dari label *AddressSaatiniLabel* ke *Unknown*.
6. Pastikan properti *Enable* tidak di checklist pada tombol *SimpanLokasiSaatIniButton* dan *ArahButton*.

Supaya lebih jelas komponen desainer dapat dilihat pada Gambar 4.11.

Dari komponen yang telah dipasang, logika aplikasinya adalah sebagai berikut :

1. Ketika *LocationSensor* mendapatkan lokasi data koordinat lokasi, letakkan pada label *LokasiSaatIni* yang telah kita desain.
2. Ketika tombol simpan lokasi saat ini di klik, simpan pada database *TinyDB* dan juga letakkan pada *Lokasi Tersimpan*.
3. Selanjutnya ketika *user* meng-klik tombol arah dari lokasi saat ini ke lokasi tersimpan, luncurkan *GoogleMaps* berikut arah ke lokasi yang telah tersimpan tadi.
4. Ketika aplikasi ditutup dan kemudian dibuka kembali, *load* data lokasi yang tersimpan dari database.

4.5 Mendapatkan Lokasi Saat Ini

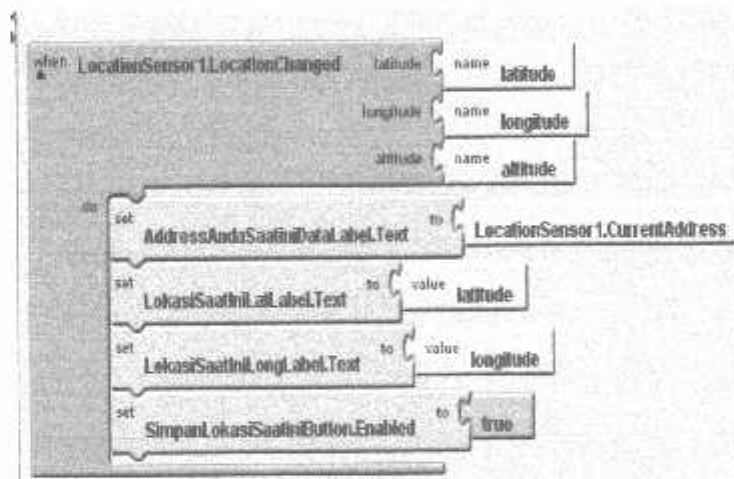
Pada saat melakukan ujicoba pembacaan koordinat, apabila pembacaan data koordinat lokasi terhalang oleh atas ruangan, sebaiknya kita keluar kehalaman atau tepat yang tidak terhalang oleh apapun. *Event block* lengkapnya dan komponen yang akan digunakan kali ini adalah dapat pada Tabel 4.5 seperti dibawah ini :

Tabel 4.5 Komponen untuk mendapatkan lokasi.

Tipe	Drawer	Fungsi
LocationSensor1. LocationChanged	LocationSensor1	Even handler dari pembacaan data koordinat
setAddressAndaSaatIniData Label. Text to	AddressAndaSaatIniData Label	Untuk menempatkan data pada update lokasi baru
LocationSensor1.CurrentAddress	LocationSensor1	Mendapatkan address jalan
setLokasiSaatIniLatLabel.Text	LokasiSaatIniLatLabel	Untuk menempatkan

		data latitude
Value latitude	My Definitions	Nilai Latitude
setLokasiSaatIniLongLabel.Text	LokasiSaatIniLongLabel	Untuk menempatkan data Longitude
Value longitude	My Definitions	Nilai Longitude
Set SimpanLokasiSaatIniButton. Enabled to	SimpanLokasiSaatIniButton	Untuk menyimpan pembacaan data lokasi
True	Logic	Kondisi true

Gambar *Block Editor* pada *Screen6* untuk mendapatkan lokasi saat ini dapat dilihat pada Gambar 4.13.



Gambar 4.13 *Block Editor* untuk mendapatkan lokasi.

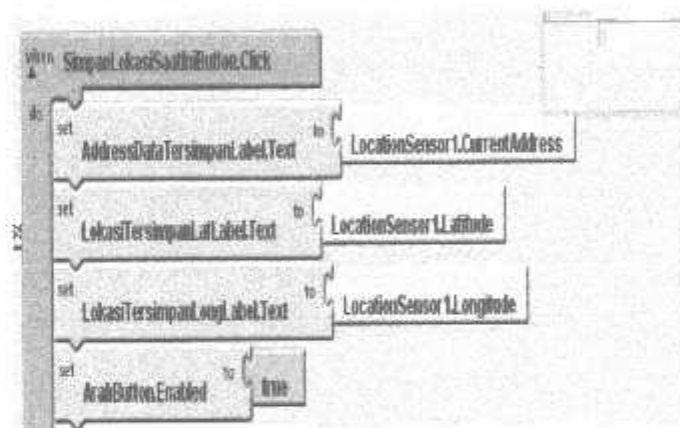
4.6 Menampilkan Data Lokasi Saat Ini

Untuk menampilkan data lokasi saat ini terlebih dulu akan membuat *event* fungsi ketika tombol **simpan lokasi saat ini** diklik, data pada label **Lokasi Saat Ini** akan tampil pada label **Lokasi Tersimpan**. *Event block* nya seperti Tabel 4.6 di bawah ini.

Tabel 4.6 *Event block* untuk mendapatkan data lokasi saat ini.

Tipe	Drawer	Fungsi
SimpanLokasiSaatIniButton. click	SimpanLokasiSaatIniButton	Tombol untuk menyimpan data
Set AddressDataTersimpanLabel. Text	AddressDataTersimpanLabel	Menempatkan data address
LocationSensor1.CurrentAddress	LocationSensor1	Mendapatkan address jalan
Set LokasiTersimpanLatLabel. Text	LokasiTersimpanLatLabel	Menempatkan data latitude
LocationSensor1.Latitude	LocationSensor1	Nilai

Untuk gambar *block event* pada tombol simpan lokasi dapat dilihat pada Gambar 4.14.



Gambar 4.14 *Event block* pada tombol simpan lokasi.

