

IMPLEMENTASI MODEL VIEW CONTROLLER (MVC) PADA APLIKASI DOA HARIAN UNTUK ANAK MUSLIM BERBASIS ANDROID

Diana¹, Febrianti²

^{1,2}Program Studi Teknik Informatika, Fakultas Teknik

Universitas Muhammadiyah Bengkulu

Jl. Bali PO BOX 118 Telp. (0736) 227665, Fax (0736)26161, Bengkulu 38119

¹anaiboel@gmail.com, ²febrianti@gmail.com

Abstract

Prayer is a means to communicate between slave by Allah in certain circumstances. In addition, as a spirit of prayer worship or religious sari as ever accepted in those sayings by Rasulullah SAW. Today many children are not aware of any various kinds of daily prayer because of their lack of understanding about the prayer in daily life and also the teaching of teachers and from the child's parents. Model-View-Controller (MVC) is a concept used to encapsulate the data along with processing (model) isolate it from manipulating (controller) and display (view) to be represented on a user interface. The purpose of this study is create design applications daily prayers for Muslim children based on Android that uses the MVC method. The methodology used in the study This is a model RAD (Rapid Application Development). From the analysis, design, testing and implementation of programs that have been carried out against Children Daily Prayer app can be inferred application runs with both fit for purpose and planning penelitian. Model View Controller can be implemented in application development and certainly more structured program or application.

Keyword: MVC, daily prayer, muslim kids, android

Abstrak

Doa merupakan salah satu sarana untuk berkomunikasi antara hamba dengan Allah SWT dalam keadaan tertentu. Di samping itu, doa sebagai roh ibadah atau sari ibadah sebagaimana yang pernah disabdakan oleh Rasulullah SAW. Saat ini banyak anak-anak yang tidak mengetahui adanya berbagai macam doa harian dikarenakan kurangnya pemahaman mereka tentang adanya doa di kehidupan sehari-hari dan juga pengajaran dari guru maupun dari orang tua si anak. Tujuan penelitian ini adalah membuat desain aplikasi do'a harian untuk anak muslim berbasis android yang menggunakan metode MVC. *Model-View-Controller* (MVC) adalah konsep yang digunakan untuk meng-enkapsulasi data bersama dengan pemrosesan (model), mengisolasi dari proses manipulasi (*controller*) dan tampilan (*view*) untuk direpresentasikan pada sebuah *user interface*. Dari hasil analisa, perancangan, pengujian dan implementasi program yang telah dilakukan terhadap aplikasi Doa Harian Anak ini disimpulkan *Model View Controller* dapat diterapkan dalam pengembangan aplikasi dan tentunya program atau aplikasi lebih terstruktur.

Kata kunci: MVC, doa harian, anak muslim, android

I. PENDAHULUAN

Pada era modern saat ini, teknologi informasi mengalami kemajuan yang begitu pesat, terutama di dunia *mobile* teknologi. Saat ini banyak produsen handal yang terus mengembangkan fungsi *handphone*. Mereka menciptakan telepon pintar yang lebih dikenal dengan sebutan *smartphone*. Teknologi *smartphone* ini tentunya didukung oleh operasi sistem yang juga terus dikembangkan supaya fitur-fitur *smartphone* terus bertambah. Sistem operasi yang sangat berkembang saat ini adalah android. Sistem operasi android memiliki banyak fitur diantaranya penggunaan sistem *database* yang sangat baik.

Segala aktifitas yang kian padat menjadikan sebagian orang memiliki tingkat mobilitas yang tinggi. Seiring dengan berkembangnya teknologi, manusia cenderung lupa akan kewajiban dalam beragama. Salah satunya membaca doa. Doa merupakan permohonan langsung kepada Allah SWT agar diberikan kebaikan, keberkahan, kemudahan, kesehatan dan jalan keluar dari kesulitan dan lain-lain [1].

Disekitar kita banyak sekali doa harian yang bisa anak-anak amalkan di dalam kehidupan sehari-hari. Saat ini banyak anak-anak yang tidak mengetahui berbagai macam doa harian dikarenakan

kurangnya informasi tentang doa baik dari guru maupun dari orang tua anak.

