

BAB II

LANDASAN TEORI

2.1. Web Camera

Webcam adalah sebuah perangkat yang dapat digunakan untuk mengambil video ataupun gambar biasa dan terkoneksi melalui *USB port*, LAN atau Wi-Fi kepada sebuah komputer, *webcam* dikenal karena fleksibilitasnya dan biaya pembuatannya yang rendah. *Webcam* umumnya digunakan agar komputer dapat digunakan sebagai *input* untuk *video conference*, *webcam* juga dapat diakses dari berbagai macam aplikasi dan perangkat (Surhone, 2010).

2.2. AT Commands

AT Commands merupakan *messages* yang digunakan untuk komunikasi antara aplikasi dengan piranti keras Wavecom untuk menangani *events* atau *services* yang berkaitan dengan GSM (Wavecom, 2004). Sehingga, dengan mengirimkan *AT Commands*, sebuah aplikasi dapat mengendalikan piranti keras SMS Gateway Wavecom termasuk untuk mengirimkan SMS. Isi atau data di dalam SMS juga ditentukan oleh aplikasi, setiap *AT Commands* yang dikirimkan akan menghasilkan pesan balik berupa status eksekusi perintah (Wavecom, 2004). Beberapa contoh *AT Commands* antara lain :

- a. *AT+CSQ* : Untuk mengetahui kuat signal dari kartu SIM yang sedang aktif.
- b. *AT+CMGS* : Untuk mengirimkan SMS ke nomor tertentu.
- c. *AT+CMGS* : Untuk melihat SMS yang tersimpan di dalam kartu SIM.



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

2.3. JPEG Format

Format gambar digital umumnya menggunakan teknik kompresi, terdapat dua buah teknik kompresi data yakni *lossless* dan *lossy*. Teknik kompresi *lossless* biasa digunakan untuk data *discrete* (harus dapat dikonstruksi ulang secara utuh) contohnya teks, sedangkan *lossy* merupakan teknik kompresi yang melibatkan hilangnya informasi, data yang dikompres menggunakan teknik ini tidak dapat dikonstruksi ulang secara utuh (Shukla, 2011).

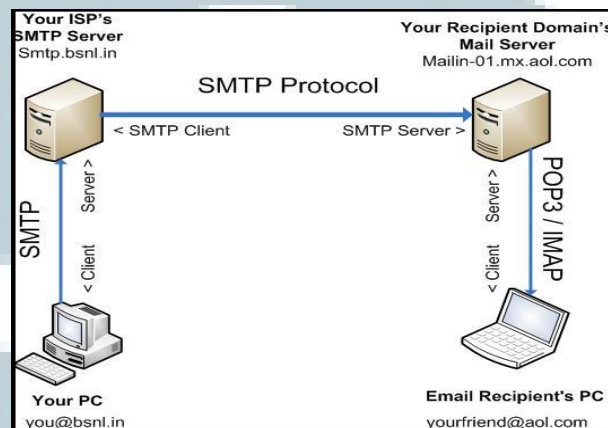
JPEG (*Joint Photographic Experts Group*) adalah format penyimpanan gambar yang menggunakan teknik kompresi *lossy*. JPEG umum digunakan untuk menyimpan gambar *photographic*. JPEG memiliki keunggulan dibandingkan dengan format gambar lainnya dari segi kompresi ukuran *file* untuk gambar *photographic*. Namun, terlepas dari keunggulan tersebut JPEG tidak cocok untuk teks dan gambar yang bukan *photographic* karena sifatnya yang *lossy* (Miano, 1999).

2.4. Protokol E-mail

SMTP (*Simple Mail Transfer Protocol*) adalah sebuah protokol yang menggunakan jaringan yang berbasis TCP/IP untuk mengirimkan sebuah e-mail dari satu *mail server* ke *mail server* lainnya, SMTP beroperasi pada port 25 dan bertugas layaknya pembawa surat, apabila SMTP gagal mengirimkan sebuah e-mail maka SMTP tidak akan berusaha mengetahui mengapa pengirimannya gagal (Dean, 2012:501).

Setelah e-mail sampai ke *mail server* penerima, protokol POP/IMAP dapat digunakan. Kedua protokol ini berguna bagi penerima e-mail untuk melakukan

retrieve terhadap e-mail. Di dalam POP, e-mail biasanya dihapus dari *server* setelah *download* sedangkan IMAP memungkinkan pengguna untuk dapat menyimpan e-mailnya tanpa pada *server* tanpa perlu *download* dan menghapusnya dari *server* (Dean, 2012:501).