4.7 Menampilkan Arah Ke Lokasi Tersimpan

Ketika user mengklik tombol Arahbutton → **Arah Dari Lokasi Saat Ini Ke Lokasi Tersimpan**, pembuat aplikasi ingin agar aplikasi membuka *GoogleMaps* dengan sebuah garis pandu arah *Direction* dari lokasi *user* berada saat itu menuju lokasi yang telah disimpan sebelumnya. Yang dalam hal ini adalah dimana kendaraan itu diparkir.

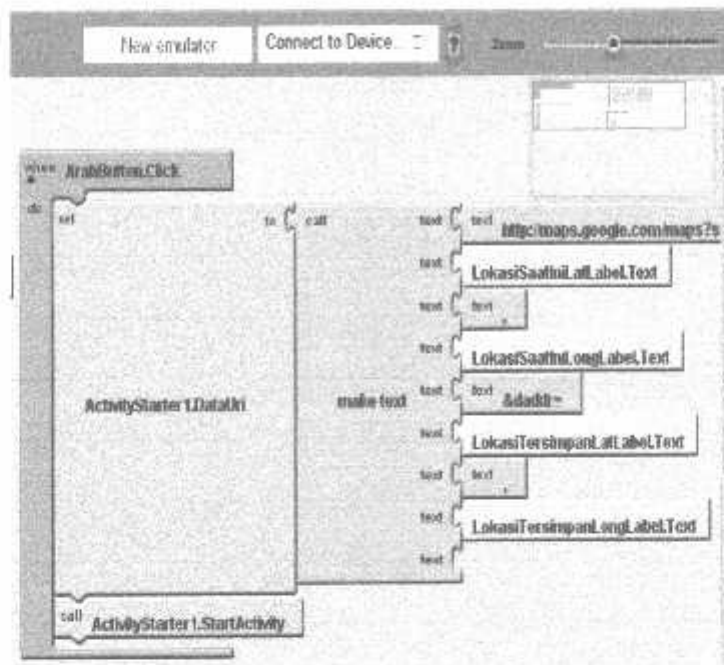
Maka dari itu, pada aplikasi ini akan membantu menghubungkan URL dengan format koordinat awal *source address* (*saddr*) dan koordinat tujuan *destination address* (*daddr*). Misalkan URL ini <http://maps.google.com/maps?saddr=-7.795442,110.368996&daddr=-7.790765,110.365856> , yang akan menampilkan peta secara langsung.

Blok yang digunakan adalah seperti Tabel 4.7 di bawah ini :

Tabel 4.7 *Event block* lengkap button arah .

Tipe	Drawer	Fungsi
ArahButton.Click	ArahButton	Tombol user
set ActivityStarter1.DataUri	ActivityStarter	URL untuk peta yang dituju
make text	Text	Menggabungkan teks
text(http://maps.google.com/maps?saddr=)	Text	Parameter alamat pertama lokasi awal
LokasiSaatIniLatLabel.Text	LokasiSaatIniLabel	Latitude lokasi saat ini
text(",")	Text	Teks koma
LokasiSaatIniLongLabel.Text	LokasiSaatIniLongLabel	Longitude lokasi saat ini
text("daddr"=)	Text	Parameter alamat kedua lokasi tujuan
LokasiTersimpanLatLabel.Text	LokasiTersimpanLatLabel	Latitude lokasi tersimpan
text(",")	Text	Teks koma
LokasiTersimpanLongLabel.Text	LokasiTersimpanLongLabel	Longitude lokasi tersimpan
ActivityStarter1.StartActivity	ActivityStarter1	Untuk membuka peta

Gambar untuk *block editor* pada tombol arah dapat dilihat pada Gambar 4.15.



Gambar 4.15 *block editor* untuk tombol arah pada *Screen6*.

Apabila mencoba pada *emulator* maka dapat ditunjukkan seperti Gambar 4.16.



Gambar 4.16 Tampilan pada *Emulator*.

4.8 Menyimpan data lokasi saat ini pada TinyDB

Pada saat aplikasi kita jalankan dan tidak ditutup, maka semua tampak sudah berfungsi. Akan tetapi apabila aplikasi kita tutup, dan jalankan kembali, maka koordinat aplikasi yang didapatkan tadi akan hilang karena belum disimpan pada sebuah database yang menyimpan secara permanen. Maka tahap ini adalah tahap untuk membuat :

1. Ketika *user* mengklik tombol **Simpan Lokasi Saat Ini**, maka data akan tersimpan pada database.
2. Ketika aplikasi diluncurkan kembali, data yang tersimpan pada database akan tetap ditampilkan pada label **Lokasi Tersimpan**.

Untuk itu disini tidak akan memodifikasi *event handler* **SimpanLokasiSaatIniButton.Click**, dan fungsi `TinyDB.StoreValue` untuk menyimpan data. Blok yang digunakan adalah seperti Tabel 4.8 :

Tabel 4.8 *Event block* menyimpan lokasi dengan TinyDB.

Tipe	Drawer	Fungsi
<code>TinyDB1.StoreValue (3)</code>	TinyDB1	Untuk menyimpan data pada database
<code>text("address")</code>	Text	Text
<code>LocationSensor1.CurrentAddress</code>	LocationSensor1	Untuk menyimpan data address pada database
<code>text("Lat")</code>	Text	Text
<code>LocationSensor1.Latitude</code>	LocationSensor1	Untuk menyimpan data latitude pada database
<code>text("Long")</code>	Text	Text
<code>LocationSensor1.Longitude</code>	LocationSensor1	Untuk menyimpan data longitude pada database

Sehingga blok lengkap dari event handler `SimpanLokasiSaatIniButton.click` sekarang dapat dilihat pada Gambar 4.17.



