

APLIKASI RUMUS MATEMATIKA SMA BERBASIS *MOBILE*

Yana Herdiana¹

¹ Teknik Informatika – Universitas Komputer Indonesia

Jl. Dipatiukur 112-114 Bandung

Email : yana.herdiana29@gmail.com¹

ABSTRAK

Salah satu pelajaran yang dipelajari di SMA yaitu pelajaran matematika. Waktu yang digunakan untuk belajar matematika di sekolah hanya 3-4 jam dalam seminggu dan waktu tersebut dirasa kurang untuk dapat memahami pelajaran matematika yang diajarkan. Pada umumnya pelajaran matematika hanya bisa didapatkan di sekolah dan tempat-tempat tertentu saja. Berdasarkan hasil perhitungan kuisioner penelitian dapat disimpulkan bahwa pelajaran matematika di SMA kurang diminati oleh siswa-siswi SMA.

Untuk membantu siswa dalam mempelajari matematika bisa dilakukan dengan beberapa cara salah satunya yaitu dengan aplikasi *mobile* berbasis android sebagai alat bantu pembelajaran matematika yang bersifat *moveble* (bisa dibawa kemana-mana). Aplikasi *mobile* yang dibangun akan memberikan informasi berupa informasi teori dan rumus, informasi contoh soal, dan informasi latihan soal yang bisa dikerjakan oleh siswa. Aplikasi ini dikhususkan untuk siswa SMA kelas X, kelas XI, dan Kelas XII.

Berdasarkan hasil pengujian, maka dapat diperoleh kesimpulan bahwa aplikasi yang dibangun dapat membantu siswa dalam mempelajari pelajaran matematika, sehingga siswa bisa menambah jam belajar diluar jam sekolah. Selain itu, dengan aplikasi ini minat belajar siswa terhadap pelajaran matematika semakin meningkat dari sebelumnya.

Kata kunci : Aplikasi, Rumus, Matematika, Android, SMA

1. PENDAHULUAN

Matematika merupakan salah satu mata pelajaran yang wajib diikuti oleh siswa yang mengenyam pendidikan di bangku SMA. Untuk dapat memahami pelajaran matematika diperlukan banyak belajar dan latihan bila dibandingkan dengan pelajaran yang lain. Waktu yang digunakan untuk belajar matematika di sekolah hanya 3-4 jam dalam seminggu dan waktu tersebut dirasa kurang untuk dapat memahami pelajaran matematika yang diajarkan. Untuk menambah waktu belajar tersebut, banyak siswa yang mengikuti bimbingan belajar

diluar sekolah dengan tambahan biaya lagi diluar biaya sekolah. Pada umumnya pelajaran matematika hanya bisa didapatkan di sekolah dan tempat-tempat tertentu saja. Berdasarkan hasil perhitungan kuisioner penelitian dapat disimpulkan bahwa pelajaran matematika di sekolah menengah atas kurang diminati oleh siswa-siswi SMA.

Dengan tuntutan seperti yang terdapat pada paragraf diatas, maka untuk membantu siswa dalam pembelajaran matematika dibutuhkan berbagai alat bantu salah satunya dengan alat bantu yang berbentuk aplikasi *mobile* yang bersifat *moveble* (mudah dibawa kemana-mana). Sistem operasi android merupakan sistem operasi *opensource* sehingga memungkinkan para pengguna dan pengembang aplikasi dapat menciptakan aplikasi berbasis *mobile* android sesuai dengan keinginan mereka sendiri. Beberapa *smartphone* yang menggunakan sistem operasi android memiliki harga yang lebih murah bila dibandingkan dengan *smartphone* yang menggunakan sistem operasi yang lain.

Maksud dari penelitian ini yaitu untuk mengembangkan aplikasi rumus matematika berbasis *mobile*, sebagai alat bantu siswa dalam mempelajari pelajaran matematika baik itu di sekolah maupun diluar sekolah.

Adapun tujuannya sebagai berikut:

1. Dapat membangun aplikasi rumus matematika berbasis *mobile* sebagai alat bantu pembelajaran matematika khususnya siswa SMA.
2. Dengan aplikasi rumus matematika SMA berbasis *mobile* yang bersifat *moveable* dan lebih simpel, maka diharapkan khususnya siswa SMA dapat belajar dimanapun dan kapanpun untuk menambah waktu belajar mereka diluar jam sekolah.
3. Dengan aplikasi rumus matematika SMA berbasis *mobile* ini diharapkan bisa meningkatkan minat siswa untuk belajar matematika.

