

PERANCANGAN APLIKASI PEMBELAJARAN BANGUN DATAR DAN RUANG UNTUK SEKOLAH DASAR MENGGUNAKAN ANDROID

Desmira¹, Didik Aribowo², Poppy Oktavia³

¹Program Studi Teknik Informatika – STIMIK NUSAMANDIRI

²Jurusan Pendidikan Teknik Elektro, Jurusan Teknik Elektro - UNTIRTA

¹desmira.dma@bsi.ac.id ²aribowo82@yahoo.co.id ³vhivhi42@rocketmail.com

Abstrak - Seiring dengan berkembangnya era zaman dan teknologi yang semakin hari semakin canggih, disini penulis ingin mengembangkan manfaat dari sebuah *handphone* genggam atau telepon seluler. Dimana awalnya hanya sebagai sebuah alat komunikasi yang nantinya akan dikembangkan menjadi sebuah tempat sarana penyimpanan informasi yang efektif dan efisien. Di dalam penulisan ini, penulis ingin mencoba membuat sebuah perancangan aplikasi yang membahas tentang bangun datar dan ruang sehingga dapat membantu anak-anak yang kesulitan dalam memahami dan mempelajari suatu bangun datar dan ruang. Aplikasi ini dibuat dengan menggunakan sistem operasi *Android*. Penulis juga mencoba untuk merancang sebuah aplikasi ponsel sendiri tanpa harus mengeluarkan biaya yang besar, karena *Android* adalah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware*, aplikasi dan *open source*. *Android* memudahkan para programmer dalam pembuatan program, karena pembuatan tampilan dibuat terpisah dengan program. *Android* juga menyediakan *plugin* untuk *Eclipse*. *Eclipse* dapat berjalan diatas beberapa sistem operasi. Dan juga dalam pembuatan aplikasi ini, penulis menggunakan komponen-komponen pada *Android SDK (Software Development Kit)* dan *API (Application Programming Interface)* yang menggunakan pemrograman *java*.

Kata Kunci : *Android, Eclipse, Java*, bangun datar dan ruang

I. PENDAHULUAN

Seiring dengan perkembangan zaman baik dari segi teknologi maupun dari segi lainnya, khususnya dibidang pendidikan maka dibutuhkan sebuah terobosan atau inovasi baru dalam membantu seorang siswa-siswi dalam belajar.

Pembelajaran matematika pada anak-anak merupakan suatu permasalahan yang sangat menarik, karena adanya perbedaan karakteristik antara hakikat anak dengan hakikat matematika[3]. Anak pada usia dini sedang mengalami perkembangan dalam tingkat berpikirnya, karena tahap berpikir mereka masih belum formal. Anak-anak dalam rentang usia 5 hingga 7 tahun, masih berada pada tahapan pra-operasional. Di lain pihak, matematika adalah ilmu yang bersifat abstrak. Mengingat adanya perbedaan karakteristik tersebut, maka diperlukan adanya metode khusus untuk menjembatani antara dunia anak yang berpikir secara konkret untuk dapat mengerti dunia matematika yang bersifat abstrak.

Mengingat luas dan begitu kompleksnya dunia aplikasi komputer, disini penulis hanya ingin menyajikan suatu aplikasi *mobile* yang sangat membantu siswa-siswi semua dalam belajar. Yaitu suatu gagasan untuk membuat sebuah Perancangan Aplikasi Pembelajaran Bangun Datar dan Ruang untuk Sekolah Dasar menggunakan *Android* yang bersifat *mobile*. Salah satu perangkat *mobile* yang kemajuannya sangat pesat adalah *handphone*, dimana dulu kegunaannya adalah hanya untuk komunikasi, sekarang *handphone* dapat dapat berguna layaknya sebuah komputer yang dapat di instal berbagai macam aplikasi. Kehadiran aplikasi *mobile* ini ditujukan sebagai pelengkap pembelajaran yang ada, serta memberikan kesempatan untuk lebih mengerti dan

memahami kembali materi yang kurang dikuasai ini dimanapun dan kapanpun itu. Hal tersebut tentu memberikan pengalaman yang berbeda bagi pengguna aplikasi ini, untuk itu pengembangan aplikasi android ini sebagai media pembelajaran matematika sangat perlu untuk dikembangkan.

