

**RANCANG BANGUN APLIKASI MONITORING RESPONSES SERVER
BERBASIS ANDROID**

SKRIPSI



**Disusun oleh:
EKO CHANDRA EDY SAPUTRA
09.18.093**



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

LEMBAR PERSETUJUAN

**RANCANG BANGUN APLIKASI MONITORING RESPONSES SERVER
BERBASIS ANDROID**

SKRIPSI

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

Disusun Oleh :

Eko Chandra Edy Saputra


09.18.093

Diperiksa dan Disetujui,

Dosen Pembimbing I

Dosen Pembimbing II


Joseph Dedy Irawan, ST, MT.
NIP. 197404162005011002


Sonny Prasetyo, ST, MT.
NIP.P. 1031000433

Mengetahui,

Ketua Program Studi Teknik Informatika S-1




Joseph Dedy Irawan, ST, MT.
NIP. 197404162005011002

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

*Teriring Ucapan Terima Kasih kepada
Ayah dan Ibu tercinta*

RANCANG BANGUN APLIKASI MONITORING RESPONSES SERVER BERBASIS ANDROID

Eko Chandra Edy Saputra

09.18.093

Jurusan Teknik Informatika S-1
Fakultas Teknologi Industri
Institut Teknologi Nasional Malang
Email : ekocs@ymail.com

Dosen Pembimbing :

- 1. Joseph Dedy Irawan, ST, MT**
- 2. Sonny Prasetyo, ST, MT**

ABSTRAK

Teknologi jaringan komputer saat ini telah berkembang pesat. Hampir semua bidang sekarang ini telah memiliki jaringan komputer, dalam mengelola jaringan komputer. Jaringan adalah sesuatu yang kompleks yang tidak pernah bisa diamati terus dengan hanya mata telanjang, diperlukan suatu sistem monitoring untuk mengetahui kondisi jaringan atau sistem yang ada.

Jika pada suatu sistem ada yang tidak berfungsi ataupun ada masalah pada security atau aplikasi dalam suatu sistem jaringan, maka sistem akan melakukan notifikasi kepada seorang network administrator atau sistem administrator melalui sistem alarm berupa email ataupun notifikasi lainnya.

Berkaitan dengan tugas administrator yaitu untuk mengelola dan menangani setiap permasalahan yang terjadi pada server dalam waktu 24 jam, diatas maka penulis mencoba untuk membuat suatu sistem monitoring server yang akan menampilkan kondisi real-time sebuah server berbasis smartphone Android. Sehingga dapat memberitahu keadaan server kepada administrator.

Kata kunci: Monitoring Server, Administrator, Android

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 MONITORING RESPONSES SERVER BERBASIS ANDROID sesuai 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.
3. Joseph Dedy Irawan, ST, MT selaku Ketua Jurusan Teknik Informatika ITN Malang.
4. Joseph Dedy Irawan, ST, MT selaku dosen pembimbing I.
5. Sonny Prasetio, 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, Agustus 2013

Penyusun

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
LEMBAR PERSEMBAHAN	iii
ABSTRAK.....	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	xi
 BAB I PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah.....	2
1.5 Metode Penelitian.....	3
1.6 Sistematika Penulisan.....	3
 BAB II LANDASAN TEORI	
2.1 Server.....	6
2.2 Sistem Monitoring Server.....	6
2.2.1 Ping.....	7
2.3 Android.....	9
2.3.1 Perkembangan Sistem Operasi Android.....	10
2.3.2 Arsitektur Android.....	13
2.4 Emulator	16
2.4.1 Alasan Dibuatnya Emulator	17

2.4.2 Kinerja Emulator	17
2.4.3 Membuat Emulator	17
2.4.4 Jenis – Jenis Emulator	17
2.4.5 Legalitas Pengguna Emulator.....	18
2.5 Perangkat Lunak Pendukung Aplikasi	18
2.5.1 Eclipse	18
2.5.2 Android SDK.....	22
2.5.3 SQLite.....	22
2.5.4 Bluestack	24

BAB III ANALISIS DAN PERANCANGAN

3.1 Pendahuluan	25
3.2 Analisis	25
3.2.1 Deskripsi Aplikasi	25
3.2.2 Analisis Jenis Aplikasi	26
3.2.3 Sasaran Pengguna Aplikasi	26
3.2.4 Analisis Sistem Aplikasi.....	26
3.2.5 Analisis Kebutuhan	26
3.3 Perancangan.....	28
3.3.1 Blok Diagram	28
3.3.2 Flowchart.....	29
3.3.3 Algoritma.....	30
3.3.4 Sistem Kerja Monitoring Server.....	31
3.4 Struktur Navigasi Aplikasi	31
3.5 Setting pada Android dan Lingkungan Kerja Eclipse	34
3.5.1 Setup Handset Smartphone Android	34
3.5.2 Lingkungan Kerja Eclipse	36

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1 Pendahuluan	40
4.2 Implementasi	40
4.2.1 Pembuatan Project Pada Eclipse	40

4.2.2 Pembuatan Menu Login	43
4.2.3 Pembuatan Menu Utama	45
4.2.4 Pembuatan Menu Monitoring	45
4.2.5 Pembuatan Menu Last Check	47
4.2.6 Pembuatan Menu Help	47
4.2.7 Publish Android Project ke APK.....	48
4.3 Pengujian Sistem	51
4.3.1 Instalasi Aplikasi Pada Smartphone Android.....	51
4.3.2 Tampilan Login	52
4.3.3 Tampilan Menu Utama	52
4.3.4 Tampilan Menu Monitoring	53
4.3.5 Tampilan Menu Last Check	54
4.3.6 Tampilan Menu Help.....	54
4.3.7 Pengujian Monitoring Server	55
4.3.8 Tampilan Hasil Last Check	57
4.3.9 Tampilan Hasil Menggunakan Interval	58
4.3.10 Pengujian Aplikasi pada Smartphone Android	59
4.3.11 Pengujian Aplikasi Berdasarkan versi SDK.....	59
4.3.12 Pengujian Fungsionalitas.....	60
4.3.13 Hasil Pengujian Sistem Monitoring Server	61

BAB V PENUTUP

5.1 Kesimpulan.....	62
5.2 Saran.....	62

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

2.1	Logo Android	9
2.2	Arsitektur Android.....	13
3.1	Diagram Blok Perencanaan	25
3.2	Blok Diagram	28
3.3	Flowchart Sistem.....	29
3.4	Flowchart Sistem Timer	30
3.5	Menu Login	32
3.6	Menu Utama	32
3.7	Menu Monitoring Server	33
3.8	Last Check	34
3.9	Menu Help	34
3.10	Gambar Setting.....	35
3.11	Pengaturan Untuk USB Debugging.....	35
3.12	Halaman Utama Eclipse	36
3.13	Android Virtual Device (AVD).....	38
3.14	Emulator Bluestacks.....	39
4.1	Membuat Project Baru di Eclipse.....	41
4.2	Nama Aplikasi Android.....	41
4.3	Activity yang akan digunakan.....	42
4.4	Tampilan awal Blank Activity.....	43
4.5	Membuat File XML baru.....	43
4.6	Pembuatan Menu Login	44
4.7	Pembuatan Menu Utama	45
4.8	Pembuatan Menu Monitoring.....	46
4.9	Pembuatan Menu Last Check.....	47
4.10	Pembuatan Menu Help	48
4.11	Memulai Ekspor Application Package	49
4.12	Memilih Project.....	49
4.13	Membuat Keystore Baru.....	49
4.14	Form untuk Membuat Key Baru.....	50

4.15	Destination APK File dan Key	50
4.16	Permission Instalasi Aplikasi Pada Android	51
4.17	Tampilan Login	52
4.18	Tampilan Menu Utama.....	53
4.19	Menu Runtime	53
4.20	Menu Last Check.....	54
4.21	Menu Help	54
4.22	Monitoring terhadap Server ITN Malang dan UNEJ	55
4.23	Monitoring terhadap Beberapa Server Down.....	56
4.24	Notifikasi dari Sistem	57
4.25	Data Last Check	57
4.26	Interval Waktu Ping Tiap 1 Menit.....	58

DAFTAR TABEL

2.1	Daftar Tipe Nilai ICMP	8
2.2	Versi dan pengembangan Android	10
2.3	Peluncuran Simultan Eclipse	22
3.1	Analisa kebutuhan perangkat keras (Hardware).....	27
3.2	Analisa kebutuhan perangkat lunak (Software)	27
4.1	Hasil Pengujian pada Smartphone.....	58
4.2	Hasil Pengujian Berdasarkan versi SDK.....	59
4.3	Pengujian Fungsionalitas.....	60
4.4	Hasil Pengujian Berdasarkan versi SDK.....	61

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi jaringan komputer saat ini telah berkembang pesat. Hampir semua bidang sekarang ini telah memiliki jaringan komputer, dalam mengelola jaringan komputer. Jaringan adalah sesuatu yang kompleks yang tidak pernah bisa diamati terus dengan hanya mata telanjang, diperlukan suatu sistem monitoring untuk mengetahui kondisi jaringan atau sistem yang ada. Hal ini dilakukan agar kondisi jaringan terjaga sehingga setiap gangguan dapat teratasi dengan segera.

Kata *network monitoring* menggambarkan bagaimana suatu sistem yang secara konstan memonitor kondisi jaringan (*router, server* maupun *device* jaringan lainnya). Jika pada suatu sistem ada yang tidak berfungsi ataupun ada masalah pada *security* atau aplikasi dalam suatu sistem jaringan, maka sistem akan melakukan notifikasi kepada seorang *network administrator* atau *system administrator* melalui sistem alarm berupa email ataupun notifikasi lainnya.

Masalah datang ketika seorang *network administrator* ataupun *system administrator* tidak bisa memonitor terus menerus, melakukan monitoring didepan komputer selama 24 jam, karena harus mengerjakan pekerjaan atau kegiatan yang lainnya. Notifikasi menggunakan *email* ataupun sebuah alarm suara dikomputer dirasa masih kurang, karena sifatnya yang harus diam didepan komputer, akibatnya penanganan masalah yang menjadi lambat dan waktu *downtime* yang menjadi lama.

Berkaitan dengan tugas *administrator* yaitu untuk mengelola dan menangani setiap permasalahan yang terjadi pada server dalam waktu 24 jam, diperlukan penanganan khusus dan secepatnya untuk menghadapi setiap permasalahan yang terjadi. Seperti kita ketahui bahwa *administrator* juga manusia biasa yang mana membutuhkan istirahat dan tidak selalu pada tempat kerjanya. Dengan keterbatasan administrator tersebut maka perlu adanya *tool* yang dapat melakukan *monitoring* terhadap server dan dapat

memberitahukan kepada administrator apabila sewaktu – waktu mengalami masalah.

Melihat permasalahan diatas maka penulis mencoba untuk membuat suatu system *monitoring* server yang akan menampilkan kondisi *real-time* sebuah server berbasis *smartphone* Android. Sehingga dapat memberitahu keadaan server kepada administrator.

1.2 Rumusan Masalah

Dari latar belakang yang telah dikemukakan sebelumnya, maka terdapat rumusan masalah sebagai berikut:

1. Bagaimana cara mempermudah administrator dalam *memonitoring* server yang dikelolanya?
2. Bagaimana aplikasi memberitahu kepada administrator apabila server mengalami *down* server?
3. Bagaimana cara membuat program *monitoring responses* server di Android.

1.3 Tujuan

Tujuan dalam pembuatan aplikasi *monitoring* server ini adalah memberikan informasi tentang keadaan server berupa *alert* (notifikasi) pada aplikasi *monitoring* server.

1.4 Batasan Masalah

Adapun batasan masalah dalam pembuatan aplikasi *monitoring* server ini, meliputi:

1. Pembuatan aplikasi ini menggunakan bahasa pemrograman Java Eclipse untuk *Smartphone* Android.
 2. Aplikasi *Monitoring* server menampilkan status berupa kondisi *UP* atau *DOWN*.
 3. Aplikasi hanya berjalan pada server yang paket *ping* tidak *disable* oleh server tersebut.
 4. Server yang akan dimonitoring adalah server HTTP atau server WEB (WWW).
-

5. Aplikasi berjalan pada *smartphone* android dan memiliki koneksi internet atau Wireless (Hot-Spot).

1.5 Metode Penelitian

Metode penelitian yang digunakan dalam pembuatan aplikasi ini adalah:

1. Tahap Studi Literatur

Pencarian dan pengumpulan literatur berupa artikel, tutorial, jurnal, buku referensi, dan sumber lain untuk mendalami konsep – konsep pembuatan, pengembangan aplikasi berbasis android yang berhubungan dengan *Monitoring Responses Server*.

2. Tahap Perancangan

Tahap ini akan dilakukan tahap analisis kebutuhan dan perancangan system dari sebuah aplikasi *monitoring* server berbasis android.

3. Tahap Implementasi dan Pengujian

Pada tahap ini akan dilakukan tahap pembuatan aplikasi dan pengujian terhadap aplikasi yang telah selesai dibuat, dengan tujuan untuk mengetahui akan adanya kesalahan atau *error* yang telah ada pada aplikasi *monitoring* server berbasis android.

4. Tahap Pembuatan Laporan

Pada tahap ini akan dilakukan penyusunan laporan dan pengumpulan dokumentasi dengan mengikuti kaidah penulisan yang benar dan sesuai dengan ketentuan – ketentuan atau sistematika yang telah ditetapkan oleh Institut Teknologi Nasional Malang.

1.6 Sistematika Penulisan

Sistematika penyusunan laporan skripsi ditujukan untuk memberikan gambaran dan uraian secara garis besar yang meliputi bab – bab sebagai berikut :

BAB I : PENDAHULUAN

Pada Bab ini membahas tentang Latar Belakang, Rumusan Masalah, Batasan Masalah, Maksud dan Tujuan Penelitian, Manfaat Penelitian, Metode Penelitian dan Sistematika Penulisan Laporan.

BAB II : LANDASAN TEORI

Pada Bab ini membahas tentang Landasan Teori yang merupakan tinjauan pustaka, menguraikan teori-teori yang mendukung judul, dan pembahasan secara detail. Landasan teori dapat berupa definisi-definisi atau model yang langsung berkaitan dengan ilmu atau masalah yang diteliti. Pada bab ini juga dituliskan tentang software (komponen) yang digunakan dalam pembuatan Program atau keperluan saat penelitian.

BAB III : ANALISIS DAN PERANCANGAN SISTEM

Bab ini berisi antara lain: Tinjauan Umum yang menguraikan tentang gambaran umum objek penelitian, misalnya gambaran umum Instansi (struktur organisasi, Pengelolaan), atau gambaran umum lokasi, serta data yang dipergunakan untuk memecahkan masalah – masalah yang dihadapi, berkaitan dengan kegiatan penelitian.

Pada Bab ini juga membahas “analisis masalah”, yang akan menguraikan tentang analisis terhadap permasalahan yang terdapat pada kasus yang sedang diteliti. Meliputi analisis terhadap masalah sistem yang sedang berjalan, analisis hasil solusinya, dan analisis kebutuhan penelitian.

BAB IV : IMPLEMENTASI DAN PEMBAHASAN

Pada Bab ini akan membahas paparan implementasi dan analisis hasil uji coba program, serta memaparkan hasil-hasil dari tahapan penelitian, dari tahap analisis, desain, implementasi desain, hasil testing dan implementasinya, berupa penjasanteoritik, baik secara kualitatif, kuantitatif, atau secara statistik. Dan Selain membandingkan dengan hasil penelitian yang masih manual.

BAB V : PENUTUP

Pada Bab ini berisi kesimpulan dan saran. Kesimpulan didapat dari ulasan data – data penelitian, menyimpulkan bukti-bukti yang diperoleh dan akhirnya menarik intisari apakah hasil yang didapat (dikerjakan), layak untuk digunakan (diimplementasikan).

BAB II LANDASAN TEORI

2.1 Server

Server adalah suatu komputer yang menjadi pengelola dan pusat bagi komputer lainnya. Oleh karena itu, komputer server harus memiliki spesifikasi yang lebih tinggi dari pada *client*.

Berikut jenis – jenis server diantaranya:

1. Server Aplikasi (*Application Server*)

Application server adalah Server yang digunakan untuk menyimpan berbagai macam aplikasi yang dapat diakses oleh *client*. Contoh dari aplikasi server yang sudah umum yaitu:

- a. *Dynamic Host Configuration Protocol (DHCP)*
- b. *Mail Server*
- c. *Server HTTP Apache* atau *server web/WWW Apache*
- d. *File Transfer Protocol (FTP)*
- e. *Domain Name System (DNS)*

2. Server Data (*Data Server*)

Server Data merupakan server yang digunakan sebagai tempat penyimpanan data baik yang digunakan *client* secara langsung maupun data yang diproses server aplikasi.

