

Pengembangan Aplikasi Rumus Matematika SMA Berbasis Android

Dody Hidayat¹, Ramli², Tenti Astria Purba³

Address: Universitas Harapan Medan/Fakultas Teknik dan Komputer, Program Studi D3 Manajemen Informatika, Indonesia^{1,2}, Universitas Harapan Medan/Fakultas Teknik dan Komputer, program Studi Teknik Informatika³

Email: hidayatdody91@gmail.com¹, ramli.brt@gmail.com²

* Corresponding author Email: hidayatody91@gmail.com

Abstrak

Rumus matematika adalah ringkasan atau ketetapan ilmu matematika yang dilambangkan dalam bentuk simbol, huruf atau angka dan dinyatakan dalam bentuk ringkas. Rumus-rumus matematika sangat membantu para siswa dalam proses mempelajari matematika. Untuk mempermudah para siswa dalam menggunakan rumus-rumus matematika dikembangkan sebuah Aplikasi Kumpulan Rumus Matematika SMA Berbasis Android dalam bahasa pemrograman Java menggunakan Eclipse IDE mengikuti alur metode SDLC Waterfall Model. Adanya Aplikasi Kumpulan Rumus Matematika SMA Berbasis Android ini dapat mempermudah para siswa dalam mencari rumus-rumus matematika dan menyajikan penggunaan rumus-rumus matematika dalam bentuk contoh soal.

Keywords - Aplikasi Kumpulan Rumus Matematika SMA berbasis Android, pemrograman Java, SDLC Waterfall Model.

1. Latar Belakang

Saat ini teknologi berkembang pesat dengan tujuan memberikan kemudahan dan kenyamanan para user dalam penggunaan teknologi informasi dengan praktis baik dari segi *hardware* maupun *software*.

Perkembangan *software* diwarnai dengan banyaknya kemunculan aplikasi maupun sistem operasi, salah satunya adalah Android. Android yang merupakan *mobile open platform* dimana tidak dibutuhkan adanya lisensi ataupun kontrak dalam pengembangan aplikasinya. Hal inilah membuatnya semakin berkembang dengan pesat dan semakin banyak pula vendor-vendor perangkat mobile menggunakan sistem operasi Android. Perangkat mobile berbasis Android sudah familiar dan banyak digunakan oleh masyarakat. Pengembangan aplikasi android bersifat *open source* dimana artinya merupakan sistem pengembangan yang dikelola oleh para pelaku pengembang yang saling bekerjasama dengan

memanfaatkan *source code* (kode sumber) yang tersedia dan tersebar dengan bebas.

Melihat perkembangan ini penulis tertarik untuk membuat sebuah Aplikasi Kumpulan Rumus Matematika SMA Berbasis Android sebagai media pembelajaran yang diharapkan efektif membantu siswa dalam belajar dan tentunya juga di sisi pengembangan aplikasi Android. Adanya Aplikasi Kumpulan Rumus Matematika SMA ini yang disematkan pada perangkat mobile akan memudahkan para siswa SMA menggunakan rumus dalam belajar matematika. Dengan adanya aplikasi ini pengguna (para siswa SMA) hanya membutuhkan sebuah perangkat *mobile smartphone* yang memiliki sistem operasi Android. Tentunya hal ini sangat menguntungkan pengguna dari sisi ekonomis dan waktu yang digunakan untuk mencari rumus-rumus tersebut secara manual seperti yang biasa dilakukan para siswa.

Dalam pengembangan aplikasi ini penulis menggunakan metode *SDLC Waterfall Model*

yaitu metode pengembangan perangkat lunak dengan pendekatan sekuensial yang meliputi tahapan perancangan system, analisa kebutuhan perangkat lunak, perancangan, pengkodean (*coding*) dan pengujian. *Waterfall Model* memberikan pendekatan-pendekatan sistematis dan berurutan bagi pengembangan piranti lunak. Untuk itu penulis mencoba menggunakan model ini pada pengembangan perangkat [1].

