

Implementasi Sistem Manajemen *Database* untuk SQLite di Sistem Android

Dimas Radityo Satrio Nugroho, Wahyu Suadi, Baskoro Adi Pratomo

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember

¹jono_51@cs.its.ac.id

²wahyu@its-sby.edu

³baskoro@if.its.ac.id

Abstract—Kemajuan teknologi yang sangat cepat pada saat ini diiringi dengan kegiatan manusia yang cepat pula. Dengan teknologi yang berkembang pada saat ini khususnya teknologi pada perangkat *mobile*. Perkembangan tersebut diiringi dengan perubahan kebiasaan *mobile-user* yang hanya menggunakan perangkat *mobile* yang biasanya hanya digunakan untuk keperluan telepon dan pesan singkat menuju ke penggunaan yang menunjang kegiatan sehari-hari, baik untuk keperluan pekerjaan maupun hiburan. Sehingga kebutuhan akan aplikasi database untuk perangkat *mobile* sangat dibutuhkan. Manusia memiliki kegiatan yang dapat dikatakan rumit. Sehingga mereka membutuhkan media dokumentasi kegiatan mereka yang banyak.

Oleh karena itu, sistem ini dibangun untuk memenuhi kebutuhan tersebut. Tujuan Dengan dibangunnya sistem ini adalah agar pengguna dapat menyimpan semua dokumentasi kegiatannya. Sistem ini dengan *platform* OS Android dan sebagai media penyimpanannya menggunakan SQLite. Sistem ini menyediakan empat fitur utama yaitu terdiri dari tiga *database* utama aplikasi ini dan satu fitur tambahan yaitu penambahan *database* baru yang dapat dibuat sesuai keinginan pengguna.

Hasil uji coba aplikasi ini adalah aplikasi ini mampu bekerja secara maksimal dalam menjalankan query tertentu dengan pemberian data sebanyak 2000 data *dummy* dengan kecepatan kompilasi 110 detik untuk syntax INSERT, 56 detik untuk UPDATE, 244 milidetik untuk SELECT, dan 45 milidetik untuk DELETE

Kata kunci: *Android, database, SQLite*

I. PENDAHULUAN

Perkembangan sistem operasi Android sebagai salah satu sistem operasi untuk *smartphone* telah menyebabkan perubahan besar pada kebiasaan *mobile-user*. Kebiasaan *mobile-user* berubah dari penggunaan perangkat *mobile* yang biasa hanya digunakan untuk keperluan telepon dan pesan singkat menuju ke penggunaan yang menunjang kegiatan sehari-hari, baik untuk keperluan pekerjaan maupun hiburan. Sehingga kebutuhan akan database untuk perangkat *mobile* sangat dibutuhkan.

Sistem manajemen database untuk android ini sebenarnya sudah tersedia, tetapi aplikasi tersebut berbayar.

Aplikasi ini dapat menyimpan semua informasi – informasi seperti daftar belanja sehari – hari, kontak, daftar film terkini. Aplikasi ini memiliki fitur *searching* yang digunakan untuk mencari kata kunci apabila *user* kesulitan mencari informasi yang diinginkan. Aplikasi ini dapat melakukan *customized views* yaitu *user* dapat menampilkan informasi dengan tampilan yang berbeda.

II. TINJAUAN PUSTAKA

A. Sejarah dan Gambaran Umum Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux². Android menyediakan platform terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada Juli 2000, Google bekerjasama dengan Android Inc., perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri Android Inc. bekerja pada Google, di antaranya Andy Rubi, Rich Miner, Nick Sears, dan Chris White. Saat itu banyak yang menganggap fungsi Android Inc. hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler. Di perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel Linux. Hal ini menunjukkan indikasi bahwa Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler.

Pada saat perilsan perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler.

Sekitar September 2007 sebuah studi melaporkan bahwa Google mengajukan hak paten aplikasi telepon seluler (akhirnya Google mengenalkan Nexus One, salah satu jenis telepon pintar yang menggunakan Android pada

sistem operasinya. Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010).

Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka, Android, perangkat mobile yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru.

Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan Android.

