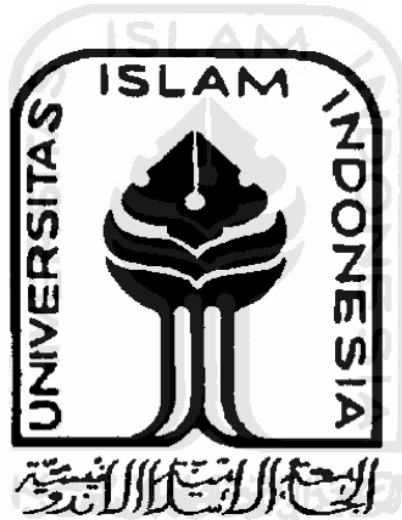


**Implementasi Pengembangan Aplikasi Reminder Berbasis
Android**

TUGAS AKHIR

**Diajukan sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Jurusan Teknik Informatika**



oleh :

Nama : Ian Jack Permana

No. Mahasiswa : 06 523 099

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2011**

**Implementasi Pengembangan Aplikasi Reminder Berbasis
Android**

TUGAS AKHIR

**Diajukan sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Jurusan Teknik Informatika**



oleh :

**Nama : Ian Jack Permana
No. Mahasiswa : 06 523 099**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2011**

LEMBAR PENGESAHAN PEMBIMBING

Implementasi Pengembangan Aplikasi Reminder Berbasis Android

TUGAS AKHIR



Nama : Ian Jack Permana

No. Mahasiswa : 06 523 099

الإسلام
Yogyakarta, 18 April 2011

Pembimbing


Hendrik, S.T., M.Eng.

LEMBAR PENGESAHAN PENGUJI
Implementasi Pengembangan Aplikasi Reminder Berbasis
Android

TUGAS AKHIR

oleh :

Nama : Ian Jack Permana

No. Mahasiswa : 06 523 099

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Informatika Fakultas
Teknologi Industri Universitas Islam Indonesia
Yogyakarta, 18 April 2011

Tim Penguji

Hendrik, ST., M.Eng.

Ketua

Yudi Prayudi, S.Si., M.Kom.

Anggota I

Nur Wijayaning Rahayu, S.Kom.

Anggota II



Mengetahui,

Ketua Jurusan Teknik Informatika
Universitas Islam Indonesia

Yudi Prayudi, S.Si., M.Kom.

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Saya yang bertanda tangan dibawah ini :


Nama : Ian Jack Permana

No. Mahasiswa : 06 523 099

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 bukanlah hasil karya saya sendiri, maka saya akan siap menanggung risiko dan konsekuensi apapun.

Demikianlah pernyataan ini saya buat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 27 April 2011



Ian Jack Permana

HALAMAN PERSEMBAHAN

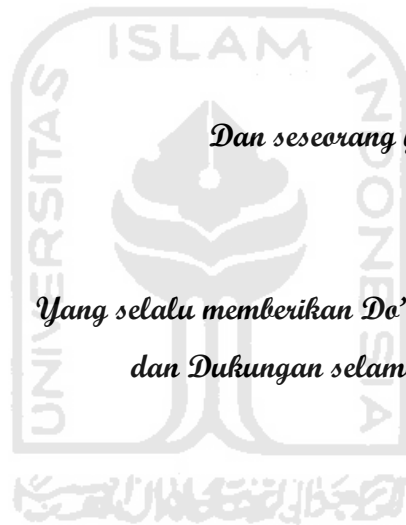
Kupersembahkan karya kecil ini sebagai tanda baktiku untuk

*Ibu dan Bapakku tercinta,
Fiti Mutiarawati dan Teguh Adi Nuryanto*

*Kakakku tercinta,
Herdian Ardi Febianto*

*Dan seseorang yang selalu mendukungku,
Komariah*

*Yang selalu memberikan Do'a, Semangat, Pengorbanan
dan Dukungan selama Aku menyelesaikan studi*



MOTTO

“Orang-orang yang beriman dan hati mereka menjadi tentram dengan mengingat Allah, ingatlah hanya dengan mengingat Allah hati mereka menjadi tentram.”

“sesungguhnya setelah kesulitan tersimpan sebuah kemudahan”

“Ilmu Allah tidaklah terhingga, seumpama dibuat pena dari seluruh pohon di bumi dan seluruh lautan sebagai tintanya, niscaya kamu tidak akan mampu menuliskan semua ilmu-Nya ”

“Sesungguhnya orang yang masuk surga adalah orang yang bermanfaat untuk orang disekitarnya”

“Jika kamu tidak memiliki apa yang kamu sukai, maka sukailah yang kamu miliki saat ini”

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalamu'alaikum Wr.Wb.

Alhamdulillah rabbil'alamin, puji syukur kehadiran Allah SWT, atas limpahan hidayah, taufiq serta 'inayah-Nya, sehingga Tugas Akhir yang berjudul “Aplikasi *Game Shooter* Berbasis Sistem Operasi Android Menggunakan Interaksi *Touchsreen*” ini dapat terselesaikan dengan baik. Shalawat serta salam semoga senantiasa tercurah atas Nabi Muhammad SAW, para kerabat, serta pengikutnya hingga hari kiamat nanti.

Laporan Tugas Akhir ini disusun untuk melengkapi salah satu syarat guna memperoleh gelar Sarjana Strata-1 (S1) Jurusan Teknik Informatika di Universitas Islam Indonesia.

Pada kesempatan ini penyusun ingin mengucapkan banyak terima kasih kepada pihak-pihak yang mempunyai andil besar dalam pelaksanaan dan penyelesaian laporan tugas akhir ini, antara lain :

Penulis menyadari bahwa selama penyusunan laporan tugas akhir ini tidak terlepas dari bantuan, bimbingan, dan dukungan baik secara material maupun moral dari berbagai pihak. Oleh karena itu penulis mengucapkan banyak terima kasih kepada:

1. Kedua orangtuaku atas segala limpahan cinta,kasih sayang,doa,dan dukungan yang tiada henti dan tidak akan pernah berhenti
2. Bapak Yudi Prayudi, S.Si., M.Kom., selaku Ketua Jurusan Teknik Informatika Fakultas Teknologi Industri.
3. Bapak Hendrik, ST., M.Eng., selaku dosen pembimbing tugas akhir penulis.
Terima kasih atas segala bimbingan, dukungan dan masukan yang telah diberikan

kepada penulis sehingga penyusunan laporan tugas akhir ini dapat terselesaikan dengan baik.

4. Seluruh dosen Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia. Terima kasih atas segala bimbingannya kepada penulis.
5. Seluruh karyawan dan staf bagian pengajaran khususnya bagian KP/TA FTI UII yang telah membantu dalam perijinan surat menyurat.
6. Bapak Kis, Bapak Anto, Bapak Halim yang telah memberikan bimbingannya walau hanya lewat dunia maya.
7. Umar Samiy, Khaerul Umam, Roni Septian, Surya Darmawan, Hari Setiaji, Rakhmat, Aryo, Willy, Adoy, Oji, Tio, terima kasih, terima kasih, dan terima kasih untuk semuanya.
8. Teman-teman FIRE dan seluruh teman-teman Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia. Terima kasih atas dukungannya kepada penulis.
9. Seluruh teman-teman dan berbagai pihak yang telah membantu baik secara langsung maupun tidak langsung kepada penulis.

Penyusun menyadari sepenuhnya bahwa laporan ini masih jauh dari sempurna, karena itu saran dan kritik sangat penyusun harapkan untuk perbaikan di masa mendatang.

Di tengah keterbatasan penyusun dalam mengerjakan laporan tugas akhir ini, penyusun berharap kiranya laporan ini bermanfaat bagi pembaca. Semoga Allah SWT membimbing dan menyertai setiap langkah kita. Amiin.

Wassalamu'alaikum Wr.Wb.

Yogyakarta, 9 April 2011

Penyusun

SARI

Perkembangan teknologi sangat berpengaruh terhadap perkembangan dunia pemrograman saat ini, dimana perubahannya akhir-akhir ini makin pesat berkembang. Seiring dengan perkembangan teknologi maka pengembangan pemrograman sekarang juga dapat diaplikasikan pada *handphone* terutama *smartphone*, dan *Operating System* (OS) *smartphone* terbaru saat ini adalah Android. Karena semua *smartphone* Android memiliki interaksi *touchscreen*, maka perlu diancang suatu aplikasi yang berjalan pada OS Android dan menggunakan interaksi berbasis *touchscreen* sebagai *interface*-nya.

Perancangan dan pembuatan Aplikasi *healthy reminder* berbasis sistem operasi Android Menggunakan Interaksi Touchsreen memerlukan adanya beberapa tahapan, yaitu : mendefinisikan masalah utama, pengumpulan data, perancangan perangkat lunak, pembuatan aplikasi, dan melakukan evaluasi dan pengujian sistem.

Aplikasi yang telah dibuat dapat berjalan pada OS Android yang bisa dijalankan dengan interaksi *touchscreen*, namun memiliki beberapa kelemahan karena memerlukan *resource memory* yang besar, dan akan lebih baik lagi jika interaksi bawaan pada *device* Android yaitu *voice command* juga digunakan sebagai salah satu metode interaksinya.

Kata Kunci : Eclipse, Android, Reminder.

TAKARIR

<i>Activity Diagram</i>	Diagram aktivitas
<i>Background</i>	Latar belakang
<i>Coding</i>	memberi kode logika pada program
<i>Crosshair</i>	garis sasaran tembak
<i>Developer</i>	Pengembang
<i>Device</i>	Piranti
<i>Emulator</i>	Pengganti perangkat lunak suatu alat yang berjalan di PC
<i>Gadget</i>	Piranti elektronik canggih
<i>Graphic User Interface</i>	Interaksi grafis untuk pengguna
<i>Input</i>	Masukan
<i>Interface</i>	Antarmuka
<i>Landscape</i>	Pemandangan
<i>Layout</i>	Lembaran
<i>Main Menu</i>	Menu utama
<i>Memory Resource</i>	Memori yang terpakai
<i>Native Application</i>	Aplikasi asli
<i>Open Source</i>	Sumber yang terbuka bebas
<i>Operating system</i>	Sistem operasi
<i>Output</i>	Keluaran
<i>Reminder</i>	Pengingat
<i>Screen Resolution</i>	Resolusi layar
<i>Script</i>	Kode naskah
<i>Smartphone</i>	Telepon cerdas
<i>Touch Screen</i>	Layar sentuh
<i>Voice Command</i>	Perintah Suara
<i>Vendor</i>	Perusahaan Penyedia

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN PEMBIMBING.....	ii
LEMBAR PENGESAHAN PENGUJI.....	iii
LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN	v
MOTTO	vi
KATA PENGANTAR	vii
SARI.....	ix
TAKARIR.....	x
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xiv
BAB I PENDAHULUAN.....	16
1.1 Latar Belakang	16
1.2 Rumusan Masalah.....	18
1.3 Batasan Masalah	18
1.4 Tujuan Penelitian	18
1.5 Manfaat Penelitian	18
1.6 Metode Penelitian	19
1.7 Sistematika Penulisan	20
BAB II LANDASAN TEORI.....	22
2.1 Perkembangan Sistem Operasi <i>Smartphone</i>	22
2.2 Sistem Operasi Android	27
2.3 Pengembangan Aplikasi Berbasis Android.....	29
2.3.1 Arsitektur Android.....	29
2.3.2 Linuk Kernel.....	30
2.3.3 Libraries	31
2.3.4 Android Runtime	31

2.3.5	Application Framework	31
2.3.6	Applications	32
2.3.7	Fundamental Aplikasi	32
2.3.7.1	Activities	33
2.3.7.2	Services	33
2.3.7.3	Broadcast Receiver	33
2.3.7.4	Content Providers	34
2.3.7.5	Layout	34
2.3.7.6	Android Manifest	34
2.3.7.7	Sumber Data/Resource	35
2.3.8	Eclipse	35
BAB III METODOLOGI		40
3.1	Metode Analisis	40
3.2	Analisis Kebutuhan	40
3.2.1	Analisis Kebutuhan Input	40
3.2.2	Analisis Kebutuhan Fungsi dan Kinerja	41
3.2.3	Analisis Kebutuhan Output	41
3.2.4	Analisis kebutuhan Perangkat Keras	41
3.2.5	Analisis Kebutuhan Perangkat Lunak	42
3.3	Perancangan Perangkat Lunak	43
3.3.1	Metode Perancangan	43
3.3.2	Hasil Perancangan	43
3.3.3	Use Case Diagram	44
3.3.4	Activity Diagram	44
3.3.5	Diagram Class	46
3.3.6	Perancangan Antarmuka	47
BAB IV HASIL DAN PEMBAHASAN		52

4.1	Batasan Implementasi	52
4.2	Tahap Pembuatan Proses.....	52
4.3	Implementasi Pembuatan Proses.....	53
4.4	Implementasi Interface.....	60
4.4.1	Implementasi Main Menu Aplikasi Reminder.....	60
4.4.2	Implementasi Interface Menu Opsi dan Sistem Inti	61
4.4.3	Implementasi Tombol Tambah.....	62
4.4.4	Implementasi Tombol Set.....	63
4.4.5	Implementasi Opsi Menitan.....	63
4.4.6	Implementasi Opsi Harian	64
4.4.7	Implementasi Opsi Mingguan.....	66
4.5	Implementasi Penggunaan Package	68
4.5.1	Package Aplikasi.....	68
4.5.2	Implementasi Code Fungsi.....	69
4.6	Pengujian Program.....	71
4.7	Analisis Kinerja Perangkat Lunak	79
4.7.1	Analisis Kesesuaian Dengan Landasan Teori.....	80
BAB V SIMPULAN DAN SARAN.....		81
Kesimpulan		81
Saran.....		81
DAFTAR PUSTAKA		82

DAFTAR GAMBAR

Gambar 2. 1 Market Share Feature Phone Vs Smartphone di AS	24
Gambar 2. 2 Pertumbuhan pasar Android di AS.....	25
Gambar 2. 3 Perkiraan pasar smartphone di AS	26
Gambar 2. 4 Arsitektur Platform Android	30
Gambar 2. 5 Membuat Project Android baru.....	36
Gambar 2. 6 Parameter Project Android	37
Gambar 2. 7 Struktur <i>project</i> aplikasi Android HelloWorld.....	39
Gambar 3. 1 Use Case Diagram.....	44
Gambar 3. 2 Activity Diagram.....	45
Gambar 3. 3 Perancangan Diagram Class.....	46
Gambar 3. 4 Perancangan Antarmuka Menu Utama	47
Gambar 3. 5 Perancangan antarmuka Set Reminder.....	48
Gambar 3. 6 Perancangan antarmuka Menu Opsi Set Reminder.....	48
Gambar 3. 7 Perancangan antarmuka Tambah	49
Gambar 3. 8 Perancangan antarmuka Set <i>Reminder</i>	49
Gambar 3. 9 Perancangan antarmuka opsi menitan.....	50
Gambar 3. 10 Perancangan antarmuka Opsi Harian	50
Gambar 3. 11 Perancangan antarmuka Opsi Mingguan.....	51
Gambar 4. 1 Tahap Pembuatan Layout Dasar Pada Droiddraw	53
Gambar 4. 2 Tahap Pembuatan <i>Layout</i> Pada <i>Aplikasi</i>	55
Gambar 4. 3 Tahap Pembuatan Menu Utama	57
Gambar 4. 4 Tahap Pengecekan <i>Emulator Engine</i>	57
Gambar 4. 5 Tahap Pembuatan <i>alarm reminder</i>	58
Gambar 4. 6 Tahap Pembuatan AndroidManifest.xml	60
Gambar 4. 7 Antarmuka aplikasi <i>healthy reminder</i>	61
Gambar 4. 8 Interface Menu Opsi dan Sistem Inti.....	62