Gambar 4.17 *Block event* menyimpan *address* lokasi

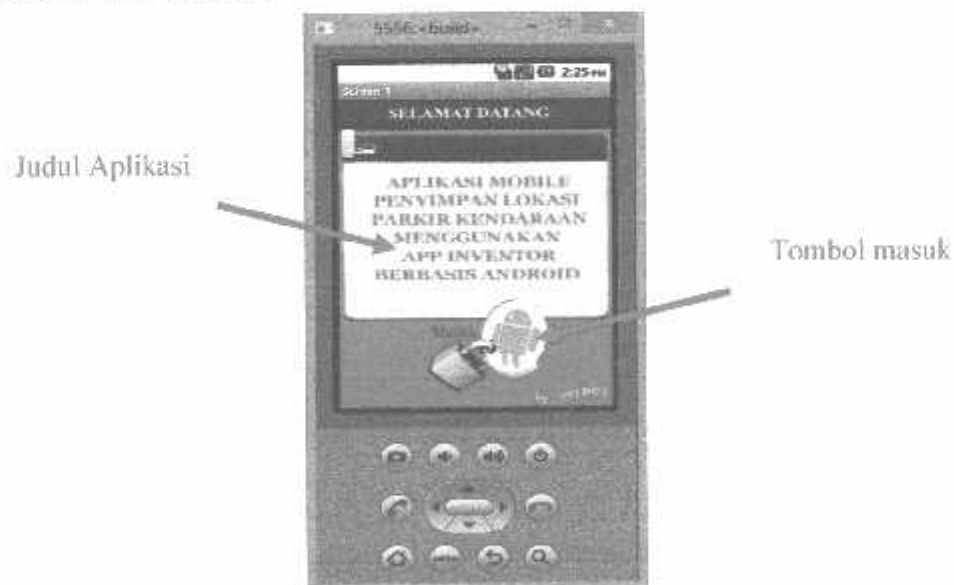
Terakhir pada *Screen6* akan membuat ketika aplikasi diluncurkan kembali, data yang tersimpan pada database akan kita tampilkan pada label **Lokasi Tersimpan**. Untuk itu akan menggunakan *event* *Screen6.Initialize*, dan juga fungsi *TinyDB.GetValue* untuk mengambil data dari database.

4.9 Pengujian Aplikasi pada *Emulator*

Setelah melalui beberapa tahapan pembuatan aplikasi, maka dihasilkan dengan tampilan sebagai berikut:

a. *Opening* atau *Intro*

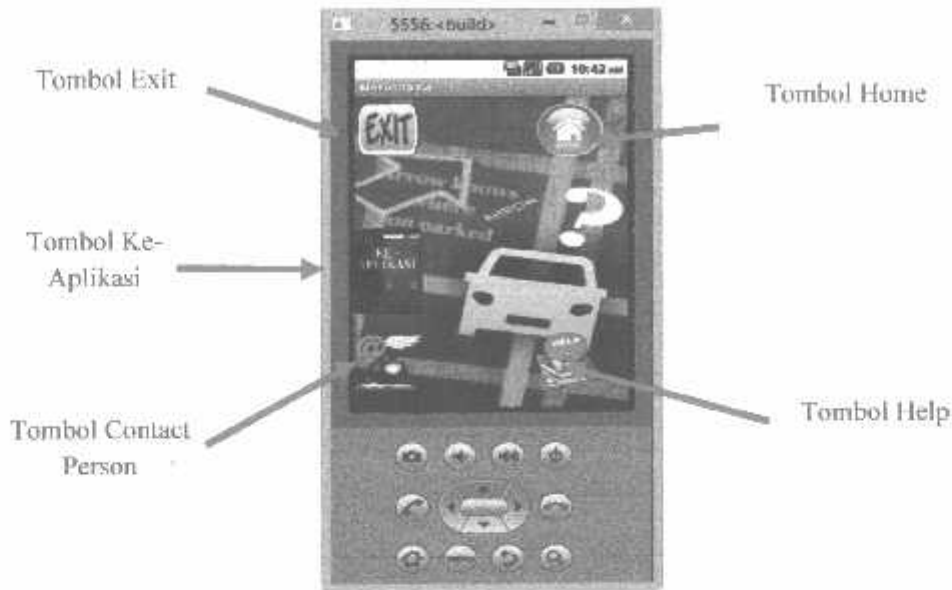
Pada *opening* atau *intro* di tampilkan judul dari aplikasi. Terdapat tombol masuk pada halaman menu utama aplikasi. Tampilan *opening* atau *intro* ditunjukkan pada Gambar 4.18. dibawah.



Gambar 4.18. Tampilan *Opening*

b. Menu Utama

Pada halaman menu utama terdapat beberapa tombol pilihan yaitu, *home*, *keAplikasi*, *contact person*, *help* dan *exit*.Tampilan pada menu utama dapat dilihat pada Gambar 4.19.



Gambar 4.19. Tampilan Menu Utama

c. Halaman KeAplikasi

Antar muka *visual* aplikasi penyimpan lokasi parkir terdiri dari beberapa komponen label dan dua buah *button*. Sekumpulan label pertama menunjukkan lokasi aktual dari ponsel/pengguna, berupa alamat dan data koordinat Lokasi. Tepat di bawahnya, terdapat sebuah tombol untuk merekam posisi actual menjadi posisi yang di ingat (*remembered*), yang artinya atributnya diberikan oleh grup Label kedua. Bila tombol ditekan aplikasi *mobile* penyimpan lokasi parkir kendaraan pada area parkir menggunakan *app inventor* berbasis *android* akan menyalin alamat dan data koordinat lokasi aktual dari grup label kedua, serta menyimpan data ini ke dalam basis data perangkat. Tombol terakhir merupakan tombol mencari rute/arah posisi sekarang ke posisi terekam dengan menggunakan *GoogleMaps*. Antar muka aplikasi ini pada *emulator* dapat dilihat pada Gambar 4.20.



Gambar 4.20. Antar muka aplikasi penyimpan lokasi kendaraan.

d. Halaman *Contact Person*

Pada halaman *contact person* menampilkan informasi mengenai *contact* dan alamat pembuat program. Jika user ingin menyampaikan kritik dan saran tentang aplikasi yang telah dibuat, maka dapat langsung menghubungi *contact person* yang ada dalam aplikasi ini. Tampilan halaman *contact person* ditunjukkan pada Gambar 4.21 dibawah ini.



