

**KODE : 123 / ILMU KOMPUTER**

**LAPORAN PENELITIAN**

**IMPLEMENTASI JAVA INTERFACE PADA PEMBUATAN  
APLIKASI MULTIMEDIA BERBASIS ANDROID**



Oleh Tim:

**EDDY NURAHARJO, ST, M.Cs**  
**AGUNG PRIHANDONO, S.Kom**  
**HENRI PUSPITASARI**  
**HARDIAN RAKHMANTO**

**0628127301 (Ketua)**  
**0607108402 (Anggota)**  
**12.01.53.0094 (Anggota)**  
**12.01.53.0096 (Anggota)**

**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS STIKUBANK SEMARANG**  
**AGUSTUS 2015**

## HALAMAN PENGESAHAN USULAN PENELITIAN

1. a. Judul Penelitian : Implementasi Java Interface Pada Pembuatan Aplikasi Multimedia Berbasis Android
- b. Jenis Penelitian : Penelitian Terapan
- c. Bidang Penelitian : 2/2.15/Application Development
- d. Tujuan Sosial Ekonomi : 20/20.05 – Information, computer and communication technologies
2. Ketua Peneliti
- a. Nama Lengkap : Eddy Nuraharjo, ST. M.Cs
- b. Jenis Kelamin : Laki-laki
- c. NIDN / NIY : 0628127301 / YU. 2.04.04.065
- d. Gol / Pangkat : III B / Penata Muda
- e. Jabatan Fungsional : Asisten Ahli
- f. Jabatan Struktural : -
- g. Fakultas/Program Studi : Teknologi Informasi / Teknik Informatika
- h. Pusat Penelitian : Lembaga Penelitian Universitas Stikubank
3. Jumlah Anggota Peneliti : 3 orang
- a. Dosen Anggota Peneliti I : Agung Prihandono, S.Kom / 0607108402
- b. Mahasiswa Anggota Peneliti I : Henri Puspitasari/12.01.53.0094
- c. Mahasiswa Anggota Peneliti II : Hardian Rakhmanto/12.01.53.0096
4. Lokasi Penelitian : Universitas Stikubank Semarang
5. Kerjasama dengan Institusi Lain
- a. Nama Institusi : -
- b. Alamat : -
- c. Telepon/Fakz/e-mail : -
6. Lama Penelitian : 3(tiga) bulan (15 Maret 2015 s/d 1 Juni 2015)
7. Pendanaan
- a. Sumber Dana : Dalam Negeri
- b. Institusi Sumber Dana : Unisbank
- c. Besar Dana : Rp. 3.000.000,-
- d. Sumber Dana Lain : -
- Jumlah : Rp. 3.000.000,-

Mengetahui,  
Dekan Fakultas Teknologi Informasi

(DR. Yohanes Suhari, M.MSI)  
NIDN. 0620106502

Semarang, 20 Agustus 2015  
Ketua Peneliti

(Eddy Nuraharjo, ST, M.Cs)  
NIDN. 0628127301

Menyetujui,  
Ketua LPPM Unisbank

(DR. Endang Tjahjaningsih, SE, M.Kom)  
NIDN. 0622055601

# IMPLEMENTASI JAVA INTEFACE PADA PEMBUATAN APLIKASI MULTIMEDIA BERBASIS ANDROID

## Abstraksi

Android merupakan sebuah sistem operasi yang sebelumnya dikembangkan khusus untuk telepon genggam. Berkembangnya teknologi internet adalah salah satu faktor yang mempengaruhi perkembangan teknologi mobile. Karena menghadirkan kemudahan bagi para pengguna perangkat elektronik untuk mengakses berbagai jenis konten melalui perangkat mobile seperti ponsel pintar, tablet, bahkan beberapa perangkat yang dapat dikategorikan sebagai bukan perangkat yang mobile seperti iTV (interactive Television), Set TV kabel, yang memungkinkan pengguna untuk dapat berinteraksi dengan perangkat TV tersebut untuk mengakses internet. Hal ini menjadikan penambahan media-media baru seperti game, animasi 3D, audio digital bahkan informasi visual dengan grafis yang sangat detil mulai ada di sekeliling kita.

Proses pembuatan aplikasi multimedia dengan konten yang bervariasi dapat dicapai dengan membuat modul-modul program. Sehingga memudahkan dalam pengembangan kode program itu sendiri. Salah satu penerapan yang dilakukan adalah dengan memanfaatkan kelas *interface* yang telah dimiliki oleh bahasa pemrograman java. Kelas ini bertugas sebagai perantara antar kelas juga dengan sistem android.

Setelah pembuatan kelas *interface*, maka selanjutnya pengembang dapat bekerja pada bagian implementasi dari kelas interface tersebut. Dengan demikian ketika terdapat perubahan pada kode teknis, pengembang tidak perlu merubah bentuk komunikasi antar kelas. Sehingga kemudahan dalam bereksperimen lebih leluasa tanpa merubah bentuk struktur aplikasi secara keseluruhan.

Kata Kunci: Kelas Interface, Android, aplikasi multimedia

## DAFTAR ISI

Lembar Judul.....	i
Lembar Pengesahan.....	ii
Lembar Abstraksi .....	iii
Daftar Isi .....	iv
Daftar Tabel .....	v
Daftar Gambar .....	vi
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah .....	2
BAB II TUJUAN DAN MANFAAT PENELITIAN .....	3
BAB III TINJAUAN PUSTAKA .....	4
BAB IV METODE PENELITIAN .....	6
BAB V HASIL DAN PEMBAHASAN .....	7
1.1 Modul-modul Aplikasi .....	7
1.2 Implementasi Aplikasi Multimedia .....	15
BAB VI PENUTUP .....	21
6.1 Simpulan .....	21
6.2 Saran .....	21
DAFTAR PUSTAKA .....	22
LAMPIRAN .....	23

## DAFTAR GAMBAR

Gambar 5.1. Orientasi arah sensor sebuah perangkat android .....	8
Gambar 5.2. Koordinat layar dan VRAM .....	10
Gambar 5.3. Penggabungan warna .....	11
Gambar 5.4. Pengkodean warna .....	11
Gambar 5.5. Penggabungan dua citra .....	12
Gambar 5.6. Penggabungan dua warna .....	13

## DAFTAR TABEL

<b>Tabel 3.1</b> Perbandingan interface dan kelas abstrak .....	5
---	---

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Teknologi mobile berkembang sangat pesat, masing-masing vendor berlomba-lomba mengusung teknologi baru, baik teknologi perangkat keras maupun perangkat lunak. Perkembangan tersebut bertujuan salah satunya untuk memudahkan dan memanjakan para calon penggunanya. Salah satu teknologi yang perkembangannya sangat pesat adalah perangkat mobile berbasis android.