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

Ada beberapa keunggulan sistem operasi Android, antara lain :

- Sistem operasi ini berlisensi terbuka, sehingga memungkinkan *vendor* perangkat keras untuk mengembangkan sistem sesuai dengan kebutuhan perangkat keras yang dimiliki. Akibatnya, terdapat berbagai macam produk yang bervariasi dari segi keunggulan harga dan perangkat keras.
- Bagi para pengembang aplikasi, disediakan *Software Development Kit (SDK)* yang lengkap dan kesemuanya disediakan dengan gratis. Hal ini berbeda dengan beberapa *framework* perangkat *mobile* yang mengharuskan pengembang aplikasi membayar untuk beberapa fitur.
- Bagi pengguna biasa, platform terbuka diterjemahkan dengan terbukanya informasi mengenai akses dan kontrol dari sebuah aplikasi yang akan diinstal. Hal ini dapat meminimalisir adanya *spyware* yang digunakan untuk mencuri data pribadi pengguna.
- Sifatnya yang terbuka memungkinkan sistem operasi ini dapat diimplementasikan pada banyak perangkat selain *smartphone* seperti pada *Tablet PC* dan perangkat keras lainnya.

Sebelum memulai membangun aplikasi berbasis Android, diperlukan beberapa perangkat, antara lain :

1. The Eclipse IDE
2. Sun's Java Development Kit (JDK)
3. The Android Software Developer's Kit (SDK)
4. The Android Developer Tool (ADT), sebuah plug-in Eclipse.

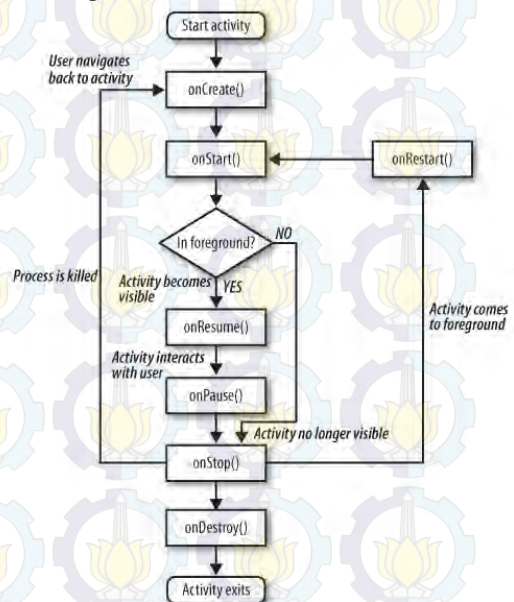
Android SDK yang ada sudah mendukung pengembangan aplikasi ada berbagai macam lingkungan sistem operasi seperti Windows (XP, Vista dan 7), Linux (Ubuntu, Mandriva, OpenSUSE, dan distro lainnya) dan Mac OS X.

C. Android Lifecycle

Pada bagian ini akan dijelaskan mengenai daur hidup atau *lifecycle* dari sistem operasi Android.

1. Activity

Android dirancang untuk memenuhi kebutuhan yang unik dari perangkat *mobile*. Sistem operasi ini dirancang dan disesuaikan dengan keterbatasan sumber daya pada perangkat *mobile*. Mekanisme pemanfaatan sumber daya yang terbatas tersebut disiasati dengan pengaturan daur hidup (*lifecycle*) dari Android Activity. Daur hidup Android dapat digambarkan pada **Gambar 1** berikut



Gambar 1 Bagan Android Lifecycle

2.

Sistem operasi ini berbasis pada *kernel* Linux yang dikenal stabil dan memiliki *access permission* yang kuat sehingga memungkinkan adanya akses pengguna yang berjenjang. Hal ini berpengaruh pada sisi keamanan dan keselamatan perangkat.

B. Sejarah dan Gambaran Umum Android

Term Frequency (tf) merupakan jumlah kemunculan kata Untuk membangun aplikasi berbasis Android, kita tidak harus memiliki perangkat mobile langsung. Kita cukup menggunakan emulator yang sudah disediakan oleh Google.

Berikut penjelasan masing-masing fase pada bagan daur hidup Android.

a. OnCreate

Fase ini terjadi pada saat suatu *activity* dibuat. Pada bagian ini, tampilan yang berhubungan langsung dengan pengguna dipanggil.

b. OnStart

Fase ini dipanggil tepat sebelum suatu *activity* terlihat di layar. Fase ini berakhir ketika *activity* yang dipanggil menjadi *foreground activity* di layar sehingga fase berpindah ke fase *onResume*.

c. OnResume

Pada fase ini, *activity* berjalan dan berinteraksi langsung dengan pengguna. Pada fase ini, aplikasi menerima input dari pengguna yang berasal dari *keyboard* dan *touch*.

d. OnPause

Fase ini dipanggil ketika aplikasi lain dalam satu sistem akan menuju ke fase *resume*. Pada fase ini, *activity* yang berjalan tidak memiliki akses ke layar, jadi sebaiknya, semua proses yang mengkonsumsi sumber daya proses dan baterai sebaiknya dihindari. Fase ini sebaiknya digunakan untuk menyimpan *state* jika aplikasi akan ditampilkan ke layar kembali.

e. OnStop

Fase ini dipanggil ketika *activity* tidak lagi terlihat karena *activity* lain sudah mengambil alih *foreground* atau karena *activity* kita sudah berakhir daur hidupnya.

f. OnDestroy

Fase ini merupakan fase berakhirnya suatu *activity*. Fase ini terjadi biasanya ketika aplikasi memanggil fungsi *finish()*.