Gambar 4.21. Tampilan *contact person* pada pengujian *emulator*.

e. Halaman *Help*

Pada halaman dari menu *help* ini menampilkan informasi tentang cara pengoperasian aplikasi penyimpan lokasi parkir. Untuk *user* yang merasa kesulitan atau bingung bagaimana cara menggunakannya maka dapat membuka halaman ini terlebih dahulu. Selain itu terdapat tombol untuk interaksi antara *user* dengan aplikasi, seperti tombol *home* untuk kembali ke halaman menu utama, tombol *exit* untuk keluar dari aplikasi. Tampilan halaman *help* pada pengujian *emulator* ditunjukkan Gambar 4.22 di bawah ini .



Gambar 4.22. Tampilan halaman *help* pada *emulator*.

4.10 Transfer Aplikasi Ke *Handphone*

Pastikan bahwa *handphone* mendukung (*Global Positioning System*) dan internet aktif. *Handphone* yang mendukung seperti X-peria X8, Galaxy mini, Galaxy Young, X-peria X10 Mini Pro, W8 walkman, X-peria Arc dan lain-lain. Setelah *handphone* mendukung untuk, maka aplikasi dapat *diinstal* ke *handphone*. Alat bantu untuk *transfer* aplikasi ke *handphone* adalah *Connectivity Adapter Cable* (kabel data), *Card Reader* (adapter dari kartu memori), dan *Bluetooth Adapter*. Setelah aplikasi ditransfer, aplikasi dapat dicoba dijalankan pada *handphone*.

4.11 Pengujian Aplikasi

Pengujian aplikasi dilakukan untuk mengetahui atau mendeteksi jika terjadi *error* pada aplikasi saat dijalankan. Pada pengujian aplikasi ini dilakukan pada *emulator* dan pada *handphone*.




4.11.1 Pengujian Pada *Handphone*

Pengujian pada *handphone* ini dilakukan untuk mengetahui apakah aplikasi penyimpan lokasi parkir kendaraan ini siap untuk digunakan di *handphone*. Pengujian ini terutama untuk mengetahui apakah tombol-tombol dan animasi yang terdapat pada aplikasi dapat berfungsi pada *handphone*.

Pada Tabel 4-32. ditunjukkan hasil pengujian aplikasi pada dimensi layar 240 x 320 pixel, 320 x 480 pixel, dan 480 x 850 pixel dengan ukuran layar yang berbeda beda.

Table 4.10. Pengujian Pixel

Dimensi Layar	Hasil Pengujian	Keterangan	Gambar
480 x 854 pixel, 4,2 inci LED-backlit LCD	Kurang Baik	Letak tombol terlalu keatas, sehingga terlihat ruang kosong pada tampilan aplikasi	
480 x 854 pixel, 3,3 inci LED-LFT	Kurang Baik	Letak tombol terlalu keatas, sehingga terlihat ruang kosong pada tampilan aplikasi	

320 x 480 pixel, 3,0 inci TFT Touchscreen	Baik	Tampilan sesuai dengan dimensi layar	
240 x 320 pixel, 3,0 inci TFT Touchscreen	Cukup Baik	Tampilan sedikit lebih kecil, sehingga ada sisi lain yang tidak terlihat untuk itu terdapat <i>scroll</i> untuk mengeser tampilan	
320 x 480 pixel, 3,27 inci TFT Touchscreen	Baik	Tampilan sesuai dengan dimensi layar	

Tabel pengujian pada *system operasi* pada *Smartphone Android* bisa dilihat pada Gambar di bawah ini :

Tabel 4.11 Pengujian untuk sistem operasi yang digunakan.

No	Nama <i>Smartphone</i>	Pengujian	
		OS (system operasi)	Keterangan
1	SONY XPERIA ARC LT 15i	V2.3 Gingerbread	OK
2	SONY XPERIA RAY	Google android V.3 Gingerbread	OK

3	SAMSUNG GALAXY YOUNG GT-S360	V2.1 Froyo	OK
4	SONY WALKMAN W8 E 161i	2.3.5 Gingerbread	OK
5	SAMSUNG GALAXY MINI 2	2.3.6 Gingerbread	OK

Aplikasi ini di uji pada 5 perangkat *android* dan diperoleh hasil pada keterangan baik memperoleh persentase yang cukup besar sehingga dapat di tarik kesimpulan bahwa aplikasi dapat berjalan di hampir semua *platform OS* (operating System) yang digunakan.

4.11.2 Pengujian Waktu untuk transfer data koordinat lokasi

Pengujian ini dilakukan untuk mengetahui berapa lama waktu yang diperlukan aplikasi untuk bisa mendapatkan posisi pengguna setelah (*Global Positioning System*) pada perangkat *Android* dihidupkan. Secara garis besar, pengujian dilakukan pada dua buah kondisi. Pada kondisi pertama, pengujian dilakukan pada lokasi yang bebas halangan terhadap langit (terbuka). Pada kondisi kedua, pengujian dilakukan pada lokasi yang atapnya menghalangi langit, namun ruangnya tidak tertutup penuh (semi terbuka).

Tabel 4.12. Hasil Pengujian terhadap waktu untuk transfer informasi koordinat.