Android merupakan sebuah sistem operasi yang sebelumnya dikembangkan khusus untuk telepon genggam. Berkembangnya teknologi internet adalah salah satu faktor yang mempengaruhi perkembangan teknologi mobile. Karena menghadirkan kemudahan bagi para pengguna perangkat elektronik untuk mengakses berbagai jenis konten melalui perangkat mobile seperti ponsel pintar, tablet, bahkan beberapa perangkat yang dapat dikategorikan sebagai bukan perangkat yang mobile seperti iTV (interactive Television), Set TV kabel yang memungkinkan pengguna untuk dapat berinteraksi dengan perangkat TV tersebut untuk mengakses internet. Hal ini menjadikan penambahan media-media baru seperti game, animasi 3D, audio digital bahkan informasi visual dengan grafis yang sangat detil mulai ada di sekeliling kita. Android masuk sebagai salah satu sistem yang mengusung pembangunan media kreatif ke level yang lebih maju yang belum pernah dirasakan oleh pengguna sebelumnya.

Android memiliki kemampuan yang setara dengan sistem operasi sebuah komputer pribadi yang berbasis pada sistem operasi Linux. Dan Java yang merupakan salah satu bahasa pemrograman paling populer di dunia. Android didasarkan pada sistem *open source*. Sehingga para pengembang punya kesempatan yang sama untuk berkreasi mengembangkan dan memanfaatkan sumber daya teknologi yang tersedia pada sebuah perangkat.

Dikarenakan teknologi yang berkembang pada suatu perangkat berbeda antara buatan vendor satu dengan vendor lainnya, bahkan versi perangkat satu dengan versi lainnya dari satu vendor yang sama. Maka dengan kemampuan bahasa pemrograman java, pengembang dapat dengan mudah membangun suatu kerangka kerja aplikasi yang dinamis. Yang dapat menyesuaikan dengan perkembangan teknologi tanpa harus merombak secara total dari aplikasi yang pernah dibuat ketika aplikasi tersebut akan dikembangkan ke versi berikutnya, atau ketika ingin menambahkan fitur-fitur baru. Salah satunya adalah dengan memanfaatkan *interface* dari bahasa pemrograman java. Penggunaan *interface* sangat membantu dalam mendesain dan membangun kerangka kerja / framework dari suatu aplikasi di android.

Selain perkembangan teknologi pada perangkat mobile, hal yang selalu berubah lainnya adalah konten. Melalui penelitian ini, bermaksud untuk menganalisis bagaimana cara android menangani konten-konten multimedia yang semakin berubah baik dari segi ukuran maupun kualitas konten. Seperti konten gambar yang memiliki resolusi beragam, atau konten audio bahkan animasi.

Dengan memanfaatkan interface yang dimiliki bahasa pemrograman java, diharapkan agar penanganan masing-masing konten dapat dilakukan secara modular. Sehingga memudahkan ketika terjadi perubahan spesifikasi konten, tidak harus merubah seluruh sistem.

## **1.2 Perumusan Masalah**

Berdasarkan latar belakang masalah yang telah dikemukakan sebelumnya, dapat dirumuskan sebagai berikut:

- a. Bagaimana sistem android menangani konten multimedia yang berupa teks, gambar, audio, hingga animasi grafis.
- b. Bagaimana pemanfaatan java interface dalam pengembangan aplikasi multimedia pada perangkat android



## **BAB II**

### **TUJUAN DAN MANFAAT PENELITIAN**

#### **2.1 Tujuan Penelitian**

Penelitian ini bertujuan untuk mengetahui penanganan sistem android terhadap konten-konten grafis dalam pembuatan suatu aplikasi multimedia dengan mengoptimalkan fungsi interface yang dimiliki oleh bahasa pemrograman java. Interface digunakan sebagai perantara komunikasi antar kelas yang berjalan pada platform android. Dibuat secara modular sehingga akan memudahkan dalam pengembangan dan pembaruan kode program.

#### **2.2 Manfaat Penelitian**

Manfaat yang ingin dicapai pada penelitian ini adalah untuk memberikan wawasan tentang bagaimana menerapkan interface pada sistem berbasis android. Sekaligus memberikan gambaran pembuatan kerangka-kerangka aplikasi multimedia.

### BAB III

#### TINJAUAN PUSTAKA

Pemrograman berorientasi objek memungkinkan kita untuk memodelkan suatu kasus di dunia nyata untuk diterapkan ke dalam sistem yang terkomputerisasi. Dengan mengetahui konsep dasar sebuah objek di dunia nyata, maka kita dapat merepresentasikan objek tersebut berdasarkan dua buah karakteristik yaitu keadaan sebuah objek dan perilakunya. Dalam konsep pemrograman ini, dikenal satu kata kunci utama yaitu *class*. *Class*/kelas merupakan sebuah purwarupa dari objek yang dibuat. Kelas berisikan definisi-definisi yang terdiri dari keadaan suatu objek ke dalam bentuk atribut dan perilaku objek ke dalam bentuk metode/*method*.

Suatu kelas dapat diakses secara langsung maupun melalui perantara sebuah *interface*. *Interface* dibuat untuk menjadi antarmuka bagi sebuah kelas dengan dunia luar. Interface berisikan definisi-definisi yang ringkas atas perilaku suatu objek yaitu kumpulan method yang saling berkaitan tetapi tidak memiliki badan program. Dengan demikian dunia luar tidak perlu tahu detil implementasi dari objek yang diakses. Hal ini menjadi salah satu kelebihan penggunaan interface pada sistem berbasis android yaitu memungkinkan kita untuk terkonsentrasi pada semantik dari aplikasi yang dibangun tanpa harus menunjukkan detil implementasi program dan memungkinkan kita untuk dapat merubah cara kita mengimplementasi secara mudah dan terpisah (Zechner dan Green, 2012).

Perantara lainnya adalah kelas abstrak. Kelas abstrak memiliki kesamaan dengan interface, yaitu untuk merancang suatu aplikasi berbasis objek tetapi bukan untuk membuat implementasi dari aplikasi yang akan digunakan. Keduanya memiliki karakteristik yang sama seperti kelas pada umumnya, yaitu berisi nama kelas, atribut dan method. Dikarenakan kelas abstrak dan interface tidak mengimplementasikan detil program, maka kelas abstrak dan interface hanya menyertakan nama-nama method tanpa badan program.