2. Service

Daur hidup *Service* mirip dengan daur hidup *Activity* yang dijelaskan pada bagian sebelumnya. Berikut penjelasan masing-masing fase.

a. OnCreate dan onStart

Service dipanggil ketika *client* memanggil fungsi *Context.startService(Intent)*. Jika *service* tidak sedang berjalan, Android akan memanggil fungsi *onCreate* yang kemudian diikuti dengan fungsi *onStart*. Jika *service* telah berjalan, fungsi *onStart* kembali dipanggil.

b. OnResume, onPause, dan onStop

Karena *Service* tidak memiliki antar muka (*interface*), maka fase *OnResume*, *onPause*, dan *onStop* tidak dibutuhkan karena proses berjalan di *background*.

c. onBind

Jika *client* membutuhkan koneksi ke sebuah *service* dia dapat memanggil fungsi *Context.bindService*. Proses ini membuat *service* jika belum berjalan akan memanggil fungsi *onCreate* tapi bukan fungsi *onStart*.

d. OnDestroy

Seperti pada *activity*, fungsi *onDestroy*. Fase ini merupakan fase berakhirnya suatu *service*. Fase ini terjadi biasanya ketika aplikasi memanggil fungsi *finish()*.

D. 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 disain tersebut bisa diraih dengan cara mengunci keseluruhan file basis data pada saat sebuah transaksi dimulai.

2. Fitur

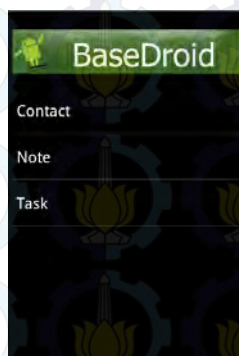
SQLite mengimplementasikan hampir seluruh elemen-elemen standar yang berlaku pada *SQL-92*, termasuk transaksi yang bersifat *atomic*, konsistensi basisdata, isolasi, dan durabilitas (dalam bahasa Inggris lebih sering disebut *ACID*), *trigger*, dan kueri-kueri yang kompleks. Tidak ada pengecekan tipe 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.

III. PERANCANGAN PERANGKAT LUNAK

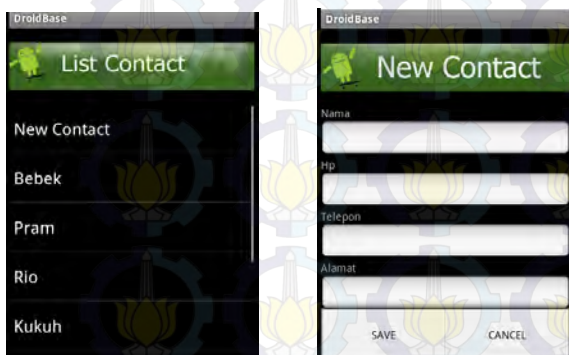
Perancangan aplikasi sistem manajemen SQLite ini nantinya akan dijalankan di *smartphone* berbasis android. Pada awal perangkat lunak dijalankan, maka akan langsung muncul pada halaman pertama yaitu pilihan *database* yang diinginkan yaitu *contact*, *note*, *task*. Pada saat pengguna memilih database kontak, pengguna dapat insert data seperti nama, alamat, nomor telepon genggam, dan telepon rumah. Setelah pengguna memasukkan isi dari kontak, kontak yang telah dimasukkan sebelumnya akan muncul pada *list view* yang telah disediakan. Ketika kontak tersebut di tekan maka akan muncul notifikasi yang didalamnya terdapat pilihan *call*, *edit*, *delete*. Pada pilihan *call* pengguna dapat langsung melakukan panggilan pada kontak yang dipilih sebelumnya.