No	Lokasi	Koordinat Tersimpan	Kondisi Lokasi	Waktu Respon (detik)	Signal
1	Jln Bulutangkis Lowokwaru Malang	-7.91291,112.62894	Terbuka	03.93	H
2	Jln Perusahaan Malang	-7.91028,112.63265	Terbuka	04.41	H
3	Perumahan Akordion Lowokwaru, Malang	-7.30891,112.78141	Terbuka	07.40	H
4	Masjid darus Sa'adah Jln Kemantren, Malang	-8.00233,112.609513	Terbuka	05.11	H

5	Ruko Dinoyo, Jln MT Haryono	-7.94343,112.610547	Terbuka	02.32	H
6	Matahari Pasar Besar Malang	-7.988103,112.63218	Semi Terbuka	10.34	H
7	Matos	-7.957173,112.61814	Semi Terbuka	15.04	II
8	Jalan Veteran, Lowokwaru	-7.959075,112.61823	Semi Terbuka	16.21	H

* II = Sinyal Internet dalam jangkauan sinyal HSDPA

Dengan melihat Tabel 4.12, dapat dihitung bahwa rata rata waktu akses data koordinat pada lokasi terbuka adalah 4,63 detik, sedangkan rata rata waktu akses data koordinat pada lokasi semi terbuka adalah 13,86 detik. Dengan begitu dapat diambil kesimpulan bahwa waktu akses data koordinat pada lokasi terbuka lebih singkat ketimbang waktu akses data koordinat pada lokasi tertutup. Hal ini dikarenakan untuk dapat bekerja secara maksimal, perangkat *Global Positioning System* harus diarahkan langsung ke langit dengan kondisi *Line of Sight* (LOS) sehingga sinyal tidak terhalang. Apabila terdapat penghalang, maka besar kemungkinan sinyal yang diterima perangkat merupakan hasil pemantulan sehingga waktu akses data koordinat menjadi lebih lama.

4.11.3 Pengujian terhadap akurasi data koordinat lokasi

Untuk mengetahui tingkat keakuratan dari data koordinat, maka pengujian dilakukan dengan *Global Positioning System* pembanding yaitu Blackberry Bold 9000. Tujuannya untuk mengetahui apakah koordinat apakah koordinat data *Global Positioning System* sama atau berbeda terhadap handphone lain.

Tabel 4.13. Hasil Pengujian dengan pembanding Blackberry Bold 9000.

No	Lokasi Pengujian	Koordinat Aplikasi yang terekam		Koordinat Pembanding Blackberry Bold 9000		Selisih Koordinat	
		Latitude	Longitude	Latitude	Longitude	Latitude	Longitude
1	Jln Bulutangkis Lowokwaru Malang	-7.91291	112.62894	-7.91290	112.62887	0.0001	0.00007
2	Jln Perusahaan Malang	-7.91028	112.63265	-7.91028	112.63263	0.0	0.00002

3	Perumahan Akordion Lowokwaru, Malang	-7.30891	112.78141	-7.30889	112.78140	0.0002	0.00001
4	Masjid datus Sa'adah Jln Kemantren, Malang	-8.00233	112.609513	-8.00230	112.609510	0.0	0.00003
5	Ruko Dinoyo, Jln MT Haryono	-7.94343	112.610547	-7.94342	112.610545	0.0001	0.00002
Kesalahan rata – rata						0.00008	0.000003

Berdasarkan hasil uji coba tingkat keakuratan data koordinat, maka dapat disimpulkan bahwa tingkat keakuratan *Global Positioning System* yang dihasilkan tidak sama antara yang ditunjukkan oleh *Global Positioning System* aplikasi dengan akurasi blackberry. *Global Positioning System* secara otomatis menunjukkan tingkat akurasi dengan rata rata 0.00008 untuk Latitude, sedangkan tingkat akurasi untuk Longitude diperoleh rata rata 0,000003. Hasil ini tidaklah mutlak karena tergantung banyak faktor, salah satunya cuaca.

BAB V PENUTUP

5.1. Kesimpulan

Setelah melakukan Implementasi dan pengujian pada “Rancang Bangun Aplikasi Mobile Penyimpan Lokasi Parkir Kendaraan Menggunakan *App Inventor* Berbasis *Android*” yang telah dibangun, maka dapat diperoleh kesimpulan bahwa:

1. Aplikasi dapat tampil optimal jika dijalankan pada handphone dengan resolusi 320x480 *pixel*, dengan ukuran layar 3 inci.
2. Hasil pengujian pada OS (*operating sistem*) dapat optimal dijalankan dengan OS dengan versi android froyo 2.2 keatas.
3. Untuk mendapatkan informasi koordinat secara tepat dan akurat sebaiknya memilih tempat parkir outdoor dan tanpa penghalang (*on sky*).
4. Waktu akses data koordinat pada lokasi terbuka lebih singkat dibandingkan akses lokasi semi terbuka. Dengan waktu rata-rata 4,63 detik untuk lokasi terbuka dan 13,86 detik untuk semi terbuka.
5. Hasil rata-rata tingkat akurasi dengan pembanding *global positioning system* pada smartphone diperoleh rata-rata rata rata 0,00008 untuk Latitude, sedangkan tingkat akurasi untuk Longitude diperoleh rata rata 0,000003. Hasil ini tidaklah mutlak karena tergantung banyak *factor*, salah satunya cuaca.

5.2. Saran

Adapun saran-saran yang dapat di berikan untuk penyempurnaan dan pengembangan sistem “Rancang Bangun Aplikasi Mobile Penyimpan Lokasi Parkir Kendaraan Menggunakan *App Inventor* Berbasis *Android*” adalah sebagai berikut :

1. Aplikasi yang telah dibuat dapat dikembangkan lagi supaya tingkat keakurasian data *global positioning system* bisa lebih sempurna dalam mendapatkan koordinat lokasi.

2. Aplikasi dapat dikembang pada tempat parkir yang berada di dalam gedung (*indoor*).
3. Aplikasi dapat dikembangkan lagi dengan membuat versi untuk segala ukuran layar selain 320 x 480 pixel dan dengan lebar 3 inci.