Gambar 4. 9 Interface Tombol Tambah.....	62
Gambar 4. 10 Interface Tombol Set.....	63
Gambar 4. 11 Interface Opsi Menitan.....	64
Gambar 4. 12 Interface Pengaturan Waktu Menitan.....	64
Gambar 4. 13 Interface Opsi Harian	65
Gambar 4. 14 Interface Pengaturan Waktu 2X Sehari.....	66
Gambar 4. 15 Interface Pengaturan Waktu 3X Sehari.....	66
Gambar 4. 16 Interface Opsi Mingguan.....	67
Gambar 4. 17 Interface Pengaturan Waktu Mingguan.....	67
Gambar 4. 18 Pemilihan Ikon Healthy Reminder.....	72
Gambar 4. 19 Pemilihan Set Reminder.....	73
Gambar 4. 20 Pemilihan Menu Insulin.....	73
Gambar 4. 21 Pilih Opsi Menitan	74
Gambar 4. 22 Pemilhan Waktu.....	74
Gambar 4. 23 Set Waktu.....	75
Gambar 4. 24 Reminder Berjalan.....	75
Gambar 4. 25 Pemilihan Ikon Healthy Reminder.....	76
Gambar 4. 26 Pemilihan Set Reminder.....	77
Gambar 4. 27 Pemilihan Menu Insulin	77
Gambar 4. 28 Pilih Opsi Menitan	78
Gambar 4. 29 Pemilhan Hari.....	78
Gambar 4. 30 Set Waktu.....	79
Gambar 4. 31 Reminder Berjalan.....	79

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi sangat berpengaruh terhadap persaingan *smartphone* saat ini,. Pada sekitar awal tahun 2005, kita telah dapat melihat persaingan ketat antar *vendor* produsen *smartphone* yang berkompetisi dalam memasarkan produknya. *Smartphone* yang populer memiliki fitur lengkap dan kaya akan aplikasi sehingga dapat memanjakan pengguna, bukan hanya sekedar sebagai media komunikasi namun dapat dimungkinkan pula pengguna untuk memutar musik sampai dengan memainkan *game*.

Imbas dari kompetisi *smartphone* kini benar-benar menguntungkan bagi konsumen maupun bagi *vendor* itu sendiri. Hal ini dapat kita lihat dengan semakin berkembangnya fitur yang semakin beragam untuk mendukung *smartphone* tertentu. Hal ini sangatlah memanjakan para pengguna *smartphone* tersebut dan dengan hal yang tidak kalah pentingnya adalah faktor harga yang semakin bersaing antara *vendor smartphone* yang ada. Di sisi *vendor* itu sendiri, kompetisi dapat memacu mereka untuk semakin menciptakan produk yang revolusioner dan pengembangan aplikasi yang bersifat kontinu[ROG11].

Pada Juli 2000, Google bekerjasama dengan Android Inc., perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri Android Inc yang bergabung dengan Google, diantaranya Andy Rubi, Rich Miner, Nick Sears, dan Chris White. Pada saat itu banyak yang menganggap fungsi Android hanyalah sebagai perangkat lunak pada telepon seluler. Pada Juli 2005, Google mengakuisisi Android, Inc. Sejak saat itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler.

Data tersebut diperoleh dari ComScore secara detail yakni pada Februari 2010 Android mendapat 9 persen pasar ponsel yang lalu menjadi 13 persen pada Mei 2010. Walau masih di peringkat keempat namun rival terdekatnya yaitu Microsoft yang hanya berbeda 0.5 persen tampaknya akan dengan mudah dilewatinya dalam beberapa bulan kedepan. Hal ini menjadikannya *platform* dengan perkembangan tercepat saat ini.

Sistem Operasi Google Android di Ponsel memang masih tergolong baru, namun demikian Sistem Operasi Android telah mengalami perkembangan yang cukup pesat. Diciptakan sebagai produk tandingan iOS, Android menunjukkan grafik perkembangan yang signifikan, ini tentu tidak terlepas dari dukungan-pabrikasi ponsel besar yang ikut menghadirkan ponsel-ponsel bersistem operasi Android.

Ketika kita melirik pasar persaingan *smartphone* berdasarkan OS yang tertanam, bukan hanya *smartphone* milik iPhone dan Android yang sebagai pemain utama namun ada juga pemain lain seperti Symbian, Linux, Palm, RIM dan Microsoft Windows Mobile dan Windows 7. Namun di tahun 2010, kita dapat melihat secara jelas bahwa persaingan hanya ada diantara Apple (IOS) dan Google (Android). Apple IOS dan Android OS adalah sistem operasi yang memperoleh 2% di pasar sementara sisanya geser ke bawah grafik pasar. Sebelumnya Android OS memiliki 7% pangsa pasar tetapi naik ke 9% hanya dalam satu tahun. Disisi lain, Apple IOS yang telah menangkap 26% dari pangsa pasar naik 2% menjadi 28%[ANO11d].

Apple dan Google berusaha keras meyakinkan calon penggunanya untuk menggunakan *smartphone* dan sistem operasi milik mereka. Pada tahun 2010 kita melihat bahwa semakin banyak ponsel Android yang dijual dibandingkan dengan iPhone. Hal ini terjadi karena Google sedang mencoba untuk menargetkan untuk kalangan generasi muda sementara Apple menargetkan untuk semua kelompok umur. Sesuai dengan catatan, bahwa lebih dari 54% pengguna ponsel Android

berada di bawah 34 tahun. Sementara hanya 47% dari pengguna Apple iPhone OS yang berusia sama.

Kini perkembangan perangkat Android sedang berkembang pesat dan sejalan dengan itu dukungan aplikasinya juga semakin bertambah. Dengan kesempatan seperti itu *developer* dituntut agar dapat mengembangkan berbagai macam aplikasi pada *platform* Android sehingga aplikasi yang ditemukan akan semakin beragam.

1.2 Rumusan Masalah

Bagaimana mengembangkan sebuah aplikasi Berbasis Android

1.3 Batasan Masalah

Batasan masalah untuk aplikasi ini adalah sebagai berikut:

1. Aplikasi akan di-debug dan diuji menggunakan emulator *Smartphone* berbasis OS Android versi 2.2 / Froyo.
2. Interaksi menggunakan *touchscreen*.
3. Aplikasi yang dibuat adalah *healthy reminder* atau pengingat.

1.4 Tujuan Penelitian

Tujuan pembuatan tugas akhir ini adalah merancang suatu aplikasi yang berjalan pada OS Android dan menggunakan interaksi berbasis *touchscreen* sebagai *interface*-nya.

1.5 Manfaat Penelitian

Berikut beberapa manfaat yang diperoleh dari pembuatan tugas akhir ini, manfaat tersebut antara lain:

1. Untuk *Developer*
 - a. Mempelajari teknik pengembangan aplikasi berbasis Android.

- b. Sebagai aplikasi *sample* untuk memaksimalkan fasilitas touchscreen pada Android.
- c. Sebagai referensi untuk membuat aplikasi pada OS Android.

1.6 Metode Penelitian

Perancangan dan pembuatan aplikasi berbasis sistem operasi Android Menggunakan Interaksi Touchsreen memerlukan adanya beberapa metodologi penelitian yang dilaksanakan demi kelancaran penelitian tindakan yang dilakukan, Metodologi penelitian tersebut antara lain:

1. Mendefinisikan masalah utama dan menetapkan tujuan. Pokok permasalahan dalam penelitian ini telah dijelaskan di rumusan masalah, tujuan penelitian pun telah dijelaskan sebelumnya.
2. Mengumpulkan data primer dan sekunder yang diperoleh dari Internet dan telaah studi pustaka yang terdapat di beberapa literatur mengenai permasalahan terkait. Data penelitian diperoleh dari beberapa sumber tertulis yang akurat kebenarannya. Sumber-sumber diperoleh dari:
 - a. Buku-buku dan literatur berkaitan dengan OS Android.
 - b. Pencarian melalui Internet.
3. Perancangan perangkat lunak. Tahapan perancangan perangkat lunak merupakan tahap awal dalam proses pembangunan suatu perangkat lunak. Dalam tahapan ini juga dijelaskan mengenai analisis kebutuhan perangkat lunak yang disertai gambaran perancangan perangkat lunak. Metode yang digunakan dalam perancangan kebutuhan perangkat lunak adalah rekayasa perangkat lunak.
4. Pengembangan aplikasi berbasis Android. Metode ini merupakan salah satu tahapan terpenting dalam penelitian, karena inti dari penelitian adalah pembuatan sebuah perangkat lunak, dalam tahapan ini perancangan yang telah dibuat sebelumnya kemudian diimplementasikan ke dalam bahasa pemrograman dan antar muka sistem. *Software* yang

dibutuhkan untuk membangun sistem antara lain:

- a. *Emulator*: Android SDK
 - b. *Script Editor* dan *Interface*: Eclipse 3.5(Galileo)
 - c. *Simulator*: ADT 0.9.4
5. Melakukan evaluasi dan pengujian sistem. Metode yang terakhir adalah pengujian dan evaluasi sistem yang selanjutnya dapat dijadikan bahan revisi apabila terdapat kesalahan dalam sistem.

1.7 Sistematika Penulisan

Sistematika penulisan untuk laporan tugas akhir ini terdiri dari 5 bab, dengan penjelasan sebagai berikut:

BAB I Pendahuluan, merupakan pembahasan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan laporan.

BAB II Landasan Teori, membahas dasar-dasar teori yang digunakan dalam penelitian serta tinjauan pustaka penelitian yang sejenis. Dasar-dasar teori yang dibahas mencakup penjelasan OS Android, peran OS Android terhadap perkembangan *smartphone*, dan penjelasan kebutuhan sistem.

BAB III Metodologi, merupakan penjelasan tentang metode analisis kebutuhan perangkat lunak, metode analisis yang digunakan serta hasil analisis, *input*, *output*, kebutuhan perangkat lunak, kebutuhan perangkat keras, antar muka (*interface*) yang dibutuhkan dan fungsi yang diharapkan. Pada bagian ini juga memuat metode perancangan perangkat lunak yang akan digunakan, hasil perancangan perangkat lunak. Selain itu juga memuat perancangan antarmuka sistem.

BAB IV Hasil dan Pembahasan, menjelaskan implementasi perangkat lunak disertai dengan kelebihan dan kekurangan aplikasi. Meliputi tampilan antar muka, bahasa pemrograman yang digunakan, prosedur bahasa pemrograman Java untuk Android, pengujian dan penjelasan fungsi-fungsi yang merupakan tahap lanjutan dari

proses menjadi suatu perangkat lunak dari sejumlah menu, proses, *input* dan *output* yang mewakili pengolahan data.

BAB V Kesimpulan dan Saran, merupakan kesimpulan berupa rangkuman-rangkuman hasil penelitian dan saran-saran mengenai keterbatasan yang ditemukan selama proses pembuatan laporan penelitian dan selama proses perancangan serta pembuatan aplikasi.



BAB II

LANDASAN TEORI

2.1 Perkembangan Sistem Operasi *Smartphone*

Sistem operasi *gadget smartphone* di dunia saat ini yang terkenal tercatat ada 7 antara lain: Android, Apple iPhone, Palm WebOS, Microsoft Window Mobile, Palm, RIM BlackBerry, dan Symbian. Masing-masing mengklaim produknya sebagai sistem operasi yang terbaik. Berkaitan dengan pasar pengguna *smartphone* itu sendiri berdasarkan data Nielsen (2010), di Amerika sekarang pertumbuhan pengguna Android cukup pesat, hanya dalam 1 tahun saja share pengguna Android menjadi hampir 20% dari sebelumnya tidak lebih dari 5%. Jika dibandingkan dengan iPhone maka perbedaan antara pengguna Android dan iPhone sangatlah tipis yaitu hanya sekitar 9% saja, dan itu termasuk pula dengan perusahaan raksasa RIM Blackberry (RIM BB) yang masih menguasai pasar *smartphone* di Amerika. Microsoft Mobile yang awalnya berambisi besar untuk mengalahkan RIM BB dan iPhone ternyata malah menurun akibat banyak penggunanya yang beralih ke sistem operasi yang baru yang bernama Android.

RIM BB memang saat ini masih menguasai pasar *smartphone* di Amerika, terutama untuk pengguna berumur 45 tahun keatas, sedangkan untuk generasi yang lebih muda (di bawah 44 tahun) masih menggunakan iPhone. Pengguna Android sendiri mempunyai kecenderungan di bawah usia 44 tahun, sedangkan Microsoft mobile masih disukai oleh generasi tua yaitu usia 55 tahun ke atas.

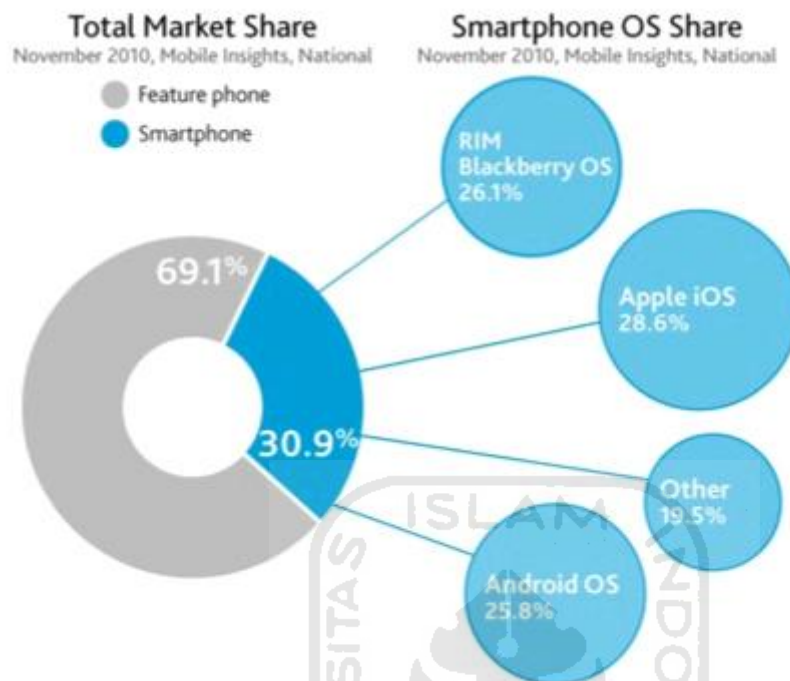
Sebagian pengguna *smartphone* berusia muda yaitu 42% penggunanya berumur 18-34 tahun dan hanya 6% penggunanya manula. Sedangkan Android sendiri penggunanya sebagian besar generasi muda, setengah penggunanya berusia 18-34 tahun. Sedangkan pengguna RIM BB dan Microsoft Mobile, sebagian besar berusia 35 tahun ke atas. Disini terlihat sekali bahwa generasi muda yang paling

mudah menerima teknologi baru, menggunakan sistem operasi yang baru dan menawarkan fitur yang beragam.