1.1 Aplikasi *Mobile*

Pengertian aplikasi adalah program yang digunakan orang untuk melakukan sesuatu pada sistem komputer. *Mobile* dapat diartikan sebagai

perpindahan yang mudah dari satu tempat ke tempat yang lain, misalnya telepon *mobile* berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi. Sistem aplikasi *mobile* merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadipemutusan atau terputusnya komunikasi. Aplikasi ini dapat diakses melalui perangkat nirkabel seperti pager, seperti telepon seluler dan PDA. Adapun karakteristik perangkat *mobile* yaitu[9]:

1. Menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.
2. Memory yang terbatas : Perangkat *mobile* juga memiliki memory yang kecil, yaitu *primary (RAM)* dan *secondary (disk)*.
3. Daya proses yang terbatas : Sistem *mobile* tidaklah setangguh rekan mereka yaitu *desktop*.
4. Mengonsumsi daya yang rendah : Perangkat *mobile* menghabiskan sedikit daya dibandingkan dengan mesin desktop
5. Kuat dan dapat diandalkan : Karena perangkat *mobile* selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapi benturan-benturan, gerakan, dan sesekali tetesan-tetesan air.
6. Konektivitas yang terbatas : Perangkat *mobile* memiliki bandwidth rendah, beberapa dari mereka bahkan tidak tersambung.

1.2 JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data[10]. JSON terbuat dari dua struktur:

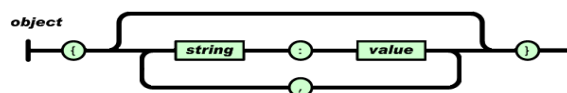
1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini

dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini.

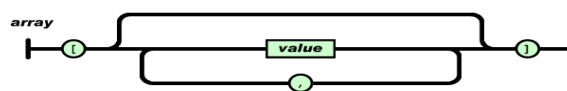
JSON menggunakan bentuk sebagai berikut:

1. **Objek** adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).



Gambar 1 JSON Object

2. **Larik** adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma).



Gambar 2 JSON Array

2. ISI PENELITIAN

2.1 Analisis Komunikasi Data

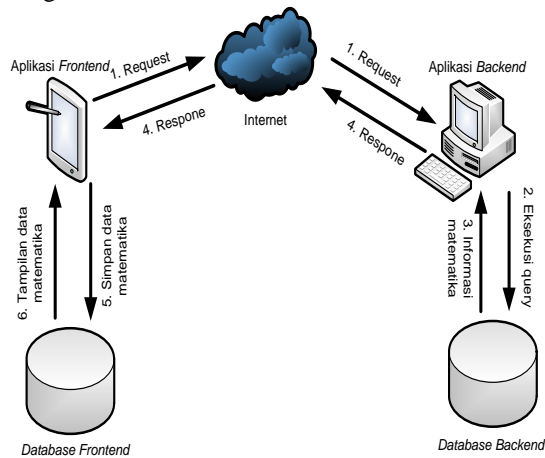
Analisis komunikasi data menjelaskan cara pengiriman dan penerimaan data antara *frontend* dan *backend*. Dalam penelitian ini pertukaran data menggunakan teknik *JSON*, dimana aplikasi *frontend* akan mengirimkan *request* kepada aplikasi *backend* melalui jaringan internet, kemudian aplikasi *backend* akan mengirim respon kepada aplikasi *frontend* sesuai dengan *request* yang dikirim oleh aplikasi *frontend*.

Pertama, aplikasi *client* akan mengirim *request* ke aplikasi *backend* melalui layanan internet, kemudian aplikasi *backend* menerima *request* tersebut. Selanjutnya aplikasi *backend* akan

melakukan eksekusi *query* terhadap *database server*. Hasil dari eksekusi yaitu informasi data materi, data teori dan rumus, data contoh soal, dan data latihan soal.