Aplikasi doa harian anak muslim berbasis *android* menggunakan *Model-View-Controller* (MVC). MVC adalah sebuah konsep yang diperkenalkan oleh penemu *Smalltalk* (Trygve Reenskaug) untuk mengenkapsulasi data bersama dengan pemrosesan (model), mengisolasi dari proses manipulasi (*controller*) dan tampilan (*view*) untuk direpresentasikan pada tampilan muka [2]. Rancang bangun sistem aplikasi doa harian anak muslim berarsitektur (MVC) ini menerapkan metodologi SDLC (*System Development Life Cycle*) dengan menggunakan pendekatan berorientasi objek. Pengembangan yang dilakukan yaitu *iterative* dan *incremental*. Hasil penelitian ini diharapkan dapat memberikan alternatif bagi para guru, orang tua dan anak-anak dalam belajar doa dan memudahkan anak-anak membaca doa-doa harian tanpa harus bingung mencari atau membeli buku dengan mengeluarkan biaya dan waktu.

II. LANDASAN TEORI

1. Doa

Doa dalam bahasa Indonesia merupakan kata serapan dari bahasa Arab, yaitu الدعاء. Doa berasal dari kata bahasa Arab (- دعاء يدعو دعا) yang berarti memohon atau meminta

(M. Khalilurrahman al-mahfani, 2006:27). Ia merupakan derivasi dari kata دعا yang merupakan fi'il madhi yang sekaligus menjadi akar katanya. Sementara kata الدعاء itu sendiri merupakan bentuk mashdar.

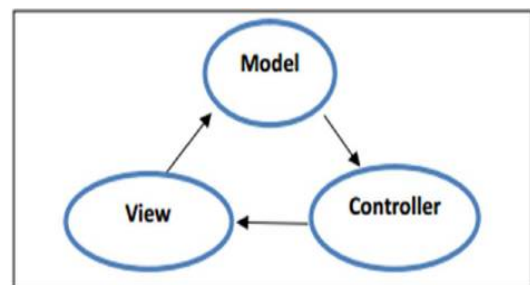
Sedang menurut istilah doa merupakan suatu media komunikasi antara seorang hamba dengan Sang Khaliq dalam rangka memohon dan meminta hajat hidup di dunia dan di akhirat, mengeluh dan mengadu atas permasalahan yang dihadapi, atau memohon perlindungan dari segala macam marabahaya [2].

2. Model-View-Controller (MVC)

Model-View-Controller (MVC) adalah sebuah konsep yang diperkenalkan oleh penemu *Smalltalk* (Trygve Reenskaug) untuk meng-enkapsulasi data bersama dengan pemrosesan (model), mengisolasi dari proses manipulasi (*controller*) dan tampilan (*view*) untuk direpresentasikan pada tampilan muka [3].

MVC mengikuti pendekatan yang paling umum dari *layering*. *Layering* hanyalah sebuah logika yang membagi kode kedalam fungsi di kelas yang berbeda. Keuntungan utama dalam pendekatan ini adalah penggunaan ulang (*reusability*) kode. Definisi dari arsitektur MVC dibagi menjadi tiga lapisan [4]:

- Model*, digunakan untuk mengelola informasi dan memberitahu pengamat ketika ada perubahan informasi. Model mengandung data dan fungsi yang berhubungan dengan pemrosesan data.
- View*, bertanggung jawab untuk pemetaan grafis sebuah perangkat.
- Controller*, menerima input dari pengguna dan mendistribusikan model dan *view* untuk melakukan aksi berdasarkan masukan tersebut. Sehingga *controller* bertanggung jawab untuk pemetaan aksi pengguna akhir terhadap respon aplikasi.



Gambar 1. Bagian MVC

Model, *View*, dan *Controller* ketiganya berkaitan erat, oleh karena itu ketiganya harus merujuk satu sama lain. Arsitektur MVC memiliki manfaat yaitu pemisahan antara model dan *view* memungkinkan beberapa *view* menggunakan model yang sama. Akibatnya komponen model sebuah aplikasi lebih mudah untuk diterapkan,

diuji dan dipelihara, karena semua akses ke model berjalan melalui komponen ini [5].

