SISTEM PERINGATAN DINI UNTUK RUMAH MENGGUNAKAN *MOTION DETECTION* MELALUI *MULTIMEDIA MESSAGING SERVICE* (MMS)

SKRIPSI

Disusun oleh :

YOSIA EKA SATYA SURYA SIMANJUNTAK
06. 12. 669

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG

2011

LEMBAR PERSETUJUAN

SISTEM PERINGATAN DINI UNTUK RUMAH MENGGUNAKAN *MOTION DETECTION* MELALUI *MULTIMEDIA MESSAGING SERVICE* (MMS)

Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Teknik

Disusun oleh :

YOSIA EKA SATYA SURYA SIMANJUNTAK

NIM : 06.12.669

Diperiksa dan Disetujui

Dosen Pembimbing	Mengetahui, Ketua Jurusan Teknik Elektro S-1
Dr. Eng. Aryuanto Soetedjo,ST.,MT	**Ir. Yusuf Ismail Nakhoda,MT**
NIP.Y.1030800417	**NIP.Y.1018800189**

JURUSAN TEKNIK ELEKTRO S-1

KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI NASIONAL MALANG

2011

SISTEM PERINGATAN DINI UNTUK RUMAH DENGAN MENGGUNAKAN *MOTION DETECTION* MELALUI *MULTIMEDIA MESSAGING SERVICE* (MMS)

Yosia Eka Satya Surya Simanjuntak

Jurusan Teknik Elektro S-1, Konsentrasi T.Komputer dan Informatika
Fakultas Teknologi Industri, Institut Teknologi Nasional Malang
Jln. Raya Karanglo Km 2 Malang
yoziedarwiz@yahoo.co.id

Dosen Pembimbing : I. Dr. Eng. Aryuanto Soetedjo,ST., MT.

Abstraksi

Handphone telah berubah dari alat telekomunikasi biasa menjadi alat serbaguna yang mempunyai berbagai fasilitas. Selain untuk berkomunikasi handphone juga dapat di gunakan sebagai kamera. dan alat pemantau. Sebagai alat pemantau handphone memiliki keunggulan yaitu mudah dibawa kemana, dapat menghubungi orang lain jika diperlukan. Pada pembuatan tugas akhir ini akan dikembangkan sistem peringatan dini untuk rumah dengan menggunakan *motion detection* melalui *Multimedia Messaging Service* yang memungkinkan user untuk memantau keadaan rumah dari jarak jauh. Sistem alarm keamanan rumah akan mengirimkan *MMS* kepada user secara otomatis jika terdeteksi adanya suatu gerakan. Sistem alarm keamanan rumah dibuat dengan menggunakan program Matlab. Berdasarkan hasil pengujian sistem, maka dapat disimpulkan bahwa sistem alarm keamanan rumah dengan menggunakan motion detection melalui *multimedia messaging service (MMS)* cocok untuk diimplementasikan di Indonesia. Hal ini disebabkan karena jaringan infrastruktur layanan *MMS* sesama operator di Indonesia sudah bekerja dengan baik. Hal ini dapat dilihat dari pengujian sistem yang membuktikan bahwa tingkat keberhasilan dari pengiriman *MMS* sesama operator yang cukup tinggi. Sehingga sistem ini cukup dapat diandalkan dalam bidang keamanan rumah.

Kata Kunci: Motion detection, MMS, GPRS

KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa yang telah memberikan rahmat dan hidayah-Nya, sehingga dapat diselesaikan skripsi yang berjudul

"SISTEM PERINGATAN DINI UNTUK RUMAH DENGAN MENGGUNAKAN *MOTION DETECTION* MELALUI *MULTIMEDIA MESSAGING SERVICE* (MMS)" ini dengan lancar. Skripsi ini merupakan persyaratan kelulusan Studi pada Jurusan Teknik Elektro S-1 Konsentrasi Teknik Komputer dan Informatika ITN Malang dan untuk mencapai gelar Sarjana Teknik.

