

# Rancang Bangun Alat Pengukur Tingkat Kepekatan Asap Berdasarkan Ringelmann Smoke Chart Pada Perangkat Bergerak

M. Choirul R. Fauzi dan Daniel Oranova Siahaan, S.Kom., M.Sc., P.D.Eng.  
Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)  
Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia  
*e-mail*: daniel@if.its.ac.id

**Abstrak**— Pencemaran udara merupakan salah satu masalah yang ditimbulkan oleh pembakaran bahan bakar pada proses industri. Untuk mencegah pencemaran udara, dilakukan kegiatan pengamatan cerobong asap oleh Badan Lingkungan Hidup. Pengamat Badan Lingkungan Hidup akan mengukur tingkat kepekatan asap dari cerobong industri untuk mengetahui apakah asap yang dihasilkan oleh cerobong industri di melebihi batas aman yang sudah ditetapkan atau tidak. Badan Lingkungan Hidup menggunakan skala Ringelmann Smoke Chart untuk menentukan tingkat kepekatan asap.

Selama ini pengukuran tingkat kepekatan asap dilakukan pengamat dengan melihat citra asap melalui teropong dan kemudian membandingkannya dengan Ringelmann Smoke Chart secara manual. Pengukuran dengan cara tersebut memiliki keterbatasan. Pengamatan tidak dapat dilakukan ketika penglihatan pengamat tidak dalam kondisi yang bagus. Selain itu data pengamatan masih ditulis tangan pada kertas laporan pengamatan, sehingga pengguna kesulitan dalam melihat perkembangan kepekatan asap dari cerobong. Untuk mengatasi permasalahan tersebut, dibuatlah aplikasi pengukuran tingkat kepekatan asap dan pencatatan data pengamatan yang berjalan pada *smartphone* berbasis Windows Phone.

Aplikasi ini mengukur tingkat kepekatan asap dengan menghitung *euclidean distance* histogram RGB (Red, Green, Blue) citra asap dengan masing-masing tingkatan pada Ringelmann Smoke Chart. Tingkat kepekatan asap diambil berdasarkan tingkatan pada Ringelmann Smoke Chart yang memiliki *euclidean distance* terendah terhadap citra asap. Selain pengukuran tingkat kepekatan asap, aplikasi ini juga memiliki beberapa fitur lain, yaitu penyimpanan data pengamatan pada *server*, menampilkan riwayat pengukuran, dan *benchmark* nilai kepekatan asap.

**Kata Kunci**— Pengukuran tingkat kepekatan asap, Ringelmann Smoke Chart, Histogram, Windows Phone.

## I. PENDAHULUAN

SEIRING dengan meningkatnya kegiatan industri, khususnya industri yang melakukan pembakaran bahan bakar, maka pencemaran udara pun semakin meningkat. Hal tersebut mendorong pemerintah untuk membuat aturan mengenai pengendalian pencemaran udara yang tertuang pada Keputusan Kepala Bapedal No. 205 Tahun 1996 [1]. Dari peraturan tersebut maka dilakukanlah pengukuran tingkat kepekatan asap oleh Badan Lingkungan Hidup berdasarkan

Ringelmann Smoke Chart.



Gambar 1 Digital Ringelmann Smoke Opacity Meter

Untuk mengetahui tingkat kepekatan asap sulit dilakukan dengan tanpa alat bantu. Salah satu alat bantu yang memudahkan untuk mengetahui tingkat kepekatan dari asap adalah Digital Ringelmann Smoke Opacity Meter. Digital Ringelmann Smoke Opacity Meter yang ditunjukkan oleh Gambar 1 merupakan sebuah alat yang terdiri dari teropong, indikator tingkat kepekatan asap dan kamera. Indikator tingkat kepekatan asap terdiri dari 5 tingkatan sesuai dengan Ringelmann Smoke Chart. Sedangkan kamera digunakan untuk mendokumentasikan citra asap yang sedang diukur tingkat kekekatannya. Cara menentukan tingkat kepekatan asap adalah pengamat melihat asap melalui teropong dan membandingkan asap tersebut sesuai dengan skala Ringlemann.