Namun di antara kelas abstrak dan interface pun punya beberapa perbedaan mendasar seperti ditunjukkan pada Tabel 1.

**Tabel 3.1.** Perbandingan interface dan kelas abstrak

No	Interface	Kelas abstrak
1	Dapat mendeklarasikan daftar method	Dapat mendeklarasikan daftar method
2	Tidak memiliki kode pada masing-masing method	Kode program pada masing-masing method dapat didefinisikan
3	Hanya mendeklarasikan konstanta	Dapat mendeklarasikan berbagai jenis tipe variabel
4	Tidak memiliki konstruktor	Dapat memiliki konstruktor
5	Mengijinkan pewarisan lebih dari satu	Hanya mengijinkan satu buah pewarisan
6	Tidak memiliki hiarki teratas	Selalu diwariskan dari kelas object
7	Mengijinkan interface induk lebih dari satu	Hanya memiliki sebuah kelas induk

## **BAB IV**

### **METODE PENELITIAN**

Metode penelitian yang akan digunakan dalam penelitian ini dapat disusun sebagai berikut:

1. Studi literatur

Untuk memperoleh dasar teori yang berkaitan dengan pemrograman berorientasi objek khususnya pemanfaatan interface dalam bahasa java dan bagaimana penerapannya pada sistem android.

2. Akuisisi data

Persiapan yang diperlukan dalam penelitian ini adalah menyediakan konten-konten pendukung multimedia. Konten tersebut berupa informasi-informasi tekstual tentang aplikasi yang akan dibuat, berkas gambar serta berkas audio.

3. Implementasi Interface

Pada tahap ini dilakukan implementasi pemakaian interface untuk membangun sebuah aplikasi multimedia dalam sistem berbasis android.

4. Analisa hasil

Analisa hasil dilakukan untuk melihat bagaimana penggunaan interface pada pembuatan aplikasi multimedia.

## BAB V

### HASIL DAN PEMBAHASAN

Interface yang menangani konten multimedia untuk aplikasi pada perangkat android dibuat sebagai antarmuka antar proses yang terdapat di dalam aplikasi. Dengan menggunakan android developer tool sebagai alat bantu dalam membuat projek dan mengkompilasi menjadi sebuah aplikasi.

Pembuatan aplikasi dipecah ke dalam beberapa bagian yang selanjutnya diintegrasikan menjadi sebuah kerangka kerja/*framework* untuk memudahkan dalam membangun komunikasi antar proses di dalam sistem dengan platform android.

#### 1.1 Modul-modul Aplikasi

Setiap modul dapat memiliki setidaknya sebuah *interface* atau lebih. Dan masing-masing *interface* memiliki setidaknya satu buah kelas yang mengimplementasikannya.

##### 1.1.1 Masukan

Modul masukan ini menangani bagaimana pengguna dapat berinteraksi dengan aplikasi melalui beberapa alat masukan antara lain fungsi sentuh, penekanan pada tombol tertentu, atau masukan berupa pembacaan sensor asccelerometer.

Pada perangkat berbasis android, terdapat tiga buah metode masukan yang dikenal yaitu layar sentuh, papan ketik dan sensor (*accelerometer*). Masing-masing masukan tersebut dapat membangkitkan berbagai *event* tertentu. Untuk layar sentuh membangkitkan event sebagai berikut:

- a. Touch-down: event yang terjadi ketika jari menyentuh layar
- b. Touch-drag: event yang terjadi ketika jari digerakkan secara mengusap di atas layar.
- c. Touch-up: event yang terjadi ketika jari diangkat/melepaskan dari layar.

Setiap event sentuh akan merekam informasi tentang posisi sentuh jari terhadap komponen antarmuka pengguna yang sedang aktif dan sebuah pointer

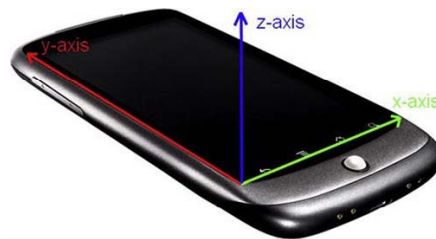
indeks yang digunakan untuk merekam ketika jari yang menyentuh layar lebih dari satu jari.

Selanjutnya untuk papan ketik/*keyboard* akan membangkitkan dua buah event yaitu:

- a. Key-down: event ketika sebuah tombol ditekan.
- b. Key-up: even ketika sebuah tombol dilepas.

Event pada papan ketik ini menyimpan informasi tentang key-code dari tombol yang ditekan beserta karakter *unicode* yang sesuai dengan tombol tersebut.

Event yang terakhir yaitu pembacaan sensor accelerometer. Sensor ini memberikan nilai keluaran yang berkaitan dengan posisi perangkat terhadap gravitasi bumi. Yaitu nilai koodrinat x, y, dan z seperti yang ditunjukkan pada gambar 5.1.



Gambar 5.1 Orientasi arah sensor sebuah perangkat android

Nilai untuk masing masing sumbu dinotasikan dengan meter perdetik kuadrat ( $m/s^2$ ). Di mana merepresentasikan bahwa sebuah benda akan melakukan percepatan pada  $9,8m/s^2$  ketika benda tersebut jatuh bebas.

Dari tiga macam masukan di atas, *interface* yang dibuat untuk keperluan tersebut ditunjukkan pada kode program berikut ini:

```
public interface Input {
    public interface Masukan {
        public static class EventTombol {
            public static final int KEY_DOWN = 0;
            public static final int KEY_UP = 1;
            public int type;
            public int keyCode;
            public char keyChar;
        }
        public static class EventSentuh {
            public static final int TOUCH_DOWN = 0;
            public static final int TOUCH_UP = 1;
            public static final int TOUCH_DRAGGED = 2;
            public int type;
            public int x, y;
            public int pointer;
        }
    }
}
```

```

public boolean isKeyPressed(int keyCode);
public boolean isTouchDown(int pointer);
public int getTouchX(int pointer);
public int getTouchY(int pointer);
public float getAccelX();
public float getAccelY();
public float getAccelZ();
public List<KeyEvent> getKeyEvents();
public List<TouchEvent> getTouchEvents();
}