Konsep *Model View Controller* (MVC) bertujuan agar sebuah aplikasi dapat mudah dipelihara oleh orang-orang di dalam tim pengembangan yang berbeda spesifikasi pekerjaan, misalnya *database administrator* (DBA) untuk mengurus masalah basis data, *blok controller* untuk programmer, dan *blok view* untuk desainer antarmuka (*interface designer*) [6].

3. *Software Development Life Cycle* (SDLC)

System Development Life Cycle (SDLC) merupakan siklus pengembangan sistem teknik (*engineering system development*). SDLC berfungsi untuk menggambarkan tahapan-tahapan utama dan langkah-langkah dari setiap tahapan yang secara garis besar terbagi dalam empat kegiatan utama, yaitu *initiation*, *analysis*, *design* dan *implementation*. Setiap kegiatan dalam SDLC dapat dijelaskan melalui tujuan (*purpose*) dan hasil kegiatannya (*deliverable*). SDLC didefinisikan oleh Departemen Kehakiman AS sebagai sebuah proses pengembangan *software* yang digunakan oleh *analyst system*, untuk mengembangkan sebuah sistem informasi. SDLC mencakup

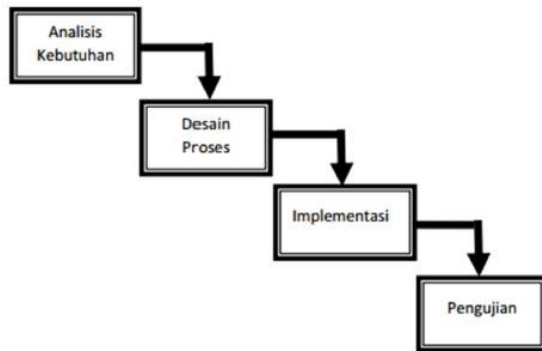
kebutuhan (*requirement*), validasi, pelatihan, kepemilikan (*user ownership*) sebuah sistem informasi yang diperoleh melalui investigasi, analisis, desain, implementasi, dan perawatan software. Metode SDLC yang meliputi beberapa fase yaitu Fase analisis, pada fase ini adalah memecahkan masalah yang terjadi dalam pembuatan aplikasi. Fase perancangan, pada fase ini adalah melakukan perancangan struktur tampilan menu dan tampilan isi doa dan audio pada aplikasi. Fase implementasi, pada fase ini adalah pembuatan aplikasi berbasis android tersebut dengan menerapkan hasil perencanaan. Fase uji coba, pada fase ini adalah dengan melakukan uji coba pada aplikasi dan uji coba kepada pengguna [7].

4. Android

Android adalah sebuah sistem operasi untuk *Smartphone* dan *Tablet*. Sistem operasi dapat diilustrasikan sebagai “jembatan” antara piranti (*device*) dan penggunanya, sehingga pengguna bisa berinteraksi dengan *device*-nya dan menjalankan aplikasi-aplikasi yang tersedia pada *device*. Di dunia personal komputer, sistem operasi yang banyak dipakai adalah *Windows*, *Mac* dan *Linux* [8].

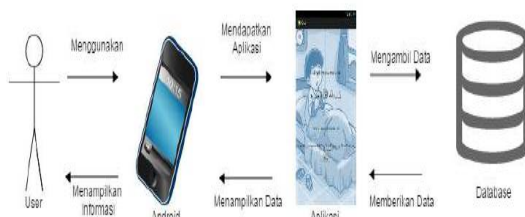
III. METODE PENELITIAN

Model pengembangan sistem yang digunakan untuk membangun aplikasi ini adalah *Rapid Application Development* (RAD).



Gambar 2. Model RAD

1. Arsitektur Sistem



Gambar 3. Arsitektur Sistem

2. Struktur Menu Aplikasi

a. Splash Screen

Splash screen merupakan tampilan awal masuk masuk aplikasi. Halaman ini berupa *splash* yang menunjukkan latar belakang *image*.

b. Menu Utama

Pada halaman menu utama terdapat empat button, yaitu menu doa, *login admin*, *help/bantuan*, *about/ tentang*.

c. Menu Doa

Pada menu ini, terdapat tampilan daftar doa-doa yang ada di *database*.

d. Login Admin

Menu ini merupakan menu yang digunakan untuk admin dalam menambahkan data doa dan hapus doa.