3. Server Proxy (*Proxy Server*)

Proxy Server adalah sebuah komputer server atau program komputer yang dapat bertindak sebagai komputer lainnya untuk melakukan *request* terhadap *content* dari *internet* atau *intranet*. *Proxy* Server bertindak sebagai *gateway* terhadap dunia internet untuk setiap komputer *client*.

2.2 Sistem Monitoring Server

Sistem *Monitoring* Server merupakan suatu proses untuk mengumpulkan data dari server. Biasanya data yang dikumpulkan merupakan data yang *real-time*. Secara garis besar tahapan dalam sebuah sistem *monitoring* server terbagi dalam tiga proses besar antara lain:

1. Proses di dalam pengumpulan data *monitoring* server (*request* dan *responses*).
2. Proses di dalam analisis data *monitoring* server.
3. Proses di dalam menampilkan data *monitoring* server pada aplikasi.

Aksi yang terjadi di antara proses – proses dalam sebuah system *monitoring* adalah berbentuk *service*, yaitu suatu proses yang terus – menerus berjalan pada interval waktu tertentu. Parameter yang digunakan untuk proses *monitoring* adalah *ping*.

2.2.1 PING

Ping adalah *software* yang berjalan di atas *protocol ICMP* untuk mengecek hubungan antara dua computer di *intranet* maupun *internet*. Ping diciptakan pada tahun 1983 oleh **Muss Mike**. Ping akan mengirimkan *Internet Control Message Protocol (ICMP) Echo Request Message* pada *ip address* computer yang dituju dan meminta *response* dari komputer tersebut. Ping juga dapat berarti program dasar yang mengijinkan satu pengguna untuk memverifikasi bahwa alamat *internet protocol* tentu ada dan dapat menerima permintaan – permintaan. (Jhuana, 2005)

Kegunaan ping antara lain sebagai berikut:

1. Mengetahui status *up/down* komputer dalam jaringan.
 2. Kita dapat memeriksa apakah sebuah komputer *up/down* menggunakan perintah ping, jika komputer tersebut memberikan *response* terhadap perintah ping yang kita berikan maka dikatakan bahwa komputer tersebut *up* atau hidup.
 3. Memonitor *availability* status komputer dalam jaringan. Ping dapat dapat digunakan sebagai *tool monitoring* availibilitas komputer dalam jaringan yang merupakan salah satu indicator kualitas jaringan yaitu dengan melakukan ping secara periodik pada computer yang dituju. Semakin kecil *downtime*, semakin bagus kualitas jaringan tersebut.
 4. Mengetahui responsifitas komunikasi sebuah jaringan. Besarnya nilai *delay* atau *latency* yang dilaporkan oleh ping menjadi indikasi
-

seberapa *responsive* komunikasi terjadi dengan komputer yang dituju. Semakin besar nilai *delay* menunjukkan semakin lamban respon yang diberikan. Sehingga nilai *delay* ini juga bisa digunakan sebagai indikator kualitas jaringan.

Contoh penggunaan Ping sebagai berikut:

- a. **Byte** menunjukkan besar *request* paket yang dikirimkan
- b. **Time** menunjukan nilai "*round trip delay*" (disebut juga sebagai *delay* atau *latency*) yang menunjukan waktu yang diperlukan paket yang anda kirimkan untuk mencapai komputer yang dituju. Nilai ini dihitung dengan membagi dua selisih waktu ping paket mulai dikirimkan dengan waktu respon dari ping paket yang diterima.
- c. **TTL** merupakan nilai "*Time To Live*" yang digunakan untuk mencegah adanya *circular routing* pada suatu jaringan. Dengan mengurangi nilai TTL awal yaitu 128 dengan nilai TTL akhir maka bias dihitung banyaknya hop yang dilalui dari komputer asal ke komputer tujuan. Setiap kali paket ping melalui sebuah *ip address* maka nilai TTL-nya akan dikurangi satu. Sehingga jika TTL mencapai nilai nol, paket ping akan di-*discard* atau di-*drop* dan hasil ping menunjukan : "*TTL expired in transit*"

Berikut daftar tipe nilai ICMP yang ditunjukkan pada tabel 2.1

Tabel 2.1 Daftar Tipe Nilai ICMP

ICMP Type	Literal
0	Echo-reply
3	Destination unreachable
4	Source-quench
5	redirect
6	Alternate-address
8	echo
9	Router-advertisement

Apabila utilitas *ping* menunjukkan hasil yang positif maka kedua komputer tersebut saling terhubung di dalam sebuah jaringan. Hasil *statistic* keadaan koneksi ditampilkan dibagian akhir.kualitas koneksi dapat dilihat dari besarnya waktu bolak – balik (*roundtrip*) dan besarnya jumlah paket yang hilang (*packet loss*). Semakin kecil kedua angka tersebut, semakin bagus kualitas koneksinya.

2.3 Android

Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti Smartphone dan komputer tablet. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak. Pada awalnya, Google Inc. mengakusisi Android Inc., yaitu pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android tersebut, dibentuklah Open Handset Alliance, yaitu konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk diantaranya ialah Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia, LG, Marvell Technology Group, Samsung Electronics, Sprint Nextel. Pada tanggal 9 Desember 2008 diumumkan bahwa 14 anggota baru akan bergabung dalam proyek pengembangan android, termasuk PacketVideo , ARM Holdings, Atheros communication, Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Logo Android ditunjukkan dalam gambar 2.1.










Gambar 2.1 Logo Android

2.3.1 Perkembangan Sistem Operasi Android

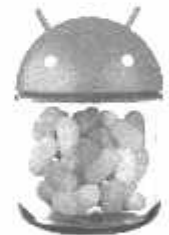
Sebagai sebuah sistem operasi terbuka yang dikembangkan oleh sebuah konsorsium yang besar, perkembangan android sangat pesat. Android memiliki versi sebagai pengembangan dari versi sebelumnya. Adapun versi dan pengembangan Android yang ditunjukkan dalam tabel 2.2.

Tabel 2.2 Versi dan pengembangan Android

Versi Android	Kode Nama	Fitur Tambahan Utama	Ukuran Layar	Tanggal Rilis	Logo
1.0	Apple Pie		320×480 HVGA	23 Sep 2008	
1.1	Banana Bread		320×480 HVGA	9 Feb 2009	
1.5	Cupcake	<ul style="list-style-type: none"> - Bluetooth A2DP, AVRCP support - Soft-keyboard with text-prediction - Record/watch videos - Gesture framework 	320×480 HVGA	30 Apr 2009	 Cupcake Android 1.5
1.6	Donut	<ul style="list-style-type: none"> - Turn-by-turn navigation 	800×480 WVGA	15 Sep 2009	 Donut Android 1.6
2.0	Eclair	<ul style="list-style-type: none"> - HTML - Digital zoom - Microsoft Exchange support - Bluetooth 2.1 - Live Wallpapers - Updated User Interface 	800×480 WVGA	26 Oct 2009	 Eclair Android 2.0/2.1
2.0.1	Eclair		800×480 WVGA	3 Dec 2009	
2.1	Eclair		800×480 WVGA	12 Jan 2010	

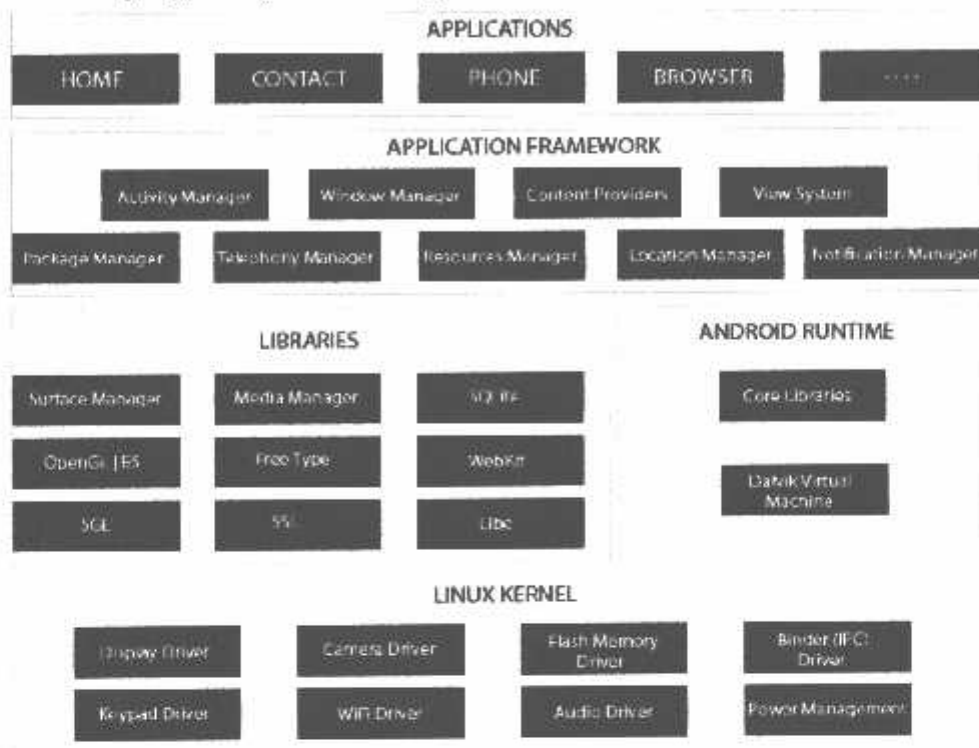
2.2	Froyo	<ul style="list-style-type: none"> - Speed improvements - JIT implementation - USB Tethering - Application installation to the expandable memory - Upload file support in the browser - Animated GIFs - Updated User Interface - Web Mobile video playback capability - Improve copy / paste 	800×480 WVGA	20 Mei 2010	 <p>Froyo Android 2.2</p>
2.3	Gingerbread	<ul style="list-style-type: none"> - Social Networking features - Near Field Communication support - NativeVoIP/SIP Support - Video call support 	1366×768 WXGA	6 Dec 2010	 <p>Gingerbread Android 2.3</p>
2.3.3	Gingerbread	<ul style="list-style-type: none"> - NFC API improvement (peer to peer communication) 	1366×768 WXGA	9 Feb 2011	
2.3.4	Gingerbread	<ul style="list-style-type: none"> - Open Accessory API - Voice or video chat using Google Talk 	1366×768 WXGA	10 Mei 2011	
2.3.5	Gingerbread	<ul style="list-style-type: none"> - Improved network performance for the Nexus S 4G - Fixed Bluetooth issues on the Samsung Galaxy S - Gmail app. improvements 	1366×768 WXGA	25 Juli 2011	
2.3.6	Gingerbread	<ul style="list-style-type: none"> - Voice search issue fixed 	1366×768 WXGA		
2.3.7	Gingerbread	<ul style="list-style-type: none"> - Google Wallet support for the Nexus S 4G 	1366×768 WXGA		
3.0	Honeycomb	<ul style="list-style-type: none"> - Multi core support - Better tablet support - Updated 3D User Interface - Google Talk video chat - Google eBooks - Private browsing 	1366×768 WXGA	22 Feb 2011	 <p>Honeycomb Android 3.0</p>

3.1	Honeycomb	<ul style="list-style-type: none"> - User Interface improvements - Open Accessory API - USB host API - Mice, joysticks, gamepads support - Resizable Home screen widgets - MTP notifications - RTP API for audio - Optimizations for a wider range of tablets 	1366×768 WXGA	10 Mei 2011
3.2	Honeycomb	<ul style="list-style-type: none"> - Compatibility display mode (zoom for fixed-sized apps) - Media sync from SD card - Facial recognition (Face Unlock) - User Interface use Hardware acceleration - Better voice recognition (dictating/Voice typing) 	1366×768 WXGA	15 Juli 2011
4.0	Ice Cream Sandwich	<ul style="list-style-type: none"> - Web browser, allows up to 16 tabs - Updated launcher (customizable) - Android Beam app to exchange data through NFC - Resizable widgets - Dual boot support - Updated File Manager 	1366×768 WXGA	19 Oct 2011
4.1	Jelly Bean	<ul style="list-style-type: none"> - Updated Virtual Keyboard - Battery Saver 	1366×768 WXGA	27 Juni 2012

Ice Cream Sandwich
4.0-4.0.4

2.3.2 Arsitektur Android

Diagram berikut menunjukkan komponen utama dari sistem operasi Android yang ditunjukkan dalam gambar 2.2.



Gambar 2.2 Arsitektur Android

1. Linux Kernel

Android bukan Linux, akan tetapi Android dibangun diatas Linux Kernel yaitu versi 2.6 sehingga kehandalannya bisa di percaya. Untuk inti sistem service Linux yang digunakan seperti keamanan, manajemen memori, proses manajemen, network, dan driver model. Seperti yang terlihat digambar, Linux Kernel menyediakan Driver Layar, Kamera, Keypad, WiFi, Flash memory, Audio, dan IPC(Interprocess Communication) untuk mengatur aplikasi dan keamanan. Kernel juga bertindak sebagai lapisan abstrak antara hardware dan software stacknya.

2. Libraries

Android menyertakan libraries C / C++ yang digunakan oleh berbagai komponen dari sistem Android. Kemampuan ini

disediakan kepada Developer aplikasi melalui Framework Aplikasi Android. Beberapa inti libraries tercantum dibawah ini :

a. System C library

Variasi dari implementasi BSD berasal dari pelaksanaan sistem standar C library (LibC), sesuai untuk perangkat embedded berbasis Linux.

b. Media Library

Packet Video berdasarkan OpenCORE; library mendukung pemutaran rekaman dan populer banyak format audio dan video, serta file gambar, termasuk MPEG4, H.264, MP3, AAC, AMR, JPG, dan PNG.

c. Surface Manager

Mengelola akses ke subsistem layar, lapisan komposit 2D dan grafis 3D dari beberapa aplikasi yang ada pada handphone.

d. LibWebCore

Mesin web modern yang powerfull yang baik browser Android embeddable web.

e. SGL

Mesin Grafis 2D.

f. 3D Library

Sebuah pelaksanaan berdasarkan OpenGL | ES 1,0 API; perpustakaan baik menggunakan perangkat keras akselerasi 3D atau yang disertakan, sangat optimal 3D software rasterizer.

g. FreeType

Bitmap dan vektor font rendering.

h. SQLite

Mesin database yang kuat dan ringan, dan penghubung tersedia untuk semua aplikasi android. SQLite adalah bentuk mini dari MySQL, SQLite lebih dikhususkan ke piranti mobile dimana dibutuhkan aplikasi yang memiliki fungsi sama tetapi memiliki ukuran yang lebih kecil.

3. Android Runtime

Android terdiri dari satu set *core libraries* yang menyediakan sebagian besar fungsi yang sama dengan terdapat pada *core libraries* bahasa pemrograman Java. Setiap aplikasi menjalankan prosesnya sendiri dalam Android, dengan masing-masing instan dari mesin virtual Dalvik. Dalvik dirancang agar perangkat dapat menjalankan multiple VMs secara efisien. Mesin virtual Dalvik 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 kedalam *.dex format oleh tool “dx” yang telah disertakan. Dalvik VM bergantung pada kernel Linux untuk, berfungsi seperti threading dan manajemen memori tingkat rendahnya.

4. Framework Aplikasi

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 mungkin akan memanfaatkan kemampuan ini. Mekanisme yang sama memungkinkan komponen untuk diganti oleh pengguna. Semua aplikasi merupakan rangkaian set layanan dan sistem, termasuk :

- a. Views, yang kaya dan ekstensible yang dapat digunakan untuk membangun aplikasi, termasuk list, grids, kotak teks, tombol, dan bahkan sebuah embeddable web.
 - b. Content Provider, yang memungkinkan aplikasi ini untuk mengakses data atau membagi data tersebut menjadi beberapa bagian. Disini juga tersedia berbagai macam jenis aplikasi yang dapat diakses secara langsung oleh pengguna Android mobile.
-

- c. Resources Manager, yang menyediakan akses ke kode non-sumber seperti string, gambar, dan tata letak file. Pada string dapat diisi dengan penjelasan dari sebuah aplikasi yang telah programmer buat.
- d. Notification Manager, yang memungkinkan semua aplikasi menampilkan alert yang bisa dikostumisasikan didalam status bar. Contoh : notifikasi chat sms yang masuk, gagal, ataupun pending, notifikasi bluetooth yang menyala, mati, atau dalam sambungan.
- e. Activity Manager, yang mengelola siklus hidup dan menyediakan navigasi umum backstack. Contoh : Home, Back, Notification, dll.