Setelah itu, aplikasi *backend* akan mengirim respon ke aplikasi *frontend*, respon tersebut berupa data hasil dari eksekusi *query* sesuai dengan *request* yang dikirim oleh aplikasi *frontend* tadi. Setelah aplikasi *frontend* menerima respon dari aplikasi *backend* maka respon tersebut akan disimpan sementara pada *database frontend* pada aplikasi *frontend* untuk ditampilkan kepada siswa.

Berikut adalah gambaran komunikasi data pada aplikasi rumus matematika SMA yang akan dibangun.



Gambar 3 Mekanisme komunikasi data aplikasi rumus matematika SMA

2.2 Analisis Parsing Data

JSON (*JavaScript Object Notation*) merupakan format yang digunakan untuk pertukaran atau parsing data seperti halnya *XML*. Formatnya berbasis teks dan mudah dibaca oleh manusia serta digunakan untuk mempresentasikan struktur data sederhana dan larik asosiatif (disebut objek). Format JSON sering digunakan untuk mentransmisikan data terstruktur melalui suatu koneksi jaringan pada suatu proses yang disebut serialisasi.

Langkah-langkah parsing data dengan *JSON*:

1. Aplikasi *backend* mengakses data dari *database*

Pada *PHP versi 5.2.x* sudah mengikutsertakan ekstensi *JSON* jadi tidak perlu lagi menambahkan alat bantu untuk penerapan *JSON* di *PHP*. Selain itu *PHP* juga sudah menyertakan dua fungsi untuk menulis dan membaca format *JSON* yaitu *json_encode* dan *json_decode* untuk mengubah variabel *PHP* kedalam format *JSON*. *PHP* akan mengakses data yang ada dalam *database* dan menampilkannya dalam bentuk *array*. Selanjutnya *array* tersebut akan di *encode* ke dalam format *JSON* dengan perintah

json_encode, maka akan menghasilkan *array* dalam bentuk format *JSON* sebagai berikut:

```
{["id_contoh": "5", "materi": "Baris dan Deret", "deretcontoh": "Diketahui a+(a+1)+(a+2)+...+50 = 1139. Jika a bilangan bulat positif, maka nilai a = ....", {"id_contoh": "8", "materi": "Baris dan Deret", "deretcontoh": "Dari deret aritmetika diketahui suku tengah 32. Jika jumlah n suku pertama deret itu 672, banyak suku deret tersebut adalah .... Jawaban!! n = 672/32 = 21"}, {"id_contoh": "10", "materi": "Baris dan Deret", "deretcontoh": "Diketahui barisan aritmatika dengan Un adalah suku ke-n. Jika U2 + U15 + U40 = 165, maka U19 = .... Jawaban : U2 + U15 + U40 = 165 --> (a + b) + (a + 39b) = 165 --> 3a + 54b = 165/3 --> a + 18b = 55 --> Jadi U19 = a + 18b = 55"}]}
```

2. Aplikasi *frontend* mengakses aplikasi *backend*
Disini aplikasi *frontend* akan mengakses alamat URL dari file *PHP* yang ada pada aplikasi *backend* dan mengambil data yang telah diubah ke dalam format *JSON* untuk disimpan sementara pada *database frontend* sebelum ditampilkan kepada pengguna (siswa). Data yang ditampilkan kepada pengguna (siswa) adalah sebagai berikut:

Diketahui $a+(a+1)+(a+2)+\dots+50 = 1139$. Jika a bilangan bulat positif, maka nilai $a = \dots$

Dari deret aritmetika diketahui suku tengah 32. Jika jumlah n suku pertama deret itu 672, banyak suku deret tersebut adalah
Jawaban!! $n = 672/32 = 21$

Diketahui barisan aritmatika dengan U_n adalah suku ke- n . Jika $U_2 + U_{15} + U_{40} = 165$, maka $U_{19} = \dots$.
Jawaban : $U_2 + U_{15} + U_{40} = 165 \rightarrow (a + b) + (a + 39b) = 165 \rightarrow 3a + 54b = 165/3 \rightarrow a + 18b = 55 \rightarrow$ Jadi $U_{19} = a + 18b = 55$

2.3 Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi Kebutuhan Perangkat Lunak berisi mengenai semua kebutuhan dalam perancangan kebutuhan perangkat lunak, diantaranya spesifikasi kebutuhan fungsional dan non-fungsional.