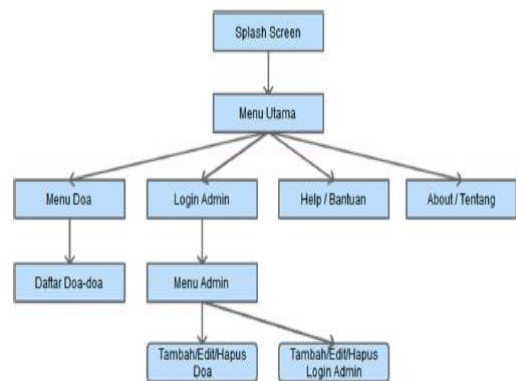
e. Help/Bantuan

Menu *help/ bantuan* digunakan untuk membantu para *user* dalam menggunakan aplikasi.

f. About/Tentang

Menu *about/ tentang* digunakan sebagai informasi yang membuat program aplikasi.

Berikut adalah Gambar struktur menu dari aplikasi yang dibangun.



Gambar 4. Struktur Menu Aplikasi

3. Rancangan Database

Nama Field	Type	Ukuran	Keterangan
id_login	integer	-	ID Login
nama_login	text	25	Nama Login
username	text	50	Nama Pengguna
password	text	50	Password Pengguna

Tabel 3.1 Tabel User

Nama Field	Type	Ukuran	Keterangan
id_doa	integer	-	ID Doa
doa	text	-	Nama Doa
kaligrafi	text	-	Gambar Doa
doa_latin	text	-	Doa huruf latin
terjemahan	text	-	Terjemahan doa
Sumber	text	50	Sumber Doa
note	text	-	Catatan
check	text	10	Tanda Doa
bookmark	text	10	Bookmark Doa

Tabel 3.2 Tabel Doa

Nama Field	Type	Ukuran	Keterangan
id_kelompok	integer	-	ID Kelompok
Kelompok_doa	text	50	Kelompok Doa

IV. HASIL DAN PEMBAHASAN

1. Hasil

Aplikasi di bangun berbasis Android dengan *Eclipse IDE* dan *Plugin ADT*. *Splash screen* (Gambar 5) adalah tampilan pertama program sebelum masuk ke menu utama. Menu utama terdiri dari empat tombol yang masing-masing memiliki fungsi berbeda (Gambar 6). Pada menu Do'a-Do'a, pengguna dapat memilih do'a harian anak-anak. Do'a-do'a tersebut di tampilkan dengan komponen *ListView android* dan populasi doa-doa merupakan data do'a-do'ayang tersimpan pada *database* (Gambar 7). Tap satu kali pada do'a yang akan ditampilkan sehingga muncul tampilan do'a seperti Gambar 8.