Selama ini pengukuran tingkat kepekatan asap masih dilakukan secara manual. Yaitu pengamat melihat asap melalui teropong dan membandingkannya dengan Ringelmann Smoke Chart. Kamera hanya digunakan untuk mendokumentasikan kegiatan pengamatan asap. Pengukuran dengan cara tersebut memiliki keterbatasan. Pengamatan tidak dapat dilakukan ketika penglihatan pengamat tidak dalam kondisi yang bagus. Selain itu data pengamatan masih dicatat dengan tulisan tangan pada kertas laporan pengamatan, sehingga pengguna kesulitan dalam melihat perkembangan kepekatan asap dari cerobong. Untuk mengatasi permasalahan di atas, maka akan dibuat sistem untuk melakukan pengukuran tingkat kepekatan asap secara otomatis dan juga pencatatan data

pengamatan. Pengukuran dengan sistem yang akan dibangun ini dilakukan dengan cara memotret asap menggunakan



Gambar 2 Ringelmann Smoke Chart

perangkat bergerak (*smartphone*) yang ditempelkan pada teropong. Kemudian citra asap tersebut diproses pada perangkat bergerak. Perangkat bergerak dipilih karena pengukuran tingkat kepekatan asap dilakukan di lapangan, sehingga pengukuran dan penyimpanan data pengukurannya bisa dilakukan dengan cepat.

## II. TINJAUAN PUSTAKA

### A. Ringelmann Smoke Chart

Ringelmann Smoke Chart merupakan sebuah kartu yang digunakan untuk menentukan tingkat kepekatan asap. Kartu Ringelmann terdiri dari 5 buah tingkatan yang diberi nomor mulai 1-5 sesuai pada Gambar 2. Kartu-kartu tersebut berwarna putih sampai hitam pekat. Selanjutnya kartu-kartu tersebut juga diberi skala dengan % kepekatan.

Cara menggunakan kartu ini sederhana saja yaitu dengan melihat asap yang keluar dari cerobong serta menyamakan warna asap dengan salah satu warna dari kartu Ringelmann sehingga tingkat kepekatan dapat diketahui [2].

### B. Windows Phone SDK 8.0

Windows Phone SDK 8.0 merupakan lingkungan pengembangan yang digunakan untuk membangun aplikasi dan permainan untuk Windows Phone 8.0 dan Windows Phone 7.5. Windows Phone SDK menyediakan Visual Studio Express edisi 2012 untuk Windows Phone yang berdiri sendiri atau sebagai sebuah *add-in* untuk Visual Studio 2012 edisi Professional, Premium atau Ultimate. Dengan SDK, pengembang dapat menggunakan keterampilan pemrograman dan kode yang telah ada untuk membangun aplikasi. Selain itu SDK mencakup beberapa *emulator* dan kakas-kakas tambahan untuk pengujian aplikasi Windows Phone sesuai kondisi dunia nyata [3].

### C. Windows Communication Foundation (WCF)

Windows Communication Foundation (WCF) adalah suatu kerangka kerja untuk membangun aplikasi berorientasi layanan. Menggunakan WCF, pengembang dapat mengirim data sebagai pesan *asynchronous* dari satu *endpoint* layanan lain.

Sebuah *endpoint* layanan dapat menjadi bagian dari layanan tersedia yang diselenggarakan oleh IIS atau dapat menjadi layanan *host* dalam sebuah aplikasi. Sebuah *endpoint* dapat menjadi klien dari layanan yang meminta data dari *endpoint* layanan. Pesan dapat sesederhana satu karakter atau kata dikirim sebagai XML, atau sekomples aliran data biner [4].