II. METODOLOGI PENELITIAN

Untuk mengimplementasikan aplikasi yang akan dibuat penulis memerlukan analisa untuk kebutuhan dalam perancangannya yaitu:

- Analisa kebutuhan. Perancangan aplikasi ini membutuhkan *hardware* dan *software* sebagai media pendukungnya. *Hardware* dan *software* yang digunakan dalam perancangan aplikasi adalah OS Windows 7 *Ultimate*, *Eclipse*, ADT (*Android Development Tools*), *Android SDK (Standart Development Kit)* dan *Java JDK (Java Development Kit)*.
- Desain. *Eclipse* merupakan editor tempat membuat desain dan *coding*, sedangkan konsep OOP (*Object Oriented Program*) adalah konsep pemrograman *java*.
- Testing. Program yang sudah dibuat penulis menggunakan AVD (*Android Virtual Device*) yang merupakan *emulator* untuk menjalankan program aplikasi *android*.
- Implementasi. Jika pengujian pada AVD tidak ada kesalahan maka aplikasi akan di install kedalam *gadget* dengan sistem operasi *Android*.

III. LANDASAN TEORI

1. Tinjauan Jurnal

Berdasarkan jurnal yang dijadikan sebagai acuan penulis pada skripsi ini, maka dijadikan beberapa jurnal sebagai jurnal referensinya.

Pembelajaran adalah proses interaksi antara peserta didik dengan pendidik dan sumber belajar pada suatu lingkungan belajar. Pembelajaran merupakan bantuan yang diberikan pendidik agar dapat terjadi proses perolehan ilmu dan pengetahuan, penguasaan kemahiran dan tabiat, serta pembentukan sikap dan kepercayaan pada peserta didik [7]. Dengan kata lain, pembelajaran adalah proses untuk membantu peserta didik agar dapat belajar dengan baik. Pembelajaran yang dilakukan saat ini masih banyak yang menggunakan pembelajaran bersifat konvensional, dimana para guru hanya berdiri menulis di papan tulis kemudian siswa diminta mencatat dan menghafal materi yang disampaikan tadi. Pembelajaran seperti ini cenderung membuat para siswa jenuh, sehingga siswa kurang berminat untuk mengikuti pembelajaran tersebut. Sehingga para siswa kurang berprestasi, dan mereka cenderung lebih suka bermain-main dari pada mengikuti pelajaran.

Android adalah sistem operasi berbasis *Linux* yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. *Android* awalnya dikembangkan oleh *Android, Inc.*, dengan dukungan finansial dari *Google*, yang kemudian membelinya pada tahun 2005 [1].

2. Konsep Dasar Program

Konsep dasar program yang digunakan dalam pembuatan *android* adalah:

a) *Android*

Android merupakan generasi baru *platform mobile*, *platform* yang memberikan pengembang untuk melakukan pengembangan sesuai dengan yang diharapkannya. Sistem operasi yang mendasari *Android* dilisensikan di bawah GNU, *General Public Lisensi Versi 2 (GPLv2)*, yang sering dikenal dengan istilah "*copyleft*" lisensi di mana setiap perbaikan pihak ketiga harus terjual di bawah terms. *Android* didistribusikan di bawah Lisensi *Apache Software (ASL/Apache2)*, yang memungkinkan untuk distribusi kedua dan seterusnya [3].

Komersialisasi pengembang (produsen handset khususnya) dapat memilih untuk meningkatkan *platform* tanpa harus memberikan perbaikan mereka ke masyarakat *Open source*. Sebaliknya, pengembang dapat keuntungan dari perangkat tambahan seperti perbaikan dan mendistribusikan aplikasi mereka dibawah skema *lisensi* apapun yang mereka inginkan.