Di pasar ponsel saat ini beredar dua kategori ponsel secara garis besar, yaitu *feature phone* dan *smartphone*. Apa yang membedakan kedua ponsel ini? Menurut wikipedia, *feature phone* adalah ponsel yang sangat minim nilai komputasinya atau memiliki kemampuan komputasi yang rendah. Biasanya ponsel yang termasuk ke kelompok *feature phone* adalah ponsel *low end*. Sementara itu, *smartphone* adalah ponsel cerdas, ponsel yang memiliki kemampuan komputasi yang tinggi. Samsung juga menambahkan kriteria bagi *smartphone* ini untuk membedakan *smartphone* buatan mereka dari Nokia yang sering mengklaim menjadi penguasa di pasar *smartphone* dengan mematok besarnya prosesor *smartphone* pada batas minimal tertentu.

Menarik untuk disimak menurut suatu tabloid yang khusus membahas ponsel, *smartphone* di Indonesia makin disukai banyak orang. Hal ini tidak lain karena harga *smartphone* sudah cukup masuk akal bagi sebagian besar orang sehingga membeli *feature phone* bisa jadi merupakan kerugian karena dengan harga yang tidak begitu mahal, sudah diperoleh *smartphone* dengan nilai komputasi yang baik yang memberikan pengalaman *gadget* yang lebih variatif, seperti contoh layanan internet super cepat.

Demikian juga di pasar Amerika. Sebuah studi yang dilakukan oleh **Nielsen Company**(2010) sebagaimana dikutip oleh mashable.com memperlihatkan pangsa pasar *smartphone* yang terus menanjak disepanjang tahun 2010 yang lalu. Nielsen juga memperkirakan bahwa di tahun 2011 ini, paling tidak di Q3, *smartphone* akan menyamai bahkan bisa melampaui *market share feature phone* di Amerika Serikat. Berikut ini adalah grafik yang menggambarkan pangsa pasar selama 2010 yang ditunjukkan pada gambar 2.1.



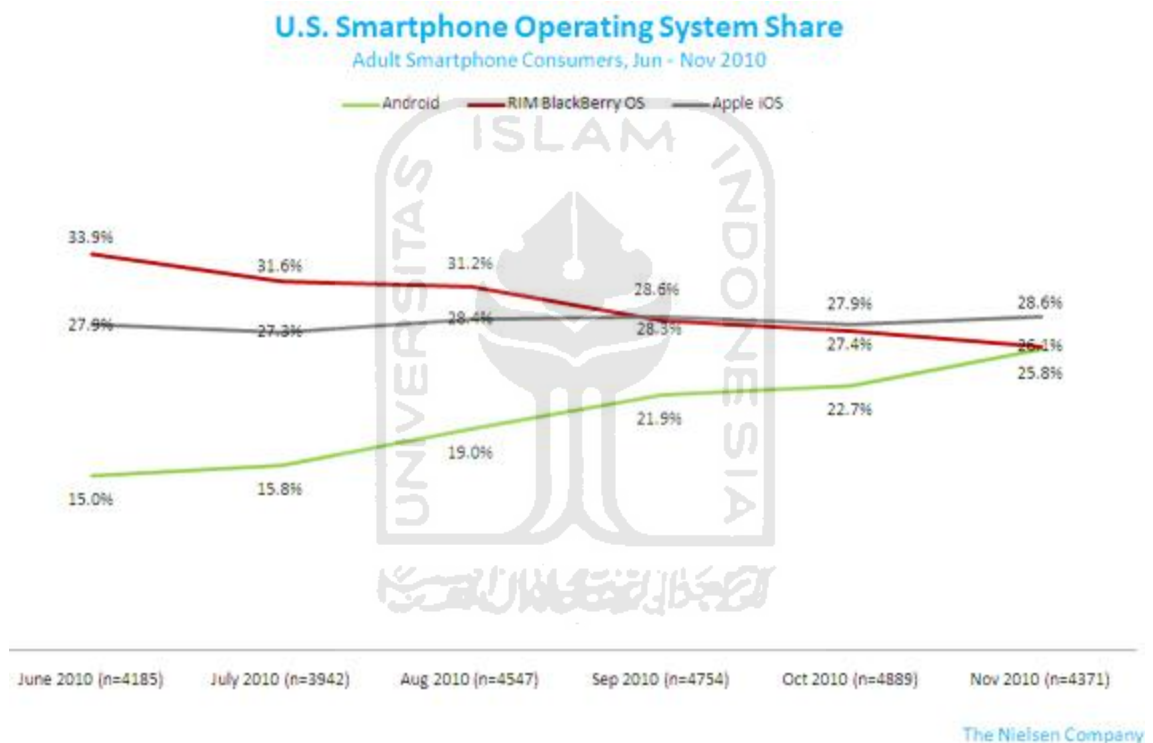
Gambar 2. 1 Market Share Feature Phone Vs Smartphone di AS

(sumber: <http://mashable.com>)

Nielsen mencatat bahwa *smartphone* merupakan pilihan pertama sebagian besar orang Amerika yang membeli gadget baru. Gaung komputer tablet seperti iPad bahkan tidak memengaruhi minat orang Amerika untuk membeli *smartphone*. Layak untuk ditelusuri, walaupun hanya memperoleh market share sebesar 30,9% di Q3 2010, angka ini merupakan lompatan yang lumayan jauh dibandingkan dengan angka di Q1 2010 yang berada di 24%. Bahkan rilis Nielsen terdahulu memberikan perkiraan angka bagi market share *smartphone* di AS yaitu sebesar 35% di Q4 2010.

Dalam pangsa pasar *smartphone* tersebut selain Symbian yang makin tidak populer, terdapat tiga nama sistem operasi yang bertarung sangat ketat yaitu iOS dari Apple, Blackberry dari RIM, dan Android dari Google. Masing-masing sistem operasi ini memiliki *market share* seperti ditunjukkan oleh grafik diatas. Pada Q3 seperti ditunjukkan gambar 2.1, iOS Apple masih menguasai pasar AS dengan 28,6%, diikuti oleh Blackberry dengan 26,1% dan Android di angka 25,8%. Yang

perlu diperhatikan bahwa pencapaian Android adalah pencapaian yang paling fenomenal. Android per-bulan Oktober 2010 yang lalu telah melampaui penjualan iPhone di AS. Bahkan Nielsen mengklaim dalam enam bulan dari Juni 2010 hingga November 2010, Android berhasil juara dalam hal *smartphone* paling banyak dibeli, yaitu sebesar 40%. Suka atau tidak meriahnya pasar *smartphone* saat ini tidak lain karena begitu pesatnya perkembangan OS Android ini. Kita bisa melihat pertumbuhan pasar Android tersebut dalam gambar 2.2[ANO11d].



Gambar 2. 2 Pertumbuhan pasar Android di AS

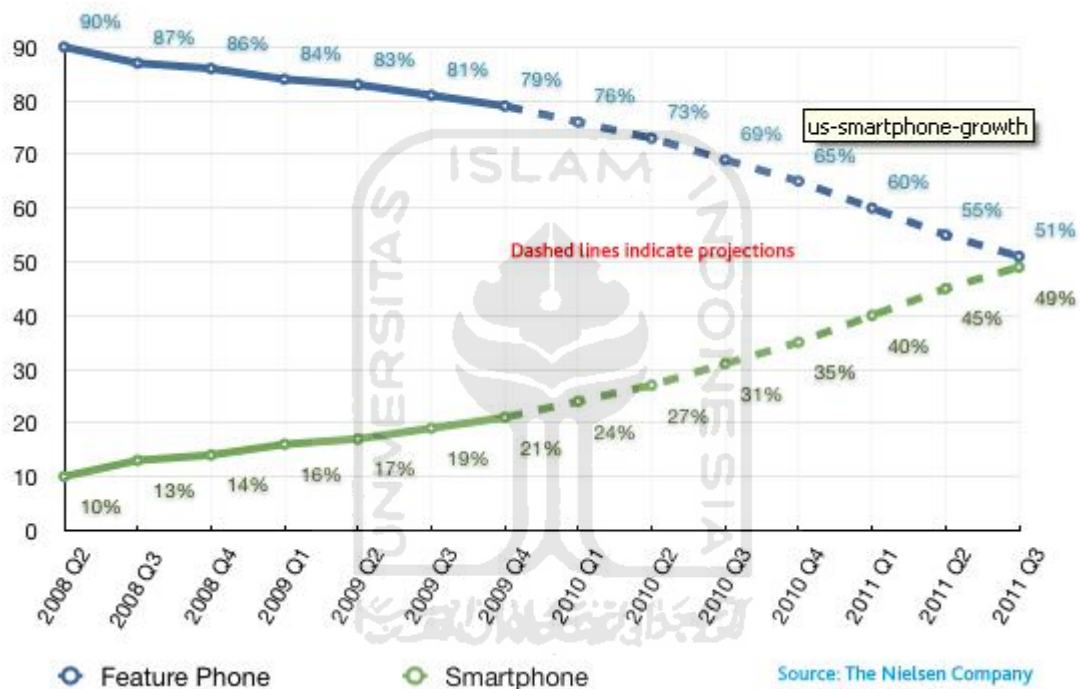
(sumber: <http://mashable.com>)

Dari gambar dapat dilihat bahwa hanya Android yang tumbuh selama tahun 2010 di AS. iOS Apple mengalami stagnasi, sedangkan Blackberry mengalami penurunan. Dengan demikian makin besarnya pangsa pasar *smartphone* tidak terlepas dari makin tumbuhnya *smartphone* berbasis Android. Nielsen juga mencatat 45%

pembeli baru lebih memilih untuk membeli *smartphone* dibandingkan dengan *feature phone*.

Nielsen memperkirakan pada tahun 2011 paling tidak di Q3, *market share feature phone* akan lebih kurang sama dengan *smartphone*. Perkiraan Nielsen tersebut dapat dilihat dalam grafik yang dijelaskan pada gambar 2.3[ANO11d].

U.S. Smartphone Penetration & Projections



Gambar 2. 3 Perkiraan pasar smartphone di AS

(sumber: <http://blog.nielsen.com>)

Pertumbuhan positif pasar *smartphone* tersebut tentu ditunjang oleh kemampuan atau fitur yang dimiliki oleh *smartphone* itu sendiri. Saat ini, sebagian besar orang tidak hanya menggunakan ponsel untuk berkomunikasi melalui pembicaraan. Banyak orang seiring dengan majunya media sosial di internet menggunakan ponsel mereka untuk berinteraksi di media sosial tersebut. Walaupun

pada beberapa *feature phone* tertentu fitur ini juga tersedia, tentu akan lebih baik dilakukan melalui *smartphone* dengan janji koneksi yang lebih beragam seperti kecepatan internet 3,5 G, Wi Fi dan lainnya. Selain itu, walaupun *feature phone* memiliki kamera tentu tidak sebaik kualitas kamera *smartphone* sehingga gambar yang diambil juga kurang baik kualitasnya dibandingkan dengan *smartphone*.

Selain hal itu beberapa *smartphone* seperti Blackberry memberikan kemudahan dalam urusan kerja para penggunanya dengan memberikan layanan e-mail terintegrasi. Banyak orang menggunakan *smartphone* miliknya untuk berinteraksi dengan pelanggannya, bahkan beberapa perusahaan tertentu memberikan *smartphone* tertentu untuk para karyawannya agar lebih mudah dalam bekerja.

Kelebihan-kelebihan *smartphone* inilah yang saya rasa memacu pertumbuhan positif pangsa pasar *smartphone* baik di AS maupun di seluruh dunia. Satu lagi seperti yang sudah disampaikan di atas, harga *smartphone* yang semakin bersahabat tentu menjadi pertimbangan penting dalam rangka menjustifikasi pilihan membeli *smartphone* dibandingkan dengan *feature phone*[ROG11].

2.2 Sistem Operasi Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan *platform open source* (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 yaitu konsorsium dari 34 perusahaan peranti keras, peranti lunak dan telekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile dan Nvidia.

Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau Google Mail Services (GMS) dan yang kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distribution (OHD).

Pada tahun 2005 Google mengakuisisi Android Inc yang pada saat itu diprakarsai oleh Andy Rubin, Rich Miner, Nick Sears dan Chris White. Kemudian pada tahun itu pula Google memulai membangun *platform* Android secara intensif.

Kemudian pada tanggal 12 November 2007 Google bersama OHA yaitu konsorium perangkat mobile terbuka, merilis Google Android SDK setelah mengumumkannya seminggu sebelumnya. Dan sambutannya sangat luar biasa, hampir semua media berita tentang IT & *Programming* memberitakan tentang dirilisnya Android SDK ini.

Ini dikarenakan ide dari *platform* Android ini sangat menarik untuk *developer* dan programer diseluruh penjuru dunia. Di sisi lain, produsen pembuat *handset* ponsel juga segera berlomba-lomba membuat handset dengan *platform* Android ini.

Google bersama dengan OHA merilis paket *software* SDK yang lengkap untuk mengembangkan aplikasi pada perangkat mobile, yaitu : Sistem Operasi, Middleware dan Aplikasi utama untuk perangkat mobile. *Programmer* dan *developer* bisa melakukan segalanya, mulai dari membuat aplikasi pengiriman SMS hanya dengan dua baris kode, hingga mengganti *event* pada *Home Screen* pada perangkat Android. Selain itu, bahkan dengan mudah kita bisa membuat dan mengkostumisasi Sistem Operasinya, atau mengganti semua aplikasi *default* Google[SON10].

Semua Aplikasi yang dibuat untuk Android akan memiliki akses yang setara dalam mengakses seluruh kemampuan *handset* tanpa membedakan apakah itu merupakan aplikasi inti atau aplikasi pihak ketiga. Dalam kata lain dengan *platform* Android ini, *programmer* atau *developer* secara penuh akan bisa mengkostumisasi perangkat Androidnya.

Android dibangun pada Linux Kernel (*Open Linux Kernel*) dengan sebuah mesin virtual yang telah didesain dan untuk mengoptimalkan penggunaan sumber daya memori dan *hardware* pada lingkungan perangkat *mobile* (*Mobile Environment*). Dalvik adalah nama dari Android Virtual Mesin, mengeksekusi file ke dalam format *Dalvik Executable* (*.dex). Sebuah format yang telah dirancang untuk

ruang penyimpanan yang efisien dan eksekusi memori yang terpetakan (*memory-mappable execution*).

Dalvik Virtual mesin (*DalvikVM*) berbasis register (*register-based*) dan dapat mengeksekusi kelas (*class*) yang telah terkompilasi pada *compiler* bahasa Java, kemudian ditransformasikan ke dalam *native format* dengan menggunakan Tool “dx” yang telah terintegrasi. Kita mungkin telah mengenal Java Virtual Mesin (*JavaVM*), yang saat ini kita temukan pada setiap kompiler *desktop*. Berbeda dengan DalvikVM, JavaVM berbasis *stack* (*Stack-based Virtual Machines*). DalvikVM memiliki keunggulan menggunakan *Registered Based*, ini karena pada prosesor ponsel telah dioptimasi untuk eksekusi berbasis register (*register-based execution*)[ANO11a].