### D. Internet Information Services (IIS)

Internet Information Services (IIS, sebelumnya Internet Information Server) adalah sebuah *web server* yang dibuat oleh Microsoft untuk digunakan oleh keluarga Windows NT. IIS mendukung HTTP, HTTPS, FTP, FTPS, SMTP dan NNTP. IIS telah menjadi bagian integral dari keluarga Windows NT sejak Windows NT 4.0, meskipun mungkin absen dari beberapa edisi, misalnya Windows XP Home Edition. IIS tidak diaktifkan secara *default* ketika Windows di-*install*. IIS Manager diakses melalui Microsoft Management Console atau Administrative Tools di Control Panel [5].

### E. Image Retrieval Based on Color Histogram

Adalah sebuah metode yang digunakan untuk mencari kemiripan antar gambar berdasarkan histogram warna berdasarkan *paper* yang ditulis oleh Jain and Vailaya [6]. Berikut ini alur dari metode yang digunakan.

1. Membaca gambar pada *database* dan mengekstrak informasi RGB dari masing-masing pixel pada gambar.
2. Membuat histogram pada *array* dengan panjang 48 yang dinormalisasi untuk masing-masing komponen pada RGB dari setiap gambar dari *database*. Dengan demikian, masing-masing gambar akan memiliki 3 histogram yang terkait dengan itu.
3. Ekstrak informasi RGB pada masing-masing *pixel* pada gambar.
4. Menghitung *euclidean distance* antara gambar-gambar yang dibandingkan.
5. Urutkan gambar dalam *database* dalam urutan menaik berdasarkan *euclidean distance*. Semakin kecil *euclidean distance*, maka semakin besar kemiripan dari dua gambar yang dibandingkan.

Pada *paper* tersebut dilakukan serangkaian pengujian dan didapatkan akurasinya sebesar 66%.

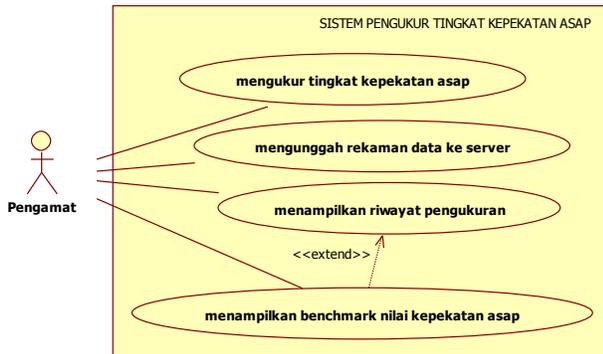
## III. ANALISIS DAN PERANCANGAN

### A. Spesifikasi Kebutuhan

Perilaku sistem dari aplikasi ini digambarkan dalam bentuk diagram *use case* yang menggambarkan hubungan

aktor dengan *use case*. Diagram *use case* aplikasi ini dapat dilihat pada Gambar 3.

Pada aplikasi ini pengguna dapat mengukur tingkat kepekatan asap dan mengunggah data pengamatannya ke *server*, mengunggah rekaman data ke *server*, menampilkan



Gambar 3 Diagram Kasus Penggunaan

riwayat pengukuran, dan menampilkan *benchmark* nilai kepekatan asap.

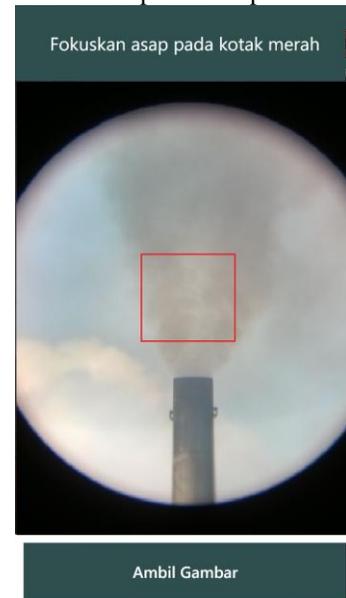
### B. Perancangan Proses Pengukuran Tingkat Kepekatan Asap