Pengembang memiliki beberapa pilihan ketika membuat aplikasi yang berbasis *android*. Kebanyakan pengembang menggunakan *Eclipse* yang tersedia secara bebas untuk merancang dan mengembangkan aplikasi *Android*. *Eclipse* adalah

IDE yang paling populer untuk pengembangan *Android*, karena memiliki *Android plug-in* yang tersedia untuk memfasilitasi pengembang *Android*. Selain itu *Eclipse* juga mendapat dukungan langsung dari *Google* untuk menjadi IDE pengembang aplikasi *Android*, ini terbukti dengan adanya penambahan *plug-in* untuk *Eclipse* untuk membuat *project android* di mana *source software* langsung dari situs resminya *Google*. Tetapi hal diatas tidak menutup kemungkinan untuk menggunakan IDE yang lain seperti *Netbeans* untuk melakukan pengembang *android*.

b) *Dalvik Virtual Machine (DVM)*

Salah satu elemen kunci dari *Android* adalah *Dalvik Virtual Machine (DVM)*. *Android* berjalan di dalam *Dalvik Virtual Machine (DVM)* bukan di *Java Virtual Machine (JVM)*, sebenarnya bnyak persamaannya dengan *Java Virtual Machine (JVM)* seperti *Java ME (Java Mobile Edition)*, tetapi *Android* menggunakan *Virtual Machine* sendiri menurut saya dikustomisasi dan dirancang untuk memastikan bahwa beberapa *feature-feature* berjalan lebih efisien pada perangkat *mobile* [3]

Dalvik Virtual Machine (DVM) adalah "*register bases*" sementara *Java Virtual Machine (JVM)* adalah "*stack based*", *DVM* didesain dan ditulis oleh *Dan Bornten* dan beberapa *engineers Google* lainnya. Jadi bisa kita katakan "*Dalvik equals (Java) == False*". *Dalvik Virtual Machine* menggunakan kernel *Linux* untuk menangani fungsionalitas tingkat rendah termasuk keamanan, *threading*, dan proses serta manajemen memori.

Semua *hardware* yang berbasis *android* dijalankan dengan menggunakan *Virtual Machine* untuk eksekusi aplikasi, pengembang tidak perlu khawatir tentang implementasi perangkat keras tertentu. *Dalvik Virtual Machine* mengeksekusi *executable file*, sebuah format yang dioptimalkan untuk memastikan memori yang digunakan sangat kecil. *The executable file* diciptakan dengan mengubah kelas bahasa *java* dan kompilasi menggunakan *tools* yang disediakan dalam SDK *Android*.

c) *Android SDK (Software Development Kit)*

Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada *platform Android* menggunakan bahasa pemrograman *Java*. *Android* merupakan subset perangkat lunak untuk ponsel yang meliputi sistem sistem operasi, *middleware* dan aplikasi kunci di *realese* oleh *Google* [6].

d) *ADT (Android Development Tools)*

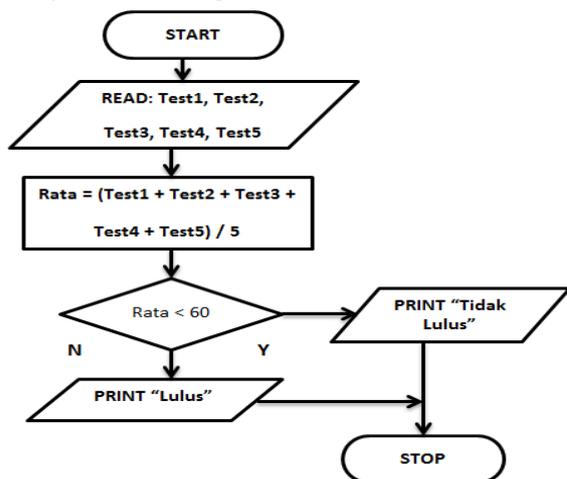
Android Development Tools (ADT) adalah *plug-in* yang didesain untuk IDE *Eclipse* yang memberikan kemudahan dalam mengembangkan aplikasi *android* dengan menggunakan IDE *Eclipse* [3].