Kemudian pada pilihan edit maka pengguna dapat merubah isi dari kontak yang dipilih yang kemudian bisa disimpan dan data yang ada didalam database di-update. Selain contact pengguna juga dapat memilih database note. Pada fitur ini pengguna dapat menuliskan catatan apapun dan disertai pula dengan tanggal penulisan. Pada database note pengguna juga dapat mencari dengan fitur search. Pada fitur ini pengguna dapat mencari berdasarkan judul note yang telah dibuat sebelumnya. Kemudian pada fitur task pengguna dapat menuliskan daftar kegiatan hari itu dengan disertai penanda yaitu checkbox. Sehingga pengguna dapat mengetahui daftar kegiatan yang sudah dilakukan maupun yang belum dilakukan. Aplikasi ini juga dapat membuat database baru. Dengan fitur ini pengguna dapat melakukan pembuatan database baru sehingga lebih fleksibel. Pengguna dapat membuat database sesuai keinginan dengan batasan atribut sebanyak 5.

Berikut merupakan rancangan antar muka sistem manajemen pada menu utama yang ditampilkan pada **Gambar 2**.

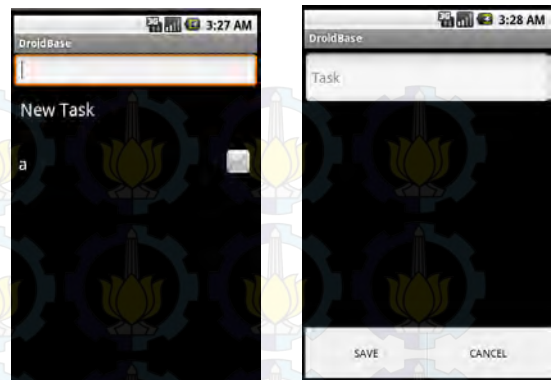


Gambar 2 Rancangan Antar Muka Menu Utama



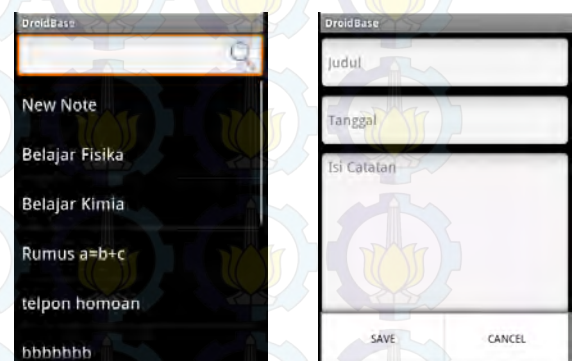
Gambar 3 Rancangan Antar Muka Pada Contact

Rancangan antar muka pada **Gambar 3** merupakan fitur contact didalamnya terdapat list contact yang merupakan daftar contact yang telah disimpan oleh pengguna. Dan juga terdapat halaman *new contact* yang berfungsi sebagai tempat pengguna memasukkan data.



Gambar 4 Rancangan Antar Pada Note

Pada **Gambar 4** merupakan tampilan pembuatan kontak baru dan list contact. Pada Gambar 4 ditampilkan field – field untuk memasukkan data.



Gambar 5 Rancangan Antar Pada Task

Rancangan antar muka pada **Gambar 5** merupakan tampilan pada Database Task. Pada gambar tersebut ditampilkan list task yang juga diikuti dengan checkbox yang berfungsi untuk menandai task mana yang telah dilakukan atau belum.



Gambar 6 Rancangan Antar Pada New Database

Rancangan antar muka pada **Gambar 6** merupakan tampilan pada pembuatan database baru. Pada gambar 3.18 ditampilkan field – field untuk inputan nama database dan nama atributnya. Pada tampilan tersebut terdapat juga tombol untuk mengatur tipe atribut yang akan dibuat.

IV. IMPLEMENTASI PERANGKAT LUNAK

Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam implementasi perangkat lunak untuk aplikasi tugas akhir ini ditampilkan pada Tabel 1.

Tabel 1 Lingkungan implementasi perangkat lunak

Perangkat Keras	Notebook	Apple MacBook dengan spesifikasi Intel Core 2 Duo @2.0 GHz, 2.0 GB DDR3 RAM, 32-Bit Operating System
	Smartphone	Sony Ericsson XPERIA 10i Android 2.1 Eclair
Perangkat Lunak	Sistem Operasi :	Linux Debian Squeeze
	Perangkat	IDE Eclipse SDK versi 3.6.2 ADT 10.0.0

Implementasi yang telah dilakukan adalah menjalankan fitur contact, note, task, dan pembuatan database baru.

V. UJI COBA DAN EVALUASI

Lingkungan uji coba terbagi kedalam dua bagian, yaitu lingkungan uji coba fungsionalitas dan uji coba performa. Lingkungan uji coba yang digunakan dalam uji coba fungsionalitas dan uji coba performa terdiri atas :

- Menggunakan 1 smartphone
- Menggunakan smartphone yang berjalan dengan aplikasi DroidBase .