2. Metode

2.1 Aplikasi Mobile

Aplikasi berasal dari kata *application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju, aplikasi mobile juga dapat di artikan sebagai sebuah aplikasi yang memungkinkan anda melakukan mobilitas dengan menggunakan perlengkapan seperti PDA, telepon seluler atau handphone [2].

2.2 SDLC *Waterfall Model*

System Development Life Cycle (SDLC) *Waterfall Mode* atau disebut juga *linear sequential model* atau *classic life cycle* adalah metode pengembangan perangkat lunak dengan pendekatan sekuensial dengan cakupan aktivitas [3].

1. Rekayasa sistem dan Analisis (Sistem Engineering and Analysis) Karena perangkat lunak adalah bagian dari sistem yang lebih besar, pekerjaan dimulai dari pembentukan kebutuhan-kebutuhan untuk seluruh elemen sistem dan kemudian memilah mana yang untuk pengembangan perangkat lunak. Hal ini penting, ketika perangkat lunak harus berkomunikasi dengan hardware, orang dan basis data.
2. Analisis kebutuhan perangkat lunak (*Software Requirements Analysis*) Pengumpulan kebutuhan dengan fokus pada perangkat lunak, yang meliputi : Domain informasi, fungsi yang dibutuhkan, unjuk kerja/performansi dan antarmuka. Hasilnya harus didokumentasi dan di review ke pelanggan
3. Perancangan (*Design*) Ada 4 atribut untuk program yaitu : Struktur Data, Arsitektur perangkat

lunak, Prosedur detil dan Karakteristik Antarmuka. Proses desain mengubah kebutuhan-kebutuhan menjadi bentuk karakteristik yang dimengerti perangkat lunak sebelum dimulai penulisan program. Desain ini harus terdokumentasi dengan baik dan menjadi bagian konfigurasi perangkat lunak.

4. Pembuatan kode (Coding) Penterjemahan perancangan ke bentuk yang dapat dimengerti oleh mesin, dengan menggunakan bahasa pemrograman
5. Pengujian (Testing) Setelah kode program selesai testing dapat dilakukan. Testing memfokuskan pada logika internal dari perangkat lunak, fungsi eksternal dan mencari segala kemungkinan kesalahan dan memeriksa apakah sesuai dengan hasil yang diinginkan.
6. Pemeliharaan (Maintenance) Merupakan bagian paling akhir dari siklus pengembangan dan dilakukan setelah perangkat lunak dipergunakan. Kegiatan :
 - a. Corrective Maintenance : Mengoreksi kesalahan pada perangkat lunak, yang baru terdeteksi pada saat perangkat lunak dipergunakan
 - b. Adaptive Maintenance : Penyesuaian dengan lingkungan baru, misalnya sistem operasi atau sebagai tuntutan atas perkembangan sistem komputer, misalnya penambahan printer driver
 - c. Perfektive Maintenance : Bila perangkat lunak sukses dipergunakan oleh pemakai. Pemeliharaan ditujukan untuk menambah kemampuannya seperti memberikan fungsi-fungsi tambahan, peningkatan kinerja dan sebagainya.

2.3 UML (Unified Modelling language)

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak [4].

UML adalah bahasa pemodelan standar atau kumpulan teknik-teknik pemodelan untuk menspesifikasi, mem-visualisasi, mengkonstruksi dan mendokumentasi hasil kerja dalam pengembangan perangkat lunak [5].

UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

2.4 JAVA

Java merupakan bahasa pemrograman berorientasi objek dan bebas platform, dikembangkan oleh SUN Micro System dengan sejumlah keunggulan yang memungkinkan Java dijadikan sebagai bahasa pengembangan enterprise [6].