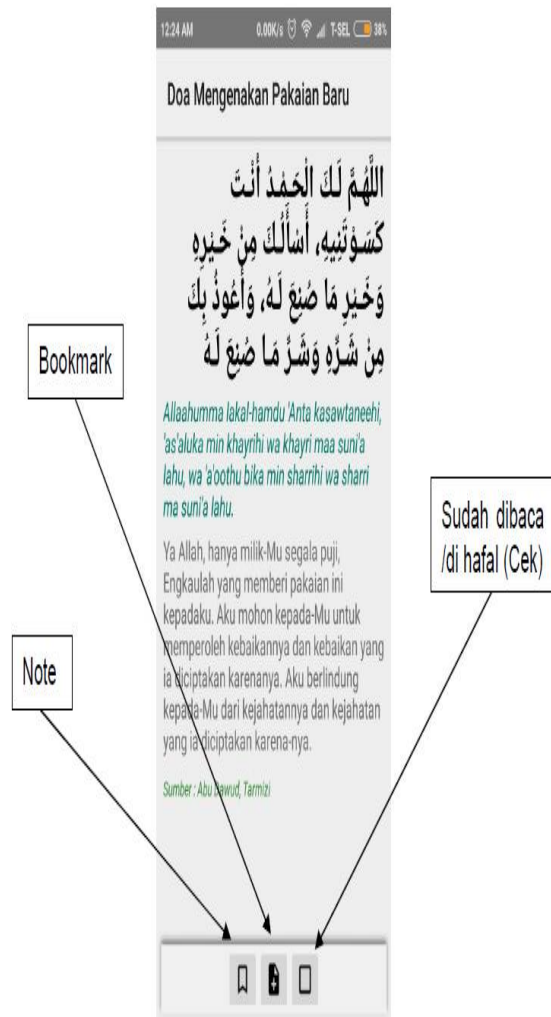
Gambar 5. Splash Screen



Gambar 6. Menu Utama



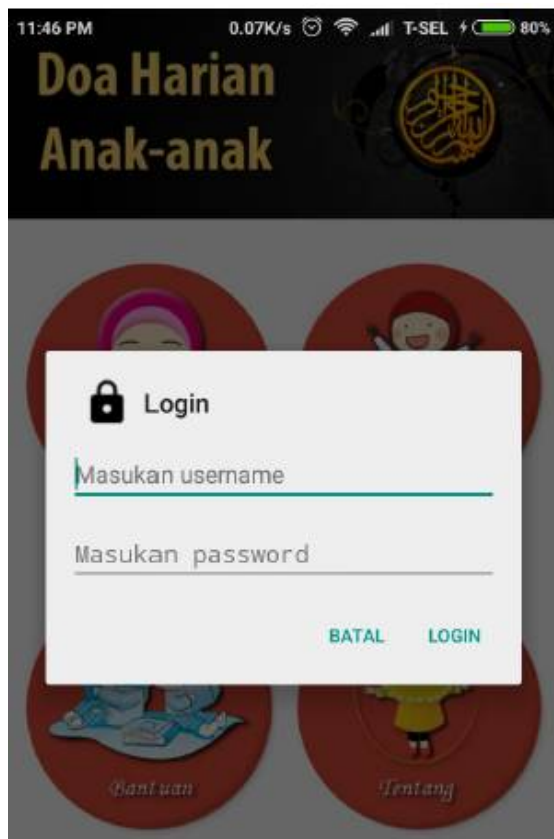
Gambar 7. Daftar Doa-doa



Gambar 8. Tampilan Doa

Pada tampilan doa, aplikasi menampilkan doa yang dipilih pada daftar do'a-do'ayang tampil sebelumnya. Pada tampilan doa ini terdapat *main tab* diposisi bawah layar yang memiliki tiga opsi untuk doa yang tampil. Tiga opsitersebut adalah *Bookmark* yang berfungsi untuk menandai doa ke dalam buku penanda, opsi *Note* yang berfungsi sebagai catatan untuk doa yang ditampilkan, dan opsi *Cek* sebagai tanda bahwa doa sudah di baca atau dihafal.

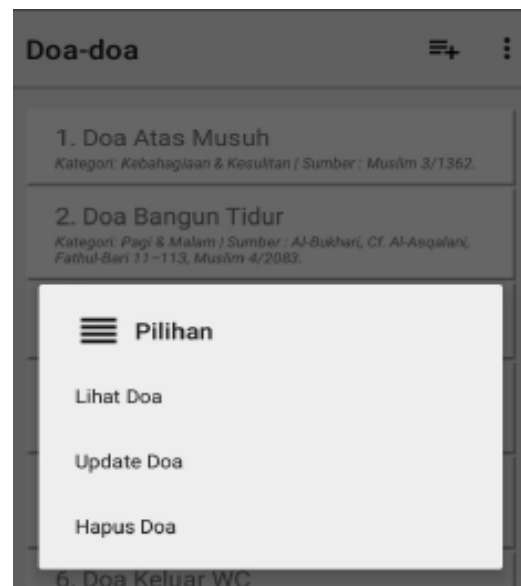
Pada Menu Admin ini, pengguna yang terdaftar sebagai admin harus memasukkan informasi akun yang di perlukan aplikasi yaitu berupa *username* dan *password* yang terdaftar sebelumnya. Setelah informasi di lengkapiselanjutnya tekan OK untuk mengecek atau memvalidasi informasi admin apakah *input* yang di masukan cocok dengan *database*. Jika login berhasilakan tampil *pop-up* untuk memilih menu admin yang akan di akses. Menuadmin tersebut dapat di lihat pada gambar di bawah ini.