3. Metode Algoritma

Algoritma merupakan Teknik penyusunan langkah-langkah penyelesaian masalah dalam bentuk kalimat dengan jumlah kata terbatas tetapi tersusun secara logis dan sistematis. Algoritma juga merupakan suatu prosedur yang jelas untuk menyelesaikan suatu persoalan dengan menggunakan langkah-langkah tertentu dan terbatas jumlahnya.

Flowchart adalah untaian simbol gambar (*chart*) yang menunjukkan aliran (*flow*) dari proses terhadap data.

Sebagai contoh menghitung nilai ratanya dan memutuskan siswa yang lulus atau tidak, membaca nilai dari 5 macam test seorang siswa kemudian menghitung nilai ratanya dan memutuskan apakah siswa ini lulus atau tidak. Siswa lulus apabila nilai ratanya adalah sebagai berikut:



Gambar 1. Diagram Flowchart

4. Pengujian Software

Didalam pengujian *software* terdapat beberapa pendekatan sebagai berikut:

- a. *Black box testing* (pengujian kotak hitam) menyimpulkan bahwa : *Black box testing* adalah tipe testing yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para tester memandang perangkat lunak seperti layaknya sebuah “kotak hitam “ yang tidak penting dilihat isinya, tapi cukup dikenai proses testing dibagian luar[5].
- b. *White box testing* (pengujian kotak putih) menyimpulkan bahwa : *White box testing* secara umum merupakan jenis testing yang lebih berkonsentrasi terhadap “isi” dari perangkat lunak itu sendiri. Jenis ini lebih banyak berkonsentrasi kepada *source code* dari perangkat lunak yang dibuat sehingga membutuhkan proses testing yang jauh lebih lama dan lebih “mahal” dikarenakan membutuhkan ketelitian dari para testeeer serta kemampuan teknis pemrograman bagi para testernya[5].

5. Peralatan Pendukung

Membuat suatu aplikasi *android* diperlukan berbagai macam pendukung yang dapat membantu

terbentuknya aplikasi mulai dari logika hingga konsep pemrogramannya.

1. OOP (*Object Oriented Program*)

OOP merupakan sebuah metodologi dalam pemrograman yang diciptakan untuk memodelkan kasus-kasus nyata ke dalam sebuah objek[4].

Sebelum melangkah lebih jauh ke pembahasan lainnya, perlu diketahui konsep dan ciri-ciri atau karakteristik dari OOP. Adapun karakteristik tersebut adalah abstraksi, pembungkusan, pewarisan, dan polimorfisme.

- a. Abstraksi (*Abstraction*). Abstraksi merupakan ciri yang paling mendasar dari OOP. Pasalnya, ini adalah suatu proses menyembunyikan kerumitan (pengabstrakan) yang terjadi dalam suatu objek sehingga pengguna objek tidak perlu untuk mengetahui detail proses yang dilakukan.
- b. Pembungkusan (*Encapsulation*). Pembungkusan artinya data-data dan method akan dibungkus menjadi paket objek yang merupakan satu-kesatuan sehingga dapat bekerja sama dalam melaksanakan tugas-tugas program tertentu.
- c. Pewarisan (*Inheritance*). Pewarisan merupakan sebuah objek yang dapat diturunkan menjadi objek baru lainnya sehingga objek baru tersebut akan mewarisi sifat dari objek induknya. Konsep inilah yang dinamakan dengan pewarisan (*inheritance*).
- d. Polimorfisme (*polymorphism*). Polimorfisme merupakan kegiatan mengungkap beberapa hal yang berbeda melalui satu cara yang sama.

2. UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak paradigma ‘berorientasi objek[2]. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipahami dan dipelajari.

Dalam aplikasi ini, program yang digunakan adalah *Activity*, *Use case*, *Sequence*, *Class*, dan *Deployment*.