```

### 1.1.2 Berkas I/O

Bagian ini menangani bagaimana aplikasi mengakses berkas eksternal yang diletakkan sebagai aset dari aplikasi. Membaca atau menulis suatu berkas merupakan hal yang mendasar dalam pembuatan aplikasi multimedia. Dalam pemrograman java, mekanisme yang sering digunakan adalah dengan membuat instan dari InputStream dan OutputStream. Adapun interface yang menangani berkas I/O ditunjukkan pada kode program berikut ini.

```

public interface FileIO {
    public InputStream readAsset(String fileName) throws IOException;
    public InputStream readFile(String fileName) throws IOException;
    public OutputStream writeFile(String fileName) throws IOException;
}

```

### 1.1.3 Audio

Bagian ini mengatur bagaimana aplikasi dalam memuat dan memainkan berkas audio. Tanpa melakukan berbagai pengolahan pada berkas audio. Dalam memainkan berkas audio, mekanismenya dibagi menjadi dua macam yaitu memainkan audio yang menjadi suara latar dari aplikasi dan memainkan suara efek. Perancangan kode interface ditunjukkan pada kode program berikut ini.

```

public interface Audio {
    public SuaraLatar newMusic(String filename);
    public SuaraEfek newSound(String filename);
}

```

```

public interface SuaraEfek {
    public void play(float volume);
    public void dispose();
}

```

```

public interface SuaraLatar {
    public void play();
    public void stop();
    public void pause();
    public void setLooping(boolean looping);
    public void setVolume(float volume);
    public boolean isPlaying();
    public boolean isStopped();
    public boolean isLooping();
    public void dispose();
}

```

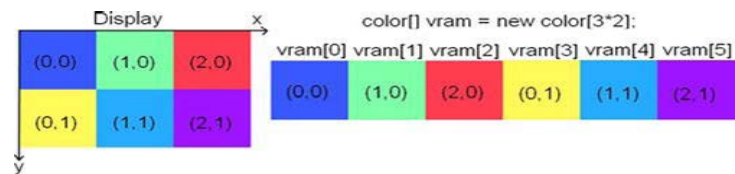
```
}
```

#### 1.1.4 Grafis

Bagian ini merupakan bagian yang kompleks berkaitan dengan aplikasi yang hendak dibuat. Namun secara khusus bertugas mengelola bagaimana aplikasi memuat dan menampilkan beragam citra ke layar. Grafis yang dibuat dalam bentuk 2 dimensi.

##### a. Koordinat Layar

Sistem koordinat pada perangkat penampil elektronik berbasiskan raster. Atau yang lebih dikenal dengan sebutan *pixel*. Pixel memiliki dua buah atribut yaitu posisi dan warna. Posisi dari suatu pixel inilah yang merepresentasikan koordinat dari layar penampil tersebut. Di mana berupa nilai diskrit. Koordinat ini dimulai dengan koordinat (0,0) dari pojok kiri atas dan selalu bernilai positif. Koordinat layar sebuah perangkat ditunjukkan pada gambar 5.2.

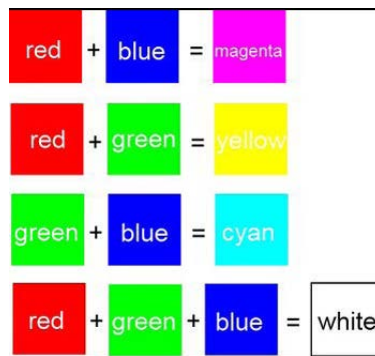


Gambar 5.2. Koordinat layar dan VRAM

##### b. Warna

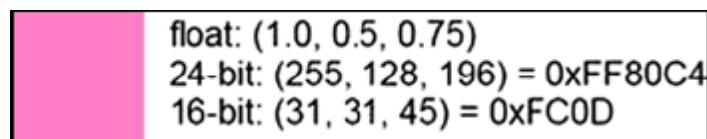
Model warna yang diterapkan pada pemrograman grafis adalah menggunakan model RGB. RGB merupakan singkatan dari warna merah, hijau dan biru. Untuk menghasilkan warna lain maka dapat diperoleh dengan mengkombinasikan nilai RGB seperti ditunjukkan pada Gambar 5.3.





Gambar 5.3. Penggabungan warna

Terdapat beberapa cara untuk mengkomputasikan warna sebuah pixel ke dalam bentuk kode yang kemudian disimpan oleh memori tergantung pada tipe data yang hendak digunakan. Dapat berupa bilangan pecahan maupun bilangan bulat positif. Tetapi hal tersebut akan mempengaruhi konsumsi memori. Salah satu contoh pengkodean warna ditunjukkan seperti pada Gambar 5.4.



Gambar 5.4. Pengkodean warna

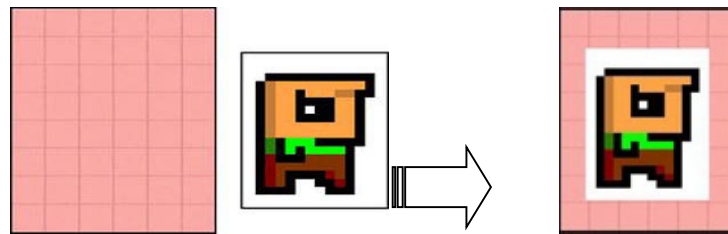
- Pengkodean RGB dengan tipe data pecahan 32-bit, sehingga satu buah pixel mendapat alokasi memori sebesar 12-byte dengan rentang nilai 0,0 – 1,0
- Pengkodean RGB 24-bit mendapat alokasi untuk satu pixel sebesar 3 atau 4 byte dengan intensitas berada pada rentang nilai 0 sampai 255. Terdapat dua buah urutan pengkodean yaitu RGB dan BGR yang dikenal dengan RGB888 atau BGR888 di mana angka delapan merepresentasikan jumlah bit untuk masing-masing elemen warna.
- Pengkodean RGB 16-bit mengalokasikan 2 byte untuk satu buah pixel. Warna merah dan biru memiliki intensitas dengan rentang nilai 0-31 sedangkan warna hijau memiliki intensitas dengan rentang nilai 0-63.

c. Format Citra dan Kompresinya

Pemilihan format dari citra yang akan digunakan sangat mempengaruhi ukuran dari citra itu sendiri. Sehingga akan berdampak pemborosan memori dan kinerja aplikasi menjadi lebih berat dalam memuat citra tersebut. Format yang paling umum dengan kompresi yang baik dan dapat diterapkan pada sistem android, adalah citra dengan tipe JPEG dan PNG.

d. Penggabungan Citra

Hal terakhir yang perlu menjadi perhatian dalam menampilkan citra ke layar adalah bagaimana menggabungkan dua buah citra. Sebuah citra ketika akan ditampilkan, di mana akan digabung dengan citra latar dari aplikasi. Contoh penggabungan ditunjukkan pada Gambar 5.4.