Java lahir karena ketidakpuasan seorang insinyur di SUN Micro System bernama James Gosling. Ia tidak puas dengan compiler C++ yang ia gunakan untuk membuat software yang di embed pada peralatan elektronik karena dinilai terlalu banyak menghasilkan bug, biaya besar, sangat bergantung terhadap platform. Gosling merasa perlu membuat compiler baru sebagai solusi terhadap sejumlah kelemahan pada C++ tersebut. Kompiler tersebut diberi nama dengan Oak. Kompiler ini mirip dengan C++ tetapi dengan sejumlah pengurangan fitur yang dianggap kurang menguntungkan dalam pengembangan. Pada tahun 1994, Oak diubah namanya menjadi Java.

2.5 ANDROID

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis Linux yang mencakup sistem operasi, middleware dan aplikasi. Android merupakan generasi baru *platform mobile* menyediakan *platform* yang terbuka bagi para pengembang untuk melakukan pengembangan sesuai kebutuhan [7][8].

Sistem operasi yang mendasari Android dilisensikan dibawah GNU, General Public Lisensi Versi 2 (GPL.v2), yang sering dikenal dengan istilah "copyleft", lisensi dimana setiap perbaikan pihak ketiga harus terus jatuh dibawah terms. Android didistribusikan dibawah lisensi Apache Software (ASL/Apache2), yang memungkinkan untuk distribusi selanjutnya. Dalam pengembangan

aplikasi berbasis Android telah banyak menggunakan Eclipse dimana Eclipse adalah IDE (Integrated Development Environment) yang mendapat dukungan langsung dari Google untuk menjadi IDE pengembangan aplikasi android. Akan tetapi tidak akan menutup kemungkinan untuk menggunakan IDE yang lain seperti Netbeans untuk melakukan pengembangan android.

3. Analisis dan Hasil

3.1 Analisis

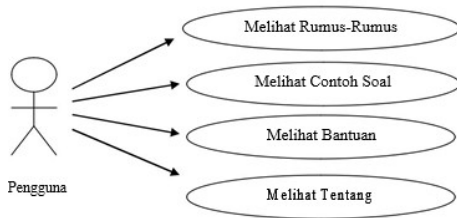
Seperti yang telah dibahas sebelumnya dalam alur kerja SDLC *Waterfall Model* diantaranya adalah tahapan analisis dan perancangan sistem. Dalam tahapan analisis akan dilakukan analisis spesifikasi kebutuhan pengguna yang digambarkan dalam bentuk *use case*. Tujuan dari tahapan ini adalah untuk mendapatkan pemahaman yang lebih mendalam tentang spesifikasi-spesifikasi kebutuhan perangkat lunak.

3.1.1 Analisa Kebutuhan

Analisa kebutuhan pengguna dilakukan untuk memperoleh fasilitas-fasilitas yang dibutuhkan pengguna dalam menggunakan perangkat lunak Aplikasi Kumpulan Rumus Matematika SMA ini. Maka berdasarkan kebutuhan pengguna dapat dihasilkan fasilitas-fasilitas sebagai berikut :

1. Fasilitas Kumpulan Rumus
Pada fasilitas ini terdapat berbagai kumpulan rumus matematika SMA yang merupakan kebutuhan utama yang harus dibuat pada aplikasi ini.
2. Fasilitas Contoh Soal
Dalam fasilitas contoh soal terdapat beberapa contoh soal yang diharapkan agar pengguna dapat lebih memahami penggunaan rumus-rumus matematika yang diaplikasikan kedalam bentuk contoh soal. Pada fasilitas ini terdapat penyelesaian soal dengan menggunakan rumus matematika yang telah dipilih sebelumnya.
3. Fasilitas Bantuan
Fasilitas ini dibuat dengan tujuan memandu *user* dalam menggunakan aplikasi ini. Pada fasilitas ini terdapat langkah-langkah cara penggunaan aplikasi ini.
4. Fasilitas Tentang
Pada fasilitas ini terdapat informasi tentang aplikasi dan pembuat aplikasi dimana diharapkan pengguna dapat

memberikan kritik dan saran untuk mengembangkan aplikasi ini menjadi lebih baik lagi dan kompleks.