- a. *Activity Diagram*. *Activity diagram* menggambarkan aliran fungsionalitas sistem. Pada tahap pemodelan bisnis, diagram aktivitas dapat digunakan untuk menunjukkan aliran kerja bisnis (*business work flow*).
- b. *Use Case Diagram*. *Use case diagram* menyajikan interaksi antara *use case* dan aktor. *Use case* menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai.
- c. *Sequence Diagram*. menjelaskan bahwa “*Sequence diagram* digunakan untuk menunjukkan aliran fungsionalitas dalam *use case*”.
- d. *Class Diagram*. *Class Diagram* menunjukkan interaksi antar kelas dalam sistem.

- e. *Deployment Diagram*. *Deployment Diagram* menampilkan rancangan fisik jaringan dimana berbagai komponen akan terdapat disana.

I. TEKNIK PENGUMPULAN DATA

Metode yang dipakai oleh penulis didalam ini dimulai dengan mengumpulkan berbagai *source code* dan materi dari internet, buku-buku program dan mencari gambar-gambar yang dibutuhkan. Adapun tahap-tahap yang dikerjakan oleh penulis dimulai dari: Metode Pengamatan (*Observation*)

1. Metode observasi merupakan perbandingan dari semua aplikasi yang saya buat dengan aplikasi yang ada sebelumnya.
2. Metode Wawancara (Interview)
Teknik pengumpulan data dengan cara mengadakan wawancara secara langsung dengan pihak-pihak yang terkait seperti guru, dengan masalah yang diteliti yaitu dengan cara Tanya jawab langsung dan sistematis dengan pengguna.
3. Metode Studi Pustaka (Library)
Suatu metode yang digunakan untuk memperoleh data dengan cara mengumpulkan data yang dilakukan secara teoritis untuk mendapatkan data dan informasi-informasi yang diperlukan dengan cara membaca, mempelajari literature yang berhubungan dengan masalah yang dibahas. Dalam penulisan dan penyusunan skripsi ini, penulis memperoleh data dari berbagai buku-buku, berkas-berkas, laporan yang berkaitan dengan judul yang diangkat sebagai referensi dari jurnal, internet dan buku yang berhubungan dengan pengumpulan data penulis butuhkan sebagai bahan perlengkapan analisa perbandingan penulis.
4. Metode Pengembangan Sistem yaitu suatu cara yang digunakan untuk melakukan suatu pendekatan sistem dasar dalam memecahkan suatu masalah.

II. ANALISA KEBUTUHAN

Tahap analisa kebutuhan mencakup *hardware*, *software*, aplikasi dan output yang digunakan adlaah sebagai berikut:

a. Komponen *Hardware*

Komputer yang digunakan mempunyai klasifikasi sebagai berikut.

1. Vendor : Notebook 10"
2. Tipe : Lenovo 20424
3. RAM : 2.00 GB
4. Processor : Intel ® Celeron ®
5. Graphic : Intel ®
6. CPU : N2830 @ 2.16GHZ
7. Os : Windows 8,1 Pro

Dalam pembuatan aplikasi android minimal *dual core* karena akan berpengaruh terhadap pembuatan virtual android yang akan digunakan.

Perangkat Gadget yang digunakan penulia mempunyai klasifikasi sebagai berikut:

- 1) Vendor : Handphone 5"

- 2) Tipe : Vandroid S5E Pro
- 3) Procesor : Cortex A7 Dual Core 1.3GHz
- 4) RAM : 512MB
- 5) Sd Internal: 4Gb
- 6) Resolution: 5"

b. Komponen *Software*

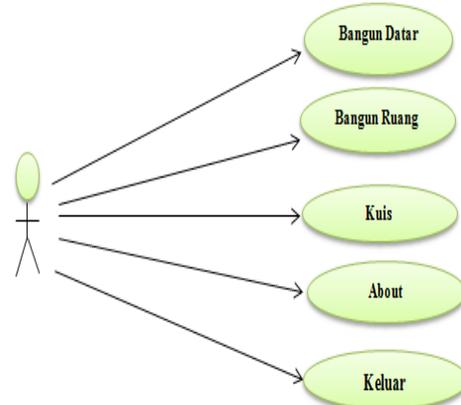
Komponen perangkat lunak yang digunakan untuk membuat *system* tersebut sebagai berikut : Penulis menggunakan sistem operasi Windows 8,1 Pro sebagai system operasi dalam perancangan aplikasi pembelajaran bangun datar dan bangun ruang ini