Keberhasilan peyelesaian laporan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu penyusun menyampaikan terima kasih kepada :

1. Bapak Ir. Soeparno Djiwo, MT, selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Sidik Noertjahjono, MT, selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. Bapak Ir. Yusuf Ismail Nakhoda, MT selaku Ketua Jurusan Teknik Elektro S-1 Institut Teknologi Nasional Malang.
4. Bapak Dr. Eng. Aryuanto Soetedjo,ST., MT. selaku Sekretaris Jurusan Teknik Elektro S-1 Institut Teknologi Nasional Malang dan selaku Dosen Pembimbing I
5. Ayah dan Ibu, saudara-saudara, sahabat-sahabat yang telah memberikan do'a restu, dorongan, semangat, dan biaya.
6. Rekan-rekan di ITN Malang.
7. Semua yang telah membantu dalam penyelesaian penyusunan skripsi ini.

Penulis telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini. Untuk itu penyusun mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini.

Harapan penyusun semoga laporan skripsi ini memberikan manfaat bagi perkembangan ilmu pengetahuan dan pembaca.

Malang, Agustus 2011

penyusun

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1. Latar Belakang

Pada saat ini perkembangan teknologi informasi (IT) sangatlah berkembang pesat. Banyak alat-alat yang dapat membuat perkerjaan manusia menjadi lebih ringan. Temasuk di dalamnya adalah alat-alat telekomunikasi *mobile*. Peralatan telekomunikasi *mobile* pada saat ini sangatlah dibutuhkan. Karena dengan adanya alat-alat telekomunikasi *mobile* tersebut. maka pengguna dapat dengan cepat mendapatkan informasi yang cepat, tepat dan akurat.

Salah satu peralatan telekomunikasi yang paling popular dan sering digunakan saat ini adalah *handphone*. Saat ini hampir semua orang memiliki *handphone*. Dari kalangan atas sampai kalangan menengah ke bawah juga dapat menggunakan peralatan telekomunikasi tersebut. Hal ini disebabkan oleh fasilitas-fasilitas yang diberikan oleh *handphone* tersebut. *Handphone* bukan hanya sebagai peralatan yang khusus untu berkomunikasi saja, namun telah menjadi sebuah alat yang serba guna yang memiliki berbagai fitur-fltur tambahan dari handphone, seperti : *koneksi dengan internet, games, memutar lagu, memutar video*, bahkan pengguna dapat melakukan *fax* langsung dari *handphone*. Pada umumnya *handphone* yang dijual sekarang telah memiliki fitur-fitur yang hampir sama. Antara lain : memiliki layar yang telah berwarna, memiliki nada dering yang polyphonic serta telah mendukung layanan *Multimedia Messaging Service* (MMS) *dan General Packet Radio Service* (GPRS). Dangan adanya fasilitas MMS tersebut, pengguna akan mendapatkan informasi yang lebih lengkap dan akurat. Karena dengan fasililas MMS ini pengguna dapat melakukan pengiriman maupun menerima data dalam bentuk gambar, suara dan teks sekaligus.

Selain teknologi informasi yang berkembang pesat, perlu juga memperhatikan sistem keamanan, mengingat banyaknya tindak kriminal yang terjadi. *Close Circuit Television* (CCTV) adalah sebuah sistem yang dapat digunakan untuk membantu menjaga keamanan dari sebuah pabrik, gedung, kantor maupun rumah pribadi. Dengan adanya sistem pemantau ini. kita dapat mengetahui setiap kejadian yang terjadi disebuah tempat hanya dengan melihat di dalam layar monitor. Namun peralatan dan sistem ini memiliki beberapa kelemahan. Selain peralatan yang sangat mahal, pengguna harus terus mengamati layar monitor. Hal ini tentunya kurang praktis

digunakan didalam rumah pribadi yang tidak memiliki satpam. Karena manusia yang mcmiliki banyak kesibukan yang lain. Untuk itu dibutuhkan alat yang dapat memonitor keadaan, dan dapat dibawa kemana-mana. Sehingga pengguna tidak perlu menyediakan waktu khusus untuk memantau. pengguna dapat melakukan aktivitasnya. Alat yang cocok dengan keadaan tersebut adalah *handphone.* Karena handphone telah memiliki fasilitas-fasilitas yang cukup lengkap sehingga dapat menyediakan informasi-informasi yang cepat, tepat dan akurat yang dibutuhkan oleh pengguna.