DAFTAR PUSTAKA

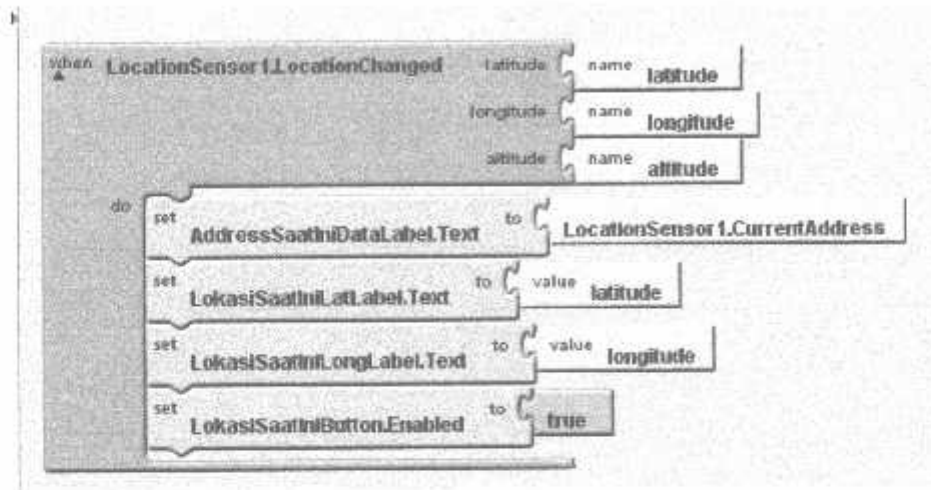
- [1] Ariyus, D. Sudarmawan. 2007, *Interaksi Manusia dan Komputer*, Yogyakarta : Andi.
- [2] Budiawan, Tiyo dan Santoso, Imam, “*Mobile Tracking GPS (Global Positioning System) melalui media SMS*”, Universitas Diponegoro, Semarang .pdf.
- [3] ELCOM, “*Hebatnya Google maps +Pintarnya Google Street*”, Penerbit Andi Yogyakarta, 2010.
- [4] Mulyadi, aka Mazadi, 2011, “*ANDROID APP INVENTOR* ”, Yogyakarta, Multimedia Center Publishing
- [5] Mulyana, Eueung, 2012 , “ *Ciptakan Sendiri Aplikasi Androidmu* “ , Yogyakarta, C.V ANDI OFFSET
- [6] Rekayasa Lalu Lintas “Parkir” BAB 7, Teknik Sipil Universitas Widyagama Malang, 2008.
- [7] Syaiful, M.Amri, “*Membangun sistem navigasi menggunakan Google Maps API* .pdf”, ITS, Surabaya.
- [8] Wibowo, Arie & Wardhani, Kusumaningati, 2008, “ *PERINTAH DASAR LINUX (BASIC COMMAND LINE)*.pdf.
- [9] Wolber, David, 2011, “ *APP INVENTOR CREATE YOUR OWN ANDROID APPS*.pdf “, O’Reilly, Cambridge
- [10] <http://beta.appinventor.mit.edu/learn/tutorial> : diakses tanggal 10 maret 2013
- [11] <http://diskusiandroid.com> diakses 2 maret 2013
- [12] [http://id.wikipedia.org/wiki/ Android_\(sistem_operasi\)](http://id.wikipedia.org/wiki/Android_(sistem_operasi)) diakses 10 maret 2013
- [13] [http://id.wikipedia.org/wiki/ Emulator](http://id.wikipedia.org/wiki/Emulator) diakses tanggal 9 maret 2013

LAMPIRAN

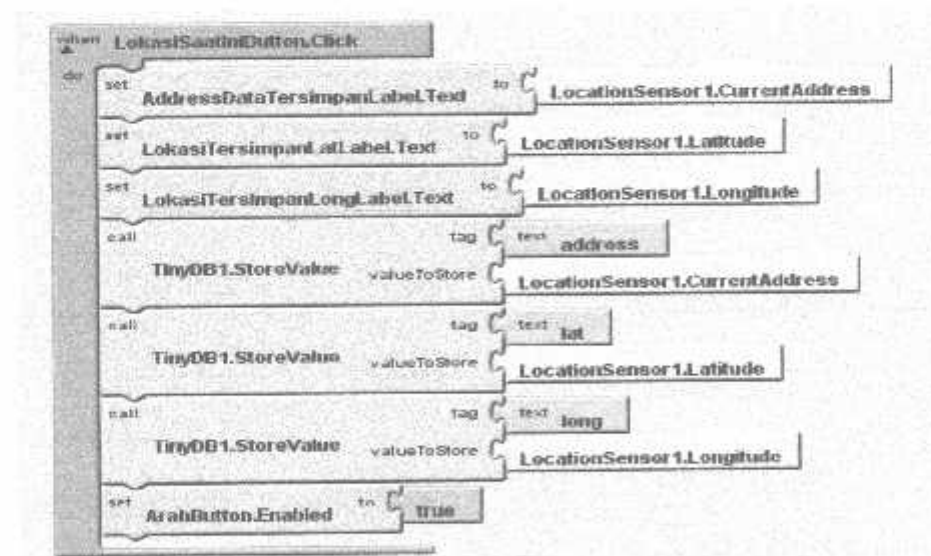
LAMPIRAN

(Block Editor)

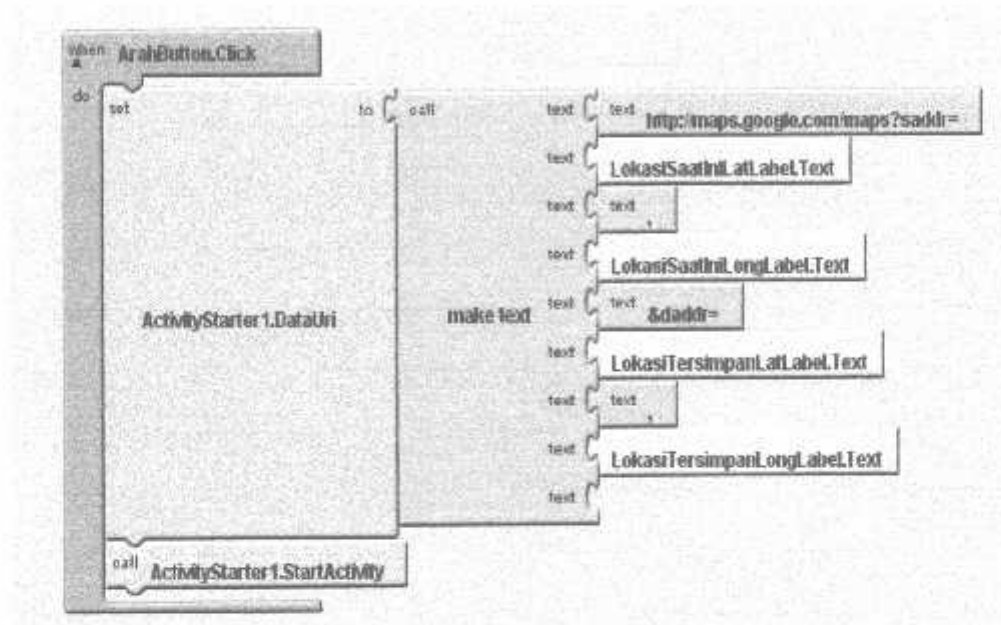
Block Editor Untuk Mendapatkan data Lokasi Saat Ini.