III. PERANCANGAN

1. Software Architecture

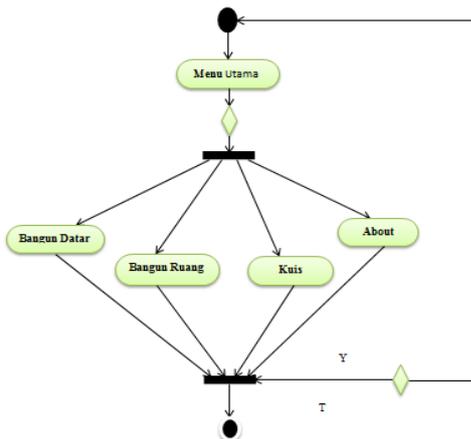
Rekayasa perangkat lunak merupakan suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu analisa kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna, desain, pengkodean, pengujian sampai pemeliharaan sistem setelah digunakan.

- a) Diagram *Use Case*. *Use case* diagram atau diagram *use case* merupakan deskripsi lengkap tentang bagaimana sistem/perangkat lunak berperilaku untuk para *actor*-nya. Dengan demikian *use case* diagram merupakan deskripsi lengkap tentang interaksi yang terjadi antara para *actor* dengan sistem/perangkat lunak yang sedang dikembangkan.



Gambar 2. Diagram *Use Case* Bangun Datar dan Ruang
Menggambarkan dimana menu awal aplikasi pertama kali pengguna akan menemui menu utama yang terdiri dari lima *button* menu yaitu *button* menu bangun datar, bangun ruang, kuis, about, dan keluar. Yang mana pada masing-masing *button* memiliki bagian tersendiri.

- b) Diagram *Activity* Aplikasi Menu Utama Bangun Datar dan Bangun Ruang

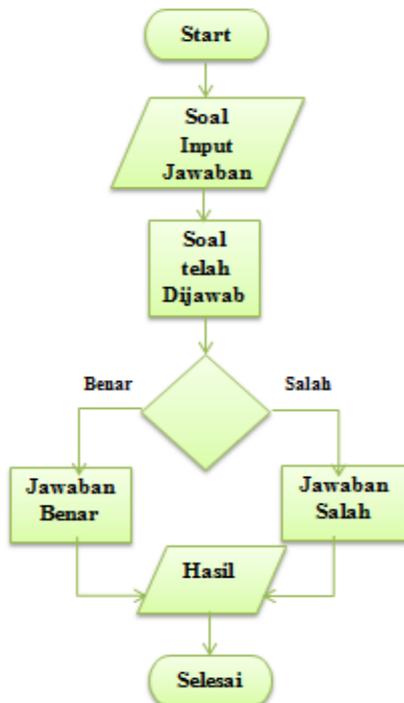


Gambar 3. Diagram Activity Menu Utama Keseluruhan

Diagram ini menggambarkan mengenai *activity* dari menu utama sendiri yang terdapat pada menu awal. Disini pengguna dapat melihat kegunaan masing-masing pada bangun datar dan bangun ruang tersebut.

2. Implementasi

Flowchart Aplikasi Pembelajaran Bangun Datar dan Bangun Ruang menggunakan bahasa pemrograman *java* yang merupakan bahasa dasar dari aplikasi Pembelajaran Bangun Datar dan Bangun Ruang, berikut adalah *flowchart* dari aplikasi Pembelajaran Bangun Datar dan Bangun Ruang pada menu kuis.