Tabel 1 Kebutuhan Fungsional

Nomor	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-F-01	Admin melakukan <i>Login</i> untuk masuk ke dalam sistem
SKPL-F-02	Sistem dapat membantu admin dalam melakukan pengolahan data teori dan rumus yang meliputi tambah data teori

	dan rumus, ubah data teori dan rumus, dan hapus data teori dan rumus.
SKPL-F-03	Sistem dapat membantu admin dalam melakukan pengolahan data contoh soal yang meliputi tambah data contoh soal, ubah data contoh soal, dan hapus data contoh soal.
SKPL-F-04	Sistem dapat membantu admin dalam melakukan pengolahan data latihan soal yang meliputi tambah data latihan soal, ubah data latihan soal, dan hapus data latihan soal.
SKPL-F-05	Sistem dapat membantu siswa dalam mencari teori dan rumus.
SKPL-F-06	Sistem dapat membantu siswa dalam mencari contoh soal.
SKPL-F-07	Sistem dapat membantu siswa dalam mencari dan mengerjakan latihan.

Tabel 1 Kebutuhan Non-Fungsional

Nomor	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-NF-01	Sistem dibangun dalam bentuk website untuk <i>backend</i> dan mobile untuk <i>frontend</i> .
SKPL-NF-02	Sistem yang dibangun bersifat online, sehingga dibutuhkan koneksi internet untuk mengaksesnya.
SKPL-NF-03	Bahasa pemrograman yang digunakan dalam membangun aplikasi ini adalah PHP dan MYSQL untuk aplikasi <i>backend</i> , sedangkan untuk aplikasi <i>frontend</i> menggunakan JAVA dan SQLite.
SKPL-NF-04	Format pertukaran data yang digunakan yaitu dengan format JSON.
SKPL-NF-05	Data yang diolah yaitu data teori dan rumus, contoh soal, dan latihan soal.

2.4 Analisis Kebutuhan Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini yaitu komputer dan *smartphone*. Komputer digunakan dalam merancang dan membangun aplikasi rumus matematika SMA. Sedangkan *smartphone* digunakan sebagai alat untuk implementasi aplikasi rumus matematika SMA yang akan dibangun pada penelitian ini.

Adapun kebutuhan spesifikasi minimum komputer dan *smartphone* yang digunakan dalam penelitian ini dapat dilihat pada tabel 3 dan tabel 4.

Tabel 3 Spesifikasi Perangkat Keras Komputer

No	Spesifikasi	
1	Prosesor	2.4 GHz
2	Memori	2 GB
3	VGA	2 GB
4	Harddisk	500 GB

Tabel 2 Spesifikasi Perangkat Keras *Smartphone*

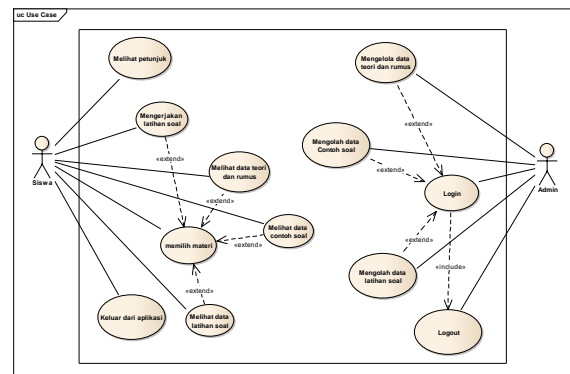
No	Spesifikasi	
1	Memori Internal	4 GB
2	Memori Eksternal	Up to 32 GB
3	Layar	4 inchi
4	Resolusi	HVGA (420x800)
5	Jaringan	Online

2.5 Use Case

Diagram *Use Case* merupakan deskripsi fungsi dari sebuah sistem dari perspektif pengguna. Diagram ini juga mendeskripsikan apa yang akan dilakukan oleh sistem.