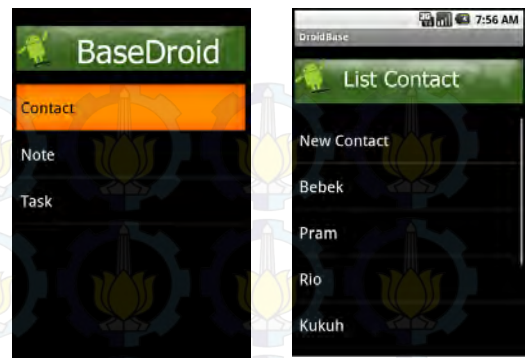
1. Uji Coba Fungsionalitas

Uji coba fungsionalitas dilakukan untuk melihat apakah fungsi-fungsi dasar aplikasi berjalan sebagai mana mestinya. Hasil uji coba ditunjukkan dengan hasil *screenshot* dari *Graphical User Interface* pada *Solver Interface* terkait.

Uji coba fungsionalitas meliputi uji coba pengguna dalam menjalankan perangkat lunak. Pengguna melakukan uji coba memasukkan data ke dalam database Contact, Note, Task. Pengguna juga dapat membuat database baru yang nama database dan atributnya dapat dibuat sesuai keinginan.

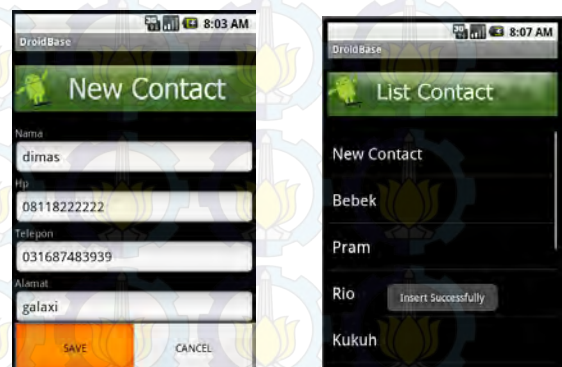
1.1. Menjalankan Fitur Database Contact

Fitur – fitur dari perangkat lunak yang dapat digunakan oleh pengguna adalah fitur tiga pilihan database utama yaitu Contact, Note, Task. Dan satu fitur pembuatan database baru. Untuk fitur database contact dimana seperti **Gambar 7** berikut



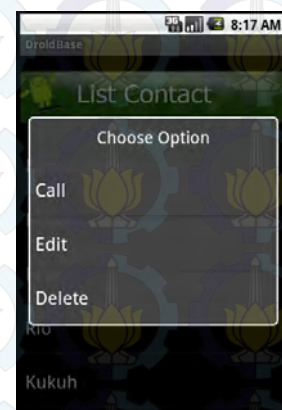
Gambar 7 Tampilan Fitur Database Contact

Setelah pengguna memilih contact, maka pengguna akan dibawa kepada halaman list contact. Dimana pada halaman list contact pengguna juga dapat membuat kontak baru dengan menekan “new contact” seperti pada **Gambar 8** berikut



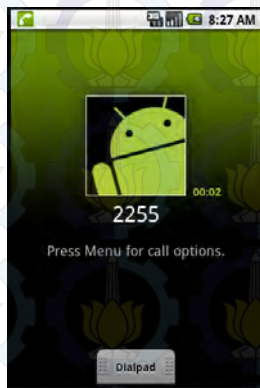
Gambar 8 Tampilan Masukan Kontak

Pada saat pengguna telah memasukkan data dan akan menyimpannya. Pengguna hanya tinggal menekan tombol *save* dan bila data berhasil disimpan maka sistem akan memunculkan peringatan *insert successfully*. Setelah fitur input data pada contact juga terdapat fitur call, edit, delete contact seperti **Gambar 9** seperti berikut



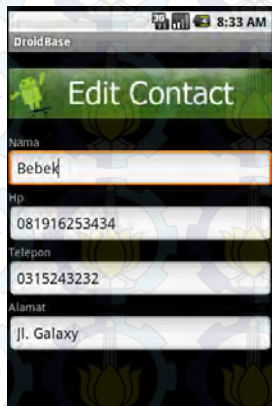
Gambar 9 Tampilan Fitur Pilihan Pada Kontak

Kemudian pada fitur call, sistem akan mengambil nomor telepon di dalam database contact dan system langsung melakukan panggilan seperti pada **Gambar 10**