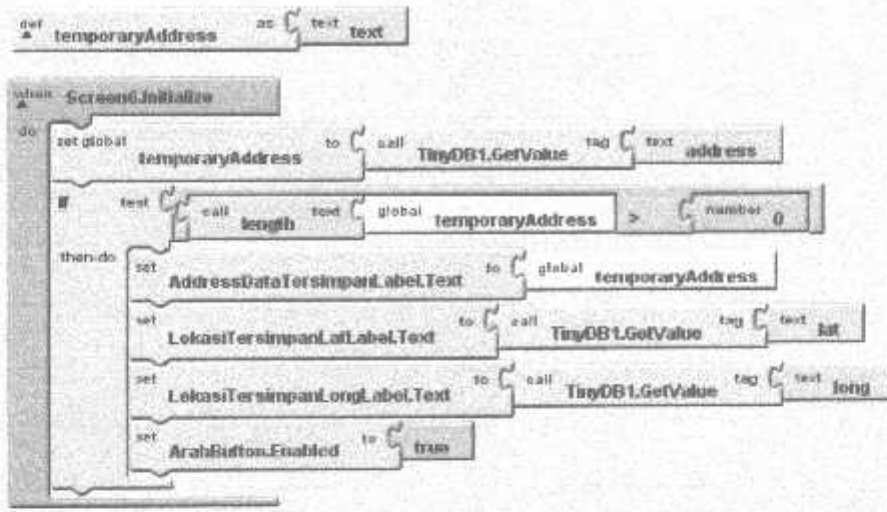
LocationSensor1.LocationChanged merupakan event handler dari pembacaan data koordinat Global positioning system. AddressSaatIniDataLabel.Text untuk mendapatkan data update lokasi baru. setLokasiSaatIniLatLabel.Text Untuk menempatkan data latitude. setLokasiSaatIniLongLabel.Text Untuk menempatkan data Longitude. SetSimpanLokasiSaatIniButton. Enable Untuk menyimpan pembacaan data lokasi.



TinyDB1.StoreValue sebanyak 3 (tiga) Untuk menyimpan data pada database.
 LocationSensor1.CurrentAddress Untuk menyimpan data address pada database.
 LocationSensor1.Latitude Untuk menyimpan data latitude pada database.
 LocationSensor1.Longitude Untuk menyimpan data longitude pada database.



set ActivityStarter1.DataUri URL untuk peta yang dituju. text(http://maps.google.com/maps?saddr=) Parameter alamat pertama lokasi awal. LokasiSaatIniLatLabel.Text Latitude lokasi saat ini. LokasiSaatIniLongLabel.Text Longitude lokasi saat ini. text("&daddr=") Parameter alamat kedua lokasi tujuan. LokasiTersimpanLatLabel.Text Latitude lokasi tersimpan. LokasiTersimpanLongLabel.Text Longitude lokasi tersimpan. ActivityStarter1.StartActivity Untuk membuka peta.



Data yang tersimpan di database akan kita tampilkan pada label **LokasiTersimpan**. Untuk itu menggunakan event `Screen1.Initialize`, dan juga fungsi `TinyDB.GetValue` untuk mengambil data dari database.




INSTITUT TEKNOLOGI NASIONAL MALANG
Fakultas Teknologi Industri
Program Studi Teknik Informatika S1
Jl. Raya Karanglo Km. 2 Malang

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama : Mochamad Nur Rofik
NIM : 09.18.012
Jurusan : Teknik Informatika S-1
Judul : Rancang Bangun Aplikasi Mobile Penyimpan Lokasi Parkir
Kendaraan Menggunakan App Inventor Berbasis Android


Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :
Hari : Jumat
Tanggal : 2 Agustus 2013
Nilai : 81,52

**Panitia Ujian Skripsi :
Ketua Majelis Penguji**



Joseph Dedy Irawan, ST, MT.
NIP. 197404162005011002

Anggota Penguji :

Dosen Penguji I


Ali Mahmudi, BEng, PhD
NIP.P. 1031000429

Dosen Penguji II


Sandy Nataly Mantja, Skom.
NIP.P. 1030800418



FORMULIR PERBAIKAN SKRIPSI

Nama : Mochamad Nur Rofik
NIM : 09.18.012
Prodi : Teknik Informatika S-1
Judul : Rancang Bangun Aplikasi Mobile Penyimpan Lokasi Parkir
Kendaraan Menggunakan App Inventor Berbasis Android

Tanggal	Penguji	Uraian	Paraf
2-8-2013	I	<ul style="list-style-type: none">- Hilangkan Kata-kata GPS dalam Laporan Skripsi- Hilangkan Persyaratan Sistem pada BAB III- Perbaiki Flowchart	
2-8-2013	II	<ul style="list-style-type: none">- Tidak membuat source code hanya menggunakan tool app inventor- Hilangkan kata-kata GPS- Revisi tulisan pada laporan- Hasil tidak maksimal karena tidak menggunakan metode pencarian- Flowchart- Pengujian jenis OS Android- Kesimpulan, saran- Daftar pustaka	

Anggota Penguji :