Gambar 2.1. Ilustrasi Protokol E-mail (<http://www.ustudy.in/node/22>)

2.5. Sistem Pewarnaan RGB

Computer graphics menggunakan *color spaces* yang berbeda-beda, monitor biasanya menggunakan RGB *color spaces* (sistem pewarnaan RGB) yang terdiri dari *red*, *green* dan *blue* (merah, hijau dan biru) sebagai tiga buah warna utama (Yuan, 2001:429). RGB merupakan sistem pewarnaan yang bersifat *additive*, kombinasi antara ketiga buah nilai pada RGB akan membentuk sebuah warna spesifik, RGB mudah untuk disimpan dan dimanipulasi sehingga umumnya juga digunakan dalam *computer graphics programming* (Yuan, 2001:430).

2.6. Sequence

Sequence atau yang biasa dikenal sebagai deret aritmatika adalah kumpulan dari angka yang disusun berdasarkan urutan tertentu, didalam kumpulan angka

tersebut terdapat sebuah *pattern* yang menentukan aturan perpindahan dalam suatu tempat pada baris angka ke tempat selanjutnya, *pattern* ini tentunya dapat diformulasikan kedalam rumus matematika yang akan menentukan aturan dari sebuah *sequence*. (Hirst, 1995).

Formula (1) adalah contoh dari rumus *sequence*, rumus *sequence* memiliki tiga buah komponen, a_n merepresentasikan nilai pada urutan ke n , d adalah nilai perbedaan (*difference*), n adalah jumlah urutan (Mathworlds).

$$a_n = a_1 + (n - 1)d \quad (1)$$

2.7. Algoritma *Frame Differences*

Frame differencing adalah sebuah teknik dimana komputer mengecek perbedaan dari dua buah *video frames*, untuk mencapai hal tersebut, algoritma *frame differencing (frame differences)* digunakan (Haritaoglu *et al.*, 2000). Algoritma ini menggunakan nilai RGB pada gambar, apabila perbedaan nilai rata-rata RGB sebuah piksel pada *frame input* dan *frame* pembanding untuk posisi yang sama melampaui batas nilai tertentu, maka *frames* dianggap berbeda, batas nilai ini dikenal juga sebagai nilai *threshold* (Widyawan *et al.*, 2012).

Secara keseluruhan, terdapat empat buah proses utama di dalam algoritma ini, yakni : *RGB calculation*, *frames comparison* dan *motion detection* (Widyawan *et al.*, 2012). Selain itu terdapat juga metode *template matching* yang diperlukan oleh algoritma di dalam menentukan *frames* pembanding. Algoritma *frame differences* yang digunakan dalam penelitian ini juga menggunakan nilai *density*. Metode *template matching*, empat buah proses utama algoritma dan *density* akan dijelaskan pada sub-subbab selanjutnya.

2.7.1. *Template Matching*

Frame differences adalah algoritma yang melakukan analisa terhadap piksel-piksel yang berada di dalam sebuah gambar. *Frame differences* bekerja dengan membandingkan piksel-piksel pada sebuah *frame* yang diambil dan *frame* pembanding. Metode yang digunakan untuk menetapkan *frame* pembanding adalah *Template Matching* (Widyawa *et al.*, 2012). Sehingga, *template matching* merupakan salah satu tahap yang berpengaruh dalam penentuan adanya deteksi gerakan atau tidak. Terdapat tiga buah *template matching*, yakni *static template matching*, *dynamic template matching* dan *dynamic and adaptive template matching* (Widyawan *et al.*, 2012). Berikut adalah penjelasan untuk tiap metode *template matching* :

1. *Static Template Matching*

Static template matching merupakan metode *template matching* yang biasanya digunakan untuk algoritma *background subtraction*, *static template matching* menggunakan *frame* pembanding yang bersifat statis (tidak berubah) (Widyawan *et al.*, 2012). Cara termudah untuk menetapkan *background image* (*static frame*) adalah dengan memperoleh gambar yang tidak memuat objek bergerak sama sekali (Bouwsman *et al.*, 2008). Contoh dari *static frame* adalah gambar dari sebuah ruangan pada saat ruangan tersebut kosong dan tidak memuat objek bergerak sama sekali.

2. *Dynamic Template Matching*

Dalam *dynamic template matching* (DTM) *background image* (*frame* pembanding) selalu berubah secara *dynamic* untuk mendapatkan gambar yang

lebih baik dari objek yang bergerak, penggunaan *templates* seperti ini membantu pendeteksian objek secara lebih kuat (Mittal, 2013). Berdasarkan hasil uji coba oleh Widyawan dan Muhammad Ihsan dalam jurnal yang berjudul *Adaptive Motion Detection Adaptive Motion Detection Algorithm using Frame Differencess and Dynamic Template Matching Method* pada tahun 2012, metode *dynamic template matching* untuk *frame* pembanding adalah *frame* pada waktu $t - 1$ menghasilkan akurasi sebesar 89.8% (Widyawan *et al.*, 2012).