5. Applications

Android telah menyertakan aplikasi inti seperti email client, SMS, kalender, peta, browser, kontak, dan lain-nya. Semua aplikasi tersebut ditulis dengan bahasa pemrograman Java. Pada layer inilah developer menempatkan aplikasi yang dibuat. Yang istimewa adalah pada Android semua aplikasi baik aplikasi inti maupun aplikasi pihak ketiga berjalan pada layer aplikasi dengan menggunakan libraries API yang sama. Ini berarti semua aplikasi yang dibuat 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.

2.4 Emulator

Emulator lebih tepatnya piranti lunak emulator memungkinkan suatu program atau piranti lunak yang dibuat pada awalnya oleh suatu sistem komputer (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.

2.4.1 Alasan Dibuatnya Emulator

Adapun beberapa alasan mengapa dibuatnya emulator, beberapa diantaranya 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 dicumulasikan di komputer personal (PC).

2.4.2 Kinerja Emulator

Kebanyakan emulator tidak mampu untuk secara sempurna mengemulate 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.

2.4.3 Membuat Emulator

Memprogram emulator adalah hal yang 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. 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.

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.

2.5 Perangkat Lunak Pendukung Aplikasi

2.5.1 Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*).

Berikut ini adalah sifat dari Eclipse:

- a. **Multi-platform:** Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
 - b. **Mult-language:** Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
-

- c. **Multi-role:** Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*. Agar dapat menggunakan Eclipse, kita harus mengenal beberapa komponen interface di Eclipse, yaitu:

- a. **Package Explorer**, merupakan jendela yang menampilkan project-project yang sudah kita buat dalam sebuah workspace. Jendela ini layaknya explore pada sistem operasi windows, yang berguna untuk mengeksplor project yang ada dalam sebuah workspace
- b. **Java Editor**, merupakan jendela yang menampilkan coding java yang sedang kita kerjakan. Dari sini kita mengetikkan perintah-perintah dalam syntax java untuk membuat sebuah aplikasi.
- c. **Console**, merupakan jendela yang menampilkan hasil output dari aplikasi yang kita buat. Agar hasil outputnya mau berjalan, tekan RUN pada toolbar. Jika yang muncul error, berarti aplikasi yang kita buat memang memiliki error, dan console ini akan menunjukkan dimana letak errornya.
- d. **Toolbar**, berisi shortcut perintah dalam bentuk icon. Seperti create new (project, folder), create new visual classes (GUI with Swing), Save, Run, Create new package, Create new class dan masih banyak lagi.

Setiap membuat sebuah project, secara otomatis eclipse akan membuat folder untuk project tersebut. Berikut beberapa penulisan yang sering dan dianjurkan untuk digunakan dalam penulisan coding java.
Project dan Class:

- Selalu diawali dengan huruf besar
- Tidak menggunakan spasi
- Tidak dimulai dengan angka
- Jika lebih dari satu kata, tiap awal kata dengan huruf besar.
Contoh: AplikasiKalkulator

Method:

- Selalu diawali dengan huruf kecil
- Tanpa menggunakan spasi
- Jangan diawali dengan angka
- Jika lebih satu kata untuk kata kedua dan selanjutnya diawali dengan huruf besar. Contoh: getAngkaAplikasiKalkulator

1. Sejarah

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak pengembangan IBM Visual Age for Java 4.0. Produk Eclipse ini diluncurkan oleh IBM pada tanggal 5 November 2001. IBM menginvestasikan US\$ 40 juta untuk pengembangannya. Sejak 5 November 2001, konsorsium Eclipse Foundation mengambil alih pengembangan Eclipse lebih lanjut.

2. Arsitektur

Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah *kernel*. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari *plug-in* yang sudah dipasang (diinstal). Ini merupakan basis dari Eclipse yang dinamakan *Rich Client Platform* (RCP). Berikut ini adalah komponen yang membentuk RCP:

- a. *Core platform*
- b. OSGi
- c. SWT (*Standard Widget Toolkit*)
- d. JFace
- e. *Eclipse Workbench*

Secara standar Eclipse selalu dilengkapi dengan JDT (*Java Development Tools*), *plug-in* yang membuat Eclipse kompatibel

untuk mengembangkan program Java, dan PDE (*Plug-in Development Environment*) untuk mengembangkan *plug-in* baru. Eclipse beserta *plug-in*-nya diimplementasikan dalam bahasa pemrograman Java.

Konsep Eclipse adalah IDE adalah

- a. terbuka (*open*),
- b. mudah diperluas (*extensible*) untuk apa saja, dan
- c. tidak untuk sesuatu yang spesifik.

Eclipse tidak saja untuk mengembangkan program Java, tetapi juga untuk

berbagai macam keperluan. Perluasan apapun cukup dengan menginstal *plug-in* yang dibutuhkan. Apabila ingin mengembangkan program C/C++ maka telah terdapat *plug-in* CDT (*C/C++ Development Tools*) yang dapat dipasang di Eclipse untuk Eclipse menjadi perangkat untuk pengembangan C/C++.

Pengembangan secara visual bukan hal yang tidak mungkin oleh Eclipse, *plug-in* UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat *plug-in* sesuai dengan keinginannya. Salah satu situs yang menawarkan *plug-in* yang gratis adalah *Eclipse downloads by project*.

3. Histori (versi-versi) Eclipse

Sejak tahun 2006, Eclipse Foundation mengkoordinasikan peluncuran Eclipse secara rutin dan simultan yang dikenal dengan nama *Simultaneous Release*. Setiap versi peluncuran terdiri dari Eclipse Platform dan juga sejumlah proyek yang terlibat dalam proyek Eclipse. Tujuan sistem ini adalah untuk menyediakan distribusi Eclipse dengan fitur-fitur dan versi yang terstandarisasi. Hal ini juga dimaksudkan untuk mempermudah *deployment* dan *maintenance* untuk sistem enterprise, serta untuk kenyamanan.

Peluncuran simultan dijadwalkan pada bulan Juni setiap tahunnya, yang ditunjukkan pada tabel 2.3.

Tabel 2.3 Peluncuran Simultan Eclipse

Kode Peluncuran	Tanggal Peluncuran	Platform	Nama Proyek
Eclipse 3.0	28 Juni 2004	3.0	
Eclipse 3.1	28 Juni 2005	3.1	
Callisto	30 Juni 2006	3.2	Callisto Projects
Europa	29 Juni 2007	3.3	Europa Projects
Ganymede	25 Juni 2008	3.4	Ganymede Projects
Galileo	24 Juni 2009	3.5	Galileo Projects
Helios	23 Juni 2010	3.6	
Juno	Juni 2012	4.2	

2.5.2 Android SDK

Android SDK (*Software Development Kit*) merupakan sebuah *tools* yang diperlukan untuk mengembangkan aplikasi berbasis Android menggunakan bahasa pemrograman Java. Pada saat ini Android SDK telah menjadi alat bantu dan API (*Application Programming Interface*) untuk mengembangkan aplikasi berbasis Android. Android SDK dapat anda lihat dan unduh pada situs resminya, yaitu <http://www.developer.android.com>. Android SDK bersifat gratis dan bebas anda distribusikan karena Android bersifat *open source*.

2.5.3 SQLite

1. Penjelasan Umum

SQLite merupakan sebuah system manajemen basis data relasional yang bersifat ACID - compliant dan memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C. SQLite merupakan proyek yang bersifat public domain yang dikerjakan oleh D. Richard Hipp. Tidak seperti pada paradigma client-server umumnya, inti SQLite bukanlah sebuah sistem yang mandiri yang berkomunikasi dengan sebuah program,

melainkan sebagai bagian integral dari sebuah program secara keseluruhan. Sehingga protokol komunikasi utama yang digunakan adalah melalui pemanggilan API secara langsung melalui bahasa pemrograman. Mekanisme seperti ini tentunya membawa keuntungan karena dapat mereduksi *overhead*, *latency times*, dan secara keseluruhan lebih sederhana. Seluruh elemen basisdata (definisi data, tabel, indeks, dan data) disimpan sebagai sebuah file. Kesederhanaan dari sisi desain tersebut bisa diraih dengan cara mengunci keseluruhan file basis data pada saat sebuah transaksi dimulai.

2. Fitur

SQLite mengimplementasikan hampir seluruh elemen-elemen standar yang berlaku pada SQL-92, termasuk transaksi yang bersifat *atomic*, konsistensi basisdata, isolasi, dan durabilitas (dalam bahasa Inggris lebih sering disebut ACID), trigger, dan kueri-kueri yang kompleks. Tidak ada pengecekan tipe sehingga data bisa dientrikan dalam bentuk string untuk sebuah kolom bertipe integer. Beberapa kalangan melihat hal ini sebagai sebuah inovasi yang menambah nilai guna dari sebuah basisdata, utamanya ketika digunakan dalam bahasa pemrograman berbasis script (PHP, Perl), sementara kalangan lain melihat hal tersebut sebagai sebuah kekurangan. Beberapa proses ataupun thread dapat berjalan secara bersamaan dan mengakses basisdata yang sama tanpa mengalami masalah. Hal ini disebabkan karena akses baca data dilakukan secara paralel. Sementara itu akses tulis data hanya bisa dilakukan jika tidak ada proses tulis lain yang sedang dilakukan; jika tidak, proses tulis tersebut akan gagal dan mengembalikan kode kesalahan (atau bisa juga secara otomatis akan mencobanya kembali sampai sejumlah nilai waktu yang ditentukan habis). Hanya saja ketika sebuah tabel temporer dibuat, mekanisme penguncian pada proses multithread akan menyebabkan masalah. Update yang terkini (versi 3.3.4) dikatakan telah memperbaiki masalah ini. Sebuah program yang mandiri dinamakan *sqlite* disediakan dan bisa digunakan untuk mengeksekusi kueri dan manajemen file-file

basisdata SQLite. Program tersebut juga merupakan contoh implementasi penulisan aplikasi yang menggunakan pustaka SQLite

2.5.4 Bluestack

BlueStack adalah suatu aplikasi yang memungkinkan untuk menjalankan aplikasi Android di Windows PC atau disebut dengan emulator. Emulator gratis ini mampu menjalankan aplikasi Android di PC tanpa modifikasi sedikitpun.

Dengan hadirnya aplikasi ini bisa dibidang memperluas pasar bagi developer aplikasi Android karena sekarang mereka dapat memasarkan aplikasinya di Windows PC. BlueStacks juga memperkenalkan Cloud Connect, sebuah layanan cloud computing yang memungkinkan sebuah Windows PC untuk terkoneksi langsung dengan perangkat mobile.

Berikut adalah beberapa fitur kunci dari "BlueStacks":

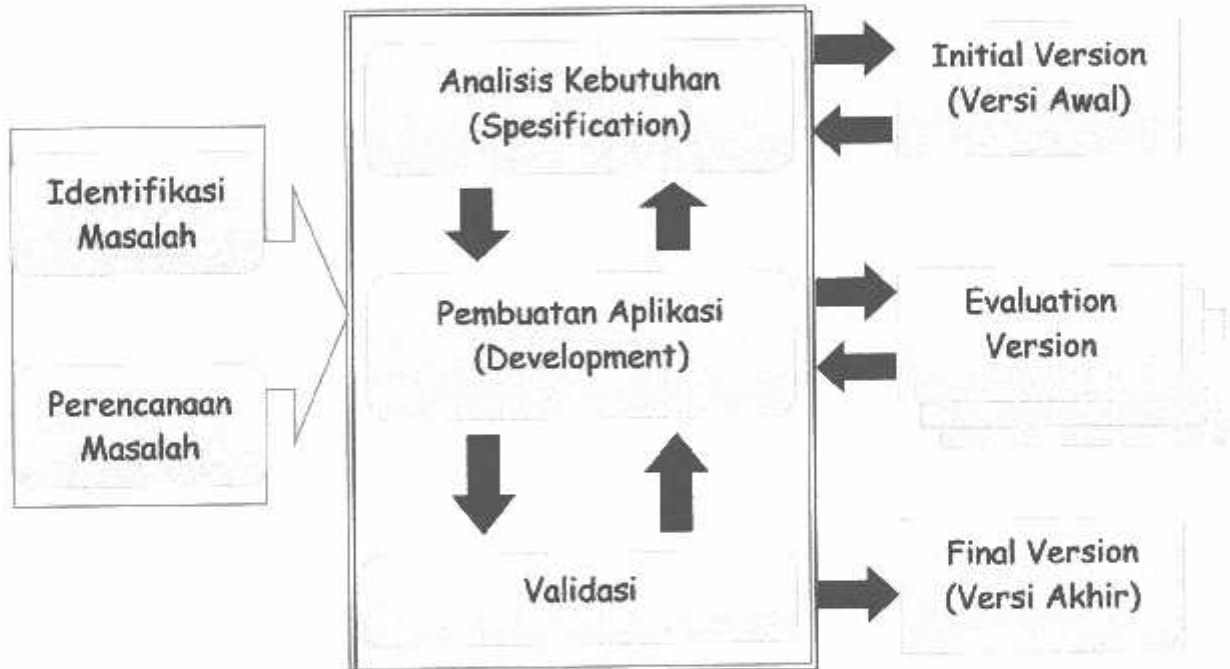
- a. Aplikasi berjalan dengan lancar dan pada mode full screen.
- b. Dapat memindahkan aplikasi favorit anda ke pc dengan menggunakan BlueStack Cloud Connect.
- c. Dapat menjalankan 10 aplikasi pre-loaded.
- d. Dapat menginstal hingga 26 lebih banyak.

Emulator ini menggunakan teknologi Layercake, yaitu suatu teknologi yang mengeksploitasi akselerator hardware untuk meningkatkan performa games Android di Windows. Awalnya, teknologi ini akan memberi kemampuan untuk menjalankan aplikasi di hardware yang sebenarnya tidak ditujukan untuk menjalankannya. Pada dasarnya mengkonversi ARM ke x86 untuk menjalankan aplikasi tersebut.

BAB III ANALISIS DAN PERANCANGAN

3.1 Pendahuluan

Analisis dan perancangan aplikasi adalah merupakan tahapan perencanaan dari seluruh proses dalam membangun aplikasi *monitoring responses server*. Analisis adalah merupakan suatu kegiatan untuk memaparkan suatu peristiwa yang dihadapi, sedangkan perancangan merupakan solusi yang dihasilkan untuk memenuhi dan memecahkan suatu masalah. Diagram blok dalam analisis dan perancangan secara keseluruhan pada aplikasi *monitoring responses server* ini dapat dilihat dalam gambar 3.1.



Gambar 3.1. Diagram Blok Perencanaan

3.2 Analisis

3.2.1 Deskripsi Aplikasi

Program aplikasi *monitoring responses server* berbasis android, bertujuan untuk memberikan informasi tentang keadaan atau status suatu server kepada administrator agar dapat melihat keadaan server dalam

kondisi *up* atau *down* meskipun sang administrator tidak sedang berada di tempat kerja.

Objek atau status yang terdapat pada program aplikasi ini, tampilkan dalam bentuk visual dan notifikasi, sehingga dapat memberitahu pengguna setiap saat.

3.2.2 Analisis Jenis Aplikasi

Jenis aplikasi yang digunakan dalam proses pembuatan program aplikasi *monitoring responses* server ini menggunakan bahasa pemrograman java dengan berorientasi berbasis objek, yaitu dengan menggunakan *graphical user interface* (GUI) dan coding untuk menjalankan setiap fitur yang ada pada program aplikasi tersebut sehingga memungkinkan bagi pengguna bisa dengan mudah untuk menggunakannya.

3.2.3 Sasaran Pengguna Aplikasi

Sasaran dalam penggunaan Aplikasi *monitoring responses* server ini ditujukan sebagai media untuk memberi informasi bagi administrator untuk dapat mengetahui informasi mengenai keadaan server yang dikelolanya dengan menggunakan media digital.

3.2.4 Analisis Sistem Aplikasi

Aplikasi *monitoring responses* server ini memiliki beberapa fungsi sistem, antara lain :

- a. Memberikan informasi tentang keadaan konektivitas suatu server (Server WEB)
- b. Notifikasi dikirim kepada pengguna dengan menampilkan informasi apabila keadaan server tersebut dalam kondisi *up* maka notifikasi berwarna hijau, dan apabila server mengalami kondisi *down* maka notifikasi yang ditampilkan berwarna merah.