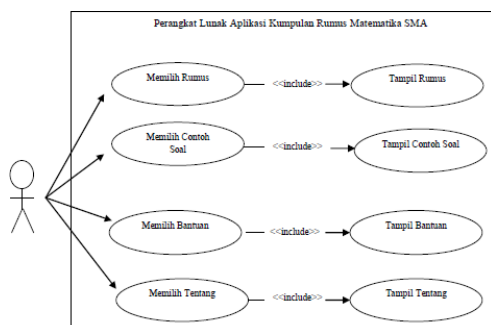
Gambar 1 Use case kebutuhan pengguna

Berdasarkan analisis terhadap kebutuhan pengguna dihasilkan beberapa point kebutuhan pengguna dalam menjalankan aplikasi yang dapat dilihat pada gambar1 use case kebutuhan pengguna diatas.

3.1.2 Perancangan Sistem

Setelah melakukan analisis terhadap kebutuhan sistem dan pengguna (user), maka tahap yang dilakukan selanjutnya adalah perancangan sistem. Berdasarkan hasil yang telah didapat pada proses analisis yang telah dilakukan sebelumnya maka dibuatlah perancangan yang mendeskripsikan sistem sebagai berikut :

1. Proses Interaksi Sistem Melihat Rumus-Rumus Melihat Contoh Soal Melihat Bantuan Melihat Tentang Pengguna Proses interaksi sistem adalah proses dimana terjadi interaksi antara pengguna dengan sistem. Pengguna akan melakukan permintaan (memilih event) sesuai dengan fasilitas yang tersedia, kemudian sistem merespon permintaan ataupun event yang dilakukan pengguna dengan menampilkan hasilnya pada user interface.



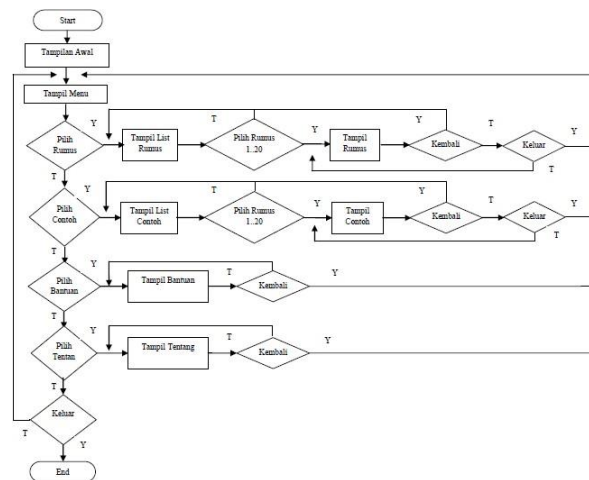
Gambar 2 Use case diagram sistem

Berdasarkan gambar 2 terlihat bahwa pengguna melakukan beberapa interaksi

dengan sistem seperti memilih rumus, memilih contoh soal, memilih bantuan dan memilih tentang. Kemudian sistem merespon dengan menghasilkan tampilan sesuai permintaan pengguna.

2. Alur Kerja Sistem

Dalam perancangan alur kerja sistem dirancang langkah-langkah proses kerja sistem yang akan dideskripsikan dalam bentuk *flowchart*. Proses alur kerja sistem dimulai dari memulai aplikasi, kemudian menampilkan menu utama dimana akan terdapat beberapa menu utama.