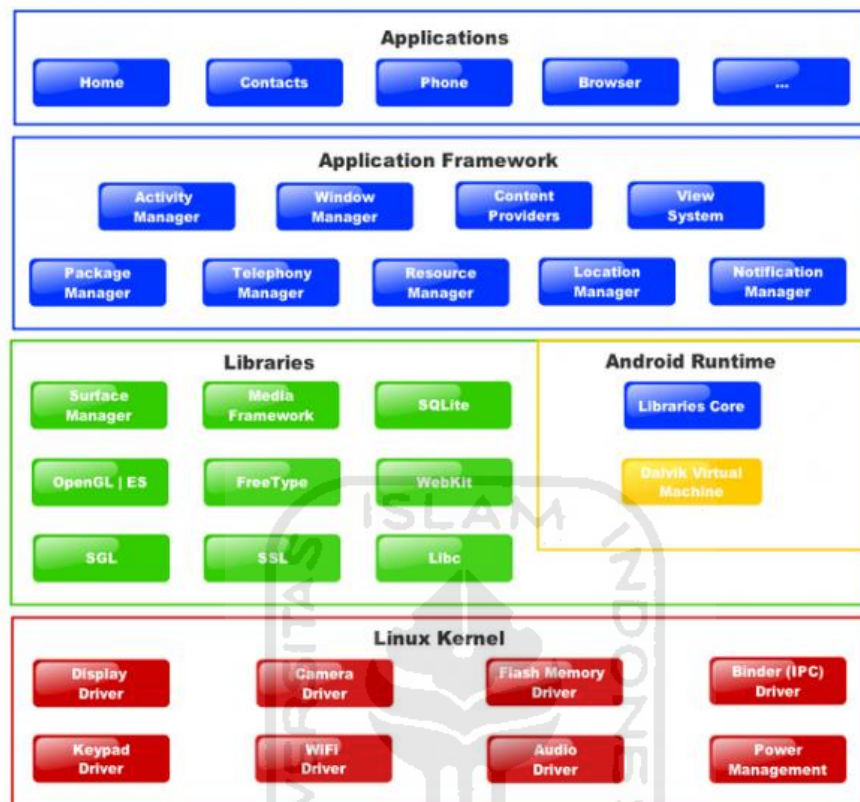
2.3 Pengembangan Aplikasi Berbasis Android

Pengembangan aplikasi adalah proses di mana sebuah aplikasi dibuat. Pengembangan aplikasi dilakukan oleh seorang developer, bisa 1 orang atau 1 perusahaan besar. Biasanya, aplikasi komersial berskala besar dibuat oleh tim pengembang dalam sebuah perusahaan.

Dalam pengembangan aplikasi dalam Android OS sendiri dapat menggunakan berbagai cara, beberapa diantaranya adalah dengan menggunakan Android SDK, bahasa pemrograman Java dan Eclipse. Penggunaan Eclipse dan Android SDK adalah untuk membentuk sebuah aplikasi dengan ekstensi .apk agar dapat diinstal di perangkat berbasis Android.

2.3.1 Arsitektur Android

Arsitektur dari sistem pada Android terdiri dari beberapa komponen. Komponen utama dari sistem operasi Android akan ditunjukkan pada gambar 2.4[MUL10].



Gambar 2. 4 Arsitektur Platform Android

2.3.2 Linuk Kernel

Android bukan Linux, akan tetapi Android dibangun di atas Linux Kernel yaitu versi 2.6 sehingga kehandalannya bisa dipercaya. Untuk inti sistem service Linux yang digunakan seperti keamanan, manajemen memori, proses manajemen, network, dan driver model. Seperti yang terlihat pada gambar 2.4, Linux Kernel menyediakan Driver Layer, Camera, Keypad, Wi-Fi, Flash Memory, Audio, dan IPC (Interprocess Communication) untuk mengatur aplikasi dan keamanan. Kernel juga bertindak sebagai lapisan abstrak antara hardware dan software stacknya [MUL10].

2.3.3 Libraries

Android menyertakan libraries C / C++ yang digunakan oleh berbagai komponen dari sistem Android. Kemampuan ini disediakan kepada developer aplikasi melalui framework aplikasi Android[MUL10].

2.3.4 Android Runtime

Android terdiri dari satu set core libraries yang menyediakan sebagian besar fungsi yang sama dengan yang terdapat pada core libraries bahasa pemrograman Java. Setiap aplikasi menjalankan prosesnya sendiri dalam Android, dengan masing-masing instan dari Dalvik VM yang mengeksekusi file dalam Dalvik executable (.dex), sebuah format yang dioptimalkan untuk memori yang kecil. Dalvik VM berbasis, berjalan dan dikompilasi oleh compiler bahasa Java yang telah ditransformasikan ke dalam .dex format oleh tool “dx” yang telah disertakan. Dalvik VM bergantung pada kernel Linux untuk berfungsi, seperti threading dan manajemen memori tingkat rendah[MUL10].

2.3.5 Application Framework

Pengembang memiliki akses penuh menuju API framework yang sama dengan yang digunakan oleh aplikasi inti. Arsitektur aplikasi dirancang agar komponen dapat digunakan kembali (reuse) dengan mudah. Setiap aplikasi dapat memanfaatkan kemampuan ini dan aplikasi yang lain mungkin akan memanfaatkan kemampuan ini sesuai dengan batasan keamanan yang didefinisikan oleh framework). Mekanisme yang sama memungkinkan komponen untuk diganti oleh pengguna. Semua aplikasi merupakan rangkaian set layanan dan sistem, termasuk:

Views

Kaya akan extensible yang dapat digunakan untuk membangun aplikasi, termasuk list, grids, kotak teks, tombol, dan bahkan sebuah embeddable web.

Content Provider

Memungkinkan aplikasi untuk mengakses data atau membagi data tersebut.

Resource Manager

Menyediakan aplikasi untuk mengakses ke kode non sumber lokal seperti string, gambar, dan tata letak file.

Notification Manager

Memungkinkan semua aplikasi menampilkan alert yang bisa dikostumasi di dalam status bar.

Activity Manager

Mengelola siklus hidup aplikasi dan menyediakan navigasi umum backstack.

2.3.6 Applications

Android telah menyertakan aplikasi inti seperti *email client*, SMS, kalender, peta, browser, kontak, dan lainnya. Semua aplikasi tersebut ditulis dengan menggunakan bahasa pemrograman Java. Pada *layer* inilah *developer* atau kita menempatkan aplikasi yang dibuat. Yang istimewa adalah pada Android semua aplikasi baik aplikasi inti (*native*) maupun aplikasi pihak ketiga berjalan semua aplikasi yang dibuat untuk Android akan memiliki akses yang setara dalam mengakses seluruh kemampuan *handset*, tanpa membedakan apakah itu merupakan aplikasi ini, *Programer* atau *developer* secara penuh akan bisa mengkostumisasi perangkat Androidnya[ANO11a].

2.3.7 Fundamental Aplikasi

Aplikasi Android ditulis dalam bahasa pemrograman Java, yaitu kode Java yang terkompilasi bersama –sama dengan data dan file *resources* yang dibutuhkan oleh aplikasi yang digabungkan oleh *aapt-tools* menjadi paket Android, sebuah file yang ditandai dengan *suffix .apk*. File inilah yang didistribusikan sebagai aplikasi dan diinstal pada perangkat mobile.

Bahasa pemrograman yang digunakan untuk membuat aplikasi pada platform Android adalah Java, namun dengan terminologi tersendiri yang sering membuat para

developer sulit memahaminya. Pada artikel ini kita akan coba memahami terminologi tersebut[ANO11b].

2.3.7.1 Activities

Sebuah activity normalnya akan menjadikan UI bagi pengguna, dimana pengguna kemudian bisa melakukan interaksi padanya. Misalnya, sebuah activity dapat memberikan daftar menu kepada pengguna, dan kemudian pengguna dapat memilih atau menampilkan foto. Sebuah aplikasi pesan teks mungkin memiliki satu activity yang menunjukkan daftar kontak untuk mengirim pesan activity kedua untuk menulis pesan pada kontak yang dipilih, dan activity lainnya untuk memeriksa pesan-pesan lama atau mengubah pengaturan. Walaupun mereka terlihat sebagai sebuah aplikasi, tetapi masing-masing activity tersebut adalah mandiri. Masing-masing diimplementasikan sebagai subkelas (sub class) dari kelas induk (base class) yang bernama activity.

2.3.7.2 Services

Service tidak memiliki visual interface (UI). Namun berjalan di latar belakang (background) untuk jangka waktu tak terbatas. Misalnya, service mungkin memainkan musik pada background, atau mungkin mengambil data melalui jaringan atau menghitung sesuatu dan memberikan hasil untuk activity yang membutuhkannya. Setiap service meng-extends kelas induk service.

2.3.7.3 Broadcast Receiver

Komponen yang tidak melakukan apapun selain menerima dan beraksi untuk menyiarkan notifikasi (announcements). Banyak broadcast berasal dari dalam sistem kode misalnya, notifikasi zona waktu yang telah berubah, bahwa baterai hampir habis, gambar telah diambil oleh kamera, atau pengguna mengubah pengaturan bahasa. Aplikasi juga dapat menginisiasi broadcast misalnya, memberikan informasi

pada aplikasi lain bahwa ada data yang telah diunduh ke perangkat dan tersedia bagi mereka untuk digunakan.

2.3.7.4 Content Providers

Membuat seta plikasi data spesifik, tersedia untuk aplikasi lain. Data dapat disimpan dalam file sistem, dalam sebuah database SQLite, atau cara lain yang bisa diterima. Content provider meng-extends kelas dasar content provider untuk menerapkan standar metod yang memungkinkan aplikasi lain untuk mengambil dan menyimpan data dari jenis yang dikontrolnya. Namun, aplikasi ini tidak memanggil metode langsung.

2.3.7.5 Layout

Secara mendasar untuk membuat layout dan memperlihatkan hirarki view adalah menggunakan file layout XML. XML memberikan struktur yang mudah dipahami seperti halnya dengan HTML. Setiap elemen XML adalah salah satu objek view seperti daun pada sebuah pohon, dan objek view group adalah cabang dari pohon tersebut.

2.3.7.6 Android Manifest

File AndroidManifest.xml diperlukan oleh setiap aplikasi Android. Diletakkan pada folder root aplikasi, dan mendeskripsikan nilai global paket aplikasi kita. Termasuk komponen yang ada pada aplikasi (activity, service, dan lainnya) yang memang diekspos untuk sistem di luar aplikasi yang membutuhkan. Termasuk juga data tiap activity kita, bagaimana menanganinya dan bagaimana mereka diluncurkan (launched).

Hal yang paling penting yang harus diperhatikan dari file AndroidManifest.xml adalah intentFilters. intentFilters menjelaskan dimana dan kapan activity dapat dijalankan. Ketika activity (atau sistem operasi) akan melakukan

sesuatu, misalnya membuka halaman web atau membuka layar kontak, itu akan membuka objek intent.

Objek intent ini akan menangani beberapa informasi yang mendiskripsikan apa yang akan dilakukan oleh aplikasi dan data apa yang dibutuhkannya. Untuk itu, selain juga beberapa informasi lain Android membandingkan informasi pada objek intent dengan intent filter, yang dibuka (exposed) oleh setiap aplikasi dan mencari activity yang tepat untuk menangani data atau aksi (action) yang dibutuhkan oleh activity (aplikasi) yang memanggilnya (caller). Jika terdapat lebih dari satu aplikasi yang menangani intent tersebut, pengguna akan ditanya melalui pesan notifikasi mana yang akan dipilih.

Selain mendeklarasikan activity aplikasi, content providers, service dan intent receivers, kita juga harus menetapkan permissions di dalam AndroidManifest.xml.

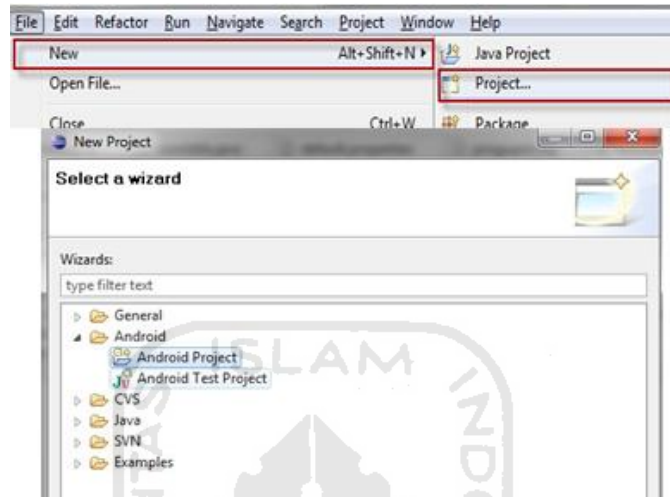
2.3.7.7 Sumber Data/Resource

Sumber daya adalah unsur-unsur eksternal yang digunakan dalam aplikasi. File-file sumber daya ditempatkan dalam folder “res” bisa berupa gambar, musik dan video. Di folder ini disimpan file xml, UI, styles serta string yang kita gunakan dalam aplikasi. Setelah semua file-file tersebut tersedia maka akan membentuk sebuah file dengan nama R.Java, file R.Java merupakan file yang digenerate secara otomatis yang mengindex semua resource dari projek kita.

2.3.8 Eclipse

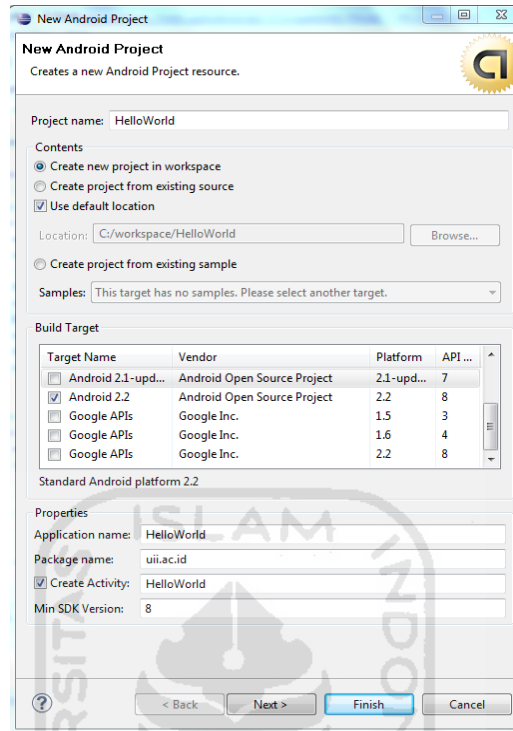
Eclipse merupakan satu dari beberapa application building tools yang berupa IDE (Integrated Development Environment) yang direkomendasikan OHA dan Google untuk mengembangkan aplikasi Android dibandingkan dengan IDE lainnya, ini terlihat dengan dirilisnya Plug-in ADT untuk Eclipse dengan. Setelah selesai mempersiapkan semua lingkungan pengembangan maka akan dilanjutkan dengan membuat project dengan Eclipse[ANO11c].

Membuat project baru dengan memilih Android Project untuk memulai pengembangan aplikasi yang terlihat pada gambar 3.5.



Gambar 2.5 Membuat Project Android baru

1. Kemudian akan dilanjutkan dengan melengkapi isian untuk project wizard tersebut. Lengkapi isian tersebut sesuai dengan gambar 2.6.