3.2.5 Analisis Kebutuhan

Kebutuhan yang digunakan untuk membuat aplikasi *monitoring responses* server, meliputi perangkat lunak dan perangkat keras yang

dihubungkan dengan sistem operasi. Perangkat keras yang digunakan untuk membuat program aplikasi *monitoring responses* server ditunjukkan dalam tabel 3.1.

Tabel 3.1 Analisa kebutuhan perangkat keras (*Hardware*)

Jenis Hardware	Jumlah	Spesifikasi		Keterangan
		Minimum	Digunakan	
PC / Notebook	1	- Proc. Intel P IV 2,5 GHz - RAM 512 MB - HDD 40 GB - VGA onboard (640x480)	- Proc. AMD A6 1,5 GHz - RAM 4 GB - HDD 500 GB - VGA ATI Radeon 1773 MB (1366x768)	PC / Notebook digunakan untuk membangun aplikasi <i>monitoring responses</i> server
Smartphone Android	1	- Processors ARM v6 600 MHz - RAM 128 MB - Memory Internal 256 MB	- Processors ARM v6 600 MHz - RAM 128 MB - Memory Internal 256 MB (Sony Ericsson XPERIA X8)	Smartphone Android digunakan sebagai perangkat untuk menjalankan aplikasi

Sedangkan kebutuhan perangkat lunak yang digunakan untuk membuat program aplikasi *monitoring responses* server ditunjukkan dalam tabel 3.2.

Tabel 3.2 Analisa kebutuhan perangkat lunak (*Software*)

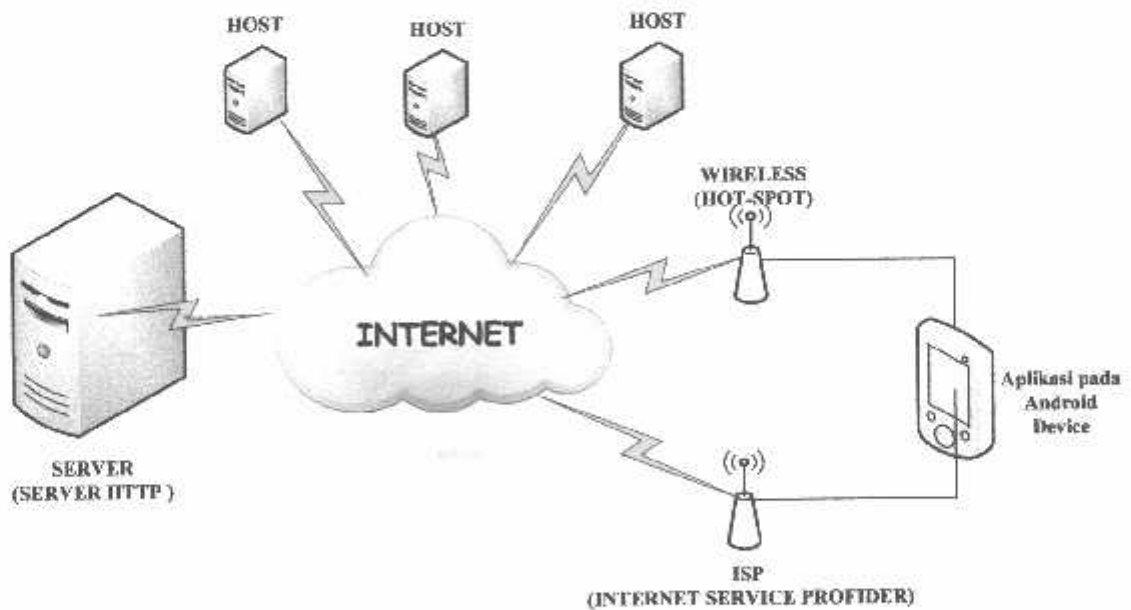
No	Kebutuhan Software	Software yang Digunakan	Keterangan
1	Operating Sistem PC / Notebook	Windows 7 Ultimate Service Pack 1	
2	Operating Sistem Smartphone Android (Mobile Device)	Android versi 2.3.7 Gingerbread	Untuk kestabilan system dan dapat mendukung berbagai aplikasi
3	Pembuatan Aplikasi	- Eclipse Helios SR 2 versi 3.6.2 - ADT (Android Development Kit) versi 20.0.3	Eclipse, ADT, JDK serta Android SDK harus terinstal untuk memulai pembuatan aplikasi berbasis

		<ul style="list-style-type: none"> - JDK (Java Development Kit) versi Android 7 - Android SDK versi 21.1.0 	
4	Emulator	<ul style="list-style-type: none"> - BlueStack HD - Eclipse Android Emulator 	Emulator digunakan untuk menjalankan program atau melakukan testing terlebih dahulu, sebelum aplikasi diinstal pada Smartphone Android
5	Pembuatan Laporan	<ul style="list-style-type: none"> - Microsoft Word 2013 - Microsoft Visio 2007 	Ms. Word digunakan untuk menyusun laporan dan Ms. Visio digunakan untuk membuat flowchart dan blok diagram

3.3 Perancangan

Desain sistem merupakan awal dari sebuah tahap perancangan program yang mana alur dan rangka dirancang disini. Desain sistem dari program ini meliputi, blok diagram, dan flowchart.

3.3.1 Blok Diagram



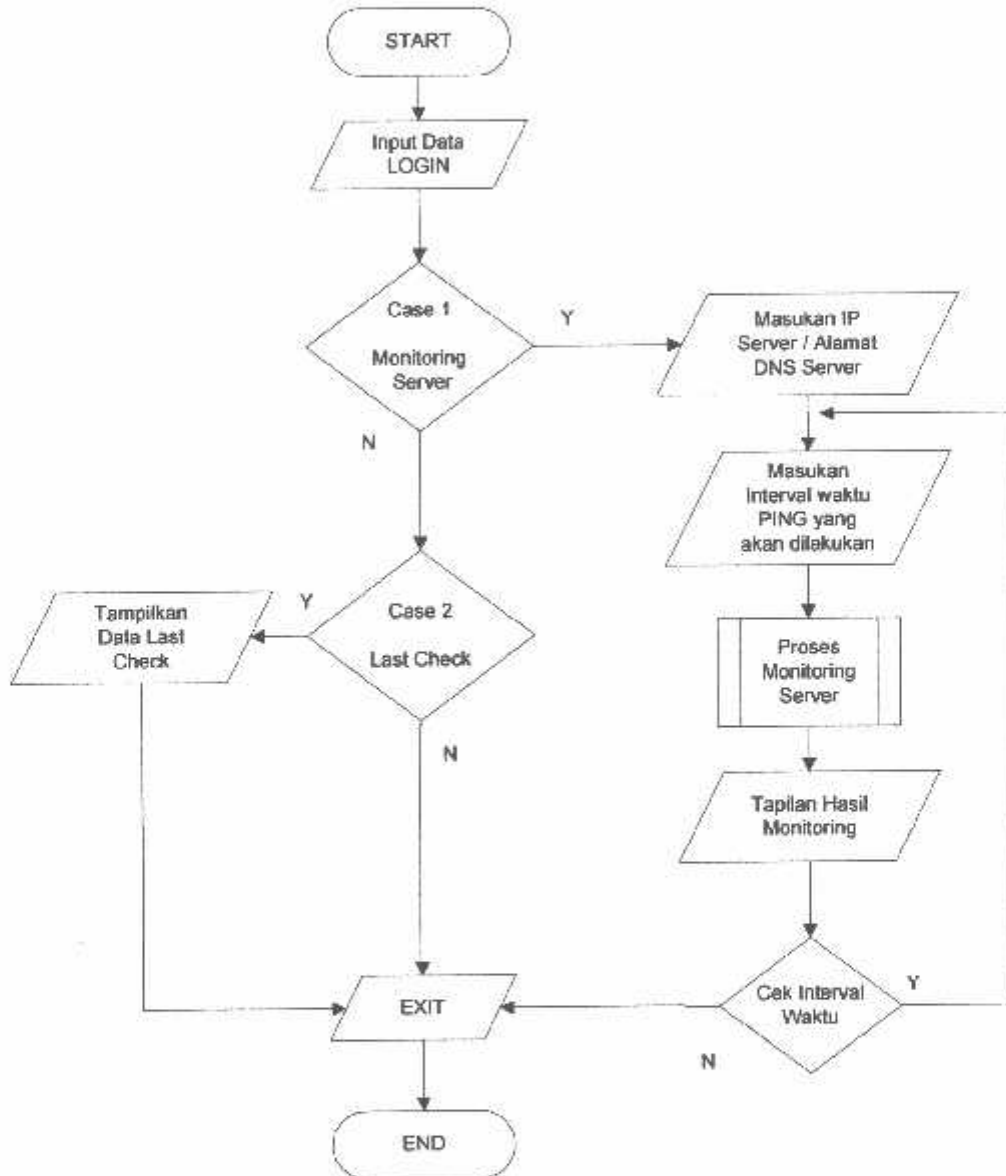
Gambar 3.2 Blok Diagram

Dalam gambar 3.2 mempunyai keterangan tentang alur kerja *monitoring* server sebagai berikut:

- a. Komputer server merupakan target yang akan dimonitoring
- b. Aplikasi berjalan pada *mobile* Android dan membutuhkan koneksi wireless atau melalui ISP untuk melakukan *monitoring (ping)*.

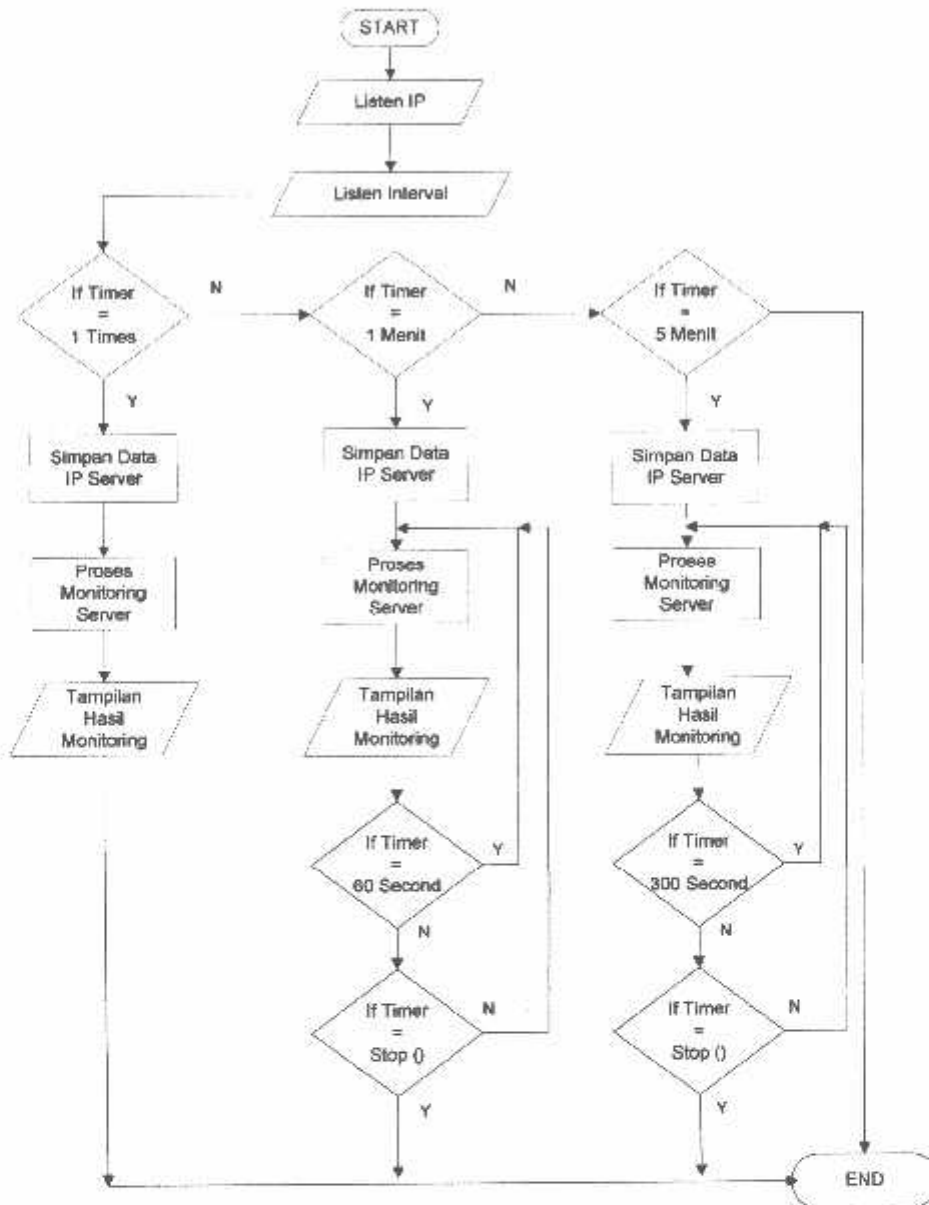
3.3.2 Flowchart

1. Flowchart Sistem



Gambar 3.3 Flowchart Sistem

2. Flowchart Berdasarkan Interval



Gambar 3.4 Flowchart Sistem *Timer (Interval)*

3.3.3 Algoritma

Dari flowchart yang ditunjukkan dalam gambar 3.3, dapat dijelaskan algoritma monitoring server sebagai berikut:

- User *login*, memasukan nama dan *password*.
- Kemudian pada tampilan utama terdapat pilihan yaitu *Monitoring*, dan *Last check*.
- Apabila *user* memilih *monitoring server*, maka *user* memasukan alamat host atau *IP Address* server yang akan dituju.

- d. Masukan interval waktu apabila ingin melakukan *ping* terhadap server secara berkala.
- e. Aplikasi akan melakukan eksekusi, sesuai data yang telah dimasukan.
- f. Kemudian menampilkan hasil monitoring dari server berupa simbol *up* atau *down*.
- g. Apabila *user* memilih *last check*, maka ditampilkan waktu kapan terakhir melakukan koneksi *ping* terhadap server.
- h. Kemudian jika memilih *exit*, maka program akan ditutup.

3.3.4 Sistem kerja *Monitoring Server*

Dalam sistem kerja monitoring server, parameter yang dipakai adalah hasil dari *ping* terhadap server yang hendak dimonitoring. Berikut *parameter* yang akan digunakan : “**ping -c 1**” + **hostip**

Keterangan:

- a. *Ping*: memanggil program *ping* untuk dieksekusi.
- b. *-c*: menentukan jumlah *echo* setelah permintaan dan *response* setelah *ping* selesai.
- c. *Hostip*: *host* atau *ip address* yang menjadi target *ping*.

3.4 Struktur Navigasi Aplikasi

Struktur navigasi pada aplikasi *monitoring responses* server terdapat beberapa menu, dan beberapa tombol yang digunakan untuk melakukan eksekusi sesuai dengan fungsi dari tombol tersebut. Menu – menu pada aplikasi *monitoring responses* server, meliputi

a. Menu Login

Pada menu ini terdapat kolom isian, yaitu nama dan *password*, kemudian terdapat tombol untuk *login* yang berfungsi untuk mengoreksi nama dan *password*, dan berfungsi untuk membuka menu selanjutnya. Menu *login* yang akan digunakan ditunjukkan dalam gambar 3.5.

Gambar 3.5 Menu Login

b. Menu Utama

Pada menu utama, terdapat beberapa pilihan tombol navigasi yang mempunyai fungsi masing – masing, kemudian dapat dipilih oleh pengguna sesuai dengan kebutuhan. Menu utama yang akan digunakan pada pembuatan program aplikasi ditunjukkan dalam gambar 3.6.

Gambar 3.6 Halaman Menu Utama

Keterangan menu utama:

1. Tombol *monitoring* server akan mengarahkan pengguna pada halaman *monitoring*.
2. Tombol *last check* berfungsi untuk menampilkan halaman *last check*.
3. Tombol *help* digunakan untuk memberi penjelasan tentang cara penggunaan dari program aplikasi.

c. Menu *Monitoring Server*

Halaman *monitoring server* ini terdapat kolom isian yaitu alamat *ip server* atau *dns server*, dan interval waktu. Dan terdapat tombol yang berfungsi untuk menjalankan eksekusi dari data yang telah diisi pada kolom isian. Menu *monitoring server* ditunjukkan dalam gambar 3.7.