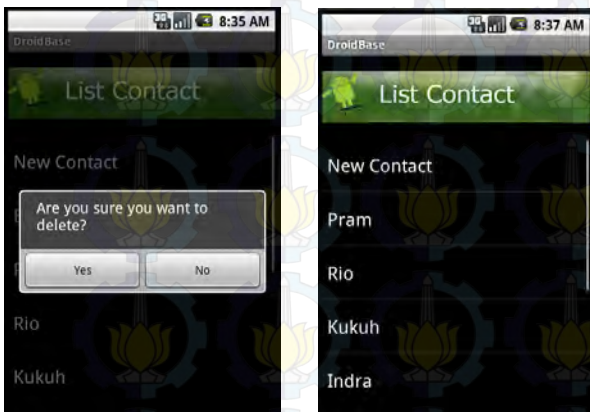
Gambar 10 Tampilan Fitur Call

Kemudian pada fitur edit, sistem akan mengambil data dari database kemudian ditampilkan sehingga pengguna dapat merubah data yang sudah pernah diinputkan sebelumnya. Seperti pada **Gambar 11**



Gambar 11 Tampilan Edit Kontak

Pada fitur delete, pengguna dapat menghapus kontak yang tidak diperlukan. Setelah kontak berhasil dihapus maka kontak tersebut akan terhapus dari list kontak Seperti pada **Gambar 12**

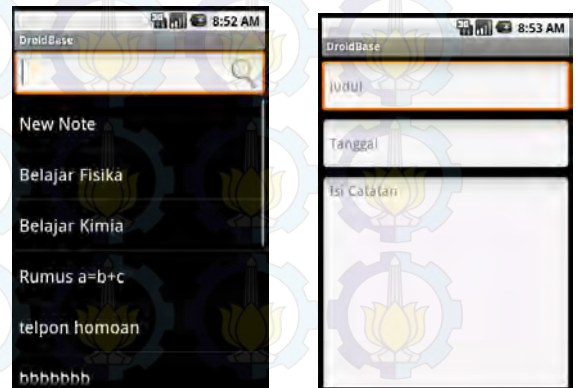


Gambar 12 Tampilan Delete Kontak

1.2. Menjalankan Fitur Database Note

Setelah pengguna memilih contact, maka pengguna akan dibawa kepada halaman list contact. Dimana pada halaman

list contact pengguna juga dapat membuat kontak baru dengan menekan “new note” seperti **Gambar 13**



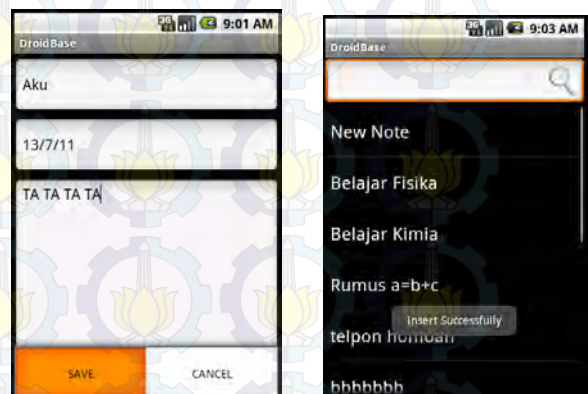
Gambar 13 Tampilan List Note dan Masukan Note

Pada saat pengguna memasukan data pada halaman new note, untuk memasukkan tanggal note dibuat pengguna tidak perlu mengetikkan tanggal yang akan dimasukkan tetapi pengguna hanya akan menginputkan tanggal melalui *date picker* seperti pada **Gambar 14**



Gambar 14 Tampilan Date Picker pada Note

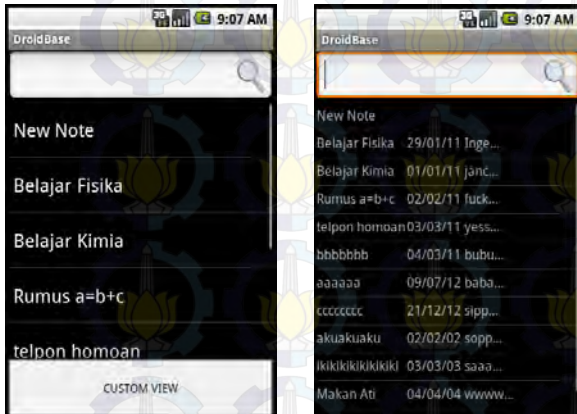
Kemudian pada saat pengguna telah selesai memasukkan data, pengguna hanya perlu memilih tombol *save* untuk menyimpan seperti pada **Gambar 15**