Gambar 9. Menu Admin

Setelah admin berhasil *login*, selanjutnya admin memilih menu apa

yang akan di akses. Jika adminmemilih data doa, maka akan muncul daftar doa-doa yang tersimpan padadatabase aplikasi. Untuk menambah doa baru dapat dilakukan denganmenyentuh tombol “ + “ di sudut kanan atas layar. Data doa juga dapat di *edit* maupun di hapus.



Gambar 10. Tampilan Tambah Doa

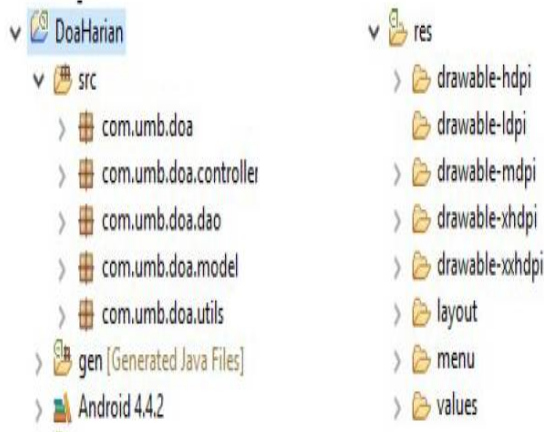
2. Pembahasan

Model View Controller di terapkan pada *package* dan *class-class* aplikasi doa harian ini.

- 1) *Class-class* model tersimpan pada *package com.umb.doa.model.**
- 2) *Class-class* pengontrol atau *controller* tersimpan pada *package com.umb.doa.controller.**
- 3) Komponen *View* atau *userinterface* tersimpan pada *package res/layout.**
- 4) *Package com.umb.doa.doa.** merupakan *package* yang berisi

class mengakses data dari *database*.
DAO adalah singkatan dari Data Akses Object.

- 5) Package *com.umb.doa.utils.** berisi *class-class* pendukung.



Gambar 8. Package Project Doa Harian

a. Model

Pada bagian model ini, terdapat *method* berupa prosedur dan *function* yang berfungsi untuk mendapatkan informasi dan mengolahnya seperti melihat, menambahkan data, mengedit dan menghapus data. Model mengakses informasi dari *database* melalui DAO. Berikut ini satu contoh Model yang ada pada aplikasi doa harian untuk anak muslim ini.

```
package com.umb.doa.model;

import ...

public class DoaMdl {
    private DataSource dao;

    public DoaMdl(Context context) {
        this.dao = new DataSource(context);
    }

    public Doa getById(int id) {
        Doa doa = new Doa();
        String selectQuery = "SELECT * FROM doa WHERE id=?";
        SQLiteDatabase db = dao.getWritableDatabase();
        Cursor cursor = db.rawQuery(selectQuery, new String[] { String.valueOf(id) });
        if (cursor.moveToNext()) {
            doa.setId(cursor.getInt(0));
            doa.setNama(cursor.getString(1));
            doa.setKategori(cursor.getString(2));
            doa.setKaligrafi(cursor.getString(3));
            doa.setLatin(cursor.getString(4));
            doa.setTerjemahan(cursor.getString(5));
            doa.setSumber(cursor.getString(6));
            doa.setNote(cursor.getString(7));
            doa.setCek(cursor.getInt(8));
            doa.setBookmark(cursor.getInt(9));
        }
        return doa;
    }
}
```

b. View

Setelah pembuatan model, di buat *View* yang berfungsi sebagai wadah atau tempat menampilkan informasi yang di hasilkan oleh model tersebut. *View* pada aplikasi android terletak pada folder *res/layout*. Berikut merupakan satu contoh *View* yang menampilkan informasi doa yang di hasilkan Model.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.umb.doa.controller.DoaCtrl" >

    <LinearLayout
        android:id="@+id/bottom_menu"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentStart="true"
        android:background="@drawable/bottom_bg"
        android:gravity="center" >