The image shows a graphical user interface for monitoring a server. It consists of a rectangular frame containing several elements:

- Two input fields at the top, one labeled 'IP Server' and one labeled 'Interval Waktu'.
- An 'OK' button located to the right of the 'Interval Waktu' field.
- A large rectangular area at the bottom, labeled '(Text View hasil ping)', intended for displaying the results of the ping command.

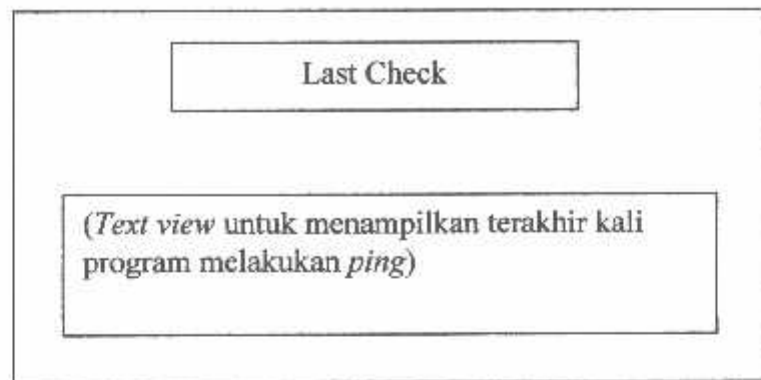
Gambar 3.7 Menu *Monitoring Server*

Keterangan menu *monitoring server*:

1. Pada kolom isian *ip server*, di masukan *ip server* yang akan menjadi target *monitoring*.
2. Pada kolom isian interval waktu, pengguna memilih waktu interval melakukan *ping*, dan interval waktu telah ditentukan oleh sistem.
3. Tombol ok digunakan untuk melakukan eksekusi,
4. *Text view* digunakan untuk menampilkan hasil dari *ping* yang telah dilakukan.

d. Menu *Last Check*

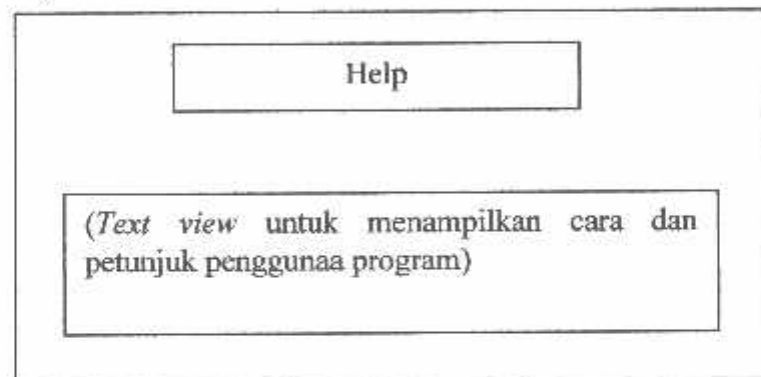
Menu *last check* digunakan untuk menampilkan waktu terakhir kali program aplikasi melakukan *ping* terhadap server. Menu *last check* ditunjukkan dalam gambar 3.8.



Gambar 3.8 Menu *Last Check*

c. Menu *Help*

Menu *help* digunakan untuk memberi penjelasan tentang cara penggunaan dari program aplikasi. Perancangan Menu *help* ditunjukkan dalam gambar 3.9.



Gambar 3.9 Menu *Help*

3.5 Setting pada Android dan Lingkungan Kerja Eclipse

3.5.1 Setup Handset Smartphone Android

Tahap ini dilakukan untuk mencoba hasil pembuatan program ke handset android. Pertama kabel data *USB*, dan menginstall driver handset Android agar terdeteksi oleh komputer. Hampir semua handset android didukung oleh java Eclipse tergantung *platform* atau versi minimum android yang akan digunakan, dan pastikan terdapat memori *SD Card* yang terpasang untuk melakukan instalasi melalui memori eksternal. Setelah itu setting pada handset Android seperti dibawah ini :

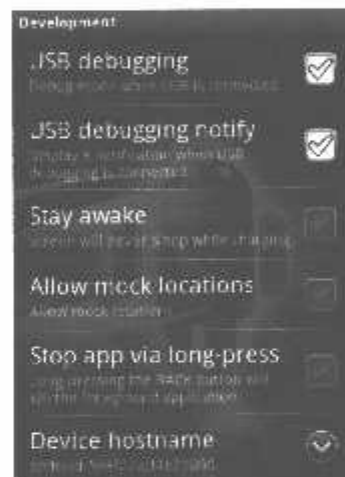
- a. Masuk pada *home screen*
- b. Pilih *Setting > applications*

- c. Pada *Unknown sources* di *ceklis*, dapat dilihat pada gambar 3.10.



Gambar 3.10 Gambar *Setting*

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



Gambar 3.11 Pengaturan untuk *USB Debugging*

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

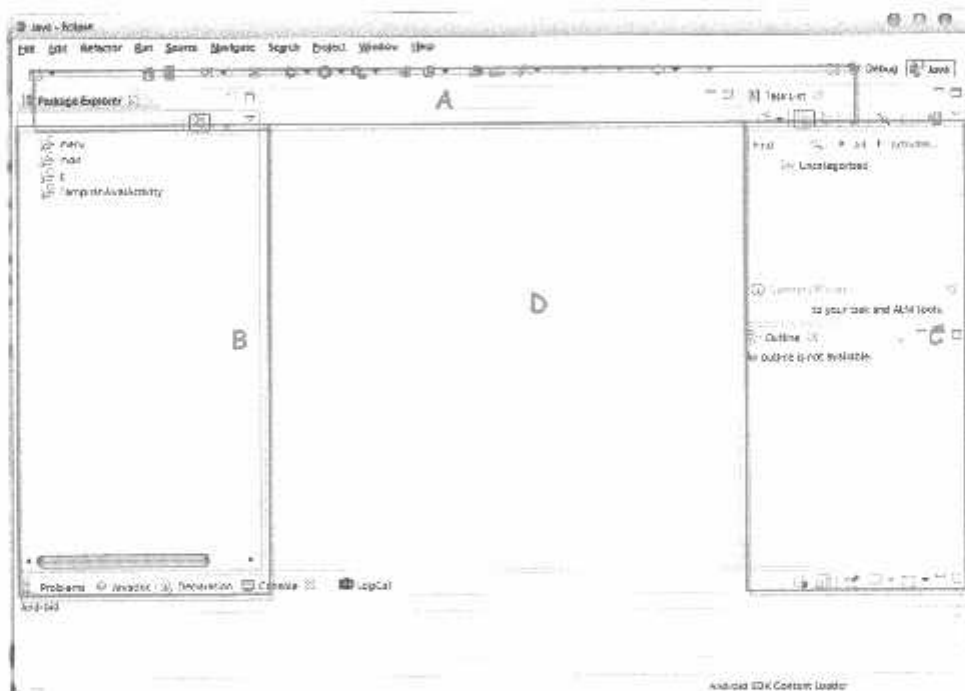
Sampai disini berarti *handset* android telah siap untuk digunakan untuk mencoba aplikasi yang telah dibuat dengan menggunakan Java Eclipse.

3.5.2 Lingkungan Kerja pada Eclipse

Sebelum melakukan pembuatan program, kita harus mengenal terlebih dahulu lingkungan kerja pada Eclipse yaitu:

1. Halaman Utama

Halaman utama pada Eclipse adalah awal membuka java Eclipse, dan terdapat beberapa komponen yang digunakan untuk membuat suatu program aplikasi. Halaman utama pada Eclipse ditunjukkan dalam gambar 3.12.



Gambar 3.12 Halaman Utama Eclipse

Keterangan :

- a. **Menu bar** – komponen yang terdapat pada menu bar yaitu:
 - *Project*, digunakan untuk membuat proyek baru, java class untuk *coding* program, dan XML untuk desain *interface* (GUI) pada Android.

- *Android SDK Manager*, berfungsi untuk instalasi *platform android* agar program aplikasi yang akan dibuat dapat digunakan pada *smartphone Android*.
 - *Execute*, digunakan untuk menjalankan program yang telah dibuat pada emulator.
- b. **Package Explorer** – Package explorer berfungsi untuk meletakkan project yang akan dibuat, dan project yang telah diimport kedalam eclipse.
 - c. **Outline** – Outline digunakan untuk meletakkan komponen – komponen yang ada pada desain *interface* (GUI) seperti *text field*, *button*, dan lainnya. Pada properties digunakan untuk inisialisasi komponen – komponen yang terdapat pada GUI.
 - d. **Work field** – Pada area work field digunakan untuk melakukan desain GUI, dan melakukan *coding* sesuai dengan program agar setiap komponen dapat berfungsi dengan baik.

2. Emulator Bluestacks (*Android Device*)

Android Devices (Emulator Bluestacks) adalah konfigurasi dari eclipse sehingga kita dapat menjalankan perangkat Android sesuai model yang dipilih, misal Android 1.5 atau 2.2. Untuk dapat menjalankan emulator Bluestacks, Anda harus terlebih dahulu memiliki Android SDK dan software tambahan yaitu Bluestacks.

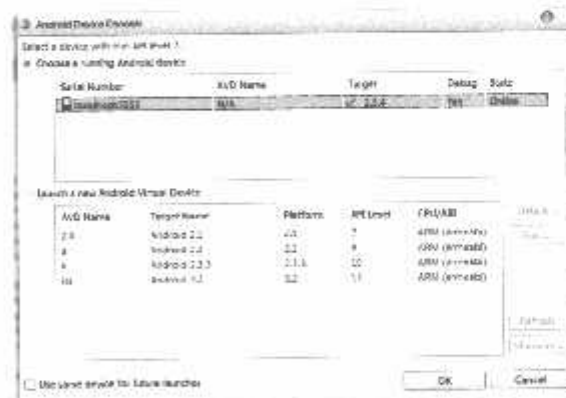
Setiap *Android Emulator Device* terdiri dari:

1. Sebuah profil perangkat keras. Anda dapat mengatur opsi untuk menentukan fitur hardware emulator. Misalnya, Anda dapat menentukan apakah menggunakan perangkat kamera, apakah menggunakan keyboard QWERTY fisik atau tidak, berapa banyak memori internal, dan lain-lain.
 2. Sebuah pemetaan versi Android. Anda dapat menentukan versi dari platform Android akan berjalan pada emulator.
 3. Pilihan lainnya. Anda dapat menentukan skin yang ingin Anda gunakan pada emulator, yang memungkinkan Anda
-

menentukan dimensi layar, tampilan, dan sebagainya. Anda juga dapat menentukan SD Card virtual untuk digunakan dengan di emulator.

Emulator juga digunakan untuk menjalankan aplikasi yang telah dibuat pada eclipse dengan tujuan untuk testing program apakah sudah berjalan dengan baik atau belum. Ini sangat bermanfaat apabila kita belum menggunakan handset langsung, karena Emulator Bluestacks ini telah terintegrasi dengan baik. Untuk dapat melakukan testing program pada emulator android, ada beberapa persiapan yang harus dilakukan, agar program yang telah dibuat pada eclipse dapat berjalan dengan baik. Berikut ini adalah persiapan untuk konfigurasi emulator Bluestacks pada eclipse:

- a. Instal dan kemudian buka Android Emulator Bluestacks.
- b. Pada eclipse program yang telah selesai dibuat, diklik kanan project yang di buat kemudian pilih **Run As > Android Application**. Dan akan muncul pengaturan untuk emulator android.
- c. Pada radio button, pilih **Choose a running Android Device** dan pada kolom **AVD**, pilih **AVD** yang tersedia > beri tanda cek (✓) pada keterangan **use same device for future launches** agar program yang akan dijalankan langsung terdeteksi pada emulator, tanpa harus melakukan setup **AVD** dari awal > klik **Ok**. Pengaturan **AVD** pada eclipse ditunjukkan dalam gambar 3.13.



Gambar 3.13 Android Virtual Device (AVD)

- d. Setelah konfigurasi selesai dilakukan dan melakukan *start*, maka akan muncul emulator android. Kemudian emulator android muncul dan dapat digunakan untuk melakukan testing program sebelum di-*install* pada *smartphone* android yang sebenarnya. Emulator yang digunakan adalah emulator Bluestacks yang ditunjukkan dalam gambar 3.14.



Gambar 3.14 Emulator BlueStacks

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1 Pendahuluan

Tahapan implementasi dan pengujian terhadap aplikasi *monitoring responses* server, dilakukan setelah seluruh proses pengumpulan data dianggap telah mencukupi untuk dilanjutkan ke dalam proses pembuatan aplikasi dengan menggunakan perangkat lunak *Eclipse Helios*, sebagai media dalam pembuatan desain *interface* dan *coding* agar aplikasi *monitoring responses* server dapat berjalan sesuai fungsi dan tujuan. Pada tahap implementasi bertujuan untuk menerapkan sistem, fungsi lainnya adalah untuk mengetahui sejauh mana keberhasilan dari rancangan yang telah dibuat.

4.2 Implementasi

Pada proses implementasi ke dalam aplikasi *monitoring responses* server, membutuhkan beberapa tahapan-tahapan, yang meliputi :

1. Memasukkan Data Visual

Hasil dari proses pengumpulan data yang berupa visual , dimasukkan ke dalam menu atau halaman pada *Eclipse* yang disesuaikan dengan setiap halaman yang mempunyai fungsi sendiri.

2. Membuat dan mengatur komponen yang akan digunakan.

Membuat dan mengatur komponen yang akan digunakan, seperti layout menu *xml* yang berfungsi untuk menampilkan data secara visual dengan pengguna dan *java class* digunakan untuk membuat kode atau *script* agar menu *xml* data berfungsi sesuai dengan tujuan.

4.2.1 Pembuatan Project Pada Eclipse

Pembuatan proyek ini, digunakan sebagai awal proses untuk membuat aplikasi *monitoring responses* server. Tahapan-tahapan yang diperlukan dalam pembuatan proyek baru untuk membuat aplikasi *monitoring responses* server, antara lain :

1. Pastikan *software* yang dibutuhkan untuk membuat aplikasi sudah terpasang pada computer / pc. *Software* yang dibutuhkan seperti

Eclipse Helios, ADT (*Android Development Kit*), JDK (*Java Development Kit*), Android SDK.

2. Jalankan Eclipse lalu menuju ke *file > New > Android Application Project*, seperti yang ditunjukkan dalam gambar 4.1.



Gambar 4.1 Membuat Proyek Baru di Eclipse

3. Pada *new android application*, isikan nama aplikasi pada *Application Name*, kemudian pilih SDK minimal yang digunakan, target SDK, Compile pada emulator, dan tema sesuai yang diinginkan setelah itu klik Next. Contoh proyek yang akan dibuat ditunjukkan dalam gambar 4.2.



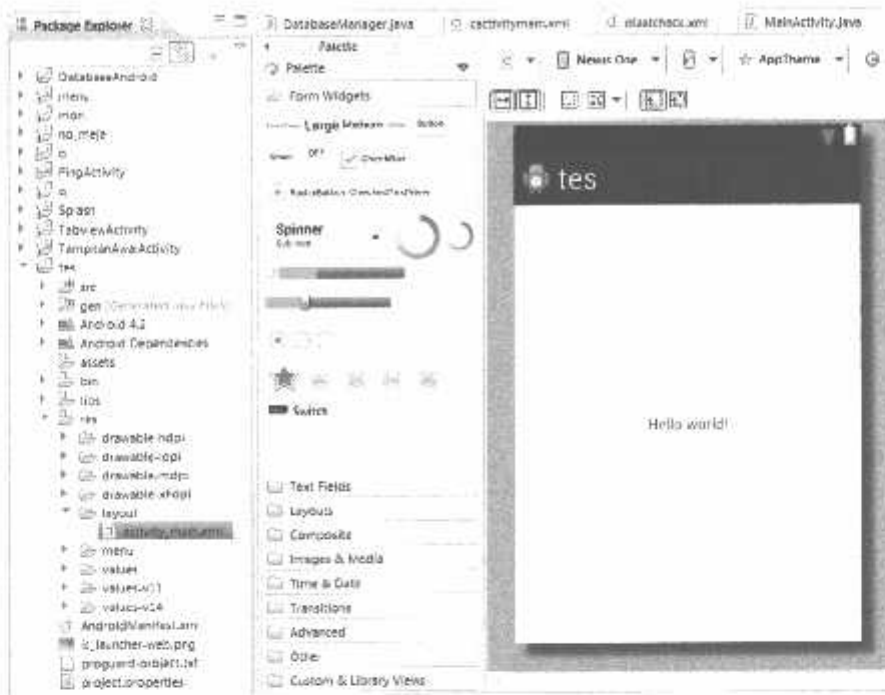
Gambar 4.2 Nama Aplikasi Android

4. Kemudian akan muncul kotak dialog *Configure Project*, klik *Next* dan muncul kotak dialog untuk memilih icon yang akan digunakan pada *smartphone* android sebagai simbol atau *launcher* aplikasi yang dibuat, setelah selesai klik *Next*. Sehingga muncul kotak dialog *Create Activity* yang ditunjukkan dalam gambar 4.3, klik *Next*.