Gambar 2. 6 Parameter Project Android

Berikut penjelasan dari setiap field yang kita isi :

a. Project Name

Isi dengan nama dari project kita. Nama project ini akan menjadi nama direktori dari project eclipse.

b. Application Name

Nama ini akan tampil sebagai *title* dari aplikasi kita yang muncul di *device* Android.

c. Package Name

Nama ini digunakan sebagai nama package untuk main activity kita. Konvensi yang digunakan sama dengan standar penamaan package di Java. Nama package haruslah unique dari semua nama package yang ada di sistem Android. Hal ini disebabkan nama package tersebut digunakan sebagai unique id untuk aplikasi. Jadi gunakan standar

penamaan yang sesuai dengan konvensi yang berlaku untuk package di Java. Penamaan package yang standar adalah dengan menggunakan nama yang sesuai dengan organisasi atau company. Misalnya kita memiliki public domain internet uii.ac.id. Maka kita dapat memberikan nama package dengan membalikkan nama domain tersebut sebagai berikut. uiii.ac.id untuk aplikasi helloworld.

d. Create Activity

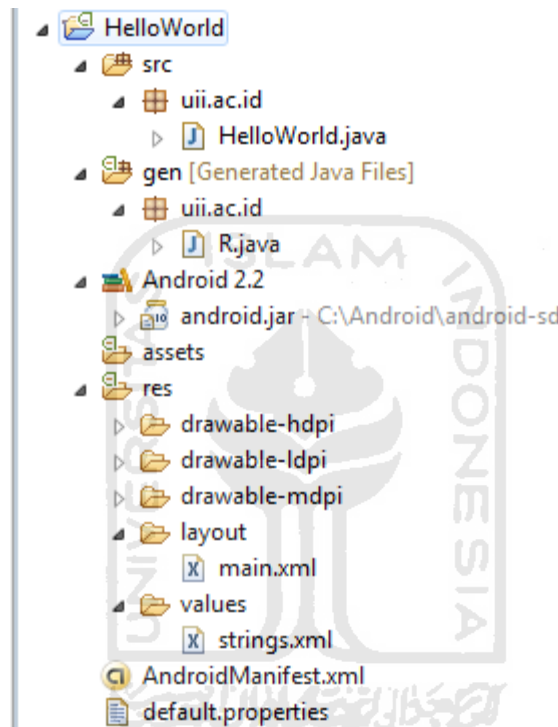
Nama ini akan diberikan untuk kelas yang akan digenerate oleh eclipse sebagai stub. Kelas ini merupakan subclass dari activity. Activity merupakan kelas yang dapat dijalankan dan melakukan task. Sama seperti main program pada program Java biasa. Activity juga dapat membuat dan menampilkan UI jika diinginkan. Kebanyakan activity digunakan untuk menampilkan grafis atau user interface sehingga orang kebanyakan menganggapnya sebagai form. Activity yang digenerate akan menjadi main activity sebagai entry awal aplikasi ini.

e. Min SDK Version

Nilai berikut akan menentukan API level yang dibutuhkan oleh aplikasi kita. Sehingga aplikasi ini tidak dapat dijalankan pada level yang lebih rendah. Ini juga digunakan untuk membatasi device platform yang digunakan sebagai target dari aplikasi kita.

2. Pada Gambar 2.7 akan dijelaskan beberapa komponen penting yang digunakan untuk aplikasi Android HelloWorld ini. Beberapa komponen tersebut adalah source code Java yaitu HelloWorld.java, XML UI layout main.xml, dan manifest file yaitu AndroidManifest.xml. Kita akan melihat sekilas isi dari HelloWorld.java Kelas HelloWorld ini adalah entry point atau program utama yang akan di load oleh Android. Kita mengamati bahwa HelloWorld merupakan subclass dari activity. Activity adalah bagian dari aplikasi yang digunakan untuk melakukan aksi. Satu aplikasi terdiri dari

beberapa activity yang saling terkait. Pada project berikut ini kita memiliki satu activity yaitu HelloWorld yang merupakan activity utama. Kita mengatakan activity utama karena activity ini akan di load pertama kali pada saat aplikasi ini dibuka.



Gambar 2. 7 Struktur *project* aplikasi Android HelloWorld

Content view dari layout yang akan ditampilkan tersebut di definisikan di main.xml. R.layout.main merupakan id yang digunakan untuk mengenali resource xml yang ada di res (resource) folder. Sesuai dengan nama identifiernya kita dapat menentukan lokasi dari file xml yang dimaksud. Kita dapat menemukannya di folder res/layout/main.xml. AndroidManifest.xml digunakan oleh Android untuk menentukan perlakuan terhadap aplikasi yang kita buat. Setiap komponen yang kita buat agar dapat dikenali oleh Android harus terlebih dahulu di daftarkan ke dalam manifest.

BAB III

METODOLOGI

3.1 Metode Analisis

Metode Analisis digunakan untuk menguraikan sistem aplikasi *healthy reminder* pada sistem operasi Android menjadi komponen-komponen untuk diidentifikasi dan dievaluasi permasalahannya. Sistem yang dianalisis adalah sistem yang berisi informasi tentang segala sesuatu yang berkaitan dengan pengekplorasian desain *Graphic User Interface* (GUI), dalam bentuk multimedia yang berupa *coding*, pengingat waktu (*alarm*), *vibrate*, dan suara yang diaplikasikan pada sistem operasi Android menggunakan interaksi *touchscreen*.

Sistem yang dianalisis adalah cara kerja aplikasi *healthy reminder* yang berjalan pada sistem operasi Android berbasis *touchsreen*. Tahap analisis ini merupakan tahapan yang paling penting dalam program yang dirancang, karena jika terjadi kesalahan dalam tahap ini akan menyebabkan terjadinya kesalahan pada tahap selanjutnya. Karena itu dibutuhkan suatu metode sebagai pedoman dalam mengembangkan sistem yang dibangun.

3.2 Analisis Kebutuhan

Dari metode dan langkah yang dilakukan maka hasil analisis yang diperoleh adalah kebutuhan *input*, fungsi dan kinerja, kebutuhan *output*, kebutuhan perangkat keras, dan kebutuhan perangkat lunak

3.2.1 Analisis Kebutuhan Input

Input adalah suatu bentuk masukan dan berupa data yang telah ada yang dibutuhkan oleh perangkat lunak sehingga dapat mencapai tujuan yang diinginkan. Masukan dari aplikasi *reminder* ini adalah masukan yang dihasilkan dari interaksi pengguna pada layar *touchscreen smartphone*.

3.2.2 Analisis Kebutuhan Fungsi dan Kinerja

Fungsi dan kinerja yang dibutuhkan pada aplikasi ini adalah sebagai berikut:

1. Membaca *input* dari *touchscreen*.
2. Pilihan menu telah tersedia.
3. Interaksi dengan objek lain, seperti menambah dan menghapus pilihan.
4. Pengaturan waktu yang dapat disesuaikan.
5. Data dikelola oleh database.

3.2.3 Analisis Kebutuhan Output

Keluaran (*output*) dari aplikasi ini adalah berupa aplikasi yang berjalan pada *device* Android menggunakan layar *touchscreen*. Objek pada aplikasi ini berupa teks, *vibrate* dan pengingat waktu disertai suara.

3.2.4 Analisis kebutuhan Perangkat Keras

Perangkat keras digunakan sebagai alat pengolah data yang bekerja secara otomatis mengolah data yang berbentuk teks, gambar, audio dan animasi. Diperlukan komputer yang memiliki kemampuan mengolah grafis yang baik. Spesifikasi komponen yang dibutuhkan akan dibagi menjadi dua bagian yaitu dari sisi pembangunan aplikasi dan dari sisi pengguna.

Adapun spesifikasi komponen perangkat keras yang diperlukan untuk pembangunan aplikasi adalah sebagai berikut:

1. Piranti input berupa *keyboard* dan *mouse*
2. Piranti output berupa monitor dengan resolusi minimal 1024x768, dan *speaker*
3. Kartu grafis yang mendukung *Direct 9*, dengan memori minimal 512MB
4. Prosesor minimal dual core dan memiliki kecepatan 2 Ghz
5. Memori RAM minimal 2GB

6. Hardisk yang memiliki ruang kosong minimal 8GB

Spesifikasi komponen perangkat keras yang diperlukan oleh pengguna yang mengakses aplikasi *healthy reminder* ini adalah sebagai berikut:

1. Piranti input berupa *touchscreen*
2. Piranti output berupa *screen* dengan resolusi minimal 240x400, dan *speaker*
3. Google Android 2.2 (Froyo) *operating system*
4. ARMv7-A *processor* 600mhz
5. Memori RAM minimal 256MB
6. SD Card yang memiliki ruang kosong minimal 20MB
7. H.264 and AAC hardware decoders

3.2.5 Analisis Kebutuhan Perangkat Lunak

Selain perangkat keras, perangkat lunak juga diperlukan dalam pembangunan aplikasi dan dalam menjalankan aplikasi. Perangkat lunak yang dibutuhkan akan dibagi menjadi dua sisi, pada sisi pembangunan aplikasi dan dari sisi pengguna aplikasi.

Perangkat lunak yang dibutuhkan pada pembangunan aplikasi adalah sebagai berikut:

1. *Sistem operasi*, sistem operasi yang dibutuhkan untuk pembangunan aplikasi adalah antara lain Windows XP, 7 (seven), Mac OSX tiger atau Leopard
2. *Android SDK dan AVD Manager*, Android SDK sebagai alat bantu dan API diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java.
3. JDK 5, digunakan untuk menjalankan Android SDK dan AVD manager beserta *emulator*-nya dan digunakan untuk menagtur pembuatan emulator Android dan mengenali koneksi USB dari PC / laptop ke *device* Android.

4. Eclipse dan ADT, merupakan salah satu *application building tools* yang berupa IDE (*Integrated Development Environment*) yang digunakan membangun keseluruhan aplikasi mulai dari menulis file XML pada Android *manifest* dan file *resource*, hingga *coding*.
5. Drooiddraw merupakan aplikasi tambahan yang digunakan untuk membantu merancang tampilan menggunakan XML.

Sedangkan perangkat lunak yang diperlukan oleh pengguna aplikasi adalah Google Android 2.2 (Froyo) *operating system*.

3.3 Perancangan Perangkat Lunak

3.3.1 Metode Perancangan

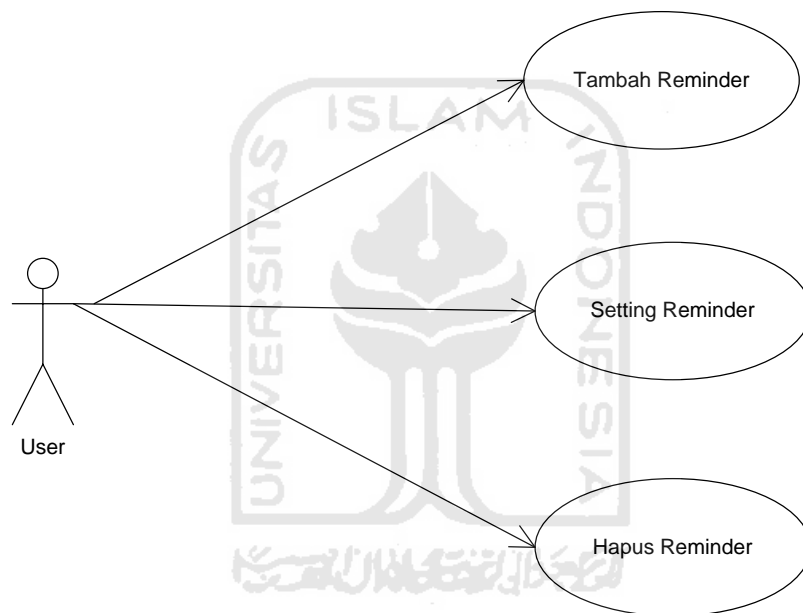
Dalam perancangan sistem ini, sistem yang dibangun digambarkan dalam salah satu bentuk diagram UML yaitu menggunakan *use case* diagram dan *activity* diagram.

3.3.2 Hasil Perancangan

Hasil pada tahap perancangan berkaitan erat dengan hasil tahap analisis. Karena pada tahap analisis telah ditemukan metode, perangkat lunak yang digunakan, serta fungsi-fungsi yang terdapat pada aplikasi ini. Hasil perancangan aplikasi akan dituangkan dalam *use case* diagram dan *activity* diagram yang bertujuan untuk menggambarkan berbagai aliran aktivitas dalam aplikasi yang sedang dirancang. Implementasi dari pengembangan aplikasi berbasis Android ini adalah berfokus pada pembangunan aplikasi *healthy reminder*.

3.3.3 Use Case Diagram

Diagram ini bertujuan untuk menggambarkan fungsi yang dibutuhkan oleh sebuah sistem. Pada bagian ini, diagram akan lebih digunakan untuk merepresentasikan interaksi-interaksi antara pengguna dan sistem. Fungsi fungsi yang digambar pada diagram ini menjelaskan pembagian hak akses terhadap sistem. Rancangan diagram ini dapat dilihat pada Gambar 3.1.



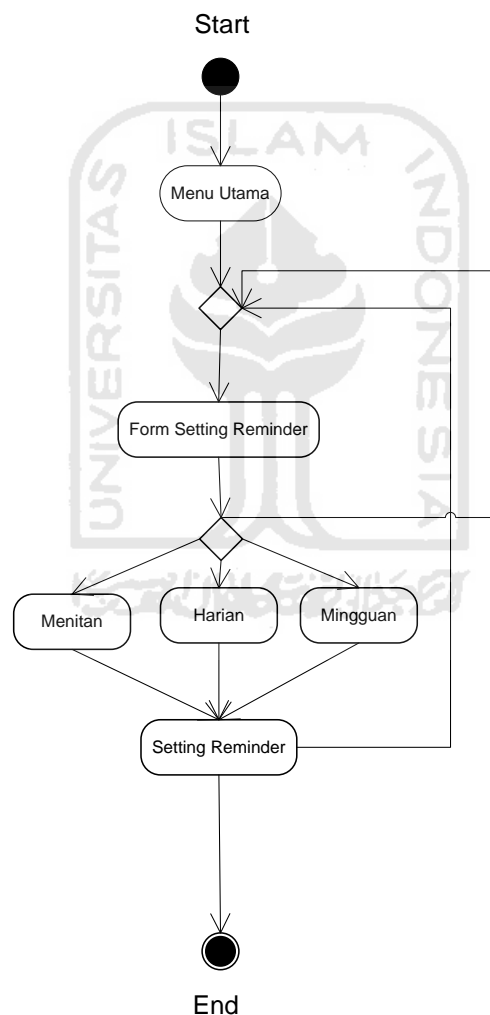
Gambar 3. 1 Use Case Diagram

3.3.4 Activity Diagram

Pada umumnya *activity diagram* tidak menampilkan secara detail urutan proses, namun hanya memberikan gambaran global bagaimana urutan prosesnya. Sehingga seringkali diagram ini digunakan untuk memodelkan aktivitas bisnis dalam level konseptual. Diagram ini sangat mirip dengan *flowchart* karena kita dapat

memodelkan suatu alur kerja dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke dalam *state*. Akan tetapi perbedaannya dengan dengan *flowchart* adalah *activity diagram* dapat mendukung perilaku paralel sedangkan *flowchart* tidak.

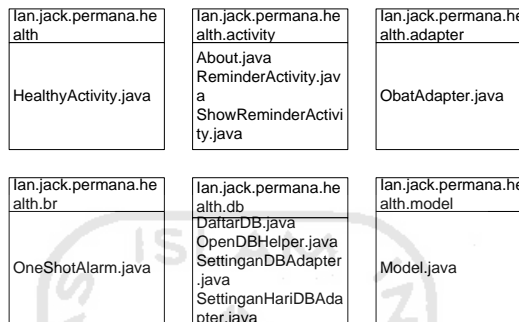
Tujuan pembuatan *activity diagram* ini adalah untuk menggambarkan semua aktivitas global yang terjadi dalam sebuah sistem. Dengan melihat *activity diagram*, pengguna dapat mengetahui apa saja yang dapat dilakukan oleh sebuah sistem yang akan dijelaskan pada Gambar 3.2.