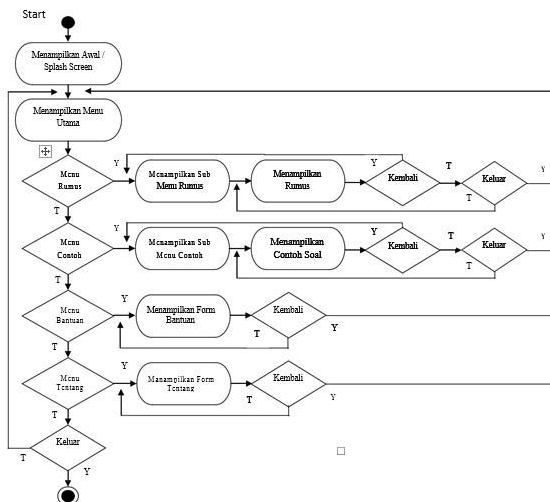


Gambar 3 Flowchart program

Pengguna akan dihadapkan pada kondisi pemilihan menu yang terdiri dari menu Rumus, menu Contoh Soal, menu Bantuan dan menu Tentang. Jika pengguna memilih salah satu menu maka akan dilanjutkan dengan alur berikutnya. Jika pengguna tidak memilih salah satu menu maka akan dihadapkan pada kondisi keluar dimana jika pengguna memilih untuk keluar dari aplikasi maka aplikasi akan dihentikan (selesai).

3. Struktur Menu

Struktur menu sistem dapat digambarkan pada *activity diagram* berikut:

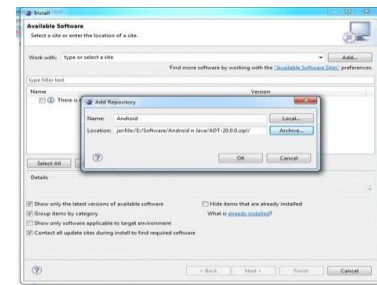


Gambar 4 Activity Diagram struktur menu

3.2 Hasil

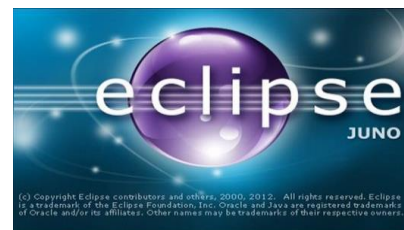
Setelah melakukan analisis dan perancangan sistem maka dilakukan tahap selanjutnya yaitu implementasi untuk mendapatkan hasil. Pada tahap implementasi dilakukan proses *coding* yaitu mengembangkan aplikasi kumpulan rumus matematika SMA menggunakan editor Eclipse dan bahasa pemrograman Java sesuai dengan perancangan yang telah dibuat sebelumnya. Dalam proses implementasi dilakukan beberapa tahap sebagai berikut :

1. Instalasi Software
Berikut adalah perangkat lunak yang diperlukan untuk membuat aplikasi Android :
 - a. Install Eclipse
 - b. Install Android SDK
2. Instalasi ADT pada Eclipse
Untuk membuat aplikasi Android menggunakan Eclipse, perlu menambahkan ADT pada Eclipse tersebut. Buka Eclipse > Help > Install New Software > Add. Pada dialog Add Site, Name diisi dengan Android dan Location dengan <https://dl-ssl.google.com/android/eclipse/> dan saat klik OK, maka akan muncul dialog Available Software kemudian diekstrak pada sistem.



Gambar 5 Tampilan instalasi ADT pada Eclipse

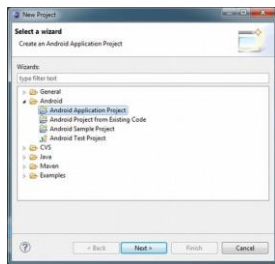
3. Menambah Platform Android
Android SDK yang telah di unduh hanya berisi tools utama saja, belum termasuk *platform* Android yang akan dijadikan sebagai media untuk mencoba menjalankan aplikasi yang dibuat di eclipse. Menambah platform Android dapat dilakukan dengan cara membuka Android SDK and AVD Manager. Kemudian memilih "Available Packages", expand "Android Repository" lalu mencentang salah satu platform yang tersedia, misalnya "SDK Platform Android 2.3, API 9 revision 2" dan mengklik "Install Selected". Proses instalasi akan memakan waktu beberapa saat tergantung dari jumlah dan besar paket yang telah dipilih.
4. Memulai Eclipse
Untuk memulai aplikasi Eclipse, jalankan aplikasi Eclipse yang ada di folder Eclipse, setelah aplikasi di jalankan maka akan muncul splash screen eclipse Seperti gambar berikutini:



Gambar 6 Proses awal eclipse dijalankan

Setelah beberapa saat splash screen berjalan, maka akan muncul tampilan untuk memilih

workspace atau folder kerja yang akan digunakan pada Eclipse. Beberapa saat setelah loading selesai, maka akan muncul tampilan Eclipse pertama. Untuk membuat project baru pada editor Eclipse dapat dilakukan dengan memilih File > New > Project. Kemudian memilih Android Application Project pada jendela New Project, klik "Next". Jika terdapat pilihan project Android, berarti ADT belum terinstal pada Eclipse.



Gambar 7 Tampilan pemilihan project

Kemudian memasukan nama project baru yang akan dibuat pada Project Name. Misalnya menuliskan project "RumusMatematika". Kemudian klik Next dan memasukan informasi sebagai berikut:

1. Application Name, masukan nama aplikasi yang akan di buat. Nantinya nama aplikasi ini yang akan menjadi judul aplikasi pada perangkat Android misalnya memasukan "Rumus Matematika" pada application name.
2. Package Name, package pada Java adalah sekumpulan source code yang dikumpulkan berdasarkan alamat tertentu. Nantinya setiap source code yang berada dalam package yang sama akan berada dalam satu folder dan setiap sub folder ditandai dengan tanda titik ".".
3. Pada Create Activity, masukan nama activity yang akan menjadi aktivitas pertama saat membuka aplikasi yang dibuat. (Activity merepresentasikan sebuah screen/tampilan tunggal pada aplikasi). Memasukan nama "SplashScreenAct" pada Create Activity.
4. Minimum SDK, versi SDK minimum yang diperlukan oleh perangkat yang ingin berjalan kiri aplikasi yang dibuat. Aplikasi dengan SDK 10 (Android 2.3.3) tidak akan bisa berjalan dalam SDK versi 8. Tetapi aplikasi dengan SDK 10 bisa dijalankan pada perangkat dengan SDK 15 (Android 4). Kemudian klik "Next". Maka akan

muncul tampilan konfigurasi. Pada menu select Build Target, pilih versi Android 2.3 atau *GingerBread*, klik Next.

Pada menu Configure Launcher Icon digunakan untuk memilih icon yang digunakan aplikasi yang di bangun. Klik "Next". Tampilan pembuatan activity Pada menu Create Activity digunakan untuk membuat activity secara default yang dimiliki Eclipse Juno. Klik "Next" Pada menu New Blank Activity digunakan untuk membuat activity secara default yang akan digunakan dalam aplikasi yang dibangun. Klik "Finish" untuk menyelesaikan pembuatan project baru dan akan muncul workbench eclipse. Setelah selesai membuat project baru maka dapat ditambahkan file, package maupun class baru misalnya dengan cara klik kanan pada project > New > Class. Setelah *project* selesai maka dilakukan proses kompilasi dan simulasi aplikasi pada *emulator* yang telah tersedia. Setelah melakukan *coding* program maka dihasilkan sebuah aplikasi kumpulan rumus matematika SMA .

Setelah *project* selesai maka dilakukan proses kompilasi dan simulasi aplikasi pada *emulator* yang telah tersedia. Setelah melakukan *coding* program maka dihasilkan sebuah aplikasi kumpulan rumus matematika SMA dengan tampilan sebagai berikut :

1. Tampilan Awal

Tampilan awal adalah tampilan yang muncul pertama sekali pada saat memulai. Aplikasi Kumpulan Rumus Matematika SMA Berbasis Android dimana terlihat tampilan awal (*splash screen*) yang berlangsung selama beberapa detik dan kemudian memunculkan tampilan menu utama. Pada sisi kanan atas form terdapat judul yang menunjukkan identitas form yang sedang tampil. Berdasarkan hasil yang diperoleh maka tampilan awal dapat dilihat pada gambar 8.