3. *Dynamic and Adaptive Template Matching*

Dalam *dynamic and adaptive template matching*, *frame* pembanding yang digunakan adalah kombinasi dari *frame* pada waktu $t - 1$, $t - n$ (Widyawan *et al.*, 2012). Sehingga, *frame* pembanding tidak hanya bersifat *dynamic* tetapi juga *adaptive* dimana *frame* pembanding tidak harus mengikuti *pattern* tertentu. *Frame* pembanding hanya akan diambil jika skenario (area yang dipantau oleh *camera*) mengalami perubahan yang signifikan (Widyawan *et al.*, 2012). *Template matching* inilah yang digunakan di dalam penelitian ini, dimana setiap kali terdeteksi gerakan maka dianggap bahwa telah terjadi perubahan skenario yang cukup signifikan dan *frame* pembanding akan diperbarui. Berdasarkan hasil uji coba oleh Widyawan dan Muhammad Ihsan dalam jurnal yang berjudul *Adaptive Motion Detection Adaptive Motion Detection Algorithm using Frame Differencess and Dynamic Template Matching Method* pada tahun 2012, metode *dynamic and adaptive template matching* menghasilkan akurasi sebesar 95.5% (Widyawan *et al.*, 2012).

2.7.2. RBG calculation

Perhitungan ini berguna untuk mendapatkan nilai rata-rata RGB pada tiap piksel yang perlu dibandingkan, baik pada *frame* pembanding maupun *frame* yang akan dibandingkan. Formula (2) dan (3) menunjukkan cara perhitungan rata-rata RGB di dalam algoritma *frame difference* (Widyawan *et al.*, 2012). $f_0(x,y)$ merepresentasikan nilai rata-rata RGB untuk *pixel* pada posisi x,y pada *frame input*, sedangkan $g_0(x,y)$ merepresentasikan nilai rata-rata RGB untuk *pixel* pada posisi x,y pada *frame* pembanding. Nilai rata-rata RGB didapat dengan menjumlahkan nilai komponen R, G dan B untuk *pixel* pada posisi x,y pada *frame* yang bersesuaian lalu dibagi dengan 3 (Widyawan *et al.*, 2012).

$$g_0(x,y) = \frac{g_R(x,y) + g_G(x,y) + g_B(x,y)}{3} \quad (2)$$

$$f_0(x,y) = \frac{f_R(x,y) + f_G(x,y) + f_B(x,y)}{3} \quad (3)$$

2.7.3. Frames Comparison

Selain nilai rata-rata RGB, untuk dapat melakukan perbandingan piksel-piksel pada *frames*, digunakan juga nilai *threshold*. Formula (4) menunjukkan perbandingan untuk menentukan apakah sebuah *pixel* terdeteksi sebagai bagian dari objek bergerak atau tidak, nilai T berfungsi sebagai nilai *threshold* (Widyawan *et al.*, 2012). $g_0(x,y)$ merepresentasikan nilai rata-rata RGB untuk *pixel* pada posisi x,y pada *frame* pembanding, T merepresentasikan nilai *threshold* dan $f_0(x,y)$ merepresentasikan nilai rata-rata RGB untuk *pixel* pada posisi x,y pada *frame input*. Apabila kondisi dibawah terpenuhi maka *pixel* tidak

terdeteksi, nilai perbedaan RGB antara *pixel* harus melewati *threshold* agar *pixel* ditandai sebagai terdeteksi.

$$(g_0(x, y) - T) \leq f_0(x, y) \leq (g_0(x, y) + T) \quad (4)$$

2.7.4. Motion Detection

Dalam algoritma *frame differences*, *motion detection* ditentukan berdasarkan persentase dari jumlah *pixel* yang terdeteksi, apabila memenuhi nilai persentase tertentu maka *motion* dianggap terjadi (Widyawan *et al.*, 2012). Oleh karena penelitian ini menggunakan algoritma *frame differences* yang menggunakan nilai *density*, satu buah *pixel* yang terdeteksi saja telah menentukan adanya *motion*.

2.7.5. Density

Pada penelitian ini, algoritma *frame differences* dimodifikasi dan menggunakan nilai *density* (kerapatan) agar penghitungan dan perbandingan nilai rata-rata RGB tidak dilakukan kepada seluruh *pixel*, melainkan hanya kepada *pixels* tertentu mengikuti sebuah pola atau *pattern*. *Density* adalah nilai persentase kerapatan *pixel* yang dicek, nilai *density* berguna untuk menentukan nilai *d* dalam rumus *series*, yaitu nilai *difference* atau jumlah *pixel* yang perlu dilewati pada saat perpindahan pengecekan *pixel* satu ke lainnya. Penggunaan nilai *density* didasarkan kepada konsep *sequence* atau deret aritmatika.