Gambar 3. 2 Activity Diagram

3.3.5 Diagram Class

Diagram ini bertujuan untuk menggambarkan perancangan dari *package-package* yang akan digunakan dalam pengembangan sistem pada aplikasi. Diagram ini akan digunakan untuk merepresentasikan interaksi *class* terhadap sistem.



Gambar 3. 3 Perancangan Diagram Class

Adapun keterangan dari *package-package* yang digunakan akan dijelaskan pada tabel 2. 1.

Tabel 2. 1 Tabel Class

NAMA CLASS	KETERANGAN
HealthyActivity	<i>Class</i> ini berfungsi untuk menampilkan halaman menu utama
About	<i>Class</i> ini berisi sekilas cerita singkat dari pengembangan aplikasi
ReminderActivity	<i>Class</i> ini berfungsi untuk mengatur settingan waktu
ShowReminderActivity	<i>Class</i> ini berfungsi untuk memunculkan alarm ke halaman <i>device</i>
ObatAdapter	<i>Class</i> ini berfungsi untuk menampilkan daftar pada menu set
OneShotAlarm	<i>Class</i> ini berfungsi sebagai akses untuk mengaktifasi <i>alarm manager</i>
DaftarDBAdapter	<i>Class</i> ini berisi data penyimpanan daftar nama <i>reminder</i>

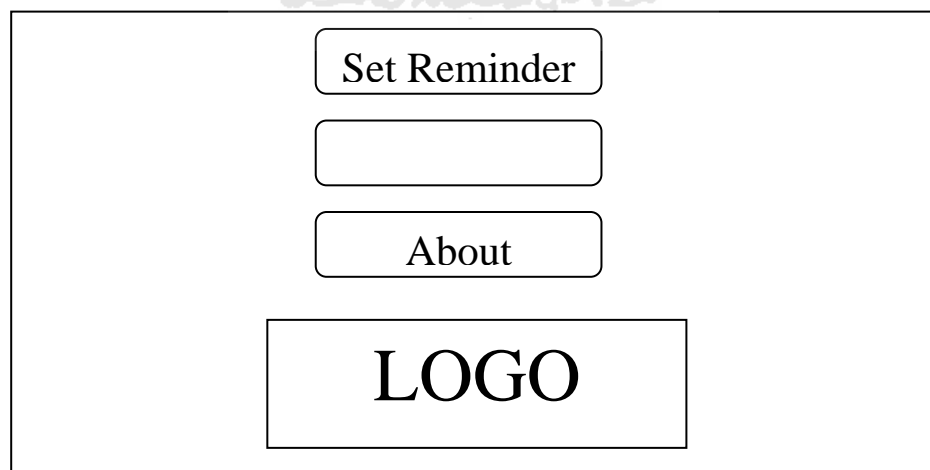
OpenDBHelper	Class ini merupakan <i>default class</i> untuk mengaktifasi data penyimpanan
SettingDBAdapter	Class ini berfungsi untuk menyimpan data setting waktu
SettinghariDBAdapter	Class ini berfungsi untuk menyimpan data setting harian
Model	Class ini bertujuan untuk memberikan model pada tampilan

3.3.6 Perancangan Antarmuka

Perancangan antarmuka bertujuan untuk memberikan kemudahan dalam mengimplementasikan perangkat lunak yang akan dibangun. Antarmuka ini juga berfungsi sebagai sarana interaksi antara manusia dan komputer. Antarmuka pada aplikasi *healthy reminder* ini merupakan gambaran antarmuka yang tampak pada halaman aplikasi.

1. Antarmuka Menu Utama :

Tampilan ini merupakan menu awal dari aplikasi yang akan dikembangkan. Menu utama terdiri dari tombol untuk mengakses aplikasi utama, about dan gambar yang akan dijelaskan pada gambar 3.2.



Gambar 3. 4 Perancangan Antarmuka Menu Utama

2. Antarmuka Set Reminder :

Menu ini merupakan menu dari fitur utama aplikasi yang akan dikembangkan. Menu ini terdiri dari nama-nama *reminder* yang akan dipilih yang akan dijelaskan pada gambar 3.3.

Nama Reminder 1
Nama Reminder 2
Nama Reminder 3
Nama Reminder 4
Nama Reminder 5
Nama Reminder ...

Gambar 3. 5 Perancangan antarmuka Set Reminder

3. Antarmuka Menu Opsi Set Reminder:

Tampilan ini merupakan menu opsi dari set *reminder* yang dapat menambah, menghapus dan mengatur *reminder* itu sendiri yang dapat dilihat pada gambar 3.3.

Tambah	Hapus	Set

Gambar 3. 6 Perancangan antarmuka Menu Opsi Set Reminder

4. Antarmuka Opsi Tambah :

Tampilan ini merupakan menu tambah yang akan menambahkan *healthy reminder* sesuai dengan keinginan, yang terdiri dari penambahan nama *reminder* dan tempat dapat dilihat pada gambar 3.4.

Nama Reminder	<input type="text"/>
Tempat	<input type="text"/>

Gambar 3. 7 Perancangan antarmuka Tambah

5. Antarmuka Set *Reminder* :

Tampilan ini akan muncul setelah menekan menu set, terdiri dari pilihan yang telah disediakan yang dijelaskan pada gambar 3.5.

<input type="radio"/>	Menitan
<input type="radio"/>	Harian
<input type="radio"/>	Mingguan
OK	Gagal

Gambar 3. 8 Perancangan antarmuka Set *Reminder*

6. Antarmuka Opsi Menitan :

Tampilan ini merupakan menu dari pilihan opsi menitan. Dapat dilihat pada gambar 3.6.

Gambar 3. 9 Perancangan antarmuka opsi menit

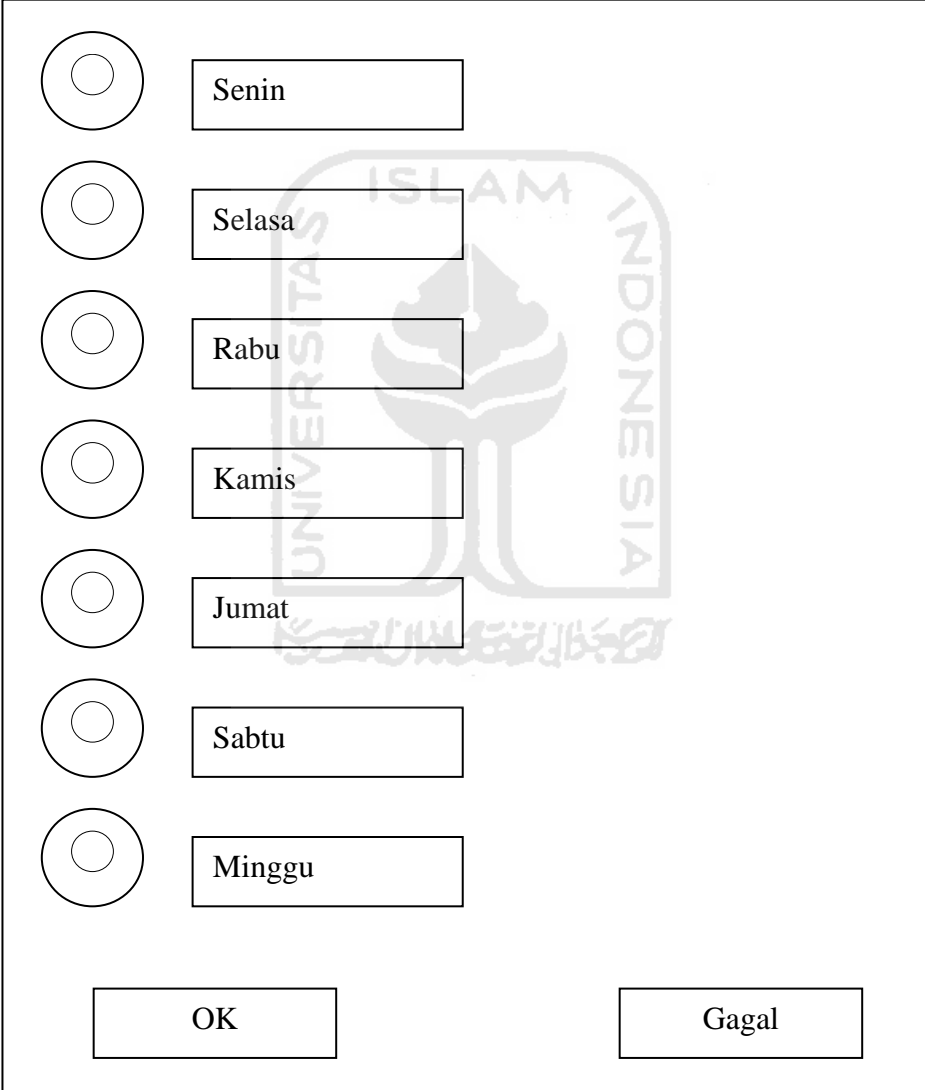
7. Antarmuka Opsi Harian :

Tampilan ini merupakan menu dari pilihan opsi harian. Dapat dilihat pada gambar 3.7.

Gambar 3. 10 Perancangan antarmuka Opsi Harian

8. Antarmuka Opsi Mingguan :

Tampilan ini merupakan menu dari pilihan opsi mingguan. Dapat dilihat pada gambar 3.8.



The image shows a user interface for selecting a weekly option. It consists of seven radio buttons arranged vertically, each followed by a text box containing a day of the week: Senin, Selasa, Rabu, Kamis, Jumat, Sabtu, and Minggu. At the bottom of the interface, there are two buttons: 'OK' on the left and 'Gagal' on the right. A faint watermark of the logo for Universitas Islam Indonesia is visible in the background of the interface.

Gambar 3. 11 Perancangan antarmuka Opsi Mingguan

BAB IV

HASIL DAN PEMBAHASAN

Hasil dan pembahasan mengutarakan tentang implementasi perangkat lunak yang meliputi batasan implementasi dan implementasi perangkat lunak berupa *layout main menu*, implementasi antarmuka, pengujian program dan pengujian menu, serta analisis kinerja aplikasi.

4.1 Batasan Implementasi

Aplikasi *healthy reminder* ini dirancang menggunakan Eclipse 0.9.5. Aplikasi *healthy reminder* ini memiliki batasan-batasan implementasi yaitu :

1. *Device*

Aplikasi ini hanya dapat berjalan pada *device* Android yang memiliki interaksi *touchscreen*.

2. *Screen Resolution*

Aplikasi ini hanya dapat berjalan pada resolusi layar minimal 240x400 (WQVGA400)..

3. *Operasting System (OS)*

Aplikasi ini berjalan pada OS Android Froyo.

4.2 Tahap Pembuatan Proses

1. Analisis data

Mengumpulkan berbagai literatur tentang Eclipse dan Android SDK pada Android dan apa saja yang diperlukan untuk mengembangkan aplikasi tersebut.

2. Pembuatan konsep

Mengkonsep alur dan tema pada aplikasi *healthy reminder* yang akan dibuat, konsep tema sangat penting pada aplikasi *healthy reminder* agar pembuatan desain antara file XML dan Java dapat terhubung dengan baik.

3. Desain

Merancang dan membuat *activity diagram* sebagai media perancangan aplikasi.

4. Pemodelan

Membuat desain XML menggunakan Droiddraw untuk digunakan sebagai visualisasi aplikasi dan tampilan GUI (*Graphical User Interface*) dari file Java dengan software Eclipse, yang dibuat menjadi file *alarm*.

5. Pengkodean dan pembuatan aplikasi *healthy reminder*

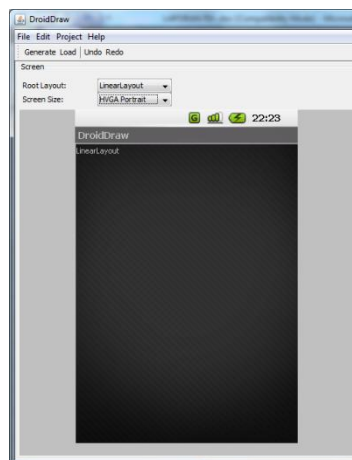
Dengan Eclipse, aplikasi *healthy reminder* akan dibangun menggunakan Java dan XML untuk membangun aplikasi Android yang berekstensi *.apk*.

6. Pengujian

Mengadakan pengujian dan menganalisis aplikasi secara sederhana.

4.3 Implementasi Pembuatan Proses

Pertama-tama yang dilakukan adalah menyiapkan *layout* pada Eclipse menggunakan Droiddraw. Layout tersebut akan dijadikan dasar aplikasi *healthy reminder*. Kemudian melakukan *setting layout* untuk disesuaikan dgn resolusi layar *device* Android, resolusi yang tepat yaitu HVGA untuk orientasi *potrait*, dapat dilihat pada Gambar 4.1



Gambar 4. 1 Tahap Pembuatan Layout Dasar Pada Droiddraw

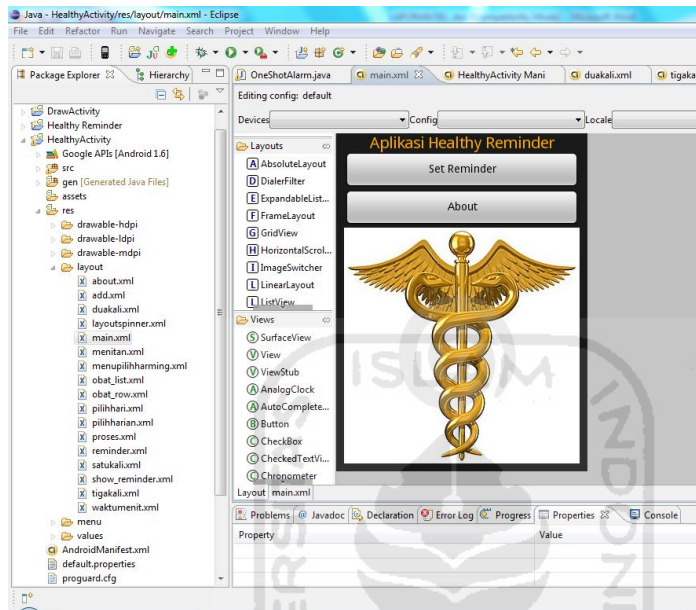
Berikutnya adalah pembuatan *layout* untuk digunakan sebagai visualisasi aplikasi, seperti logo, teks, tombol, dan warna. Pembuatan dilakukan pada Eclipse XML agar dapat dimunculkan sebagai tampilan, dengan contoh kode sebagai berikut

:

```
<?xml version="1.0" encoding="utf-8"?>
<!-- File XML untuk membentuk layout atau tampilan pada aplikasi
Android-->
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="wrap_content"
    android:layout_height="fill_parent"
    android:layout_gravity="center">
    <!--TextView merupakan salah satu dari bermacam-macam bentuk
widget yang
    disediakan SDK Android berupa field text yang dapat diisi oleh
user-->
<TextView android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Aplikasi Healthy Reminder"
    android:textColor="@color/cOrange"
    android:gravity="center"
    android:layout_gravity="center_vertical"
    android:id="@+id/btnViewApotek"
    android:textSize="20sp">
</TextView>
    <!--Button merupakan widget yang berfungsi untuk memberikan
efek tombol yang
    dapat memberikan akses menuju activity lain-->
<Button android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Set Reminder"
    android:id="@+id/btnSetReminder">
</Button>
<Button android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="About"
    android:id="@+id/btnViewAbout">
</Button>
    <!--ImageView adalah widget yang dapat menampilkan gambar yang
dipanggil dari folder drawable-->
<ImageView
    android:id="@+id/imageView1"
    android:layout_height="wrap_content"
    android:src="@drawable/medical"
    android:layout_width="fill_parent">
</ImageView>
```


</LinearLayout>

Adapun tahap pembuatan *layout* dapat dilihat pada Gambar 4.2.