Proses pengukuran tingkat kepekatan asap menggunakan metode *image retrieval based on color histogram*. Dengan menggunakan metode tersebut tingkat kepekatan asap didapatkan dengan mencari *euclidean distance* dari histogram RGB yang paling kecil antara citra asap dengan masing-masing skala pada Ringelmann Smoke Chart. Citra asap dan masing-masing skala Ringelmann Smoke Chart yang dibandingkan harus sama-sama berukuran 512x512 *pixel*. Foto asap yang didapatkan dari kamera *smartphone* berukuran 2048x1536 *pixel*. Sehingga untuk mendapatkan *euclidean distance* dari citra asap dan skala Ringelmann Smoke Chart, citra asap dipotong menjadi berukuran 512x512 *pixel*. Citra asap yang dipotong tersebut sesuai dengan citra yang terdapat pada kotak merah pada antarmuka kamera yang ditunjukkan oleh gambar Gambar 4.

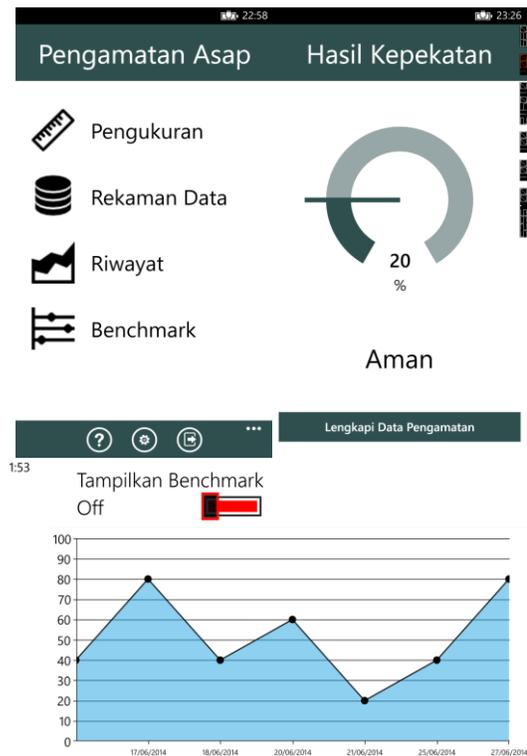
Proses untuk mendapatkan tingkat kepekatan asap adalah sebagai berikut:

1. Memotong citra asap menjadi ukuran 512x512 *pixel* sesuai dengan posisi kotak merah pada antarmuka kamera.
2. Mengekstrak informasi histogram RGB dari citra asap yang berukuran 512x512 *pixel* tersebut.
3. Melakukan normalisasi terhadap histogram RGB dari citra asap dan memasukkannya pada *array* dengan panjang 16 pada masing-masing komponen RGB.
4. Mengekstrak informasi histogram RGB dari citra masing-masing skala Ringelmann Smoke Chart.
5. Melakukan normalisasi terhadap histogram RGB dari citra masing-masing skala Ringelmann Smoke Chart dan memasukkannya pada *array* dengan panjang 16 pada masing-masing komponen RGB.

6. Menghitung *euclidean distance* antara histogram RGB citra asap dengan histogram RGB citra masing-masing skala pada Ringelmann Smoke Chart.
7. Tingkat kepekatan asap didapatkan dari skala pada Ringelmann Smoke Chart yang memiliki *euclidean distance* terkecil terhadap citra asap.



Gambar 4 Antarmuka Kamera



Gambar 5 Tampilan Antarmuka Aplikasi

## IV. IMPLEMENTASI

### A. Implementasi Antarmuka

Halaman antarmuka ini menggunakan komponen antarmuka bertipe *XAML* untuk membangunnya. Aplikasi ini memiliki 17 halaman antarmuka. Tampilan antarmuka aplikasi dapat dilihat pada Gambar 5.