1.2. Rumusan Masalah

Untuk membuat sistem pemantau keadaan ada beberapa hal yang perlu diperhatikan :

- Menentukan pendeteksian gerak dengan adanya perubahan pixel dari gambar yang ditangkap. Serta menentukan sensitivitas yang tepat, Sehingga jika terjadi pergerakan, program akan mcndeteksinya.
- Untuk transfer data dari komputer ke handphone dapat digunakan *kabel data, infrared, Bluetooth.* Namun jarak untuk dapat melakukan pentransferan data tersebut sangatlah pendek. Hanya sekitar 1—10 meter saja. Untuk mengatasi masalah tersebut digunakan fasilitas *Multimedia Messaging Service (MMS)* untuk mentransfer data ke *handphone* yang diinginkan walaupun letaknya jauh.
- Pengiriman *MMS (Multimedia Message Service)* dari *komputer* ke *Handphone* yang dituju Diperlukan koneksi *GPRS* ke *MMSC (multimedia messaging service center).* Kualitas koneksi *GPRS* tergantung dari layanan operator yang digunakan. Biaya koneksi *GPRS* tergantung dari operator yang digunakan.

1.3. Tujuan

Tujuan dari pembuatan tugas akhir ini adalah untuk mengaplikasikan bidang telematika ke dalam pembuatan alat *motion detection* dengan menggunakan *webcam* yang dihubungkan dengan *komputer.* Sehingga user dapat mengetahui keberadaan seseorang dalam ruangan yang telah disetting sebelumnya. Dengan adanya alat ini diharapkan dapat membantu beberapa pihak untuk dapat meningkatkan sistem keamanannya.

1.4. Batasan Masalah

Berdasarkan rumusan masalah yang telah diuraikan sebelumnya, maka pada pelaksanaan skripsi ini dibatasi pada:

- Pengambilan *Motion detection* (deteksi gerak) yang dihasilkan oleh webcam (kamera web) yang ditempatkan pada posisi tetap
- Pengiriman deteksi gambar menggunakan handphone melalui MMS.
- *Motion detection* (deteksi gerak) yang diimplementasikan tidak mengenali jenis obyek yang bergerak didalam citra yang dideteksi.
- Jarak pandang pemantauan ruangan terbatas berdasarkan pada jenis *webcam* yang digunakan.

1.5. Metodologi

Metodologi yang digunakan dalam pelaksanaan skripsi ini adalah sebagai berikut :

1. Studi literatur, dengan cara mempelajari berbagai macam referensi, baik tulisan ilmiah, paper, artikel maupun textbook mengenai image-processing (pengolahan citra digital) serta metode-metode yang akan dipakai dalam motion detection (deteksi gerak) dan mengirimnya dengan MMS .
2. Analisis masalah untuk menentukan masalah yang dihadapi dan solusinya.
3. Analisis dan perancangan perangkat lunak untuk mendapatkan gambaran perangkat lunak yang akan dibangun dan spesifikasinya.
4. Implementasi perangkat lunak sesuai dengan hasil rancangan yang telah dibuat.
5. Pengujian dan analisis hasil uji perangkat lunak yang telah dibangun.
6. Perbaikan untuk memperbaiki kesalahan-kesalahan yang mungkin terjadi pada program, laporan, dan dokumentasi teknik.

1.6. Sistematika Penulisan

Penyusunan laporan skripsi ini menggunakan kerangka pembahasan yang terbentuk dalam susunan bab, yang dapat dijelaskan sebagai berikut :

BAB I PENDAHULUAN

Bab ini merupakan dasar penyusunan laporan skripsi yang di dalamnya berisi tentang latar belakang masalah, rumusan masalah, tujuan skripsi, batasan masalah, metodologi pengembangan sistem, dan sistematika pembahasan skripsi.

BAB II DASAR TEORI

Bab ini berisi tentang permasalahan yang berhubungan dengan penelitian yang dilakukan pada skripsi ini.