Gambar 4.3 *Activity* yang akan digunakan

5. Kemudian akan muncul kotak dialog *New Blank Activity*, karena pada dialog *Create Activity* sebelumnya telah dipilih *blank activity*. Tekan *Finish*, dan proyek yang telah dibuat akan muncul pada *Package Explorer* dalam Eclipse. Proyek baru yang telah dibuat ditunjukkan dalam gambar 4.4.

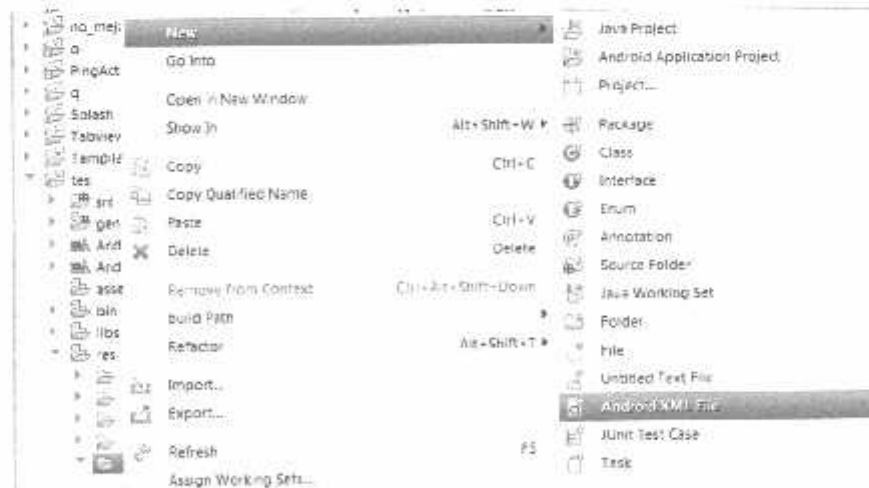


Gambar 4.4 Tampilan awal *Blank Activity*

4.2.2 Pembuatan Menu Login

Tahapan yang diperlukan dalam pembuatan menu login, antara lain:

1. Membuat xml baru dengan cara masuk ke folder *res > layout*, klik kanan folder *layout > new > Android XML File*, seperti yang ditunjukkan dalam gambar 4.5.



Gambar 4.5 Membuat File XML baru

2. Setelah itu akan muncul kotak dialog, isikan nama file xml tersebut, contoh : LoginForm.xml. Klik *Finish* apabila telah selesai.
3. Kemudian *drag and drop* content pada Pallete, yaitu:
 - *TextView*, yang berfungsi untuk menampilkan text pada xml.
 - *ImageView*, berfungsi untuk menampilkan gambar. Gambar – gambar yang akan ditampilkan pada xml disimpan dalam folder *res > drawable*.
 - *EditText*, berfungsi untuk memasukan data. Pada menu login, dimasukan 2 buah *EditText* untuk memasukan nama dan password.
 - *Button*, digunakan untuk melakukan eksekusi. *Button* pada menu login berfungsi untuk mencocokkan data nama dan password, kemudian apabila data yang dimasukan cocok maka button akan menavigasi pengguna untuk masuk ke halaman atau menu selanjutnya.

Setelah dimasukan semua content yang dibutuhkan, ubahlah dan modifikasi menu login sesuai dengan keinginan. Menu login pada proyek kali ini ditunjukan dalam gambar 4.6.



Gambar 4.6 Pembuatan Menu Login

4.2.3 Pembuatan Menu Utama

Setelah selesai membuat menu login, langkah selanjutnya membuat menu utama. Dalam membuat menu utama ada beberapa tahapan, yaitu:

1. Membuat xml baru, kemudian akan muncul kotak dialog, isikan nama file xml tersebut, contoh : Menu.xml. Klik *Finish* apabila telah selesai.
2. Kemudian *drag and drop* content pada Pallete, yaitu:
 - *TextView*, yang digunakan untuk menampilkan judul pada menu utama.
 - *Button*, digunakan untuk melakukan navigasi ke menu selanjutnya, sesuai dengan pilihan. *Button* pada menu utama ada 3 buah, yaitu tombol *Monitoring*, tombol *Last Check* dan tombol *Help*.

Setelah memasukan komponen – komponen yang dibutuhkan lakukan modifikasi sesuai dengan keinginan. Menu utama yang digunakan pada proyek ini ditunjukkan dalam gambar 4.7.



Gambar 4.7 Pembuatan Menu Utama

4.2.4 Pembuatan Menu Monitoring

Menu monitoring ini digunakan untuk melakukan *monitoring ping* terhadap server atau *host*. Dan langkah – langkah untuk membuat menu monitoring sebagai berikut:

1. Membuat xml baru, kemudian akan muncul kotak dialog, isikan nama file xml tersebut, contoh : ActivityMain.xml. Klik *Finish* apabila telah selesai.
2. Kemudian *drag and drop* content pada Pallete yang dibutuhkan, yaitu:

- *TextView*, yang digunakan untuk menampilkan label. Ambil 4 buah *textview*, masing – masing akan digunakan untuk menampilkan label untuk *edittext*, menampilkan hasil untuk status *up* dan *down*, dan untuk menampilkan waktu melakukan proses monitoring.
- *EditText*, digunakan untuk memasukkan alamat server atau *host* yang akan dimonitoring.
- *Button*, digunakan untuk melakukan proses ping.
- *ImageView*, berfungsi untuk menampilkan gambar atau image yang ada dalam database maupun dalam folder yang telah disediakan oleh eclipse. Pada menu ini digunakan 2 buah *imageview*, yang berfungsi untuk memberikan informasi keadaan server atau *host* dalam status *up* atau *down*.

Setelah memasukkan komponen – komponen yang dibutuhkan lakukan modifikasi sesuai dengan keinginan. Menu utama yang digunakan pada proyek ini ditunjukkan dalam gambar 4.8.



Gambar 4.8 Pembuatan Menu Monitoring

4.2.5 Pembuatan Menu Last Check

Pembuatan menu *last check* ini digunakan untuk menampilkan data alamat *host* atau server yang telah dimonitoring sebelumnya. Langkah – langkah dalam pembuatan menu *last check* yaitu:

1. Membuat xml baru, kemudian akan muncul kotak dialog, isikan nama file xml tersebut, contoh : LastCheck.xml. Klik *Finish* apabila telah selesai.

2. Kemudian *drag and drop* content pada Pallete, yaitu:

- *TextView*, yang digunakan untuk menampilkan judul pada menu.

Setelah memasukan komponen – komponen yang dibutuhkan lakukan modifikasi sesuai dengan keinginan. Menu *Last check* yang digunakan pada proyek ini ditunjukkan dalam gambar 4.9.



Gambar 4.9 Pembuatan menu Last Check

4.2.6 Pembuatan Menu Help

Pembuatan menu *help* digunakan untuk memberi petunjuk bagi pengguna agar dapat mengenal seluruh komponen yang terdapat dalam aplikasi *monitoring server* ini. Langkah – langkah dalam pembuatan menu *help* yaitu:

1. Membuat xml baru, kemudian akan muncul kotak dialog, isikan nama file xml tersebut, contoh : Help.xml. Klik *Finish* apabila telah selesai.

2. Kemudian *drag and drop* content pada Pallete, yaitu:

- *WebView*, digunakan untuk menampilkan petunjuk kepada pengguna. Penggunaan *WebView* berfungsi agar teks yang ditampilkan tertata dengan rapi seperti tampilan pada HTML.

Setelah memasukan komponen – komponen yang dibutuhkan lakukan modifikasi sesuai dengan keinginan. Menu *Help* yang digunakan pada proyek ini ditunjukkan dalam gambar 4.10.



Gambar 4.10 Pembuatan Menu *Help*

4.2.7 Publish Android Project ke APK

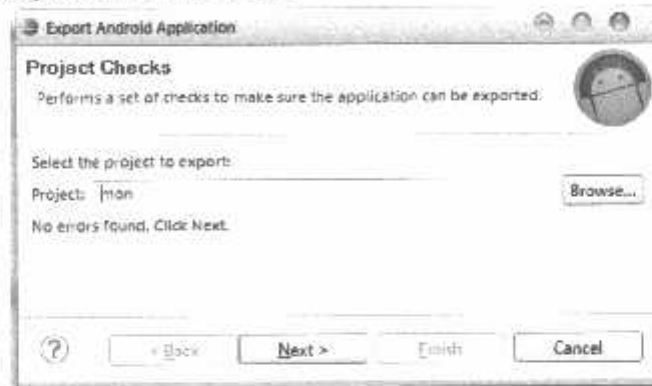
Setelah selesai membuat menu menu yang dibutuhkan dan *coding* telah selesai dilakukan, langkah selanjutnya adalah mem-*publish android project* ke dalam format APK. *Project Android* yang dibuat dapat di-*compile* atau di-*publish* menjadi bentuk APK yang dapat dipasang ke *Handphone* Android. Karena jika langsung *Export to APK* dengan cara biasa dengan menu yang tersedia di Eclipse atau IDE yang dipakai, terkadang APK yang dihasilkan tidak bisa diinstal atau dipasang ke *Handphone* Android. Waktu proses instalasi akan muncul pesan error “**Application Not Installed**“. Maka diperlukan untuk mem-*publish* project android ke APK agar dapat dipasang dan digunakan.

1. Klik kanan Project yang ingin di-*publish*, lalu pilih **Android Tools**, lalu pilih **Export Signed Application Package**. Seperti yang ditunjukan dalam gambar 4.11.



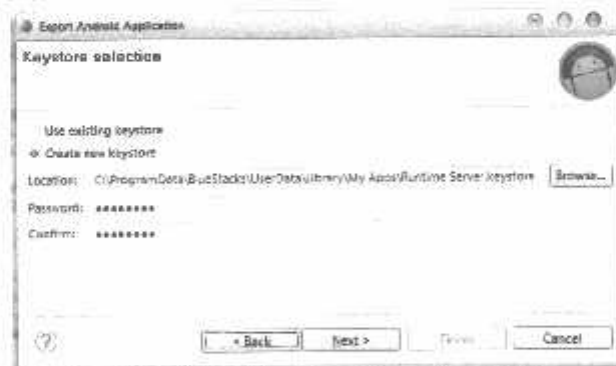
Gambar 4.11 Memulai *Eksport Application Package*

2. Akan muncul tampilan seperti pada gambar 4.12, pilih project yang akan di-eksport dan tekan Next.



Gambar 4.12 Memilih Project

3. *Create new keystore* jika ingin membuat tempat penyimpanan baru untuk *keystore*, klik *Browse* untuk menentukan tempat penyimpanan *keystore*. Beri nama file dengan cara *namafile.keystore*. Kemudian berikan *Password* dan *Confirm* masing-masing minimal 6 digit agar dapat lanjut ke proses selanjutnya, seperti dalam gambar 4.13. Setelah itu klik *Next*.



Gambar 4.13 Membuat *Keystore* Baru

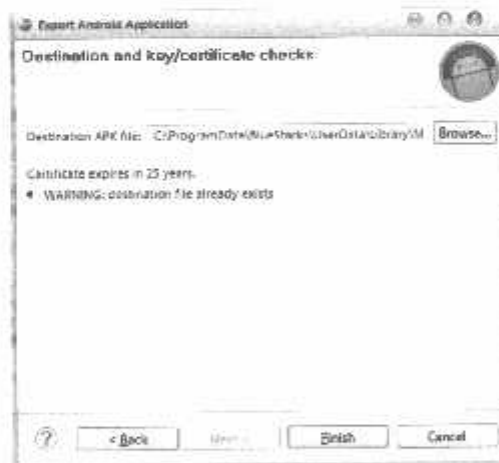
4. Pada *window* (Jendela) *Key Creation* isilah kotak kosong yang tampil pada layar, yang ditunjukkan dalam gambar 4.14. Adapun kotak kosong yang wajib diisi hanya 5 yaitu *Alias*, *Password*, *Confirm*, *Validity*

(years) dan *First and Last Name*. Isi kotak kosong Alias dengan nama file aplikasi, berikan *Password* dan *Confirm* masing-masing minimal 6 digit. Isi *Validity (years)* untuk *Expired* jangka lama minimal 25 tahun. Dan terakhir isi *First and Last Name*. Jika sudah klik *Next*.



Gambar 4.14 Form untuk Membuat Key Baru

5. Pada jendela *Destination and key/certificate checks*, klik *Browse* untuk menentukan letak file .apk. Kemudian beri nama file apk, misal : *Monitoring Runtime Server.apk*. Kemudian klik *Save*, dilanjutkan dengan *Finish*. Peletakan file apk ditunjukkan dalam gambar 4.15.



Gambar 4.15 Destination APK File dan Key

4.3 Pengujian Sistem

Pengujian sistem merupakan tahap uji coba terhadap sistem yang telah dibuat. Pengujian sistem dilakukan bertujuan untuk mengidentifikasi masalah pada sistem bila terjadi kesalahan pada sistem dan bertujuan untuk mengetahui sejauh mana keberhasilan aplikasi ini bisa diterapkan dalam system. Adapun yang diharapkan berjalan pada aplikasi monitoring server berbasis android ini adalah sebagai berikut:

1. Aplikasi mendapatkan hasil ping dan menampilkannya pada layar.
2. Aplikasi dapat menyimpan alamat server atau *host* sebagai *history*.
3. Aplikasi menampilkan notifikasi berupa alert jika server tujuan tidak dapat diakses melalui ping.
4. Aplikasi menampilkan waktu terakhir melakukan ping terhadap server.

Hasil pengujian sistem telah cukup berhasil dalam menampilkan data yang diolah, dan dapat memberikan informasi sesuai dengan tujuan. Aplikasi *monitoring responses* server ini telah diuji pada *Android Virtual Device* (Emulator Android) di Eclipse dan juga pada *smartphone* Android.

4.3.1 Instalasi Aplikasi Pada Smartphone Android

Instalasi aplikasi pada perangkat android, yang pertama dilakukan adalah memindahkan file aplikasi yang telah diubah formatnya menjadi (.apk) ke memori eksternal perangkat android. Langkah selanjutnya dengan menggunakan aplikasi eksplorer (*file explorer*) untuk membuka file yang telah dipindahkan pada memori eksternal dan melakukan instalasi aplikasi. Instalasi aplikasi pada perangkat android ditunjukkan dalam gambar 4.16.



Gambar 4.16 Permission Instalasi Aplikasi Pada Android

4.3.2 Tampilan Login

Tampilan login adalah tampilan yang muncul pada saat pengguna memulai aplikasi, pada menu login, pengguna diwajibkan untuk memasukan nama dan password yang telah ditentukan dalam aplikasi untuk masuk kedalam menu utama. Setelah memasukan nama dan password tekan tombol login untuk masuk ke menu utama aplikasi. Menu login yang dijalankan pada perangkat *smartphone* Android yaitu Sony Xperia X8 dan ditunjukkan dalam gambar 4.17.



Gambar 4.17 Tampilan Login

4.3.3 Tampilan Menu Utama

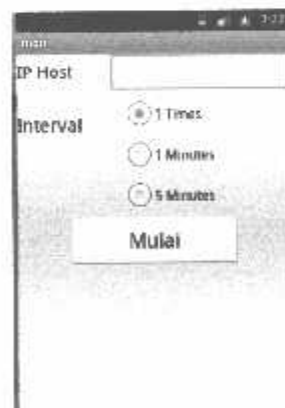
Tampilan menu utama terdiri dari beberapa tombol yang berfungsi untuk melakukan navigasi ke menu selanjutnya, yaitu tombol *runtime*, *last check* dan *Exit* yang akan muncul setelah menekan tombol *option menu*. Tombol *Runtime* digunakan untuk menuju halaman *runtime server*. Tombol *Last Check* digunakan untuk menuju halaman *last check*. Dan tombol *exit* digunakan untuk keluar dari aplikasi. Tampilan menu utama ditunjukkan dalam gambar 4.18.