Sequence adalah sebuah *pattern* yang menentukan aturan perpindahan suatu tempat pada baris angka ke tempat selanjutnya, *pattern* ini tentunya dapat diformulasikan kedalam rumus matematika yang akan menentukan aturan dari sebuah *sequence*. (Hirst, 1995). Rumus *sequence* memiliki tiga buah komponen,

a_n merepresentasikan nilai pada urutan ke n , d adalah nilai perbedaan (*difference*), n adalah jumlah urutan (Mathworlds). Rumus *sequence* yang digunakan dalam penentuan posisi *pixel* yang dicek ditunjukkan pada formula (5).

$$a_n = (d * n) - (d - 1) \quad (5)$$

Oleh karena posisi piksel bersifat dua dimensi, formula (5) berlaku untuk perhitungan kolom dan baris sebagai posisi *pixel* yang perlu dicek. Pada implementasinya, hal ini dicapai dengan menggunakan perulangan. Sehingga, formula (5) digunakan untuk menentukan posisi x, y (kolom, baris) dari *pixel*. Formula (6) dan (7) menunjukkan penggunaan formula (5) untuk penghitungan kolom dan baris dari *pixel* yang perlu dicek. Formula (6) dan (7) juga mengikuti ketentuan dalam mendefinisikan rumus *sequence* dimana terdapat tiga buah komponen, yakni a_n , d dan n (Mathworlds).

$$a_{nkolom} = (dkolom * nkolom) - (dkolom - 1) \quad (6)$$

$$a_{nbaris} = (dbaris * nbaris) - (dbaris - 1) \quad (7)$$

a_{nkolom} merepresentasikan nilai kolom pada urutan ke $nkolom$, $dkolom$ adalah nilai perbedaan (*difference*) yang didapat dengan mengalikan nilai *density* dengan lebar gambar dan $nkolom$ adalah jumlah urutan dalam perhitungan kolom. Sedangkan, a_{nbaris} merepresentasikan nilai baris pada urutan ke $nbaris$, $dbaris$ adalah nilai perbedaan (*difference*) yang didapat dengan mengalikan nilai *density* dengan tinggi gambar dan $nbaris$ adalah jumlah urutan dalam perhitungan baris.

Sebagai contoh, sebuah gambar memiliki ukuran 250x250 dan nilai *density* yang digunakan adalah 2%, sehingga nilai $dkolom$ dan $dbaris$ adalah 5.

Perulangan dimulai dari orienasi baris lalu kolom. Sehingga, formula (7) digunakan dimulai dari nilai $nbaris$ adalah 1. Maka dari itu baris pertama yang dicek adalah baris kesatu. Selanjutnya di dalam baris kesatu ini, dilakukan juga perhitungan untuk posisi kolom selama nilai kolom belum melebihi lebar gambar (dalam satuan piksel). Formula (6) lalu digunakan dimulai dari $nkolom$ adalah 1. Sehingga, piksel pertama yang akan dicek adalah pada posisi (1, 1) selanjutnya nilai $nkolom$ akan bertambah 1 di setiap perulangan sehingga posisi piksel selanjutnya yang dicek antara lain (1, 6), (1, 11) dan seterusnya. Apabila perhitungan kolom untuk sebuah baris telah selesai, maka perhitungan baris akan dilanjutkan dengan nilai $nbaris$ yang telah bertambah satu, perhitungan kolom untuk baris baru ini juga akan dilakukan kembali dari nilai $nkolom$ adalah 1, sehingga piksel-piksel yang kemudian dicek adalah piksel pada posisi (6, 1), (6, 6), (6, 11) dan seterusnya. Perhitungan posisi baris dilakukan selama nilai baris belum melebihi tinggi gambar.

2.8. AVICAP Library

Avicap merupakan *library* yang disediakan oleh Microsoft untuk pemrosesan audio dan video *streams*. *Library* ini disertakan di dalam Sistem Operasi Windows. AVIcap mendukung *streaming video capture* dan *single-frame capture*, selain itu AVIcap juga memungkinkan pengguna untuk mengendalikan *video sources* termasuk *Media Control Interface (MCI) devices* sehingga user dapat mengontrol *devices* melalui aplikasi dan melakukan dari proses *capture* dari *video sources* atau *frame captures* (Microsoft Developer Network).