Tabel 5 Aktor sistem

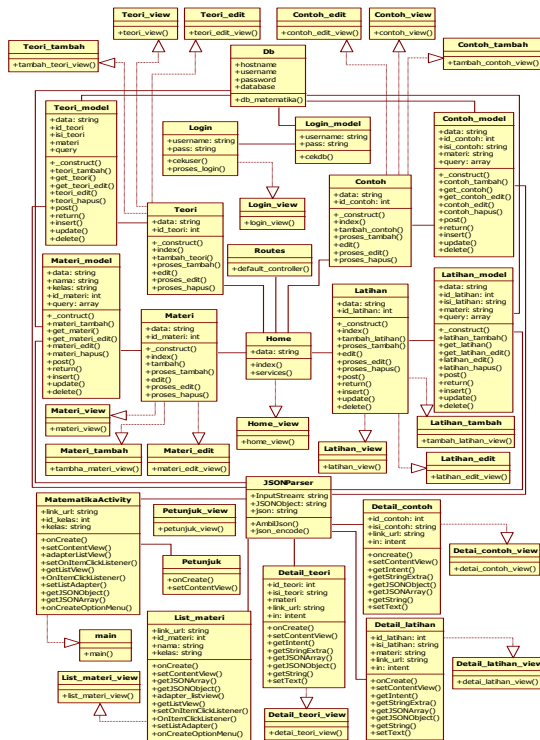
No	Aktor	Deskripsi
1	Admin	Aktor ini merupakan orang yang mengelola data yang berhubungan dengan aplikasi rumus matematika SMA yang akan dibangun
2	Siswa	Aktor ini merupakan pengguna dari aplikasi rumus matematika SMA yang akan dibangun.



Gambar 4 Use Case

2.6 Class Diagram

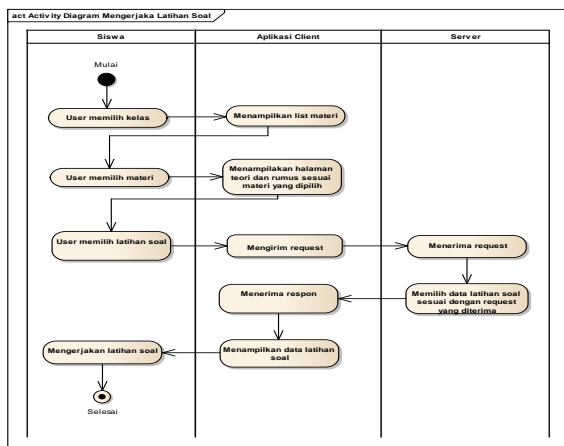
Diagram kelas (*Class diagram*) merupakan diagram struktural yang memodelkan sekumpulan kelas, interface, kolaborasi dan relasinya. Diagram kelas digambarkan dengan kotak yang pada dasarnya terbagi atas tiga bagian, yaitu: nama kelas, atribut, dan operasi. Diagram kelas digunakan untuk menggambarkan proses statik dari aplikasi rumus matematika SMA ini. Berikut ini adalah kelas diagram dari aplikasi rumus matematika SMA.



Gambar 5 Class Diagram

2.7 Activity Diagram

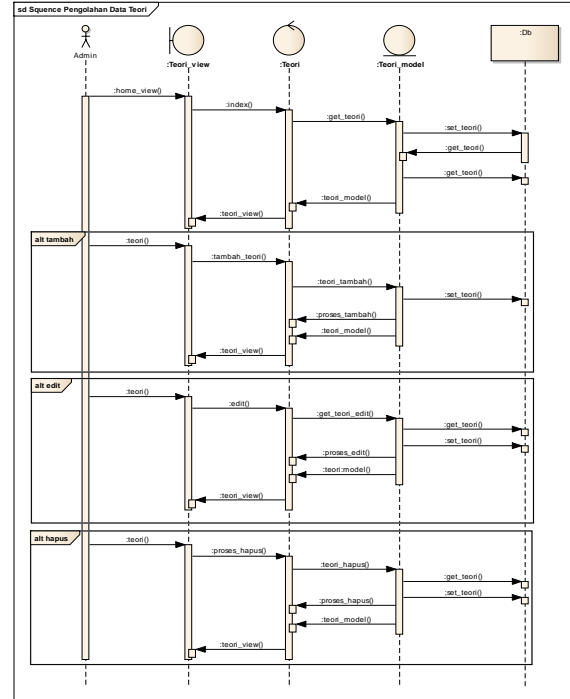
Activity Diagram merupakan suatu diagram yang mendeskripsikan logika procedural, proses bisnis, dan aliran kerja dalam banyak kasus. *Activity Diagram* memiliki peranan seperti halnya *flowchart*, akan tetapi perbedaan antara *activity diagram* dengan *flowchart* yaitu bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. *Activity Diagram* memodelkan event-event yang terjadi didalam suatu *Use Case* dan digunakan untuk pemodelan aspek dinamis dari sistem.