### B. Implementasi Pengukuran Tingkat Kepekatan Asap

Implementasi pengukuran tingkat kepekatan asap ditunjukkan oleh Kode Sumber 1. Fungsi *CropImage* merupakan fungsi untuk memotong citra menjadi berukuran  $512 \times 512$  pixel. Fungsi *ConvertToWbmp* merupakan fungsi untuk mengubah skala Ringelmann Smoke Chart berupa .png menjadi bertipe *WriteableBitmap*. Fungsi *getQuant* merupakan fungsi untuk menormalisasikan histogram RGB ke dalam array dengan panjang 16. Fungsi *GetCumulativeHistogram* merupakan fungsi untuk menggabungkan masing komponen RGB ke dalam 1 array. Fungsi *GetEuclideanDistance* merupakan fungsi untuk menghitung *euclidean distance* antara 2 histogram RGB. Fungsi *GetOpacity* merupakan fungsi untuk mendapatkan nilai kepekatan asap berdasarkan *euclidean distance* yang terkecil.

```
private const int QUANT = 16;

public static void GetOpacity(WriteableBitmap
ImageSource)
{
    DataSingleton.Instance.ChartHistogram = new
int[5][];
    DataSingleton.Instance.ChartHistogram[0] =
OpacityMeasurement.GetCumulativeHistogram(OpacityMea
surement.ConvertToWbmp("20.png"));
    DataSingleton.Instance.ChartHistogram[1] =
OpacityMeasurement.GetCumulativeHistogram(OpacityMea
surement.ConvertToWbmp("40.png"));
    DataSingleton.Instance.ChartHistogram[2] =
OpacityMeasurement.GetCumulativeHistogram(OpacityMea
surement.ConvertToWbmp("60.png"));
    DataSingleton.Instance.ChartHistogram[3] =
OpacityMeasurement.GetCumulativeHistogram(OpacityMea
surement.ConvertToWbmp("80.png"));
    DataSingleton.Instance.ChartHistogram[4] =
OpacityMeasurement.GetCumulativeHistogram(OpacityMea
surement.ConvertToWbmp("100.png"));
    int[] chart = { 20, 40, 60, 80, 100 };

    int[] smokeImage =
GetCumulativeHistogram(CropImage(ImageSource, 768,
512, 512, 512));

    double min = Double.MaxValue;
    int index = -1;
    for (int i = 0; i < 5; i++)
    {
        double has =
GetEuclideanDistance(DataSingleton.Instance.ChartHis
togram[i], smokeImage);
        if (has < min)
        {
            min = has;
            index = i;
        }
    }
}
```

```
DataSingleton.Instance.RingelmannOpacity =
chart[index];
}

private static WriteableBitmap
CropImage(WriteableBitmap source,
int xOffset, int yOffset,
int width, int height)
{
    var sourceWidth = source.PixelWidth;
    var result = new WriteableBitmap(width, height);

    for (var x = 0; x <= height - 1; x++)
    {
        var sourceIndex = xOffset + (yOffset + x) *
sourceWidth;
        var destinationIndex = x * width;
        Array.Copy(source.Pixels, sourceIndex,
result.Pixels, destinationIndex, width);
    }
    DataSingleton.Instance.Pic = result;
    return result;
}

public static WriteableBitmap ConvertToWbmp(string
filename)
{
    StreamResourceInfo sri =
Application.GetResourceStream(new
Uri("Images/"+filename, UriKind.Relative));
    BitmapImage src = new BitmapImage();
    src.SetSource(sri.Stream);
    WriteableBitmap bitmapImage = new
WriteableBitmap(src);
    return bitmapImage;
}

public static int[]
GetCumulativeHistogram(WriteableBitmap bitmapImage)
{
    int[][] arr = getQuant(bitmapImage);
    int[] r = new int[QUANT];
    int[] g = new int[QUANT];
    int[] b = new int[QUANT];
    for (int j = 0; j < arr[0].Length; j++)
    {
        r[arr[0][j]]++;
        g[arr[1][j]]++;
        b[arr[2][j]]++;
    }
    int[] output = new int[r.Length + g.Length +
b.Length];
    r.CopyTo(output, 0);
    g.CopyTo(output, r.Length);
    b.CopyTo(output, r.Length + g.Length);
    return output;
}

private static int[][] getQuant(WriteableBitmap img)
{
    int[][] feature = new int[3][];
    int[] r = new int[img.Pixels.Length];
    int[] g = new int[img.Pixels.Length];
    int[] b = new int[img.Pixels.Length];

    for (int i = 0; i < img.Pixels.Length; i++)
    {
        var pixelbytes =
BitConverter.GetBytes(img.Pixels[i]);
        r[i] = pixelbytes[0] * (QUANT - 1) / 255;
        g[i] = pixelbytes[1] * (QUANT - 1) / 255;
        b[i] = pixelbytes[2] * (QUANT - 1) / 255;
    }

    feature[0] = r;
    feature[1] = g;
    feature[2] = b;
    return feature;
}
```

```
private static double GetEuclideanDistance(int[]
array1, int[] array2)
{
    double Sum = 0.0;
    for (int i = 0; i < array1.Length; i++)
    {
        Sum = Sum + Math.Pow((array1[i] - array2[i]),
2.0);
    }
    return Math.Sqrt(Sum);
}
```