BAB III ANALISA DAN PERANCANGAN SISTEM

Bab ini berisi tentang tahap analisis yaitu identifikasi dan analisis masalah dan analisis kebutuhan sistem untuk menyelesaikan masalah yang dihadapi berdasarkan teori pada Bab II. Bab ini juga berisi hasil perancangan program yaitu proses kelanjutan dari tahap analisis.

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Dari perancangan sistem maka akan dilakukan pengujian dari pembuatan aplikasi yang berdasarkan pada hasil dari analisis serta perancangan sistem pada Bab III. Dari program yang telah dibuat maka akan dilakukan pengujian dan analisa untuk mendapatkan kesimpulan dari kerja sistem.

BAB VI PENUTUP

Bab ini merupakan bab terakhir dari laporan skripsi ini yang berisi tentang kesimpulan dari hasil pembahasan pada perancangan dan kesimpulan pengujian akhir sistem yang dibuat serta saran-saran untuk penyempurnaan dalam pengembangannya.

BAB II
LANDASAN TEORI

2.1. Pengolahan Citra *Digital*

Image processing adalah suatu metode yang digunakan untuk mengolah gambar sehingga menghasilkan gambar lain yang sesuai dengan keinginan kita[1]. Pengambilan gambar biasanya dilakukan dengan kamera *video digital* atau alat lain yang biasanya digunakan untuk mentrasnfer gambar (*scanner*, kamera *digital*).

Pengolahan gambar *digital* atau *Digital Image Processing* (DIP) adalah bidang yang berkembang sangat pesat sejalan dengan kemajuan teknologi pada industri saat ini. Fungsi utama dari *Digital Image Processing* adalah untuk memperbaiki kualitas dari gambar hingga dapat dilihat lebih jelas tanpa ada ketegangan pada mata, karena informasi penting diekstrak dari gambar yang dihasilkan harus jelas sehingga didapatkan gambar yang terbaik. Selain itu DIP digunakan untuk memproses data yang diperoleh dalam persepsi mesin, yaitu prosedur-prosedur yang digunakan untuk mengekstraksi informasi dari gambar, informasi dalam bentuk yang cocok untuk proses komputer.

Keuntungan menggunakan DIP adalah presisi, yaitu pada masing-masing proses fotografi, disini terdapat penurunan kualitas gambar dan sinyal elektrik yang terdegradasi akibat keterbatasan komponen elektrik, dalam kondisi ini DIP dapat menjaga hasil gambar tetap presisi. Keuntungan yang lain adalah fleksibilitas, yaitu penggunaan yang lebih besar, sebuah gambar dapat di *magnified*, *reduced*, atau *rotated*, *kontras*, *brightness* dapat diubah.

Selain keuntungan DIP juga memiliki kekurangan yaitu kecepatan dan mahal, banyak operasi yang digunakan oleh DIP lebih lambat dan lebih mahal dibandingkan operasi optik atau elektrikal lainnya dan *resources* untuk menghitung bisa mahal.

Tujuan dari pengolahan citra adalah memperbaiki informasi pada gambar sehingga mudah terbaca atau memerpbaiki kualitas dari gambar itu sendiri.

Pengolahan citra mempunyai dua tujuan utama, yaitu sebagai berikut :

1. Memperbaiki kualitas citra, dimana citra yang dihasilkan dapat menampilkan informasi secara jelas. Hal ini berarti manusia sebagai pengolah informasi.
2. Mengekstraksi informasi yang menonjol pada suatu citra, dimana hasilnya adalah informasi citra dimana manusia mendapatkan informasi ciri dari citra secara numerik atau komputer melakun interpretasi terhadap informasi yang ada pada citra melalui besar-besaran data yang dapat dibedakan secara jelas.

2.1.1. Citra *Digital*

Citra atau *image* adalah angka (*image is just a number*), dari segi estetika, citra atau gambar adalah kumpulan warna yang bisa terlihat indah, memiliki pola, berbentuk abstrak dan lain sebagainya[2]. Citra dapat berupa foto udara, penampang lintang (*cross section*), dari suatu benda, gambar wajah, hasil *tomografi* otak dan lain sebagainya.