Gambar 5.5. Penggabungan dua citra

Terlihat bahwa citra hasil mengandung efek warna putih pada tepian citra dari karakter asli. Permasalahan ini dapat dipecahkan dengan mekanisme penggabungan gambar *alpha* (*alpha blending*). Pada kasus yang ditunjukkan gambar xx secara teknis disebut sebagai *alpha masking*.

Dari tiga format pengkodean warna yang telah dijelaskan sebelumnya, kita dapat menyimpan pengkodean RGB 24-bit pada 32-bit bilangan bulat. Sehingga diperoleh 8 bit yang belum digunakan. 8 bit tersebut kemudian dimanfaatkan untuk menyimpan nilai *alpha*, dengan rentang nilai dari 0 sampai 255. Di mana nilai 0 adalah transparan dan nilai 255 berwarna normal. Pengkodean ini kemudian dikenal dengan ARGB8888 atau BGRA8888.

Salah satu Algoritma yang digunakan untuk menggabungkan dua buah citra dapat menggunakan persamaan 5.1. Namun untuk kesederhanaan logika komputasi, persamaan ini hanya digunakan untuk mengkomputasi nilai pixel R, G dan B tanpa mengkomputasi nilai pixel alpha.

$$\begin{aligned} \text{red} &= \text{src.red} * \text{src.alpha} + \text{dst.red} * (1 - \text{src.alpha}) \\ \text{blue} &= \text{src.green} * \text{src.alpha} + \text{dst.green} * (1 - \text{src.alpha}) \\ \text{green} &= \text{src.blue} * \text{src.alpha} + \text{dst.blue} * (1 - \text{src.alpha}) \dots\dots\dots (5.1) \end{aligned}$$

*src* dan *dst* adalah nilai pixel dari citra asal dan citra tujuan yang hendak digabung. Contoh penerapan persamaan 5.1 ditunjukkan pada Gambar 5.5. Di mana mengilustrasikan penggabungan dua buah citra dengan warna merah muda dan hijau terang, menghasilkan citra dengan warna hijau gelap.



Gambar 5.6. Penggabungan dua warna

Jika diperhatikan, bahwa persamaan 5.1 tersebut memiliki komputasi perkalian yang banyak. Jika memungkinkan hal ini harus dihindari dalam proses penggabungan citra terlebih jika media komputasi berupa perangkat mobile. Kita dapat melakukan proses pra-perkalian antara nilai pixel RGB dengan nilai *alpha*-nya.

Untuk menampilkan citra dengan API grafis harus memperhatikan persamaan yang sesuai. Citra yang hendak digunakan pun harus dipastikan apakah mengandung nilai *alpha* atau tidak. Berkaitan dengan hal ini, API grafis yang dimiliki sistem Android memberikan kemudahan dalam proses penggabungan citra.

Perancangan kode interface dibuat berdasarkan kebutuhan mengolah grafis sebagai berikut ini:

- Memuat citra dari media penyimpan kemudian menyimpannya ke memori
- Membersihkan *framebuffer* dan menyiapkannya dengan sebuah warna sehingga tidak ada sisa dari frame sebelumnya.
- Mengatur nilai pixel pada area pixel tertentu dengan warna tertentu
- Gambar garis dan persegi panjang pada *framebuffer*

- Gambar citra yang telah dimuat ke dalam memori sebelumnya pada *framebuffer*
- Baca ukuran dimensi dari *framebuffer*

Dari kebutuhan di atas, kemudian dibuat dalam bentuk kode interface seperti pada kode berikut ini.

```
public interface Graphics {
    public static enum PixmapFormat {
        ARGB8888, ARGB4444, RGB565
    }
    public Pixmap newPixmap(String fileName, PixmapFormat format);
    public void clear(int color);
    public void drawPixel(int x, int y, int color);
    public void drawLine(int x, int y, int x2, int y2, int color);
    public void drawRect(int x, int y, int width, int height, int color);
    public void drawPixmap(Pixmap pixmap, int x, int y, int srcX, int srcY,
        int srcWidth, int srcHeight);
    public void drawPixmap(Pixmap pixmap, int x, int y);
    public int getWidth();
    public int getHeight();
}
```

```
public interface Pixmap {
    public int getWidth();
    public int getHeight();
    public PixmapFormat getFormat();
    public void dispose();
}
```

### 1.1.5 Pengelola Tampilan

Modul masukan ini Bertanggungjawab untuk mengelola pembuatan tampilan dan mengatur bagaimana menutup sebuah tampilan, memberhentikan atau menjalankan kembali sebuah tampilan. Sekaligus sebagai program utama yang menggabungkan interface-interface dan digunakan untuk membuat sebuah aplikasi multimedia.

Adapun fitur yang diperlukan dalam membangun aplikasi multimedia adalah sebagai berikut:

- Menyiapkan window dan antarmuka pengguna dan mengesetnya agar dapat menangkap window dan *event* masukan.
- Jalankan thread utama
- Rekam setiap aktivitas tampilan layar
- Berikan akses penuh untuk masing-masing modul lainnya

Kode interface untuk rancangan di atas ditunjukkan pada kode program berikut ini

```

public interface ApMultimedia {
    public Input getInput();
    public FileIO getFileIO();
    public Graphics getGraphics();
    public Audio getAudio();
    public void setScreen(Screen screen);
    public Screen getCurrentScreen();
    public Screen getStartScreen();
}

```

Bagian terakhir adalah pembuatan kelas abstrak Layar. Pemilihan penggunaan kelas abstrak daripada menggunakan kelas interface adalah untuk proses implementasi pemakaian layar yang hanya dapat menampilkan satu tampilan dalam satu layar.

```

public abstract class Layar {
    protected final ApMultimedia apmult;
    public Screen(ApMultimedia apmult) {
        this.apmult = apmult;
    }
    public abstract void update(float deltaTime);
    public abstract void present(float deltaTime);
    public abstract void pause();
    public abstract void resume();
    public abstract void dispose();
}