Kode Sumber 1 Sintaks Pengukuran Tingkat Kepekatan Asap

Tabel 1  
Kesimpulan Pengujian Fungsionalitas

ID	Nama	Skenario	Hasil
UJ.UC-0001	Pengujian fitur mengukur tingkat kepekatan asap	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
UJ.UC-0002	Pengujian fitur mengunggah rekaman data ke <i>server</i>	Skenario 1	Berhasil
UJ.UC-0003	Pengujian fitur menampilkan riwayat pengukuran	Skenario 1	Berhasil
UJ.UC-0004	Pengujian fitur menampilkan <i>benchmark</i> nilai kepekatan asap	Skenario 1	Berhasil

## V. PENGUJIAN & EVALUASI

### A. Pengujian Fungsionalitas

Pengujian aplikasi menggunakan metode *black box* untuk mengetahui apakah keluaran dari aplikasi yang telah dibangun sesuai. Pengujian fungsionalitas sistem dilakukan dengan menyiapkan sejumlah skenario sebagai tolok ukur keberhasilan pengujian. Kesimpulan dari pengujian fungsionalitas dapat dilihat pada Tabel 1.

### B. Pengujian Kegunaan & Kemudahan

Berdasarkan hasil dari kuesioner yang diisi oleh pengguna maka didapatkan kesimpulan bahwa aplikasi ini cukup membantu pengguna dalam mengukur tingkat kepekatan asap, mencatat data pengamatan, melihat riwayat pengamatan, dan melihat *benchmark* kepekatan asap. Persentase dari hasil pengujian kegunaan dapat dilihat pada Tabel 2.

Tabel 2  
Hasil Pengujian Kegunaan & Kemudahan

No.	Kegunaan	Persentase
1.	Pengukuran tingkat kepekatan asap dapat dilakukan dengan mudah menggunakan aplikasi ini.	80%
2.	Pengisian data pengamatan melalui aplikasi ini lebih mudah dan cepat dibandingkan dengan pengisian secara manual pada kertas laporan pengamatan.	88%
3.	Fitur rekaman data membantu pengguna untuk tidak mengulang proses	88%

	pengukuran dan pengisian data pengamatan, apabila koneksi internet sedang bermasalah.	
4.	Fitur histori / riwayat mempermudah pengguna untuk <i>tracking</i> perkembangan kepekatan asap cerobong.	87%
5.	<i>Benchmark</i> dapat ditampilkan dengan baik pada aplikasi ini.	81%
Rata-rata		85%