Gambar 8 Tampilan awal aplikasi

2. Tampilan Menu Utama

Setelah tampilan awal muncul selama beberapa detik maka selanjutnya tampilan yang muncul adalah tampilan Menu Utama. Pada tampilan menu utama terdapat beberapa button menu utama seperti Rumus, Contoh Soal, Bantuan dan Tentang. Tampilan menu utama yang dihasilkan dapat dilihat pada gambar 9.



Gambar 9 Tampilan menu utama

3. Tampilan List Rumus

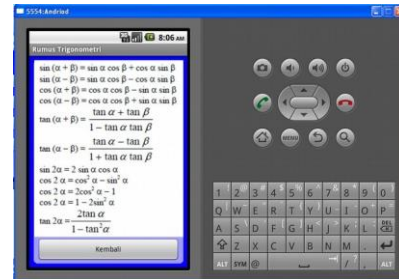
Pada tampilan list rumus terdapat beberapa daftar/list rumus matematika SMA seperti rumus trigonometri, logika matematika, matriks integral, turunan. Tampilan list rumus dapat dilihat pada gambar 10 berikut :



Gambar 10 Tampilan list rumus

4. Tampilan Rumus

Pada tampilan rumus akan muncul rumus matematika sesuai dengan rumus yang telah dipilih pada tampilan list rumus. Tampilan rumus dapat dilihat pada gambar 11 di bawah ini :



Gambar 11 Tampilan rumus

5. Tampilan List Contoh Soal

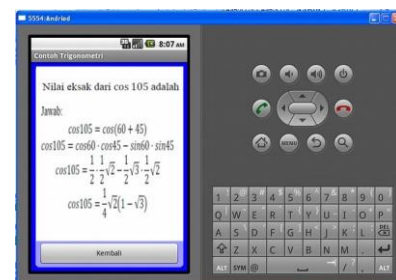
Tampilan list contoh soal sama dengan tampilan list rumus dimana terdapat pilhan rumus yang mewakili contoh soal. Tampilan list contoh soal dapat dilihat pada gambar 12 berikut :



Gambar 12 Tampilan list contoh soal

6. Tampilan Contoh Soal

Pada tampilan contoh soal terdapat soal sesuai dengan rumus yang telah dipilih dimana terdapat penyelesaiannya menggunakan rumus.



[6] Fikri, Rijalul. 2005. *Pemrograman Java*. Yogyakarta: Penertbit ANDI.

[7] Hermawan, Stephanus. 2011. *Mudah Membuat Aplikasi Android*. Yogyakarta: Penerbit ANDI.

[8] Sulihati.A, “Aplikasi Akademik online Berbasis Mobile Android Pada Universitas Tama Jagakarsa”,
Jurnal Sains dan Teknologi Teknik Utama, Vol XI,
Gambar 13 Tampilan contoh soal

Nomor 1, April 2016, ISSN 1978-001X

References

- [1] Pressman, Roger. 2000. *Software Engineering: Practitioner's Approach (Terjemahan)*. Yogyakarta : Penerbit ANDI.
- [2] Kosidin, Farizah.N.R, “Pemodelan Aplikasi Mobile Reminder Berbasis Android”, SENTIKA 2016, ISSN 2089-9815
- [3] Firmansyah.Y, Udi, “ Penerapan Metode SDLC Waterfall Dalam Pembuatan Sistem Informasi Akademik Berbasis Web Studi Kasus Pondok Pesantren Al-Habi Sholeh Kabupaten Kubu Raya, Kalimantan Barat”, Jurnal Teknologi & Manajemen Informatika, Vol. 14 No 1 2018
- [4] Pudjo, Prabowo. 2011. *Menggunakan UML*. Bandung: INFORMATIKA.
- [5] HM, Jogiyanto. 2005. *Analisis dan Desain*. Yogyakarta: Penerbit ANDI