Gambar 6 Activity Diagram Mengerjakan Soal

2.8 Sequence Diagram

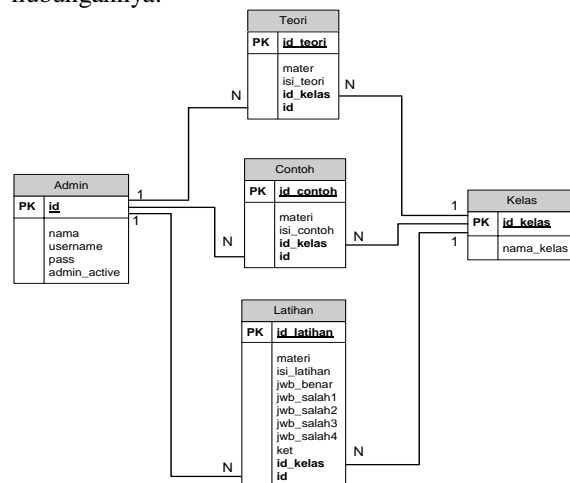
Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh objek dan pesan yang diletakkan diantara objek-objek ini didalam *use cas*.



Gambar 7 Pengolahan data Teori dan Rumus

2.9 Skema Relasi

Relasi antara tabel merupakan gabungan antar file yang mempunyai kunci utama yang sama, sehingga *file-file* tersebut menjadi satu kesatuan yang dihubungkan oleh *field* kunci (*Primary Key*). Pada proses ini elemen-elemen data dikelompokkan menjadi satu *file* database beserta entitas dan hubungannya.



Gambar 8 Skema relasi

2.10 Implementasi Basisdata

Implementasi basisdata diambil berdasarkan perancangan data yang dibuat sebelumnya. Secara fisik, implementasi basisdata diimplementasikan menggunakan perangkat lunak MySQL sebagai *database server* yang ditempatkan pada aplikasi *backend*. Tabel-tabel berikut menggambarkan struktur tabel yang diimplementasikan pada *database server* dan *database frontend*.

a. Database Server

1. Tabel Kelas

Tabel 6 Tabel kelas

```
CREATE TABLE `kelas` (
  `id_kelas` int(3) NOT NULL auto_increment,
  `nama_kelas` varchar(15) NOT NULL,
  PRIMARY KEY (`id_kelas`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

2. Tabel Teori

Tabel 7 Tabel teori

```
CREATE TABLE `teori` (
  `id_teor` int(3) NOT NULL auto_increment,
  `materi` varchar(100) NOT NULL,
  `isi_teor` text NOT NULL,
  `id_kelas` int(3) NOT NULL,
  `id` int(3) NOT NULL default '1',
  PRIMARY KEY (`id_teor`),
  KEY `teori-fk1` (`id_kelas`),
  KEY `teori-fk2` (`id`),
  CONSTRAINT `teori-fk1` FOREIGN KEY (`id_kelas`) REFERENCES `kelas` (`id_kelas`),
  CONSTRAINT `teori-fk2` FOREIGN KEY (`id`) REFERENCES `admin` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

3. Tabel Contoh

Tabel 8 Tabel contoh

```
CREATE TABLE `contoh` (
  `id_contoh` int(3) NOT NULL auto_increment,
  `materi` varchar(100) NOT NULL,
  `isi_contoh` text NOT NULL,
  `id_kelas` int(3) NOT NULL,
  `id` int(3) NOT NULL default '1',
  PRIMARY KEY (`id_contoh`),
  KEY `contoh-fk1` (`id_kelas`),
  KEY `contoh-fk2` (`id`),
  CONSTRAINT `contoh-fk1` FOREIGN KEY (`id_kelas`) REFERENCES `kelas` (`id_kelas`),
  CONSTRAINT `contoh-fk2` FOREIGN KEY (`id`) REFERENCES `admin` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

```
fk2` FOREIGN KEY (`id`)
REFERENCES `admin` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