Gambar 4.18 Tampilan Menu Utama

4.3.4 Tampilan Menu Monitoring

Tampilan menu *Monitoring* digunakan untuk melakukan *monitoring ping* terhadap server atau *host*. Ada beberapa komponen yang terdapat pada menu *monitoring* ini, meliputi *edittext* yang berfungsi untuk memasukkan alamat server atau *host*, *button* mulai yang digunakan untuk melakukan ping berdasarkan alamat yang dimasukkan pada *edittext* sebelumnya. Dan terdapat beberapa *textview* yang berfungsi untuk menampilkan hasil *ping* yang telah diproses, dan menampilkan data alamat server yang telah dimonitoring sebelumnya. Tampilan menu *monitoring* ditunjukkan dalam gambar 4.19.



Gambar 4.19 Menu *Monitoring*

4.3.5 Tampilan Menu Last Check

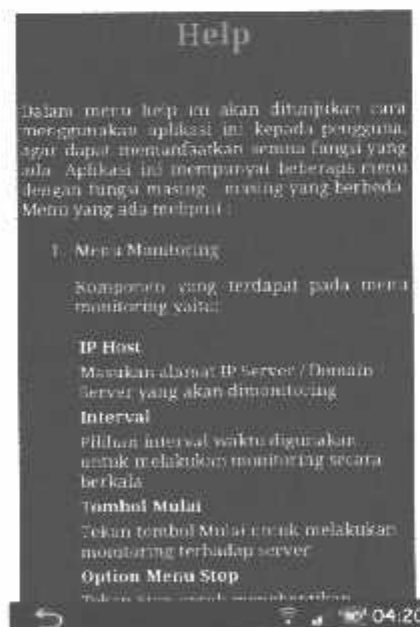
Tampilan menu Last Check digunakan untuk menampilkan alamat server atau host yang telah dilakukan monitoring, kemudian disimpan dalam basis data SQLite dan ditampilkan pada menu last check. Tampilan menu last check ditunjukkan pada gambar 4.20.



Gambar 4.20 Menu *Last Check*

4.3.6 Tampilan Menu Help

Tampilan menu help yang berfungsi untuk memberi petunjuk kepada pengguna, agar dapat menggunakan aplikasi ini dengan baik. Hasil menu help ditunjukkan dalam gambar 4.21



Gambar 4.21 Menu Help

4.3.7 Pengujian Monitoring Server

Pengujian terhadap server perlu dilakukan agar dapat mengetahui informasi tentang keadaan alamat server. Pada menu runtime masukan alamat server yang akan dimonitoring. Kemudian aplikasi akan melakukan monitoring terhadap server dengan kondisi *up* dan *down*.

1. Kondisi Server *UP*

Berikut ini dilakukan pengujian terhadap beberapa *server* yaitu ITN Malang dan Universitas Jember (UNEJ), yang ditunjukkan dalam gambar 4.22, dengan status *UP*.




Gambar 4.22 Monitoring terhadap Server ITN Malang dan UNEJ

Dari gambar 4.22 telah berhasil dimonitoring dengan:

- Alamat : www.itn.ac.id
www.unej.ac.id
- Status : UP
- Pengecekan : 2013-08-20, 08-32-27
2013-08-20, 08:24:45

Keterangan:

Status

- Jika text server berwarna hijau dan icon  menyatakan Server *UP*.

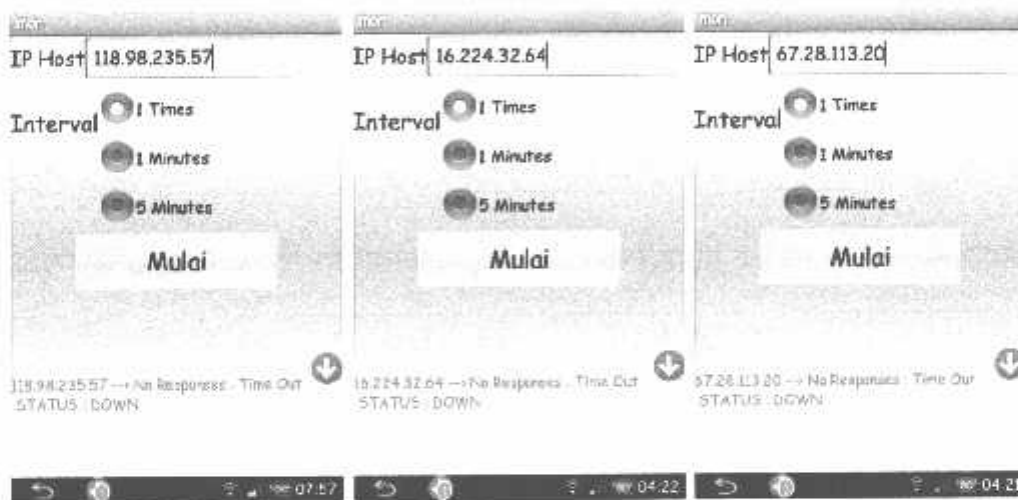
Pengecekan

- Menampilkan tanggal dan waktu dengan format: Tahun-Bulan-Tanggal, Jam:Menit:Detik.

2. Kondisi Server *Down*

Sedangkan pengujian terhadap server dengan keadaan *DOWN*, ditunjukkan dalam gambar 4.23. Dan server yang dituju adalah server yang telah non-aktif atau mati. Beberapa server down yang digunakan untuk dimonitoring yaitu :

- 118.98.235.57 milik bidikmisi.dikti.go.id (DIKTI)
- 16.224.32.64 milik Digital Equipment Coporation (DEC)
- 67.28.113.20 milik Yahoo




Gambar 4.23 Monitoring terhadap beberapa Server Down

Dari gambar 4.23, telah berhasil dimonitoring dengan:

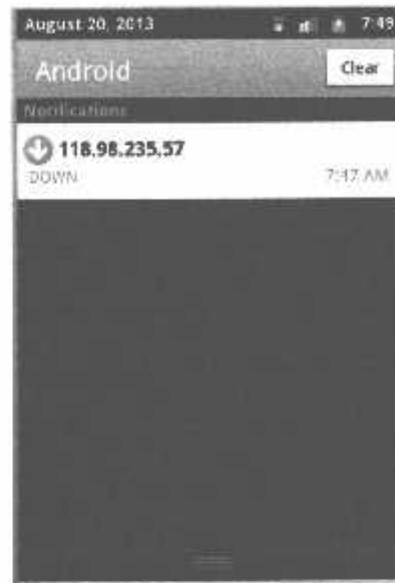
- Alamat : bidikmisi.dikti.go.id (118.98.235.57)
16.224.32.64 (Digital Equipment Coporation)
67.28.113.20 (Yahoo)
- Status : *Down*

Keterangan:

Status

- Jika text server berwarna merah dan icon  menyatakan Server *DOWN*.

Apabila terjadi keadaan down pada server, maka system akan memberikan notifikasi berupa alert kepada pengguna. Notifikasi akan muncul pada status bar yang terletak pada atas atau dibawah layar smartphone, yang ditunjukkan dalam gambar 4.24.



Gambar 4.24 Notifikasi dari Sistem

4.3.8 Tampilan Hasil Last Check

Tampilan last check digunakan untuk menampilkan data alamat host atau server yang telah dimonitoring sebelumnya, alamat host tersebut dimasukan kedalam database kemudian ditampilkan kembali. Hasil data last check ditunjukkan dalam gambar 4.25.



Gambar 4.25 Data Last Check

4.3.9 Tampilan Hasil Menggunakan Interval

Tampilan hasil interval dilakukan untuk mengetahui program aplikasi dapat berjalan sesuai dengan fungsinya, yaitu melakukan proses monitoring tiap waktu yang telah ditentukan oleh pengguna atau *user*. Untuk memberikan informasi bahwa interval dapat berjalan dengan baik, terdapat pemberitahuan berupa *teks toast* atau teks yang akan muncul dan hilang dengan sendirinya untuk memberitahukan kepada pengguna. Proses menggunakan interval waktu ditunjukkan dalam gambar 4.26.



Gambar 4.26 Inteval Waktu Ping Tiap 1 Menit

4.3.10 Pengujian Aplikasi pada Smartphone Android

Aplikasi *Monitoring Responses Server* berbasis android telah dibuat dan kemudian diuji pada *smartphone android*, agar dapat mengetahui bahwa aplikasi yang telah dibuat dapat berjalan dengan baik pada *smartphone android*. Pengujian dilakukan pada perangkat yang berbeda – beda, seperti Sistem Operasi Android, Resolusi, dan Tipe Smartphone Android. Hasil pengujian aplikasi ditunjukkan dalam tabel 4.1.

Tabel 4.1 Hasil Pengujian Pada Smartphone

Perangkat Android	Operating System	Resolusi	Hasil Pengujian	
			Berhasil	Gagal
Sony Ericsson Xperia X8	Android OS v2.1 (Éclair)	320 x 480 pixel, 3 inchi	√	-

Sony Neo L	Android OS v4.0.3 (Ice Cream Sandwich)	480 x 854 pixel, 4 inchi	√	-
Samsung Galaxi Fit	Android OS v2.2 (Frozen Yogurt)	230 x 320 pixel, 3,31 inchi	√	-
Samsung Galaxi Tab 2 7.0	Android OS v4.0.4 (Ice Cream Sandwich)	600 x 1024 pixel, 7,0 inchi	√	-

Keterangan: √ : Berhasil

X : Gagal

4.3.11 Pengujian Aplikasi Berdasarkan versi SDK

Pengujian aplikasi *Monitoring Responses Server* berbasis android yang telah dibuat dan diuji terhadap SDK Version yang berbeda, agar dapat mengetahui bahwa aplikasi yang telah dibuat dapat berjalan dengan baik pada tiap versi SDK yang berbeda. Pengujian dilakukan pada perangkat Smartphone Android, yang berbeda versi SDK berdasarkan Sistem Operasi Android ditunjukkan dalam tabel 4.2 .

Tabel 4.2 Hasil Pengujian Berdasarkan versi SDK

Perangkat Android	Operating System	SDK Version	Hasil Pengujian	
			Berhasil	Gagal
Sony Ericsson Xperia X8	Éclair	API 7 Android 2.1	√	-
Samsung Galaxi Fit	Frozen Yogurt	API 8 Android 2.2	√	-
Sony Neo L	Ice Cream Sandwich	API 15 Android 4.0.3	√	-
Samsung Galaxi Tab 2 7.0	Ice Cream Sandwich	API 15 Android 4.0.4	√	-

4.3.12 Pengujian Fungsionalitas

Pengujian fungsionalitas pada aplikasi dilakukan agar dapat mengetahui bahwa komponen pada aplikasi yang telah dibuat berfungsi dengan baik pada tiap Smartphone Android. Hasil pengujian fungsionalitas ditunjukkan dalam tabel 4.3.

Tabel 4.3 Hasil Pengujian Fungsionalitas pada Aplikasi

Komponen	Hasil Pengujian	
	Berfungsi	Gagal
Edit Text Nama	√	-
Edit Text Password	√	-
Edit Text IP Host / Server	√	-
Tombol Login	√	-
Tombol Monitoring	√	-
Tombol Last Check	√	-
Tombol Help	√	-
Tombol Mulai	√	-
Radio Button 1 Times	√	-
Radio Button 1 Menit	√	-
Radio Button 5 Menit	√	-
Option Menu Tombol Exit	√	-

4.3.13 Hasil Pengujian Sistem Monitoring Server

Dari hasil pengujian system pada aplikasi monitoring server yang telah dilakukan, maka didapatkan hasil yang ditunjukkan dalam tabel 4.4.

Tabel 4.4 Hasil Pengujian Sistem pada Aplikasi

Tujuan Pengujian	Metode Pengujian	Hasil Pengujian	Keterangan
Mendapatkan status server	Monitoring (ping) terhadap server	Aplikasi mendapatkan status server yang dimonitoring	Berhasil
Mendapatkan notifikasi jika server down	Monitoring (ping) terhadap server yang sudah mati	Aplikasi memberikan notifikasi berupa alert	Berhasil
Mendapatkan alamat terakhir melakukan ping terhadap server	Mengakses data / informasi dari database SQLite	Aplikasi memberikan informasi tentang record terakhir melakukan ping	Berhasil
Aplikasi melakukan monitoring secara periodik	Interval diset 1 menit	Aplikasi melakukan monitoring terhadap server setiap 1 menit	Berhasil

Dari percobaan yang telah dilakukan, yaitu sebanyak 4 kali pengujian dengan hasil "**berhasil**". Perhitungan persentase keberhasilan dilakukan dengan cara berikut:

$$P/HP * 100\%$$

Keterangan: P : Pengujian yang dilakukan

HP : Hasil pengujian

Jadi, diperoleh persentase keberhasilan: $4/4 * 100\% = 100\%$ (berhasil). Dari 4 kali percobaan, didapat hasil pengujian sebanyak 3 keberhasilan. Dari jumlah pengujian dibagi dengan hasil pengujian (berhasil) kemudian dikalikan dengan 100%, maka akan didapatkan hasil persentase keberhasilan program aplikasi monitoring runtime server.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari pembahasan yang telah dikemukakan oleh penulis dalam membuat rancang bangun aplikasi monitoring runtime server dan pembuatan laporan Skripsi ini, dapat diambil kesimpulan:

1. Dengan membuat sebuah tool (aplikasi) monitoring runtime server dapat membantu administrator dalam mengawasi server yang dikelolanya.
2. Adanya system notifikasi pada aplikasi dapat memberikan pemberitahuan secara otomatis kepada administrator apabila suatu waktu server sedang down.

5.2 Saran

1. Untuk pengembangan selanjutnya aplikasi monitoring runtime server tidak hanya memonitoring runtime server, tetapi dapat juga memonitoring lainnya seperti service yang sedang berjalan, performa server dan lain sebagainya.
2. Menambahkan fitur multiple server agar dapat memonitoring beberapa server dalam waktu bersamaan.

DAFTAR PUSTAKA

1. Akbarul, Arif Huda. (2012). 24 Jam!! Pintar Pemrograman Android. Yogyakarta: Penerbit Andi.
2. Chan, Syahrial. (2010). Membuat Aplikasi Client / Server & WEB. Elex Media.
3. Jhuana, Gheyb Ohara (2005). Aplikasi Sistem Monitoring Berbasis WEB Untuk Open Cluster. Jurnal STT Telkom.
4. Radityo, Dimas Satrio Nuugroho, Wahyu Suadi, Baskoro Adi Pratomo (2011). Implementasi Sistem Manajemen Database untuk SQLite di Sistem Android. Jurnal ITS Surabaya.
5. Safaat, N. s. (2011). Pemrograman Aplikasi Mobile Smartphone dan tablet PC. Bandung: Informatika
6. Siregar, Ivan Michael, (2011). Membongkar Source Code berbagai Aplikasi Android. Yogyakarta: Gava Media
7. Wahana Komputer, (2012). Membuat Aplikasi Android untuk Tablet dan Handphone. Elex Media.
8. Android Developer, (2007). Android Developer Guide Retrieved. Februari 2013, dari <http://www.developer.android.com>
9. Syamsudin, M. (2011). 60 Menit Belajar Sistem Monitoring Jaringan. Google Book Publisher (E-Book).