Dari segi ilmiah, citra adalah gambar 3-dimensi (3D) dari suatu fungsi, biasanya intensitas warna sebagai fungsi *spatial* x dan y. Di komputer, warna dapat dinyatakan, misalnya sebagai angka dalam bentuk skala RGB. Karena citra adalah angka, maka citra dapat diproses secara *digital.*

Image adalah sebuah gambar, foto yang ditampilkan atau bentuk lain yang memberikan *representasi visual* tentang sebuah obyek atau pemandangan. Pada DIP sebuah gambar bilangan *array* 2 dimensi, yang setiap barisnya adalah *representasi pixel* pada gambar setiap barisnya. Ukuran gambar biasanya 256 x 256, 512 x 512, 1024 x 1024. Minimum nilai *pixel* = 0 (hitam), maksimum = 255 (putih) dan bilangan antara 0 s.d. 255 *merepresentasikan* derajat keabuan. Gambar berwarna dapat direpresentasikan dengan array 2D *Red, Green* dan *Blue* atau 3D. Komputer membutuhkan *memory* lebih banyak untuk data ini rata-rata 3 kali data *storage.*

2.1.2. Model Citra

Oleh karena citra merupakan matrik dua dimensi dari fungsi intensitas cahaya, maka referensi citra menggunakan dua variabel yang menunjuk posisi pada bidang dengan sebuah fungsi intensitas cahaya yang dapat dituliskan sebagai f(x,y) dimana f adalah nilai amplitudo pada koordinat spasial (x,y)[2]. Karena cahaya merupakan salah

satu bentuk energi, f(x,y) tidak berharga nol atau negatif dan merupakan bilangan berhingga, yang dalam pernyataan matematis adalah sebagai berikut, 0<f(x.y).

2.1.3. Macam-Macam Citra

Gambar tidak berwarna atau hitam putih dikenal juga sebagai gambar dengan derajat keabuan (citra *gray level*)[2]. Derajat keabuan yang dimiliki ini bisa beragam mulai dari 2 derajat keabuan (yaitu 0 dan 1) yang dikenal juga sebagai citra *monochrome*, 16 derajat keabuan dan 256 derajat keabuan. Semakin besar jumlah derajat keabuan yang dimiliki maka semakin halus cira tersebut.

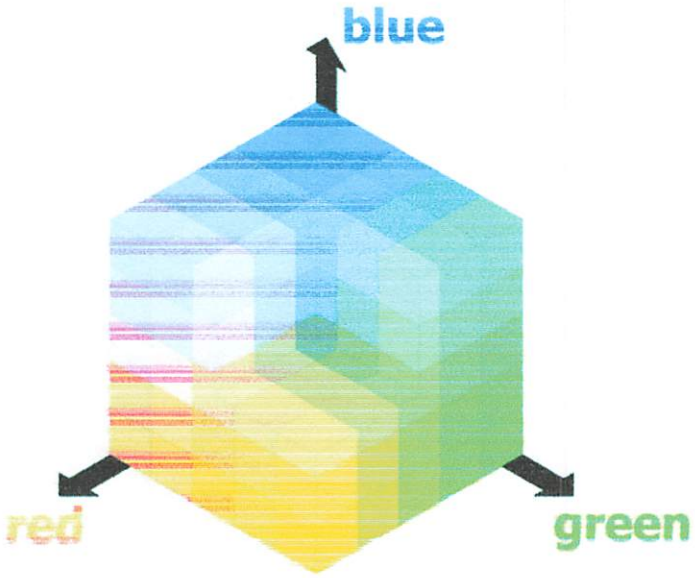
Dalam sebuah citra *monochrome*, sebuah pixel diwakili oleh 1 bit data yang berisikan data tentang derajat keabuan yang dimiliki pixel tersebut. Data akan berisi 1 bila pixel tersebut berwarna putih dan data akan berisi nilai 0 bila pixel tersebut berwarna hitam.