Gambar 4. 2 Tahap Pembuatan *Layout* Pada *Aplikasi*

Selanjutnya, penataan dan penggabungan *layout* dengan file Java agar lebih mudah mengecek dan memilih GUI yang akan digunakan, dan tempat terjadinya aktivitas pada aplikasi, seperti : menu utama, menu opsi, tambah, set *reminder*, *about* dan lain-lain. Dengan contoh kode sebagai berikut :

```
package ian.jackpermana.health;

import ian.jackpermana.health.activity.About;
import ian.jackpermana.health.activity.ReminderActivity;
import ian.jackpermana.health.R;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class HealthyActivity extends Activity implements
OnClickListener {
    /** Called when the activity is first created. */
```

```

//Fungsi yang dipanggil pertama kali ketika memanggil objek activity
@Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        Button btnSetReminder = (Button)
findViewById(R.id.btnSetReminder);
        btnSetReminder.setOnClickListener(this);
        Button btnFindApotek = (Button)
findViewById(R.id.btnFindApotek);
        btnFindApotek.setOnClickListener(this);
        Button btnViewAbout = (Button)
findViewById(R.id.btnViewAbout);
        btnViewAbout.setOnClickListener(this);
    }
//Fungsi untuk memanggil event (disentuh di layar (klik))
@Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.btnSetReminder:
                startActivity(new Intent(this,
ReminderActivity.class));
                break;
            case R.id.btnViewAbout:
                startActivity(new Intent(this, About.class));
                break;
            default:
                break;
        }
    }
}

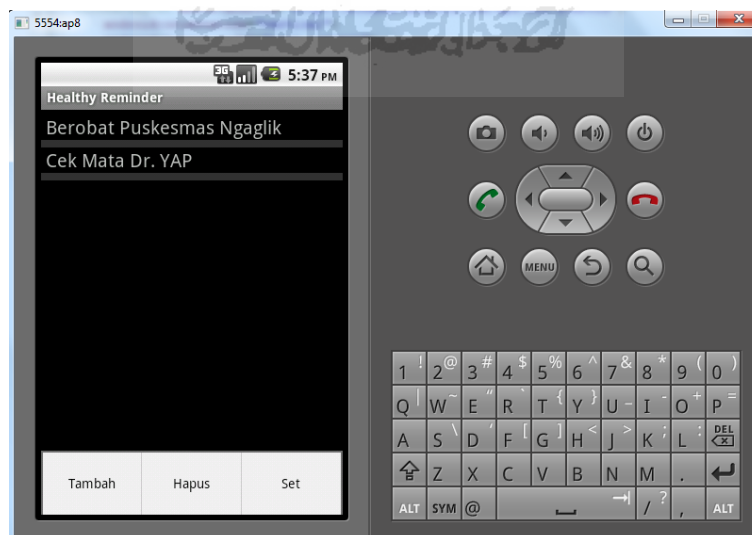
```

Penataan *frame* ini bertujuan agar saat setiap *event* dalam suatu *frame* selesai dieksekusi maka sebagaimana dapat dengan mudah berpindah ke *frame* dengan *event* yang lain, dilihat pada Gambar 4.3



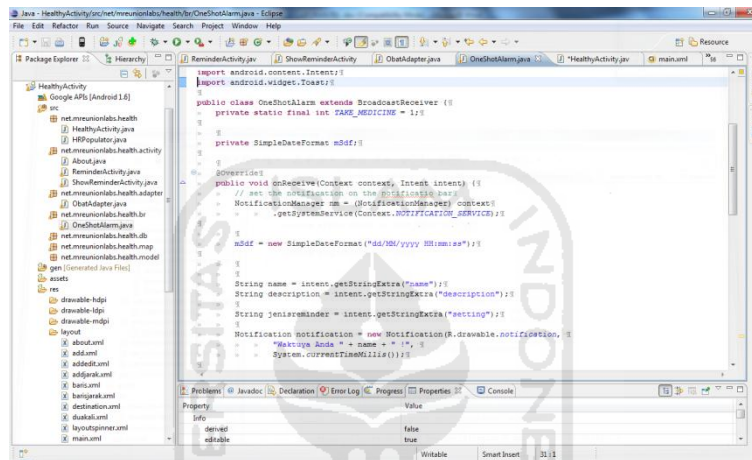
Gambar 4. 3 Tahap Pembuatan Menu Utama

Kemudian tahap berikutnya adalah pengecekan *Emulator Engine* pada Eclipse. Pengecekan *Emulator Engine* akan menggunakan AVD yang mewakili *device* agar lebih mudah dan cepat dalam pengujian *trial and error*-nya. Tahap pengecekan dapat dilihat pada gambar 4.4 :



Gambar 4. 4 Tahap Pengecekan *Emulator Engine*

Tahap selanjutnya adalah membuat *alarm reminder*, ini adalah inti dari aplikasi *reminder* yang akan dibuat, disini akan dibuat keterangan waktu, nama *reminder*, tempat *reminder*. Disini juga akan dibuat percabangan *activity* antara menu opsi menitan hingga menu opsi mingguan Adapun tahap pembuatannya dapat dilihat pada gambar 4.5.



Gambar 4.5 Tahap Pembuatan *alarm reminder*

Adapun contoh kodenya adalah sebagai berikut :

```

private SimpleDateFormat mSdf;

//Service-service yang ada pada Android untuk memanggil
fungsi-fungsi yang sifatnya built-in pada Android
@Override
public void onReceive(Context context, Intent intent) {
    // memunculkan notifikasi ke bar pada layar Android
    NotificationManager nm = (NotificationManager) context
        .getSystemService(Context.NOTIFICATION_SERVICE);

    mSdf = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");

    String name = intent.getStringExtra("name");
    String description =
intent.getStringExtra("description");

```

```

String jenisreminder = intent.getStringExtra("setting");

Notification notification = new
Notification(R.drawable.notification,
            "Waktuya Anda " + name + " !",
            System.currentTimeMillis());

```

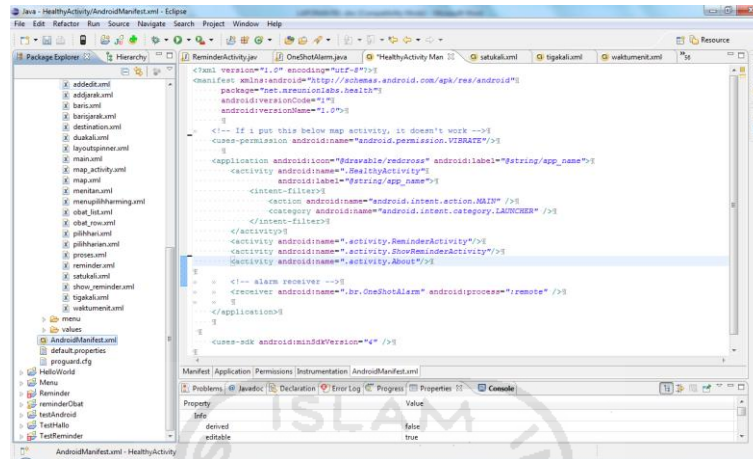
Tahap berikutnya adalah pembuatan *manifest* sebagai *permission* untuk mengaktifkan semua *activity* agar dapat dijalankan, karena tanpa adanya *permission* maka kelas yang dibuat akan menjadi tidak berfungsi. Pembuatan file ini ada pada `AndroidManifest.xml` dengan contoh kode sebagai berikut :

```

<?xml version="1.0" encoding="utf-8"?>
<!--File Manifest merupakan halaman XML yang memebrikan akses pada
setiap activity-->
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.mreunionlabs.health"
    android:versionCode="1"
    android:versionName="1.0">
    <!--uses-permission menjelaskan tentang security permission,
yang harus diberikan agar activity dapat beroperasi secara benar-->
    <uses-permission android:name="android.permission.VIBRATE"/>
    <!-- Elemen root yang berisi deklarasi dari level aplikasi
komponen yang terdapat pada paket aplikasi -->
    <application android:icon="@drawable/redcross"
android:label="@string/app_name">
        <!--hal mendasar yang pada sebuah aplikasi untuk berinteraksi
dengan user-->
        <activity android:name=".HealthyActivity"
            android:label="@string/app_name">
            <!--mendeklarasikan inten seperti apa yang didukung oleh
komponen-->
            <intent-filter>
                <!--tipe aksi-->
                <action android:name="android.intent.action.MAIN" />
                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".activity.ReminderActivity"/>
        <activity android:name=".activity.ShowReminderActivity"/>
        <activity android:name=".activity.About"/>
        <!-- alarm receiver -->
        <receiver android:name=".br.OneShotAlarm"
android:process=":remote" />
    </application>
    <uses-sdk android:minSdkVersion="4" />
</manifest>

```

Hasil *layout* dapat dilihat pada gambar 4.6 :



Gambar 4. 6 Tahap Pembuatan AndroidManifest.xml

4.4 Implementasi Interface

Implementasi *Interface* aplikasi *healthy reminder* ini adalah *Main Menu*, beserta *Interface* menu opsi dan sistem inti menggunakan *device* SAMSUNG GALAXY S P9000, dengan penjelasan sebagai berikut.

4.4.1 Implementasi Main Menu Aplikasi Reminder

Implementasi *main menu* aplikasi *healthy reminder* merupakan tampilan utama saat membuka aplikasi *healthy reminder* ini. Pada tampilan ini, *user* disuguhkan beberapa tombol menu, yaitu :

1. Menu Set Reminder : Berfungsi untuk mulai aplikasi *healthy reminder*.
2. Menu About : Berisi sekilas tentang aplikasi *healthy reminder*.

Implementasi *main menu*, dapat dilihat pada gambar 4.7.



Gambar 4. 7 Antarmuka aplikasi *healthy reminder*

4.4.2 Implementasi Interface Menu Opsi dan Sistem Inti

Implementasi *interface* pada aplikasi *healthy reminder* merupakan desain antarmuka yang mempunyai beberapa *interface* utama, yaitu :

1. Tombol Tambah : Berfungsi untuk menambahkan *reminder* sesuai keinginan *user*.
2. Tombol Hapus : Berfungsi untuk menghapus *reminder* yang telah disediakan..
3. Tombol Set : Berfungsi untuk mengatur *reminder* sesuai dengan pilihan yang tersedia.

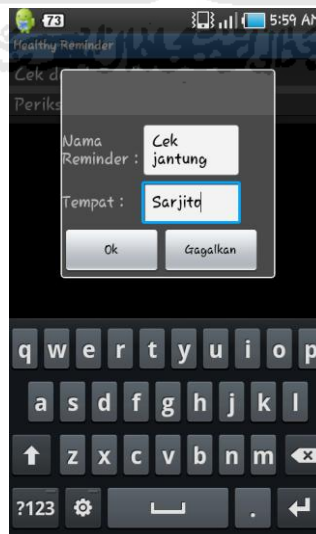
Implementasi menu opsi dan sistem inti dalam aplikasi *healthy reminder*, dapat dilihat pada Gambar 4.8 berikut.



Gambar 4. 8 Interface Menu Opsi dan Sistem Inti

4.4.3 Implementasi Tombol Tambah

Implementasi tombol tambah ini merupakan tombol untuk menambahkan *reminder* sesuai dengan keinginan dan kebutuhan *user* berupa nama dan tempat. Dapat dilihat pada gambar 4.9.



Gambar 4. 9 Interface Tombol Tambah

4.4.4 Implementasi Tombol Set

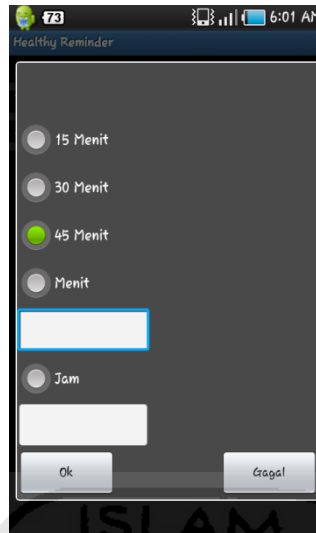
Implementasi tombol set adalah untuk mengatur waktu dari *reminder* yang telah dipilih oleh *user*. Tombol set terdiri dari 3 opsi yaitu opsi menitan, harian, dan mingguan seperti yang terlihat pada gambar 4.10.



Gambar 4. 10 Interface Tombol Set

4.4.5 Implementasi Opsi Menitan

Implementasi dari opsi menitan ini digunakan untuk keperluan *reminder* dalam jangka waktu menit yaitu 15 menit, 30 menit, 45 menit, atau mengisi sendiri jumlah menit atau jam sesuai dengan kebutuhan *user*. Seperti yang terlihat pada gambar 4.9.



Gambar 4. 11 Interface Opsi Menitan

Kemudian setelah menekan tombol ok akan diberikan pilihan waktu untuk memulai *reminder* yang telah dipilih. Pemilihan waktu ini terdiri dari jam dan menit serta pemilihan untuk pagi hari atau malam hari. Dapat dilihat pada gambar 4.12



Gambar 4. 12 Interface Pengaturan Waktu Menitan

4.4.6 Implementasi Opsi Harian

Implementasi dari opsi harian ini digunakan untuk keperluan *reminder* setiap hari. *Reminder* ini akan mengulang setiap harinya setelah *user* menetapkan waktunya. Opsi ini terdiri dari 2 pilihan yaitu 2X sehari dan 3X sehari yang terlihat pada gambar 4. 13.



Gambar 4. 13 Interface Opsi Harian

Kemudian *user* akan mengatur waktu dari opsi-opsi yang ditunjukkan pada gambar 4. 13, yaitu :

1. Pilihan 2X sehari,

Pilihan ini akan memberikan 2 pengaturan waktu *reminder* kepada *user* yang akan dijalankan pada hari saat pengaturan *reminder* dipilih. Contoh akan dijelaskan pada gambar 4. 14.



Gambar 4. 14 Interface Pengaturan Waktu 2X Sehari

2. Pilihan 3X sehari,

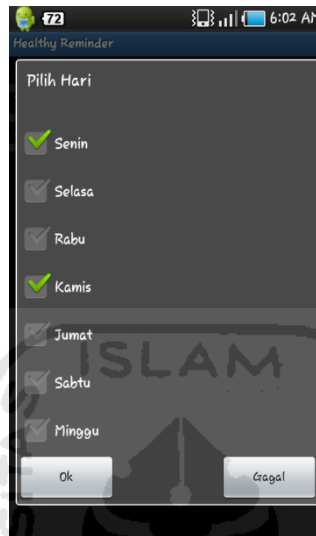
Sama halnya pilihan 2X sehari, pilihan ini akan memberikan 3 pengaturan waktu *reminder* kepada *user* yang akan dijalankan pada hari yang sama setelah pengaturan ditetapkan. Dapat dilihat pada gambar 4.15.



Gambar 4. 15 Interface Pengaturan Waktu 3X Sehari

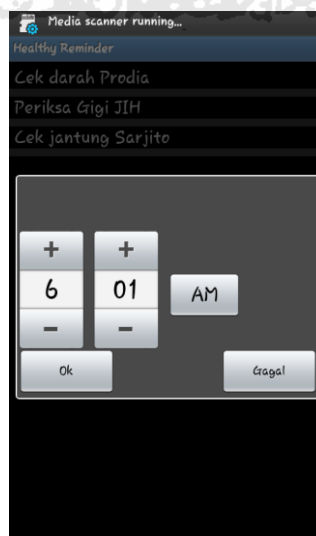
4.4.7 Implementasi Opsi Mingguan

Opsi ini akan memberikan beberapa pilihan hari, *user* hanya tinggal memilih salah satu dari hari yang sudah disediakan, mulai dari hari senin hingga minggu atau lebih dari satu sesuai dengan kebutuhan *user*. Dapat dilihat pada gambar 4.16.