Dosen Penguji I

Ali Mahmudi, BEng, PhD
NIP.P. 1031000429

Dosen Penguji II

Sandy Nataly Mantja, Skom.
NIP.P. 1030800418

Mengetahui

Dosen Pembimbing I

Joseph Dedy Irawan, ST, MT.
NIP. 197404162005011002

Dosen Pembimbing II

Survo Adi Wibowo, ST, MT.
NIP. 1031000438

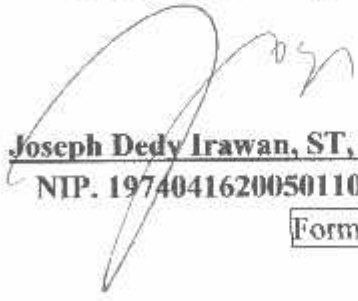


FORMULIR BIMBINGAN SKRIPSI

Nama : Mochamad Nur Rofik
Nim : 09.18.012
Masa Bimbingan : 11 Mei 2013 s/d 11 Nopember 2013
Judul Skripsi : Rancang Bangun Aplikasi Mobile Penyimpan Lokasi Parkir
Kendaraan Menggunakan App Inventor Berbasis Android

NO	TANGGAL	URAIAN	PARAF PEMBIMBING
1		Cek BAB I – II	
2		Cek BAB III - IV	
3	16-7-2013	Revisi Kesimpulan	
4	18-7-2013	Acc laporan	
5	22-7-2013	Acc makalah seminar hasil	
6	24-7-2013	Acc seminar hasil	
7			
8			
9			
10			

Malang, 30 Juli 2013
Dosen Pembimbing I


Joseph Dedy Irawan, ST, MT.
NIP. 197404162005011022

Form S-4b




INSTITUT TEKNOLOGI NASIONAL MALANG
Fakultas Teknologi Industri
Program Studi Teknik Informatika S1
Jl. Raya Karanglo Km. 2 Malang

FORMULIR BIMBINGAN SKRIPSI

Nama : Mochamad Nur Rofik
Nim : 09.18.012
Masa Bimbingan : 11 Mei 2013 s/d 11 Nopember 2013
Judul Skripsi : Rancang Bangun Aplikasi Mobile Penyimpan Lokasi Parkir
Kendaraan Menggunkan App Inventor Berbasis Android

NO	TANGGAL	URAIAN	PARAF PEMBIMBING
1	10-6-2013	Pengecekan program setengah jadi	
2	24-6-2013	Revisi program + cek BAB I, II	
3	1-7-2013	BAB II, III + revisi BAB III	
4	15-7-2013	BAB III + revisi	
5	16-7-2013	Penambahan pengujian pada OS, User, dan hanphone	
6	21-7-2013	Makalah seminar hasil	
7	24-7-2013	Acc makalah seminar hasil	
8	1-8-2013	Acc laporan	

Malang, 30 Juli 2013
Dosen Pembimbing II


Suryo Adi Wibowo, ST, MT.
NIP. 1031000438

Form S-4b



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigora-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karangas, Km 2 Telp. (0341) 417633 Fax. (0341) 417634 Malang

Nomor : ITN-78/T.INF/TA/2013
Lampiran : -
Perihal : Bimbingan Skripsi

11 Mei 2013

Kepada : Yth. Bpk/Ibu Joseph Dedy Irawan, ST, MT
Dosen Pembimbing Program Studi Teknik Informatika S1
Institut Teknologi Nasional
Malang

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : MOH NUR ROFIK
Nim : 0918012
Prodi : Teknik Informatika S1
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Bpk/Ibu selama masa waktu 6 (enam) bulan, terhitung mulai tanggal :

11 Mei 2013 – 11 Nopember 2013

Sebagai satu syarat untuk menempuh Ujian Sarjana Teknik. Program Studi Teknik Informatika S1.
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.



Mengetahui
Program Studi Teknik Informatika S1
Ketua,

Joseph Dedy Irawan, ST, MT
NIP : 197404162005021002

Form S-4a



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BN (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417635 Fax. (0341) 417634 Malang

Nomor : ITN-78/T.JNF/TA/2013
Lampiran : -
Perihal : Bimbingan Skripsi

11 Mei 2013

Kepada : Yth. Bpk/Ibu Suryo Adi Wibowo, ST.
Dosen Pembimbing Program Studi Teknik Informatika S1
Institut Teknologi Nasional
M a l a n g

Dengan hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

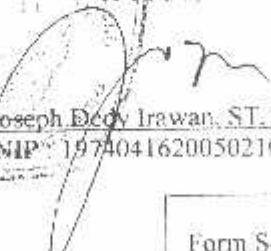
Name : MOH NUR ROFIK
Nim : 0918012
Prodi : Teknik Informatika S1
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Bpk/Ibu selama masa waktu 6 (enam) bulan, terhitung mulai tanggal :

11 Mei 2013 – 11 Nopember 2013

Sebagai satu syarat untuk menempuh Ujian Sarjana Teknik, Program Studi Teknik Informatika S1.
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Program Studi Teknik Informatika S1
Ketua,
Joseph Dedy Irawan, ST, MT
NIP. 197404162005021002



Form S-4a



FORMULIR PERBAIKAN UJIAN SKRIPSI

Dalam pelaksanaan Ujian Skripsi Jenjang Strata 1 Jurusan Teknik Informatika, maka perlu adanya perbaikan untuk mahasiswa :

Nama : MOCHA NUR ROFIN
NIM : 0918021
Perbaikan Meliputi :

1. TIDAK MEMBUAT SOURCE CODE PROGRAM HANYA MENGGUNAKAN TOOL APP (MATERI) DAN LINK ICE GOOGLE MAP
2. HILANGKAN KATA GPE
3. PERUBAHAN LIHAT LAPORAN
4. SEHINGGA HASIL TIDAK MAKSIMAL KARENA TIDAK MENGGUNAKAN METODE PEMCARIAN
5. PERUBAHAN SALAH
6. PERUBAHAN JENIS OF ANDROID PADA BAWAH
7. PERUBAHAN IPRAK
8. HAFAR RUSTAK
9. DEMO UJIAN

Malang, 2-8-13


(SANDY MASALE)