Gambar 15 Tampilan Date Picker pada Note

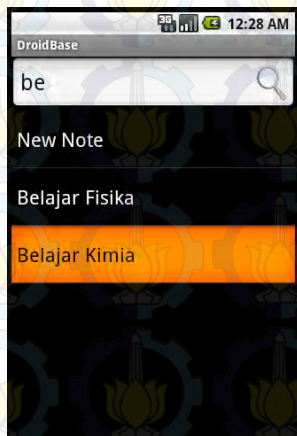
Setelah data berhasil disimpan akan terlihat seperti Gambar 5.9. Dan note yang disimpan akan ditampilkan pada list

note. kemudian pada note juga terdapat customized view yang merupakan tampilan grid view seperti pada **Gambar 16**



Gambar 16 Tampilan Customized View

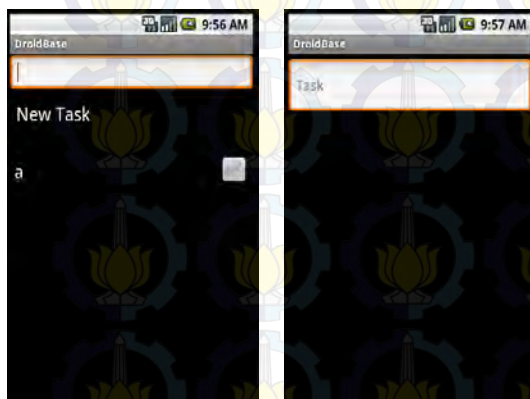
Pada database note juga terdapat fungsi search. Dengan fungsi search ini pengguna dapat lebih mudah mencari nama note yang dicari seperti pada **Gambar 17**



Gambar 17 Tampilan Search Pada Note

1.3. Menjalankan Fitur Database Note

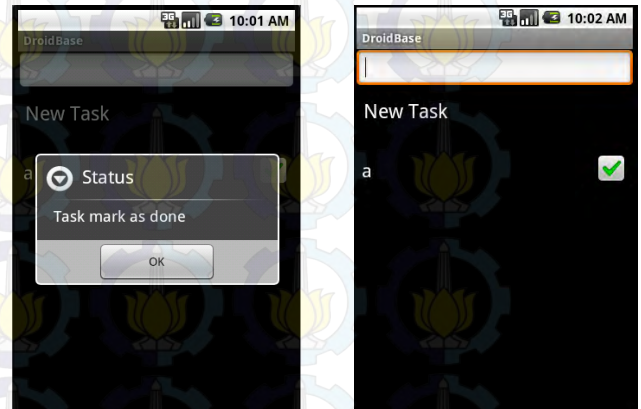
Setelah pengguna memilih contact, maka pengguna akan dibawa kepada halaman list contact. Dimana pada halaman list task pengguna juga dapat membuat kontak baru dengan menekan *new task* seperti **Gambar 18**



Gambar 18 Tampilan Fitur Database Task

Pada saat pengguna memasukkan data dan data tersebut telah disimpan maka task yang baru saja dibuat akan muncul di list

task dan otomatis terdapat checkbox di sebelah kanannya. Fungsi checkbox disini adalah sebagai penanda bahwa tersebut sudah dilakukan atau tidak seperti **Gambar 19**

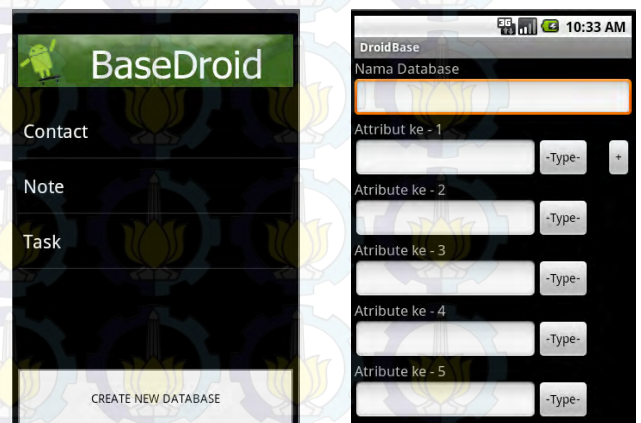


Gambar 19 Tampilan Fitur Database Task

Pada **Gambar 19** ditampilkan bahwa ketika pengguna menekan nama task ketika status undone maka akan keluar dialog box Task mark as done begitu pula sebaliknya.