```

## 1.2 Implementasi Aplikasi Multimedia

Implementasi dari masing-masing interface adalah sebagai berikut

### a. Implementasi Masukan

```

public class ImplInput implements Input {
    AccelerometerHandler accelHandler;
    KeyboardHandler keyHandler;
    TouchHandler touchHandler;
    public ImplInput(Context context, View view, float scaleX, float scaleY) {
        accelHandler = new AccelerometerHandler(context);
        keyHandler = new KeyboardHandler(view);
        if (Integer.parseInt(VERSION.SDK) < 5)
            touchHandler = new SingleTouchHandler(view, scaleX, scaleY);
        else
            touchHandler = new MultiTouchHandler(view, scaleX, scaleY);
    }
    public boolean isKeyPressed(int keyCode) {
        return keyHandler.isKeyPressed(keyCode);
    }
    public boolean isTouchDown(int pointer) {
        return touchHandler.isTouchDown(pointer);
    }
    public int getTouchX(int pointer) {
        return touchHandler.getTouchX(pointer);
    }
    public int getTouchY(int pointer) {
        return touchHandler.getTouchY(pointer);
    }
    public float getAccelX() {
        return accelHandler.getAccelX();
    }
    public float getAccelY() {
        return accelHandler.getAccelY();
    }
    public float getAccelZ() {
        return accelHandler.getAccelZ();
    }
    public List <TouchEvent> getTouchEvents() {
        return touchHandler.getTouchEvents();
    }
    public List <KeyEvent> getKeyEvents() {
        return keyHandler.getKeyEvents();
    }
}

```

```
}  
}
```

## b. Implementasi Berkas I/O

```
public class ImplFileIO implements FileIO {  
    Context context;  
    AssetManager assets;  
    String externalStoragePath;  
    public ImplFileIO(Context context) {  
        this.context = context;  
        this.assets = context.getAssets();  
        this.externalStoragePath = Environment.getExternalStorageDirectory()  
            .getAbsolutePath() + File.separator;  
    }  
    public InputStream readAsset(String fileName) throws IOException {  
        return assets.open(fileName);  
    }  
    public InputStream readFile(String fileName) throws IOException {  
        return new FileInputStream(externalStoragePath + fileName);  
    }  
    public OutputStream writeFile(String fileName) throws IOException {  
        return new FileOutputStream(externalStoragePath + fileName);  
    }  
    public SharedPreferences getPreferences() {  
        return PreferenceManager.getDefaultSharedPreferences(context);  
    }  
}
```

## c. Implementasi Audio, SuaraEfek, SuaraLatar

```
public class ImplAudio implements Audio {  
    AssetManager assets;  
    SoundPool soundPool;  
    public ImplAudio(Activity activity) {  
        activity.setVolumeControlStream(AudioManager.STREAM_MUSIC);  
        this.assets = activity.getAssets();  
        this.soundPool = new SoundPool(20, AudioManager.STREAM_MUSIC, 0);  
    }  
    @Override  
    public Music newMusic(String filename) {  
        try {  
            AssetFileDescriptor assetDescriptor = assets.openFd(filename);  
            return new ImplMusic(assetDescriptor);  
        } catch (IOException e) {  
            throw new RuntimeException("Couldn't load music '" + filename + "'");  
        }  
    }  
    @Override  
    public Sound newSound(String filename) {  
        try {  
            AssetFileDescriptor assetDescriptor = assets.openFd(filename);  
            int soundId = soundPool.load(assetDescriptor, 0);  
            return new ImplSound(soundPool, soundId);  
        } catch (IOException e) {  
            throw new RuntimeException("Couldn't load sound '" + filename + "'");  
        }  
    }  
}
```

```
public class ImplSound implements SuaraEfek {  
    int soundId;  
    SoundPool soundPool;  
    public ImplSound(SoundPool soundPool, int soundId) {  
        this.soundId = soundId;  
        this.soundPool = soundPool;  
    }  
    @Override  
    public void play(float volume) {  
        soundPool.play(soundId, volume, volume, 0, 0, 1);  
    }  
    @Override  
    public void dispose() {  
        soundPool.unload(soundId);  
    }  
}
```

```
}
```

```
public class ImplMusic implements SuaraLatar, OnCompletionListener {

    MediaPlayer mediaPlayer;
    boolean isPrepared = false;

    public ImplMusic(AssetFileDescriptor assetDescriptor) {
        mediaPlayer = new MediaPlayer();
        try {
            mediaPlayer.setDataSource(assetDescriptor.getFileDescriptor(),
                assetDescriptor.getStartOffset(),
                assetDescriptor.getLength());
            mediaPlayer.prepare();
            isPrepared = true;
            mediaPlayer.setOnCompletionListener(this);
        } catch (Exception e) {
            throw new RuntimeException("Couldn't load music");
        }
    }

    @Override
    public void play() {
        if (mediaPlayer.isPlaying())
            return;
        try {
            synchronized (this) {
                if (!isPrepared)
                    mediaPlayer.prepare();
                mediaPlayer.start();
            }
        } catch (IllegalStateException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    @Override
    public void stop() {
        mediaPlayer.stop();
        synchronized (this) {
            isPrepared = false;
        }
    }

    @Override
    public void pause() {
        if (mediaPlayer.isPlaying())
            mediaPlayer.pause();
    }

    @Override
    public void setLooping(boolean isLooping) {
        mediaPlayer.setLooping(isLooping);
    }

    @Override
    public void setVolume(float volume) {
        mediaPlayer.setVolume(volume, volume);
    }

    @Override
    public boolean isPlaying() {
        return mediaPlayer.isPlaying();
    }

    @Override
    public boolean isStopped() {
        return !isPrepared;
    }

    @Override
    public boolean isLooping() {
        return mediaPlayer.isLooping();
    }

    @Override
    public void dispose() {
        if (mediaPlayer.isPlaying())
            mediaPlayer.stop();
        mediaPlayer.release();
    }

    @Override
```

```

        public void onCompletion(MediaPlayer arg0) {
            synchronized (this) {
                isPrepared = false;
            }
        }
    }
}