LAMPIRAN





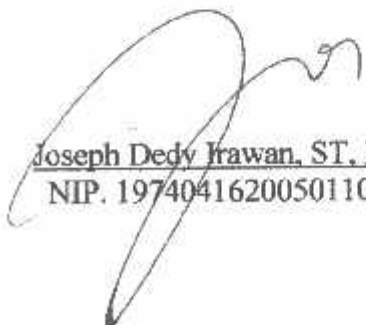
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Karanglo Km. 2 Malang

FORMULIR BIMBINGAN SKRIPSI

Nama : Eko Chandra Edy Saputra
NIM : 09.18.093
Masa Bimbingan : 11 Mei – 11 Nopember 2013
Judul : Rancang Bangun Aplikasi Monitoring Responses
Server Berbasis Android

No.	Tanggal	Uraian	Paraf Pembimbing
1	30 Mei 2013	- Revisi Bab I dan Bab II	
2	26 Juli 2013	- ACC Bab I dan Bab II	
3	29 Juli 2013	- ACC Bab III dan Bab IV	
4	30 Juli 2013	- ACC Bab V - ACC Makalah Seminar Hasil	
5	14 Agustus 2013	- Revisi Judul "Rancang Bangun Aplikasi Monitoring Responses Server Berbasis Android" - ACC Kompre	

Malang, 21 Agustus 2013
Dosen Pembimbing


Joseph Dedy Irawan, ST, MT.
NIP. 197404162005011002



INTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Karanglo Km. 2 Malang

FORMULIR BIMBINGAN SKRIPSI

Nama : Eko Chandra Edy Saputra
NIM : 09.18.093
Masa Bimbingan : 11 Mei – 11 Nopember 2013
Judul : Rancang Bangun Aplikasi Monitoring Responses
Server Berbasis Android

No.	Tanggal	Uraian	Paraf Pembimbing
1	30 Mei 2013	- Algoritma - Tampilan pada Android dan Eclipse pada Bab III	
2	24 Juli 2013	- Flowchart - Penggunaan Emulator	
3	26 Juli 2013	- Revisi Bab III dan Bab IV - ACC Bab I dan Bab II	
4	29 Juli 2013	- ACC Bab III dan Bab IV - Revisi Bab V - Buat Makalah Seminar Hasil	
5	30 Juli 2013	- ACC Bab V - ACC Makalah Seminar Hasil	
6	14 Agustus 2013	- ACC Kompre	

Malang, 21 Agustus 2013
Dosen Pembimbing

Sonny Prasctio, ST, MT.
NIP.P. 1031000433



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Karanglo Km. 2 Malang

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

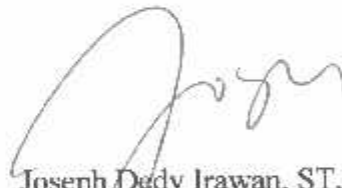
Nama : Eko Chandra Edy Saputra
NIM : 09.18.093
Jurusan : Teknik Informatika S-1
Judul : Rancang Bangun Aplikasi Monitoring Responses Server
Berbasis Android

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1)
pada :

Hari : Jum'at
Tanggal : 15 Agustus 2013
Nilai : 85,65 (A)

Panitia Ujian Skripsi :

Ketua Majelis Penguji



Joseph Dedy Irawan, ST, MT
NIP.19740416 200501 1 002

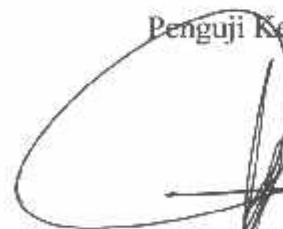
Anggota Penguji :

Penguji Pertama



Michael Ardita, ST, MT.
NIP.P.1031000434

Penguji Kedua



Suryo Adi Wijowo, ST, MT.
NIP.P. 1031000438



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Raya Karanglo Km. 2 Malang

FORMULIR PERBAIKAN SKRIPSI

Nama : Eko Chandra Edy Saputra
NIM : 09.18.093
Jurusan : Teknik Informatika S-1
Judul : Rancang Bangun Aplikasi Monitoring Responses Server Berbasis Android

Tanggal	Penguji	Uraian	Paraf
15 Agustus 2013	I	<ul style="list-style-type: none">- Flowchart Bab III sesuaikan dengan aplikasi yang telah dibuat.- Lengkapi flowchart dengan pengecekan waktu masing – masing interval (1 times, 1 minutes, 5 minutes).- Kesimpulan: Perhitungan pada point 3 dipindah ke Bab IV.- AVD pada Bab III diganti dengan Bluestacks- Tambahkan teori / pustaka pada Bab II tentang SQLite.	
15 Agustus 2013	II	<ul style="list-style-type: none">- Pengujian terhadap server lain- Flowchart terhadap system timer- Pengujian terhadap fungsionalitas, resolusi, system dan terhadap versi SDK yang berbeda.- Help menu	

Anggota Penguji :

Penguji Pertama

Michael Ardita, ST, MT.
NIP.P.1031000434

Penguji Kedua

Suryo Adi Wibowo, ST, MT.
NIP.P. 1031000438

Dosen Pembimbing:

Dosen Pembimbing I

Joseph Dedy Irawan, ST, MT.
NIP.19740416 200501 1 002

Dosen Pembimbing II

Sonny Prasetyo, ST, MT.
NIP.P.1031000433

SURAT PERNYATAAN KEASLIAN SKRIPSI

Yang bertanda tangan di bawah ini, saya:

Nama : Eko Chandra Edy Saputra

NIM : 0918093

Program Studi : Teknik Informatika S-1

Fakultas : Teknologi Industri

Judul Skripsi : Rancang Bangun Aplikasi Monitoring Responses Server Berbasis
Android

Menyatakan bahwa skripsi saya ini asli dan belum pernah diajukan untuk mendapatkan gelar akademik sajarna baik di Institut Teknologi Nasional Malang maupun di Perguruan Tinggi lain.

Dalam Skripsi ini tidak terdapat karya atau pendapat yang pernah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama dan dicantumkan dalam daftar rujukan. Apabila dikemudian hari ada klaim dari pihak lain maka akan menjadi tanggung jawab saya sendiri bukan tanggung jawab dosen pembimbing atau pengelola Fakultas Teknologi Industri dan saya bersedia menerima sanksi akademik sesuai dengan peraturan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sebenarnya tanpa ada paksaan dari pihak manapun.

Malang, Agustus 2013

Yang menyatakan,


METERAI
TEMPEL
REPUBLIK INDONESIA
R226ABF710908297
6000 DJP

Eko Chandra Edy Saputra
NIM. 0918093



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 Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 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
M a l a n g

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

Nama : EKO CHANDRA EDY SAPUTRA
Nim : 0918093
Prodi : Teknik Informatika S1
Fakultas : Teknologi Industri

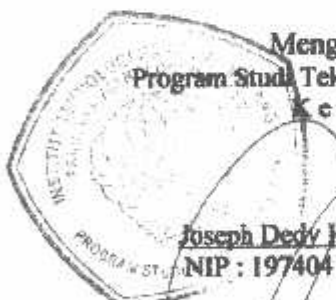

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Bpk/Ibu selama masa waktu 6 (enam) bulan, terhitung mulai tanggal ;

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. BNI (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) 417636 Fax. (0341) 417634 Malang

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

11 Mei 2013

Kepada : Yth. Bpk/Ibu Sonny Prasetio, ST, MT.
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 :

Nama : EKO CHANDRA EDY SAPUTRA
Nim : 0918093
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

Source Code Untuk Halaman Login

```
package com.example.mon;
import com.example.mon.R;
import com.example.mon.menu.a_menu;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import android.view.View.OnClickListener;

public class a_LoginForm extends Activity implements
OnClickListener {
    Button login;
    AlertDialog alert;
    EditText name;
    EditText pass;
    String isi_pass, username;

    public void onCreate(Bundle savedInstanceState) {
        // kindie savedInstanceState = null;
        super.onCreate(savedInstanceState);
        setContentView(R.layout.a_loginform);

        login = (Button) this.findViewById(R.id.btnlogin);
        login.setOnClickListener(this);

        name = (EditText) findViewById (R.id.name);
        name.getText();

        pass = (EditText) findViewById (R.id.pass);
        pass.getText();

        username = "root";
        isi_pass = "root";

        name.setText("");
        pass.setText("");
    }

    public void onClick(View tombol_act) {
        // TODO Auto-generated method stub
        if (tombol_act == login) {
```

```

        if (pass.getText().toString().equals(isi pass) &&
name.getText().toString().equals(username)) {
            Toast.makeText(this, "Nama dan Pass Anda
Benar", Toast.LENGTH_LONG).show();

            Intent i = new
Intent(getApplicationContext(), a_menu.class);
            startActivity(i);
        } else {
            AlertDialog.Builder pesan = new
AlertDialog.Builder(this);
            pesan.setMessage("Nama dan Password Anda
Salah");

            .setCancelable(false).setPositiveButton("OK",
new
DialogInterface.OnClickListener() {

                @Override
                public void
onClick(DialogInterface dialog,
                    int which) {
// TODO Auto-
generated method stub
                    name.setText("");
                    pass.setText("");
                }
            });
            alert = pesan.create();
            alert.show();
        }
    }

    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.login, menu);
        return true;
    }

    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.mExit:
                // TODO Auto-generated method stub
                break;
        }
        return super.onOptionsItemSelected(item);
    }
}
}

```

Source Code Untuk Halaman Menu Utama

```
package com.example.mon.menu;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.DialogInterface.OnClickListener;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import com.example.mon.R;
import com.example.mon.cek.LastCheck;
import com.example.mon.run.MainActivity;

public class a_menu extends Activity {
    Button Run;
    Button Cek;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.bmenu);
        // TODO Auto-generated method stub
        Cek = (Button) findViewById(R.id.btnCek);
        Run = (Button) findViewById(R.id.btnRun);
        Run.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                // Launching All products Activity
                Intent i = new
Intent(getApplicationContext(), MainActivity.class);
                startActivity(i);
                //setContentView(R.layout.cashiermain);
            }
        });
        Cek.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                // Launching All products Activity
                Intent i = new
Intent(getApplicationContext(), LastCheck.class);
                startActivity(i);
                //setContentView(R.layout.diastcheck);
            }
        });
    }
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.menu, menu);
    }
}
```

```

        return true;
    }

    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.exit:
                keluarYN();
                break;
        }
        return super.onOptionsItemSelected(item);
    }

    private void keluarYN() {
        // TODO Auto-generated method stub
        AlertDialog.Builder ad=new
AlertDialog.Builder(a_menu.this);
        ad.setTitle("Konfirmasi");
        ad.setMessage("Apakah benar ingin keluar?");
        ad.setPositiveButton("OK",new OnClickListener(){
            private AlertDialog show;
            public void onClick(DialogInterface dialog,
int which) {
                keluar();
            }
            private void keluar() {
                finish();
            }
        });
        ad.setNegativeButton("No",new OnClickListener(){
            public void onClick(DialogInterface arg0, int
arg1) {
                });
        ad.show();
    }
}

```

Coding Halaman Runtime

```
package com.example.mon.run;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.InetAddress;
import java.net.UnknownHostException;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Timer;
import java.util.TimerTask;

import com.example.mon.R;
import android.os.Bundle;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.text.Editable;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;
import android.widget.Toast;

@SuppressLint("SimpleDateFormat")
public class MainActivity extends Activity
{
    NotificationManager notificationManager;
    static final int uniqueID = 1284885;
    EditText edit;
    RadioGroup rgl;
    RadioButton rb1, rb2, rb3;

    ImageView img1, img2;
    TextView text, text2, text3, text4;
    Button muba;
    DatabaseManager db;
    TableLayout tabledata;
    TableRow rowtab;
```

```

TextView lastcek;

Timer timer = new Timer();
long starttime = 0;

public void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    db = new DatabaseManager(this);
    lastcek = (TextView)findViewById(R.id.cek_id);
    tabeldata = (TableLayout)findViewById(R.id.tblLayout1);
    edit = (EditText)findViewById(R.id.editText1);
    rg1 = (RadioGroup)findViewById(R.id.radioGroup1);
    rb1 = (RadioButton) findViewById (R.id.radio0);
    rb2 = (RadioButton) findViewById (R.id.radio1);
    rb3 = (RadioButton) findViewById (R.id.radio2);
    text1 = (TextView)findViewById(R.id.textView3);
    text2 = (TextView)findViewById(R.id.textView4);
    text3 = (TextView)findViewById(R.id.textView5);
    text4 = (TextView)findViewById(R.id.textView2);
    img1 = (ImageView)findViewById(R.id.imageView1);
    img2 = (ImageView)findViewById(R.id.imageView2);
    btn1 = (Button) findViewById(R.id.btnMula1);
    img1.setVisibility(View.INVISIBLE);
    img2.setVisibility(View.INVISIBLE);

    btn1.setOnClickListener(new klik());
}
class klik implements Button.OnClickListener{
public void onClick(View v){
    int id = rg1.getCheckedRadioButtonId();
    if (id==R.id.radio0){
        simpanData();
        ping();
    }else if (id==R.id.radio1){
        simpanData();
        satuMenit();
    }else if (id==R.id.radio2){
        simpanData();
        limaMenit();
    }else {

    }
}
}

```


Fungsi Untuk Monitoring Terhadap Server satu kali

```
private void ping() {
    // TODO Auto-generated method stub
    text2.setText("");
    text3.setText("");
    text4.setText("");
    img1.setVisibility(View.INVISIBLE);
    img2.setVisibility(View.INVISIBLE);

    Editable host = edit.getText();
    try {
        String pingCmd = "ping -c 1" + host;
        String pingResult = "";
        java.lang.Runtime r =
java.lang.Runtime.getRuntime();
        java.lang.Process p = r.exec(pingCmd);
        BufferedReader in = new BufferedReader(new
InputStreamReader(p.getInputStream()));
        String inputLine;
        while ((inputLine = in.readLine()) != null) {
            System.out.println(inputLine);
            text.setText(inputLine + "\n\n");
            pingResult += inputLine;
            text.setText(pingResult);
        }
        in.close();
        InetAddress i;
        i = null;
        try{
            i = InetAddress.getByHost(host.toString());
            Calendar c = Calendar.getInstance();
            System.out.println("Current time =>
"+c.getTime());

            SimpleDateFormat df = new
SimpleDateFormat("yyyy MM-dd HH:mm:ss");
            String formattedDate = df.format(c.getTime());

            if (i.isReachable(10000)){
                text2.append("\n" + host + " -->
STATUS : UP");
                text4.setText(formattedDate);
                img1.setVisibility(View.VISIBLE);
            }
            else {
                text3.append("\n" + host + " --> No
Responses : Time Out \n STATUS : DOWN");
                img2.setVisibility(View.VISIBLE);
                tampilkanNotifikasi();
            }
        } catch (UnknownHostException e ){
```

```

        e.printStackTrace();
        Toast.makeText(MainActivity.this, "\n IP
Address /: Alamat Host Salah", Toast.LENGTH_SHORT).show();
    }
}
catch (IOException e) {
    System.out.println(e);
}
}
}

```

Fungsi Timer Untuk Interval Waktu Monitoring Setiap 1 Menit

```

public void satuMenit() {
    // TODO Auto-generated method stub
    timer = new Timer();
    timer.schedule(new TimerTask() {

        @Override
        public void run() {
            MainActivity.this.runOnUiThread(new Runnable() {

                public void run() {

                    Toast.makeText(MainActivity.this, "\n Interval Tiap 1 Menit",
                    Toast.LENGTH_LONG).show();
                    text2.setText("");
                    text3.setText("");
                    text4.setText("");
                    img1.setVisibility(View.INVISIBLE);
                    img2.setVisibility(View.INVISIBLE);

                    Editable host = edit.getText();
                    try {
                        String pingCmd = "ping -c 1" + host;
                        String pingResult = "";
                        java.lang.Runtime r = java.lang.Runtime.getRuntime();
                        java.lang.Process p = r.exec(pingCmd);
                        BufferedReader in = new BufferedReader(new
                            InputStreamReader(p.getInputStream()));
                        String inputLine;
                        while ((inputLine = in.readLine()) != null) {
                            System.out.println(inputLine);
                            text.setText(inputLine + "\n\n");
                            pingResult += inputLine;
                            text.setText(pingResult);
                        }
                    }
                }
            });
        }
    }, 1000);
}

```

```

        in.close();
        InetAddress i;
        i = null;
        try{
            i = InetAddress.getByHost(host.toString());
            Calendar c = Calendar.getInstance();
            System.out.println("Current time -> "+c.getTime());
            SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss");
            String formattedDate = df.format(c.getTime());
            if (i.isReachable(90000)){
                text2.append("\n" + host + " --> STATUS : UP");
                text4.setText(formattedDate);
                img1.setVisibility(View.VISIBLE);

                //updateTable();
            }else {
                text3.append("\n" + host + " --> No Responses :
Time Out \n STATUS : DOWN");
                img2.setVisibility(View.VISIBLE);
                tampilkanNotifikasi();
            }
        } catch (UnknownHostException e ){
            Toast.makeText(MainActivity.this, "\n IP Address /
Alamat Host Salah", Toast.LENGTH_SHORT).show();
        }
        catch (IOException e) {
            System.out.println(e);
        }
    });
}
}, 100, 60000);

```

Source Code untuk Menu Last Check

```

package com.example.mon.cdk;

import java.util.ArrayList;
import com.example.mon.R;
import com.example.mon.run.DatabaseManager;
import android.app.Activity;
import android.os.Bundle;

import android.widget.Button;

import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;
import android.widget.Toast;

```