4. Tabel Latihan

Tabel 9 Tabel latihan

```
CREATE TABLE `contoh` (
  `id_contoh` int(3) NOT NULL auto_increment,
  `materi` varchar(100) NOT NULL,
  `isi_contoh` text NOT NULL,
  `id_kelas` int(3) NOT NULL,
  `id` int(3) NOT NULL default '1',
  PRIMARY KEY (`id_contoh`),
  KEY `contoh-fk1` (`id_kelas`),
  KEY `contoh-fk2` (`id`),
  CONSTRAINT `contoh-fk1` FOREIGN KEY (`id_kelas`) REFERENCES `kelas` (`id_kelas`),
  CONSTRAINT `contoh-fk2` FOREIGN KEY (`id`) REFERENCES `admin` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

5. Tabel Admin

Tabel 10 Tabel admin

```
CREATE TABLE `admin` (
  `id` int(3) NOT NULL auto_increment,
  `nama` varchar(30) NOT NULL,
  `username` varchar(30) NOT NULL,
  `pass` varchar(30) NOT NULL,
  `admin_active` tinyint(1) NOT NULL default '1',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

a. Database Frontend

1. Tabel Teori

Tabel 11 Tabel teori (Frontend)

```
private static final String
CREATE_TABLE_TEORI = "CREATE
TABLE " + TABLE_TEORI + "(" +
KEY_ID_TEORI + " INTEGER
PRIMARY KEY," + KEY_MATERI + "
TEXT," + KEY_ISI_TEORI + " TEXT" +
")";
```

2. Tabel Contoh

Tabel 12 Tabel contoh (Frontend)

```
private static final String
```

```
CREATE_TABLE_CONTOH = "CREATE
TABLE " + TABLE_CONTOH + "(" +
KEY_ID_CONTOH + " INTEGER
PRIMARY KEY," + KEY_MATERI + "
TEXT," + KEY_ISI_CONTOH + " TEXT"
+ ")";
```

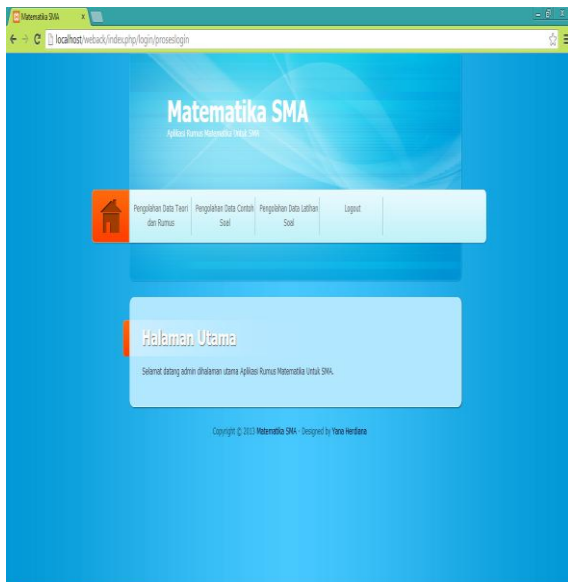
3. Tabel Latihan

Tabel 13 Tabel latihan (Frontend)

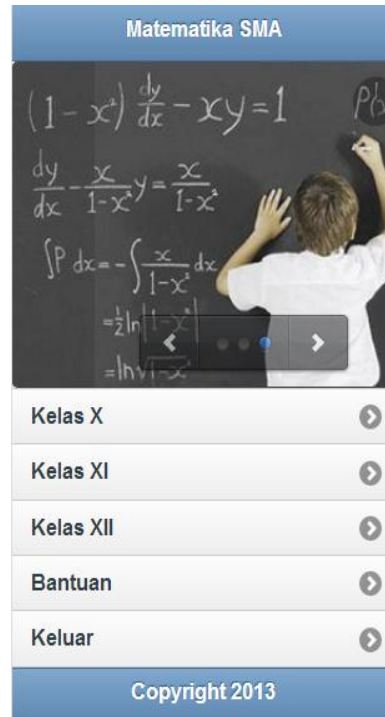
```
private static final String
CREATE_TABLE_LATIHAN =
"CREATE TABLE " + TABLE_LATIHAN
+ "(" + KEY_ID_LATIHAN + " INTEGER
PRIMARY KEY," + KEY_MATERI + "
TEXT," + KEY_ISI_LATIHAN + "
TEXT," + KEY_JWB_BENAR + " TEXT,"
+ KEY_JWB_SALAH1 + " TEXT," +
KEY_JWB_SALAH2 + " TEXT," +
KEY_JWB_SALAH3 + " TEXT," +
KEY_JWB_SALAH4 + " TEXT," +
KEY_KET + " TEXT" + ")";
```