```

#### d. Implementasi Grafis

```

public class ImplGraphics implements Graphics{
    AssetManager assets;
    Bitmap frameBuffer;
    Canvas canvas;
    Paint paint;
    Rect srcRect = new Rect();
    Rect dstRect = new Rect();
    public ImplGraphics(AssetManager assets, Bitmap frameBuffer) {
        this.assets = assets;
        this.frameBuffer = frameBuffer;
        this.canvas = new Canvas(frameBuffer);
        this.paint = new Paint();
    }
    public Pixmap newPixmap(String fileName, PixmapFormat format) {
        Config config = null;
        if (format == PixmapFormat.RGB565)
            config = Config.RGB_565;
        else if (format == PixmapFormat.ARGB4444)
            config = Config.ARGB_4444;
        else
            config = Config.ARGB_8888;
        Options options = new Options();
        options.inPreferredConfig = config;
        InputStream in = null;
        Bitmap bitmap = null;
        try {
            in = assets.open(fileName);
            bitmap = BitmapFactory.decodeStream(in);
            if (bitmap == null)
                throw new RuntimeException("Couldn't load bitmap from asset '"
                    + fileName + "'");
        } catch (IOException e) {
            throw new RuntimeException("Couldn't load bitmap from asset '"
                + fileName + "'");
        } finally {
            if (in != null) {
                try {
                    in.close();
                } catch (IOException e) {
                }
            }
        }
        if (bitmap.getConfig() == Config.RGB_565)
            format = PixmapFormat.RGB565;
        else if (bitmap.getConfig() == Config.ARGB_4444)
            format = PixmapFormat.ARGB4444;
        else
            format = PixmapFormat.ARGB8888;
        return new AndroidPixmap(bitmap, format);
    }

    public void clear(int color) {
        canvas.drawRGB((color & 0xff0000) >> 16, (color & 0xff00) >> 8,
            (color & 0xff));
    }

    public void drawPixel(int x, int y, int color) {
        paint.setColor(color);
        canvas.drawPoint(x, y, paint);
    }

    public void drawLine(int x, int y, int x2, int y2, int color) {
        paint.setColor(color);
        canvas.drawLine(x, y, x2, y2, paint);
    }

    public void drawRect(int x, int y, int width, int height, int color) {
        paint.setColor(color);
        paint.setStyle(Style.FILL);
    }
}

```



```

        canvas.drawRect(x, y, x + width - 1, y + width - 1, paint);
    }

    public void drawPixmap(Pixmap pixmap, int x, int y, int srcX, int srcY,
        int srcWidth, int srcHeight) {
        srcRect.left = srcX;
        srcRect.top = srcY;
        srcRect.right = srcX + srcWidth - 1;
        srcRect.bottom = srcY + srcHeight - 1;
        dstRect.left = x;
        dstRect.top = y;
        dstRect.right = x + srcWidth - 1;
        dstRect.bottom = y + srcHeight - 1;
        canvas.drawBitmap(((AndroidPixmap) pixmap).bitmap, srcRect, dstRect, null);
    }

    public void drawPixmap(Pixmap pixmap, int x, int y) {
        canvas.drawBitmap(((AndroidPixmap)pixmap).bitmap, x, y, null);
    }

    public int getWidth() {
        return framebuffer.getWidth();
    }
    public int getHeight() {
        return framebuffer.getHeight();
    }
}

```

### e. Implementasi Layar

```

public class AndroidApMultimedia extends Activity implements ApMultimedia{
    AndroidFastRenderView renderView;
    Graphics graphics;
    Audio audio;
    Input input;
    FileIO fileIO;
    Layar screen;
    WakeLock wakeLock;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //setContentView(R.layout.activity_main);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);
        boolean isLandscape = getResources().getConfiguration().orientation ==
            Configuration.ORIENTATION_LANDSCAPE;
        int framebufferWidth = isLandscape ? 480 : 320;
        int framebufferHeight = isLandscape ? 320 : 480;
        Bitmap framebuffer = Bitmap.createBitmap(framebufferWidth,
            framebufferHeight, Config.RGB_565);
        float scaleX = (float) framebufferWidth
            / getWindowManager().getDefaultDisplay().getWidth();
        float scaleY = (float) framebufferHeight
            / getWindowManager().getDefaultDisplay().getHeight();
        renderView = new AndroidFastRenderView(this, framebuffer);
        graphics = new AndroidGraphics(getAssets(), framebuffer);
        fileIO = new AndroidFileIO(this);
        audio = new AndroidAudio(this);
        input = new AndroidInput(this, renderView, scaleX, scaleY);
        screen = getStartScreen();
        setContentView(renderView);
        PowerManager powerManager = (PowerManager) getSystemService(Context.POWER_SERVICE);
        wakeLock = powerManager.newWakeLock(PowerManager.FULL_WAKE_LOCK, "GLGame");
    }

    @Override
    public void onResume() {
        super.onResume();
        wakeLock.acquire();
        screen.resume();
        renderView.resume();
    }

    @Override
    public void onPause() {
        super.onPause();
        wakeLock.release();
        renderView.pause();
        screen.pause();
    }
}

```

```
        if (isFinishing())
            screen.dispose();
    }
    @Override
    public Input getInput() {
        return input;
    }
    @Override
    public FileIO getFileIO() {
        return fileIO;
    }
    @Override
    public Graphics getGraphics() {
        return graphics;
    }

    @Override
    public Audio getAudio() {
        // TODO Auto-generated method stub
        return audio;
    }
    @Override
    public void setScreen(Screen screen) {
        if (screen == null)
            throw new IllegalArgumentException("Screen must not be null");
        this.screen.pause();
        this.screen.dispose();
        screen.resume();
        screen.update(0);
        this.screen = screen;
    }
    @Override
    public Screen getCurrentScreen() {
        return screen;
    }
    @Override
    public Screen getStartScreen() {
        return null;
    }
}
```

## **BAB VI**

### **PENUTUP**

#### **6.1 Simpulan**

- a. Pembuatan aplikasi secara modular memudahkan dalam konsentrasi pengembangan kode utama
- b. Penerapan interface java membantu dalam pemrograman modular
- c. Penanganan konten multimedia harus dilakukan secara terpisah dan spesifik
- d. Untuk penanganan konten berupa animasi grafis belum sepenuhnya diimplementasikan

#### **6.2 Saran**

Untuk penelitian selanjutnya dapat dikembangkan berupa:

- a. Pengembangan aplikasi dapat diarahkan dengan menggunakan *framework*
- b. Aplikasi multimedia yang dibuat dapat dibuat lebih implementatif dan lebih menarik.