1.4. Menjalankan Fitur Pembuatan Database Baru

Pada fitur pembuatan database baru ini pengguna dapat membuat database sesuai keinginan user seperti **Gambar 20**



Gambar 20 Tampilan Fitur Database Task

Pada **Gambar 20** ditampilkan bahwa pada menu awal pengguna dapat memilih menu *CREATE NEW DATABASE* untuk masuk ke halaman masukan database baru. Pada halaman masukan database baru terdapat field seperti nama database dan atribut. Untuk atribut sendiri jumlahnya dibatasi sampai dengan 5. Bila lebih dari itu akan muncul peringatan "*max attribute 5*" seperti pada **Gambar 21**



Gambar 21 Tampilan Masukan Fitur New Database

Pada **Gambar 21** ditampilkan pada saat pengguna memasukkan data. Atribut juga dapat diatur formatnya. Disini format yang disediakan hanya 2 yaitu *TEXT* dan *NUMERIC*

2. Uji Coba Performa

Lingkungan uji coba yang digunakan dalam uji coba performa terdiri atas :

- Uji performa dengan inputan sebanyak 500, 1000, 2000.
- Uji performa syntax query yang terdiri dari :
 - INSERT
 - UPDATE
 - DELETE
 - SELECT

Uji performa ini dilakukan pada masing – masing query sebanyak tiga kali percobaan. Setelah dilakukan tiga kali percobaan tersebut kemudian telah didapatkan hasil rata – rata kecepatan kompilasi pada query – query tersebut.

a) INSERT

Jumlah Data Rata - Rata	Waktu Rata – Rata (detik)
500	13,56
1000	38,35
2000	110,34

b) UPDATE

Jumlah Data Rata - Rata	Waktu Rata – Rata(detik)
500	10,10
1000	22,08
2000	55,98

c) SELECT

Jumlah Data Rata - Rata	Waktu Rata – rata (milidetik)
500	26,33
1000	57,33
2000	244,33

d) DELETE

Jumlah Data Rata - Rata	Waktu Rata – Rata (milidetik)
500	32
1000	22,32
2000	45,33

VI. KESIMPULAN DAN SARAN

Berdasarkan aplikasi yang telah dibuat beserta uji coba yang telah dilakukan, maka dapat ditarik kesimpulan sebagai berikut.

- Sistem yang dibuat sudah dapat memenuhi kebutuhan untuk melakukan proses penyimpanan data pada database yang telah disediakan yaitu Contact, Note, dan Task.
- Aplikasi yang dibuat memiliki fitur pembuatan database baru. Yang dimaksud pembuatan database baru adalah pengguna dapat membuat database baru sesuai keinginan dimana nama database dan atribut beserta jenis datanya dapat dibuat sesuai keinginan user. Begitu pula jumlah atribut yang akan dibuat. Jumlahnya dapat diatur sesuai keinginan dengan batas maksimal 5 atribut.
- Pada uji coba kecepatan pada syntax INSERT didapatkan waktu 13,56 detik untuk 500 data, 38,35 untuk 1000 data, dan 110,34 detik untuk 2000 data.
- Pada uji coba kecepatan pada syntax UPDATE didapatkan waktu 10,10 detik untuk 500 data, 22,08

- detik untuk 1000 data, dan 55,98 detik untuk 2000 data.
5. Pada uji coba kecepatan pada syntax SELECT didapatkan waktu 26,33 milidetik untuk 500 data, 57,33 milidetik untuk 1000 data, dan 244,33 milidetik untuk 2000 data.
 6. Pada uji coba kecepatan pada syntax DELETE didapatkan waktu 10,10 milidetik untuk 500 data, 22,08 milidetik untuk 1000 data, dan 55,98 milidetik untuk 2000 data

VII. DAFTAR PUSTAKA

1. OSSFest, Tim. Sejarah Android (Sistem Operasi);2010. [Diakses tanggal 7 Juni 2011]. <<http://ossfest-indonesia.web.id/2010/10/sejarah-android-sistem-operasi/>>.
2. DiMarzio, Jerome. Android : A Programmer's Guide. New York : The McGraw-Hill Companies; 2008.
3. Rogers, Rick, John Lombardo. Android Application Development, 1st Edition. New York : O'Reilly Media, Inc; 2009.
4. Brady, Patrick. Android Anatomy and Physiology; 2008. [Diakses tanggal 10 Juni 2011]. <<http://sites.google.com/site/io/anatomy-physiology-of-an-android>>.
5. Morrill, Dan. Inside the Android Application Framework; 2008. [Diakses tanggal 10 Juni 2011] <<http://sites.google.com/site/io/inside-the-android-application-framework>>.