Tabel 3  
Akurasi Pengukuran Oleh Aplikasi

Pengamat	Jumlah Pengukuran yang Sama dengan Aplikasi	Jumlah Foto	Akurasi
Pengamat 1	30	42	71%
Pengamat 2	31	42	74%
Pengamat 3	31	42	74%
Pengamat 4	30	42	71%
Pengamat 5	30	42	71%
Rata-rata			72%

### C. Pengujian Hasil Pengukuran Tingkat Kepekatan Asap

Evaluasi hasil pengukuran tingkat kepekatan asap dilakukan dengan membandingkan hasil pengukuran yang dilakukan oleh aplikasi dengan hasil pengukuran yang dilakukan dengan membandingkan antara foto asap dan skala pada Ringelmann Smoke Chart secara manual. Pengukuran secara manual dilakukan oleh 5 pengamat. Dari masing-masing pengukuran oleh pengamat didapatkan akurasi dengan membagi jumlah foto yang memiliki hasil yang sama antara pengukuran aplikasi dan pengukuran manual dengan jumlah foto yang diuji. Dari akurasi tersebut dirata-rata dan didapatkan hasil akurasinya adalah 72%. Dengan nilai akurasi tersebut, dapat dikatakan aplikasi dapat mengimplementasikan metode *Image Retrieval Based on Color Histogram* dengan baik, di mana akurasi hasil pengujian metode tersebut pada *paper* sebesar 66% [6]. Hasil akurasi pengukuran aplikasi dibandingkan dengan pengukuran secara manual ditunjukkan oleh Tabel 3.

## VI. KESIMPULAN & SARAN

Berdasarkan proses perancangan, implementasi dan pengujian yang dilakukan, maka kesimpulan-kesimpulan yang dapat diambil adalah:

1. Pengukuran tingkat kepekatan asap berdasarkan Ringelmann Smoke Chart dapat dilakukan dengan mudah menggunakan aplikasi ini. Aplikasi dapat mengimplementasikan metode pengukuran *Image Retrieval Based on Color Histogram* dengan akurasi sebesar 72%.
2. Penyimpanan data pengukuran tingkat kepekatan asap

dapat dilakukan dengan mudah menggunakan aplikasi ini. Selain fitur penyimpanan, aplikasi juga dapat mengimplementasikan fitur riwayat pengukuran dan benchmark nilai kepekatan asap.

Saran untuk pengembangan dan perbaikan sistem di masa yang akan datang sesuai hasil perancangan, implementasi dan uji coba yang telah dikerjakan adalah aplikasi dapat dijalankan pada platform lain selain Windows Phone.

#### DAFTAR PUSTAKA

- [1] BAPEDAL, "Kep-205/BAPEDAL/07/1996 tentang Pedoman Teknis Pengendalian Pencemaran Udara," 1996.
- [2] T. Kurniawan, "Pengukuran kepekatan asap dengan kartu Ringelmann," [Online]. Available: <http://infokelistrikan.blogspot.com/>. [Accessed 4 Maret 2014].
- [3] Microsoft, "Download Windows Phone SDK 8.0 from Official Microsoft Download Center," 30 October 2012. [Online]. Available: Available: <http://www.microsoft.com/en-us/download/details.aspx?id=35471>. [Accessed 4 Maret 2014].
- [4] Microsoft, "What Is Windows Communication Foundation," [Online]. Available: [http://msdn.microsoft.com/en-us/library/ms731082\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/ms731082(v=vs.110).aspx). [Accessed 4 Maret 2014].
- [5] Techopedia, "Internet Information Services (IIS)," [Online]. Available: <http://www.techopedia.com/definition/24953/internet-information-services-iis>. [Accessed 5 Maret 2014].
- [6] R. Chakravarti and X. Meng, "A Study of Color Histogram Based Image Retrieval," *2009 Sixth International Conference on Information Technology: New Generations*, 2009.