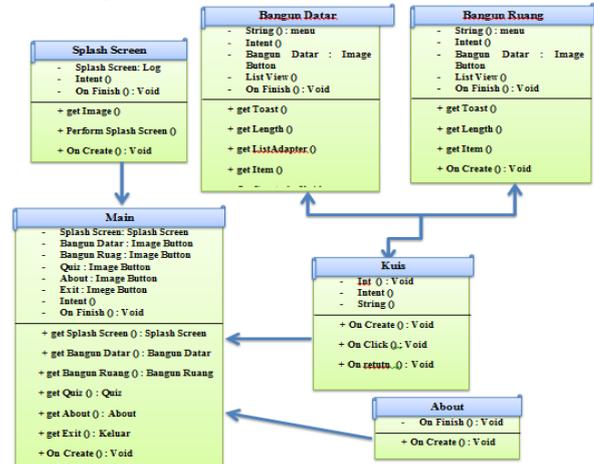


Gambar 4. Flowchart Aplikasi pada menu kuis

Eclipse merupakan salah satu *software* yang menggunakan bahasa dasar *java* yang membentuk OOP bersifat (*Object Oriented Programming*), artinya sebuah program yang mempunyai objek-objek tertentu dalam pemrograman. Selain itu *eclipse* adalah sebuah program yang bersifat visual, artinya mempunyai tampilan grafik-grafik (*Grafik User Interface*) yang mudah dimengerti oleh pemula sekalipun.

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Pada *class* diagram dapat disimpulkan bahwa ketika program dijalankan, akan ditampilkan GUI, dan ketika *user* tersebut menekan karakter pada *keyboard* maka program akan langsung menjalankan *class* bangun datar dan bangun ruang.



Gambar 5. Diagram Class

Antarmuka pengguna (*user interface*) merupakan mekanisme komunikasi Antara pengguna (*user*). Tujuan utamanya adalah mengkomunikasikan fitur-fitur yang terdapat pada system agar *user* mengerti dan dapat menggunakan aplikasi dengan baik.

Arsitek *user interface* pada aplikasi android meliputi *Activity* dan *user interface* yang terdiri dari komponen. Semuanya yang berhubungan dengan *user interface* pada aplikasi android terdapat pada lokasi *res/layout/filename.xml*. Adapun *interface* yang digunakan dalam aplikasi ini adalah sebagai berikut:



Gambar 6. Tampilan Loading

Pada tampilan menu utama, terdapat lima *button* dan penulis meletakkan tombol *button* agar pengguna lebih mudah memilih. Penulis juga menggunakan gambar latar berwarna agar lebih terlihat menarik dan *button*nya juga menggunakan gambar agar terlihat lebih menarik.

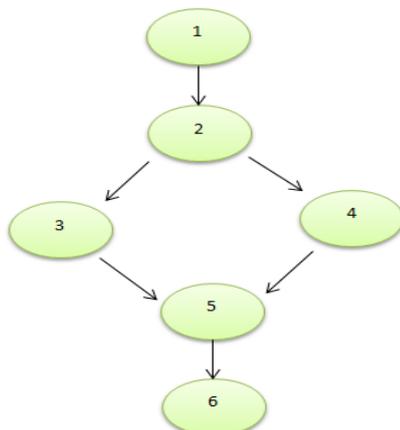


Gambar 7. Tampilan Menu Utama

5.1. Testing

Testing menggunakan *white box* dan *black box* untuk pengujian *whitebox*. Dengan menggunakan skema diagram alir, berikut merupakan diagram alir dari aplikasi pembelajaran bangun datar dan bangun ruang pada menu kuis.

1. White Box



Gambar 8. Skema Diagram Air

Kompleksitas siklomatis dari grafik alir dapat diperoleh dengan perhitungan:

$$V(G) = E - N + 2$$

Dimana:

V(G) = Jumlah region

E = jumlah Edge yang ditentukan dengan gambar panah

N = Jumlah simpul grafik alir ditentukan dengan gambar lingkaran

Sehingga didapat:

$$V(G) = 6 - 6 + 2 = 2$$

V(G) < 10 berarti memenuhi syarat kekompleksitasian siklomatisnya.