        <ImageButton
            android:id="@+id/btn_bookmark"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:contentDescription="@string/app_name"
            android:src="@drawable/unbookmark" />
```

c. Controller

Controller bekerja sesuai dengan *request* yang di terima dari *View*. Ketika user memberi perintah melalui *View*, *Controller* bekerja sesuai intruksi terhubung ke model dan

mengembalikan *value* ke *View*. Pada gambar di atas di jelaskan kode pada *class DoaCtrl* atau singkatan dari *Doa Controller*. Pada baris pertama menjelaskan bahwa *class* ini berada pada *package Controller* selanjutnya mengimport *library android* lalu memberi nama *class* yaitu *DoaCtrl*. Pada *class controller*, *class DoaCtrl* meng-inherit *Activity* dimana *Activity* ini merupakan *class* yang manajemen aktivitas *Layout* sehingga membuktikan bahwa *class Controller* ini terhubung dengan *View*. Di dalam *class* tersebut juga di deklarasikan *class DoaMdl* yang merupakan *Model* yang sebelumnya telah di jelaskan dan membuat *Controller* ini terhubung dengan *Model*. Pada *method prosedur onCreate()* di jelaskan proses pengambilan informasi dari *Model* dan di tampilkan ke *View*.

Baris kode `setContentView(R.layout.lihat_doa);` merupakan penentuan *View* apa yang terhubung dengan *Controller* ini, pada kode tersebut menentukan layout `lihat_doa` sebagai *View* dari *Controller* ini.

```
package com.umb.doa.controller;

import ...

public class DoaCtrl extends Activity {
    private int idDoa;
    private DoaMdl model = new DoaMdl(this);
    private Doa doa;
    private TextView kaligrafi, latin, terjemahan, sumber;
    private ImageButton bookmark, note, cek;
    private int bookmarkState = 0;
    private int cekState = 0;
    private String oldNote;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.lihat_doa);
        Bundle b = getIntent().getExtras();
        if (b != null) {
            idDoa = b.getInt("id");
            doa = model.getById(idDoa);
            this.setTitle(doa.getNama());
        }
        bookmarkState = doa.getBookmark();
        cekState = doa.getCek();
        oldNote = doa.getNote();
        kaligrafi = (TextView) findViewById(R.id.txt_kaligrafi);
        latin = (TextView) findViewById(R.id.txt_latin);
        terjemahan = (TextView) findViewById(R.id.txt_terjemahan);
        sumber = (TextView) findViewById(R.id.txt_sumber);
        bookmark = (ImageButton) findViewById(R.id.btn_bookmark);
        note = (ImageButton) findViewById(R.id.btn_add_note);
        cek = (ImageButton) findViewById(R.id.btn_cek);
        setup();
        setBookmark();
    }
}
```

V. KESIMPULAN

Dari hasil analisa, perancangan, pengujian dan implementasi program yang telah dilakukan terhadap aplikasi Doa Harian Anak, dapat disimpulkan sebagai berikut :

1. Metode *Model View Controller* (MVC) dapat di terapkan dalam pengembangan aplikasi dan tentunya program atau aplikasi lebih terstruktur.
2. Hasil pengujian dari berbagai aspek aplikasi telah sesuai dengan perencanaan dan tujuan penelitian.

REFERENSI

- [1] Prakosa ZRS. 2016. *Aplikasi Doa Sehari-hari Bagi Umat Muslim Berbasis Android*. Fakultas Teknologi Informasi. Universitas Stikubank. Semarang.
- [2] Karnoto M. *Aplikasi Pembelajaran Doa Harian dan Hadist Berbasis Android pada SDN 37 Pangkal Pinang*.

- [3] Deacon, J., 2009. *Model-View-Controller*.
<http://www.jdl.co.uk/briefings/index>.
Diakses tanggal 10 Agustus 2017.
- [4] Burbeck, S., 1992. *Application Programmings in Smaltalk's 80TM: How To Use MVC*. [Online]
<http://stwww.cs.illinois.edu/users/smarch/stdocs/mvc.html>.
Diakses tanggal 14 Agustus 2017.
- [5] Balani, N., 2002. *Web services architecture using MVC style*.
[http://www.webifysolutions.com?subject=Web services architecture using MVC style](http://www.webifysolutions.com?subject=Web+services+architecture+using+MVC+style).
Diakses tanggal 14 Agustus 2017.
- [6] Rosa A.S dan M.Shalahuddin. 2011. *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek*. Bandung.
- [7] Nurul AI. 2014. *Aplikasi Doa-Doa Harian Anak Muslim Berbasis Android*.
- [8] Satyaputra A dan Aritonang EM. 2014. *Beginning Android Programming with ADT Bundle*. Jakarta. PT.Elex Media Komputindo.