Gambar 4. 16 Interface Opsi Mingguan

Setelah memilih hari yang diinginkan, kemudian *user* akan mengatur waktu (jam, menit, siang atau malam) untuk keperluan *reminder* yang akan ditampilkan. Seperti yang terlihat pada gambar 4.17.



Gambar 4. 17 Interface Pengaturan Waktu Mingguan

4.5 Implementasi Penggunaan Package

4.5.1 Package Aplikasi

Dalam pengembangan aplikasi *reminder* menggunakan 6 package untuk mempermudah dalam mengelola *class*. Berikut ini merupakan deskripsi dari *package* yang dikembangkan pada aplikasi *reminder*.

Tabel 4. 1 Tabel Implementasi Package Aplikasi

Nama Fungsi	Nama Class	Keterangan
OnCreate	<ul style="list-style-type: none"> • HealthyActivity • About • ShowReminder • ReminderActivity 	Fungsi ini digunakan pertama kali untuk memanggil objek activity
OnClick	HealthyActivity	Fungsi ini digunakan untuk memanggil event (click)
OnCreateOptionsMenu	ReminderActivity	Fungsi ini digunakan untuk memunculkan menu pilihan ketika click tombol menu
tampil	ReminderActivity	Fungsi ini untuk menampilkan layout daftar
pilihhari	ReminderActivity	Fungsi ini digunakan untuk memberikan pilihan hari pada menu mingguan
pilihsetting	ReminderActivity	Fungsi ini digunakan untuk menampilkan pilihan pada set
tambah	ReminderActivity	Fungsi untuk menambahkan nama pada daftar
setContentview	ReminderActivity	Fungsi untuk memanggil XML layout
findViewById	ReminderActivity	Fungsi untuk memanggil ID pada folder <i>resource</i>

delete	ReminderActivity	Fungsi untuk menghapus daftar yang sudah ada
TimePickerDialog DatePickerDialog	ReminderActivity	Fungsi untuk memanggil fungsi <i>time</i> pada device
setAlarm	ReminderActivity	Fungsi untuk membuat data berupa <i>alarm</i>
onReceive	OneShotAlarm	Fungsi untuk service <i>alarm</i> yang built ini pada Android
notification	ReminderActivity	Fungsi untuk memunculkan notifikasi pada <i>device</i>

4.5.2 Implementasi Code Fungsi

Adapun penjabaran kode dari fungsi-fungsi dari *class* yang digunakan pada pengembangan aplikasi adalah sebagai berikut:

1. onCreate :

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    Button btnSetReminder = (Button)
    findViewById(R.id.btnSetReminder);
    btnSetReminder.setOnClickListener(this);
    Button btnViewAbout = (Button)
    findViewById(R.id.btnViewAbout);
    btnViewAbout.setOnClickListener(this);
}
```

2. onClick:

```
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.btnSetReminder:
            startActivity(new Intent(this,
ReminderActivity.class));
            break;
        case R.id.btnViewAbout:
            startActivity(new Intent(this, About.class));
            break;
        default:
            break;
    }
}
```

3. onCreateOptionsMenu:

```

public boolean onCreateOptionsMenu(Menu menu) {
    menu.add(Menu.NONE, ADD_ID, Menu.NONE, "Tambah");
    menu.add(Menu.NONE, DELETE_ID, Menu.NONE, "Hapus");
    menu.add(Menu.NONE, SET_ID, Menu.NONE, "Set");
    return super.onCreateOptionsMenu(menu);
}

```

4. tampil:

```

private void tampil(){
    mItems = new ArrayList<Model>();
    populate();
    ListView obatList = (ListView)
    findViewById(R.id.obat_list);
    mObatAdapter = new Adapter(this, R.layout.obat_row,
    mItems);

    obatList.setAdapter(mObatAdapter);
    obatList.setOnItemClickListener(this);
}

```

5. pilihhari:

```

private void pilihhari(){
    dialoghari = new Dialog (this);
    dialoghari.setTitle("Pilih Hari");
    dialoghari.setContentview(R.layout.pilihhari);
    dialoghari.show();
    chkSen = (CheckBox)dialoghari.findViewById(R.id.senin);
    chkSen.setOnClickListener(this);
    chkSel
    (CheckBox)dialoghari.findViewById(R.id.selasa);
}

```

6. pilihsetting:

```

private void pilihsetting(){
    dialogsetting = new Dialog (this);
    dialogsetting.setTitle("Pilih Setting");
    dialogsetting.setContentview(R.layout.menupiliharming);
    dialogsetting.show();
    Button btnOk =
    (Button)dialogsetting.findViewById(R.id.ok);
    btnOk.setOnClickListener(this);
}

```

7. tambah:

```

private void tambah(){
    dialog= new Dialog(this);
    dialog.setTitle("");
    dialog.setContentview(R.layout.add);
    dialog.show();
    Button btnOk = (Button)dialog.findViewById(R.id.Ok);
    btnOk.setOnClickListener(this);
}

```

8. setContentView:


```
dialogharian.setContentView(R.layout.pilihharian);
```

9. findViewById:

```
Button btnOokk=(Button)dialogharian.findViewById(R.id.ookk);
```

10. delete:

```
private void delete() {
    mDaftarDBAdapter.deleteItem(mObat.getName());
    alarmManager.cancel(sn);
    tampil();
}
```

11. TimePickerDialog/DatePickerDialog:

```
private TimePickerDialog.OnTimeSetListener mTimeSetListener =
new TimePickerDialog.OnTimeSetListener() {
```

12. setAlarm:

```
protected void setAlarm() {
    final Calendar cal = Calendar.getInstance();
    cal.set(Calendar.HOUR_OF_DAY, mHour);
    cal.set(Calendar.MINUTE, mMin);
}
```

13. onReceive:

```
public void onReceive(Context context, Intent intent) {
    NotificationManager nm = (NotificationManager) context
        .getSystemService(Context.NOTIFICATION_SERVICE);
    mSdf = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");
    String name = intent.getStringExtra("name");
    String description =
        intent.getStringExtra("description");
    String jenisreminder =
        intent.getStringExtra("setting");
}
```

14. notification:

```
Notification notification = new
Notification(R.drawable.notification,
    "Waktunya Anda " + name + "!",
    System.currentTimeMillis());

notification.defaults = Notification.DEFAULT_ALL;
notification.flags =
Notification.FLAG_AUTO_CANCEL;
```

4.6 Pengujian Program

Untuk menguji aplikasi *healthy reminder* adalah dengan melakukan beberapa uji coba dalam bentuk cerita singkat atau skenario. Adapun pengujian aplikasi *healthy reminder* akan dicontohkan dalam 2 skenario, yaitu:

1. Penggunaan insulin,

Ada seseorang bernama Fulan, dia mengidap penyakit diabetes selama hampir 1 tahun. Kemudian setelah berkonsultasi dengan dokter Fulan pun diberi cara pengobatan dengan menyuntikan insulin secara berkala dalam setiap 1,5 jam. Waktu penyuntikan insulin tidak boleh terlambat karena akan mengurangi keefektifan dari insulin tersebut. Dengan kesibukan Fulan saati ini, maka ia memutuskan untuk membuat *reminder* pada perangkat Android yang ia miliki. Adapun cara pembuatan *reminder* yang dibutuhkan Fulan adalah sebagai berikut:

- a. Memilih ikon *healthy reminder* pada menu aplikasi seperti yang terlihat pada gambar 4.18.



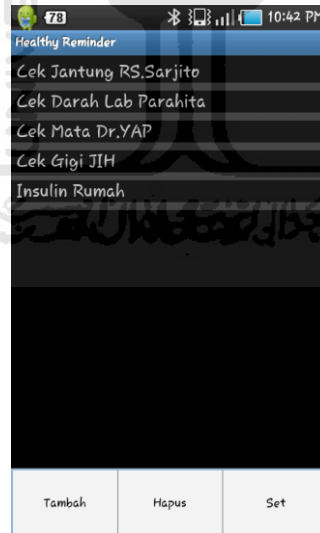
Gambar 4. 18 Pemilihan Ikon Healthy Reminder

- b. Pilih set reminder pada menu utama seperti yang terlihat pada gambar 4.19.



Gambar 4. 19 Pemilihan Set Reminder

- c. Pilih menu Insulin dan tekan tombol *option* seperti yang terlihat pada gambar 4.20.



Gambar 4. 20 Pemilihan Menu Insulin

- d. Pilih opsi Menitan seperti yang terlihat pada gambar 4.21.



Gambar 4. 21 Pilih Opsi Menitan

- e. Pilih waktu yang dibutuhkan seperti yang terlihat pada gambar 4.22.



Gambar 4. 22 Pemilhan Waktu

- f. Set Waktu seperti yang terlihat pada gambar 4.23.



Gambar 4. 23 Set Waktu

- g. Reminder berjalan seperti yang terlihat pada gambar 4.24.



Gambar 4. 24 Reminder Berjalan

2. Pemeriksaan mata secara berkala,

Ada seseorang bernama Falun, dia mengalami gangguan penglihatan sebelah kiri. Kemudian Ia pun datang ke RS YAP untuk melakukan pengecekan matanya. Dokter memberitahukan bahwa Falun terkena

katarak pada mata sebelah kirinya, namun belum begitu parah sehingga dokter menyarankan untuk melakukan pemeriksaan secara rutin yaitu 2 kali seminggu pada hari Senin dan Jumat pukul 13.00 WIB. Dengan kesibukan aktifitas yang sedemikian banyaknya waktu yang Falun punya tidaklah banyak. Dengan bantuan aplikasi *healthy reminder* Falun pun membuat *reminder* pada perangkat Androidnya agar dapat mengingatkan jadwal pemeriksaan mata pada hari Senin dan Jumat dengan tahapan sebagai berikut :

- a. Memilih ikon *healthy reminder* pada menu aplikasi seperti yang terlihat pada gambar 4.25.



Gambar 4. 25 Pemilihan Ikon Healthy Reminder

- b. Pilih set reminder pada menu utama seperti yang terlihat pada gambar 4.26.



Gambar 4. 26 Pemilihan Set Reminder

- c. Pilih menu Cek Mata DR YAP dan tekan tombol *option* seperti yang terlihat pada gambar 4.27.



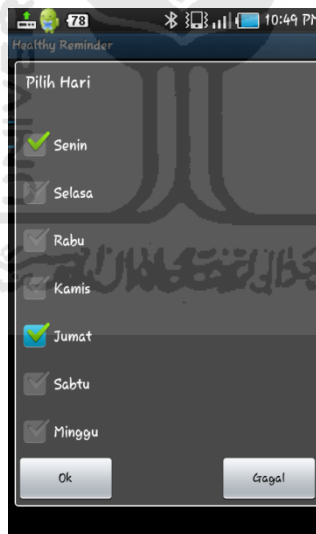
Gambar 4. 27 Pemilihan Menu Insulin

- d. Pilih opsi Mingguan seperti yang terlihat pada gambar 2.28.



Gambar 4. 28 Pilih Opsi Menitan

- e. Pilih hari yang ditentukan seperti yang terlihat pada gambar 4.29.



Gambar 4. 29 Pemilihan Hari

- f. Set Waktu seperti yang terlihat pada gambar 4.30.



Gambar 4. 30 Set Waktu

- g. Reminder berjalan seperti yang terlihat pada gambar 4.31.



Gambar 4. 31 Reminder Berjalan

4.7 Analisis Kinerja Perangkat Lunak

Untuk menguji sejauh aplikasi *healthy reminder* dengan pengguna setelah diberi tindakan oleh pengguna ke dalam Android.

4.7.1 Analisis Kesesuaian Dengan Landasan Teori

Sesuai dengan landasan teori dalam urutan pembuatan aplikasi *healthy reminder*, analisis yang dilakukan terhadap teori pembuatan *reminder* pada aplikasi ini yang dibangun adalah

1. Jenis aplikasi

Aplikasi yang dibuat berupa *reminder* yang berjalan pada *device* dengan sistem operasi Android.

2. Interaksi

Pengguna berinteraksi dengan objek-objek menggunakan interaksi *touchscreen*, mulai dari tombol utama pada aplikasi *healthy reminder* yang digunakan untuk memilih menu, hingga tombol *radiobutton* pada pemilihan opsi dan tombol *checkbox*.

3. Tipe Ekstensi Aplikasi

Aplikasi yang dibangun dengan ekstensi *file* yang bertipe *.apk* (untuk Sistem Operasi Android). Sehingga memudahkan pengguna, karena dapat dijadikan *native application* pada sistem operasi tersebut.

BAB V

SIMPULAN DAN SARAN

Kesimpulan

Berdasarkan hasil penelitian, analisis, perancangan sistem dan pembuatan program sampai dengan tahap penyelesaian, maka dapat ditarik beberapa kesimpulan antara lain sebagai berikut :

1. Aplikasi *healthy reminder* berbasis sistem operasi Android menggunakan interaksi *touchscreen* telah berhasil dikembangkan dan sudah bisa dijalankan pada *device* Android menggunakan interaksi *touchscreen*.
2. Aplikasi yang berjalan menggunakan *sound* dan *vibrate* agar dapat berjalan ketika dieksekusi.

Saran

Berdasarkan kekurangan dan keterbatasan yang ada di dalam Aplikasi *healthy reminder* ini, maka disarankan :

1. Dalam pengembangan berikutnya dapat digunakan *voice command* sebagai interaksi aplikasi selain *touchscreen*, karena semua *device* Android telah dilengkapi dengan *voice command*.
2. Tampilan yang interaktif dibatasi pada aplikasi ini karena keterbatasan *resource memory*, untuk kedepannya bisa ditambahkan dengan interaksi-interaksi dan gambar yang lebih menarik.

DAFTAR PUSTAKA

- [ANO11a] *What Is Android?*, diakses pada tanggal 25 Februari 2011 dari <http://developer.android.com/guide/basics/what-is-android.html>
- [ANO11b] *Application Fundamental*, diakses pada tanggal 25 Februari 2011 dari <http://developer.android.com/guide/topics/fundamentals.html>
- [ANO11c] *Development Resources*, diakses pada tanggal 25 Februari 2011 dari <http://developer.android.com/guide/topics/fundamentals.html>
- [ANO11d] *Mobile Snapshot: Smartphones Now 28% of U.S. Cellphone Market*, diakses pada tanggal 25 Februari 2011 dari http://blog.nielsen.com/nielsenwire/online_mobile/mobile-snapshot-smartphones-now-28-of-u-s-cellphone-market
- [MUL10] Mulyadi, ST, 2010. *Membuat Aplikasi Untuk Android*. Multimedia Center Publishing. Yogyakarta.
- [ROG11] Roger Entner, *Smartphone to Overtake Feature Phones in U.S. by 2011*, diakses 25 Februari 2011 dari <http://blog.nielsen.com/nielsenwire/consumer/smartphones-to-overtake-feature-phones-in-u-s-by-2011/>
- [SON10] Sony Hanifudin, *Sekilas Mengenal Android OS*, <http://www.jagatreview.com/2010/08/android-os/>, diakses 17 Desember 2010.