## DAFTAR PUSTAKA

- Friesen, J., 2010. Learn Java for Android Development. Apress Publishing, New York.
- Lemay, L., Perkins, C., L., 1996. Teach Yourself Java in 21 days. Sams.net Publishing, Indianapolis.
- Mahmoud, O., 2001. Learning Wireless Java. O'reilly & Association, California.
- Nugroho, Y., 2009. Handout: Pengenalan Bahasa Java – Inheritance Interface, ITB, Bogor
- Zechner, M., Green, R., 2012. Begining Android Games, Second Edition. Apress Media, New York
- [\\_http://docs.oracle.com/javase/tutorial/java/conceps/interface.html](http://docs.oracle.com/javase/tutorial/java/conceps/interface.html) diakses pada 5 Mei 2015 pukul 10.10wib.

## LAMPIRAN I:

### Personalia Penelitian

Personalia yang terlibat di dalam penelitian ini adalah mereka yang sesuai dengan bidang yang diteliti dan bisa melakukan penelitian dengan memperhatikan beban tugas lain yang harus dikerjakannya. Adapun personil yang terlibat dalam penelitian ini adalah sebagai berikut:

#### 1. Ketua Peneliti

- a. Nama Lengkap : Eddy Nuraharjo, ST, M.Cs
- b. Jenis Kelamin : Laki-laki
- c. NIY/NIDN : YU.2.04.04.065
- d. Disiplin Ilmu : Ilmu Komputer
- e. Pangkat/Golongan : Penata Muda / IIIB
- f. Fakultas/Program Studi : Teknologi Informasi/Teknik Informatika
- g. Waktu Penelitian : 12 jam / minggu

#### 2. Anggota Peneliti

##### 2.1. Anggota Peneliti 1

- a. Nama Lengkap : Agung Prihandono, S.Kom
- b. Jenis Kelamin : Laki-laki
- c. NIY/NIDN : YU.2.09.11.080
- d. Disiplin Ilmu : Ilmu Komputer
- e. Pangkat/Golongan : Tenaga Pengajar / IIIB
- f. Fakultas/Program Studi : Teknologi Informasi/Teknik Informatika
- g. Waktu Penelitian : 12 jam / minggu

##### 2.2. Anggota Peneliti 2

- a. Nama Lengkap : Henri Puspitasari
- b. NIM : 12.01.53.0094
- c. Jabatan Akademik : Mahasiswa
- d. Fakultas/Program Studi : Teknologi Informasi/Teknik Informatika
- e. Bidang Keahlian : Teknik Informatika

**LAMPIRAN III:**

**DAFTAR RIWAYAT HIDUP**

**TIM PENELITIAN**

**KETUA :**

- a. Nama : Eddy Nuraharjo, ST, M.Cs
- b. NIY/NIDN : YU. 2.04.04.065 / 0628127301
- c. Jenis Kelamin : Pria
- d. Pangkat / Golongan : Penata Muda / III B
- e. Jabatan Fungsional : Asisten Ahli
- f. Bidang Keahlian : Ilmu Komputer
- g. Fakultas / Progdi : Teknologi Informasi / Teknik Informatika
- h. Pengalaman Penelitian

No.	Tahun	Judul Penelitian	Keterangan
1	2013	Implementasi Metode <i>Center Plotting of Image Pixels</i> Untuk Mendeteksi Warna Citra Bidang Datar 2-D	Ketua
2	2012	Rekayasa Sistem Informasi Pemotongan Kayu Menggunakan Algoritma Greedy dan Perhitungan Pendapatannya	Anggota
3	2012	Implementasi Morphology Concept and Technique dalam pengolahan citra digital untuk menentukan batas obyek dan latar belakang citra	Ketua
4	2011	Rancang Bangun Sistem Monitoring Tingkat Ketinggian Air Bendungan Berbasis Mikrokontroler	Anggota
5	2010	Implementasi Image Statistic Method pada Pengolahan Citra Digital	Ketua
6	2009	Analisis Graphic User Interface Berbasis Matlab dalam Pengolahan Citra Digital	Ketua

Semarang, 11 Mei 2015

(Eddy Nuraharjo, ST, M.Cs)

## DAFTAR RIWAYAT HIDUP

### DOSEN ANGGOTA 1:

- a. Nama : Agung Prihandono, S.Kom
- b. NIY / NIDN : YU.2.09.11.080 / 0607108402
- c. Jenis Kelamin : Laki-laki
- d. Fakultas/Progdi : Teknologi informasi / Teknik informatika
- e. Pangkat / Golongan : Tenaga Pengajar / IIIa
- f. Jabatan Fungsional : Tenaga Pengajar
- g. Pengalaman Penelitian :

<b>No</b>	<b>Tahun</b>	<b>Judul Penelitian</b>	<b>Keterangan</b>
1	2009	Analisis Graphical User Interface Berbasis Matlab dalam Pengolahan Citra Digital	Anggota
2	2010	Pengenalan Sidik Jari dengan metode minutiae	Anggota
3	2013	Aplikasi Sistem Persamaan Linier Berbasis GUI	Anggota
4	2014	Analisa Sistem Pencacah Objek Gambar berbasis GUI-DE Matlab	Anggota
5	2015	Rekayasa Sistem Kendali Generator Sinyal XR-2206 Berbasis Arduino UNO R3	Anggota

Semarang, 11 Mei 2015

(Agung Prihandono, S.Kom)

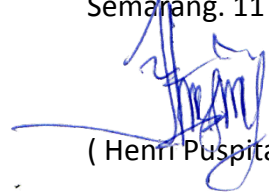
## DAFTAR RIWAYAT HIDUP

### MAHASISWA ANGGOTA I :

- a. Nama : Henri Puspitasari
- b. NIM : 12.01.53.0094
- c. Jenis Kelamin : Wanita
- d. Fakultas / Progdil : Teknologi Informasi / Teknik Informatika
- e. Bidang Keahlian : Teknik Informatika
- f. Pengalaman Penelitian:

<i>No</i>	<i>Tahun</i>	<i>Judul Penelitian</i>	<i>Keterangan</i>
1	-	-	-

Semarang. 11 Mei 2015

  
( Henri Puspitasari )



## DAFTAR RIWAYAT HIDUP

### MAHASISWA ANGGOTA II :

- a. Nama : Hardian Rakhmanto
- b. NIM : 12.01.53.0096
- c. Jenis Kelamin : Laki-laki
- d. Fakultas / Progdil : Teknologi Informasi / Teknik Informatika
- e. Bidang Keahlian : Teknik Informatika
- f. Pengalaman Penelitian:

<i>No</i>	<i>Tahun</i>	<i>Judul Penelitian</i>	<i>Keterangan</i>
1	-	-	-

Semarang. 11 Mei 2015

  
( Hardian Rakhmanto )