2.11 Implementasi antarmuka

Implementasi antarmuka berisi pemaparan dari setiap tampilan pada aplikasi rumus matematika SMA berbasis *mobile* yang telah dibangun. Adapun gambar dari tampilan implementasi antarmuka aplikasi rumus matematika SMA berbasis *mobile* tertera pada halaman Lampiran A Implementasi Antarmuka.



Gambar 10 Halaman Utama Backend



Gambar 11 Halaman Utama Frontend

2.12 Pengujian Sistem

Rencana pengujian merupakan tahapan pengujian fungsional yang terdapat dalam sistem untuk mengetahui apakah sistem yang dibangun sudah sesuai dengan yang diharapkan atau tidak. Berikut adalah tabel rencana pengujian aplikasi *frontend* dan *backend*.

Tabel 14 Pengujian List Materi

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Menu : "Kelas X"	Menampilkan list materi Kelas X	List materi Kelas X ditampilkan	[√] diterima [] ditolak
Menu : "Kelas XI"	Menampilkan list materi Kelas XI	List materi Kelas XI ditampilkan	[√] diterima [] ditolak
Menu : "Kelas XII"	Menampilkan list materi Kelas XII	List materi Kelas XII ditampilkan	[√] diterima [] ditolak

Berdasarkan hasil pengujian *alpha* yang telah dilakukan dapat ditarik kesimpulan bahwa aplikasi rumus matematika SMA berbasis *mobile* sudah berjalan cukup maksimal sesuai dengan tujuan dan perancangan awal. Tetapi tidak menutup kemungkinan dapat terjadi kesalahan pada saat aplikasi digunakan, baik itu kesalahan pengguna, kesalahan pada perangkat yang digunakan, ataupun

kesalahan yang lainnya. Sehingga membutuhkan proses *maintenance* agar aplikasi tetap berjalan maksimal dan sesuai dengan harapan.

3. PENUTUP

Kesimpulan dari penelitian skripsi yang berjudul: “Pembangunan Aplikasi Rumus Matematika SMA Berbasis *Mobile*” adalah bahwa aplikasi rumus matematika SMA yang dibangun dapat menjadi salah satu alat bantu siswa dalam mempelajari pelajaran matematika. Dengan adanya aplikasi rumus matematika SMA yang bisa bersifat *moveble* maka siswa bisa belajar dimanapun dan kapanpun khususnya di luar jam sekolah. Dengan adanya aplikasi rumus matematika SMA ini membantu membuat siswa termotivasi untuk belajar matematika.

DAFTAR PUSTAKA

- [1] Lukito, Agung. Sisworo. 2013. *Matematika (Untuk SMA/MA)*. Politeknik Negeri Media Kreatif. Jakarta.
- [2] Safaat, Nazrudin. 2012. *Pemrograman Aplikasi Mobile Smartphone Dan Tablet PC Berbasis Android*. Informatika. Bandung.
- [3] Aziz, Abdul. 2009. *Rumus Jitu Matematika SMA*. PT Indonesia Tera. Yogyakarta.
- [4] Harianto, Bambang, Ir.,MT. 2004. *Rekayasa Sistem Berorientasi Objek*. Informatika. Bandung.
- [5] Pudjo W, Prabowo. Herlawati. 2011. *Menggunakan UML*. Informatika. Bandung.
- [6] Harianto, Bambang. 2004. *Sistem Manajemen Basisdata*. Informatika. Bandung.
- [7] Sholih. 2006. *Pemodelan Sistem Informasi Berorientasi Objek Dengan UML*. Informatika. Bandung.
- [8] Murya, Yosef. 2013. *Pemrograman Android Black Box*. Jasakom. Jakarta.
- [9] Dharma K. Akhmad. 2013. *Kolaborasi Dahsyat ANDROID dengan PHP dan MySQL*. Lokomedia. Yogyakarta.
- [10] JSON. “*Pengenalan JSON*”. <http://www.json.org/json-id.html> (diakses tanggal 12 November 2013).