Baris set yang dihasilkan dari jalur independent adalah sebagai berikut:

a. 1 - 2 - 3 - 5 - 6

b. 1 - 2 - 4 - 5 - 6

Ketika aplikasi dijalankan, maka terlihat bahwa satu set baris yang dihasilkan adalah 1 - 2 - 3 - 5 - 6 - 1 - 2 - 4 - 5 - 6 dan terlihat bahwa simpul telah dieksekusi satu kali.

2. Black Box

Pengujian selanjutnya dilakukan untuk memastikan bahwa suatu *event* atau masukan menjelaskan proses yang tepat dan menghasilkan *output* yang sesuai dengan rancangan.

Tabel 1 Pengujian Black Box

Input	Proses	Output	Validasi
Button Splash	Menampilkan Layout Splash	Tampil Layout Splash	Sesuai
Button Main	Menampilkan Main menu utama	Tampil Layout Main Menu Utama	Sesuai
Button Bangun Datar	Menampilkan ListView Bangun Datar	Tampil Layout ListView Bangun Datar	Sesuai
Button Bangun Ruang	Menampilkan ListView Bangun Ruang	Tampil Layout ListView Bangun Ruang	Sesuai
Button Kuis	Menampilkan Layout Kuis	Tampil Layout Kuis	Sesuai
Button About	Menampilkan Layout About	Tampil Layout About	Sesuai
Tombol Exit	Menampilkan Message box keluar	Tampil message box pilihan keluar atau tidak	Sesuai

V. KESIMPULAN

Setelah aplikasi ini dibuat dapat disimpulkan bahwa :

1. Android merupakan aplikasi yang sudah tak asing lagi dikalangan masyarakat umum sekarang apalagi bagi anak-anak, baik anak kecil, maupun orang dewasa.
2. Android merupakan Sistem Operasi yang mendukung penulisan *script* dalam bahasa Java.
3. Android merupakan penggabungan antara *script* XML dan Java.
4. Tidak terlalu sulit dalam mendesain dan mengaplikasikan aplikasi tersebut di dalam sistem berbasis android.
5. Tidak semua orang (pelajar) dapat dengan mudah mempelajari dan membedakan bentuk-bentuk dari bangun datar dan bangun ruang itu sendiri.
6. Dapat membantu dan mempermudah seseorang atau pelajar dalam mempelajari, memahami, dan mengetahui dasar-dasar dari bentuk bangun datar dan bangun ruang itu sendiri.

VI. DAFTAR PUSTAKA

[1].DamarullahWandy, Amir Hamzah, dan Uning Lestari. 2013. Aplikasi Pengenalan dan pembelajaran bahasa korea (*Hangeul*) Berbasis Android. ISSN: 2338-6304. Vol 1. No. 1 Desember 2013

[2] Hermawan, Stephanus, *Mudah Membuat Aplikasi Android*, Andi Publiser, Jakarta, 2011.

[3].Nugroho, Adi. 2010. *Rekayasa Perangkat Lunak Berorientasi Objek dengan Metode USDP*. Yogyakarta: Andi

[4].Nurani, Ainindan Nugrahardi Ramadhani. 2014. *Perancangan buku Interaktif jarimatika penjumlahan dan pengurangan sebagai altenatif pembelajaran matematika untuk anak usia 5-7 tahun*. ISSN: 2337-3520. Vol 3. No. 1 2014

- [5].Raharjo, Budi. 2010.Teknik pemrograman pascal. Bandung: Informatika
- [6].Rizky,Soetam. 2011. Konsep Dasar Rekayasa Perangkat Lunak. Jakarta:Prestasi Pustaka.
- [7].Safaat, Nazruddin.2014. Android Pemrograman aplikasi *mobile smartphone* dan *tablet pc* berbasis android. Bandung: informatika
- [8].Setiawan, Eko. 2014. Perancangan media pembelajaran interaktif berbasis komputer untuk pembelajaran matematika tingkat SMP dengan metode *computer assisted instruction*. ISSN: 2301-9425. Vol VI. No.2 April 2014