Citra yang memiliki 16 derajat keabuan (mulai dari 0 yang mewakili warna hitam sampai dengan 15 yang mewakili warna putih) direpresentasikan oleh 4 bit data, sedangkan citra dengan 256 derajat keabuan (nilai dari 0 yang mewakili warna hitam sampai dengan 255 yang mewakili warna putih) direpresentasikan oleh 8 bit data.

Dalam citra berwarna, jumlah warna bisa beragam mulai dari 16, 256, 65536 atau 16 juta warna, yang masing-masing direpresentasikan oleh 4,8,16 atau 24 bit data untuk setiap pixelnya. Warna yang ada terdiri dari tiga komponen utama yaitu nilai merah (*red*), nilai hijau (*green*) dan nilai biru (*blue*). Paduan ketiga komponen utama pembentuk warna ini dikenal sebagai RGB *color*.

2.1.4. Warna

Dasar dari pengolahan adalah pengolahan warna RGB pada posisi tertentu. Ruang warna adalah dasar warna dari *display* monitor komputer. Garis sepanjang titik hitam (0,0,0) RGB hingga titik abu (1,1,1) RGB disebut dengan titik keabuan atau *grayscale*. Jika skala r,g,b kita beri harga antara 0 dan 1, maka semua definisi warna akan memiliki nilai yang berbeda-beda. Sebagai perumpamaan nilai tersebut berada dalam kubus seperti pada gambar 2.1.



Gambar 2.1. Kubus warna

Pada proses penjumlahan warna, warna yang dideskripsikan dengan RGB adalah pemetaan yang mengacu pada panjang gelombang dari RGB. Pemetaan tersebut menghasilkan warna sekitar 16 juta warna (256^3= 16.777.216) yaitu 256(8 bit) untuk masing-masing R,G,B.

2.1.5. Pixel

Salah satu komponen gambar yang menentukan resolusi dari gambar tersebut adalah pixel. Sebagai contoh suatu gambar memiliki spesifikasi resolusi sebesar 400 x 300 hal ini berarti gambar ini memiliki panjang pixel horisontalnya pixel gambar tersebut adalah 120.000.

Dalam pengolahan citra, pixel memegang peranan yang sangat penting. Pixel p pada koordinat (x,y) mempunyai 4 tetangga vertical dan horizontal, yang koordinatnya seperti pada persamaan (2.1) :

(x+1,y+1),(x-1,y),(x,y+1), dan (x, y-1) (2.1)

Kumpulan pixel diatas disebut 4-neighbours of p dapat dinyatakan sebagai N4(p), kecuali jika p(x,y) posisinya terletak digaris batas gambar, sehingga jumlah pixel tetangga tidak terdiri dari 4 tetangga, selain 4 tetangga diatas, p juga mempunyai 4 tetangga diagonal, yaitu seperti pada persamaan (2.2) :

(x+1,y+1), (x+1,y-1), (x-1,y+1), dan (x-1,y-1) (2.2)

Pixel merupakan suatu konsep yang sangat penting untuk menyatakan batas-batas dari suatu objek serta bagian-bagian daerah kecil dari suatu gambar. Sehingga pertimbangan apakah dua pixel dihubungkan atau tidak, dibutuhkan beberapa kriteria. Diantaranya apakah kedua pixel mempunyai prinsip kedekatan yang sesuai dengan konsep yang telah ditentukan.

2.1.6. Teknik Pengolahan Citra Digital

Teknik pengolahan citra dibedakan menjadi 3 tingkat pengolahan[4], yaitu :

1. Tahap 1 *Low Level Processing*, merupakan pengolahan cira, paling dasar dalam pengolahan citra, sebagai contohnya : pengurangan derau (*noise reduction*), perbaikan citra (*image enhancement*) dan resolusi citra (*image restoration*).
2. Tahap 2 *Mid Level Processing* pengolahan citra ini meliputi segmentasi pada citra, deteksi objek dan klasifikasi objek secara terpisah.
3. Tahap 3 *High Level Processing* merupakan tahap analisa citra.

   Dari ketiga tahap diatas, dapat dinyatakan suatu gambaran mengenai teknik-teknik pengolahan citra digital.

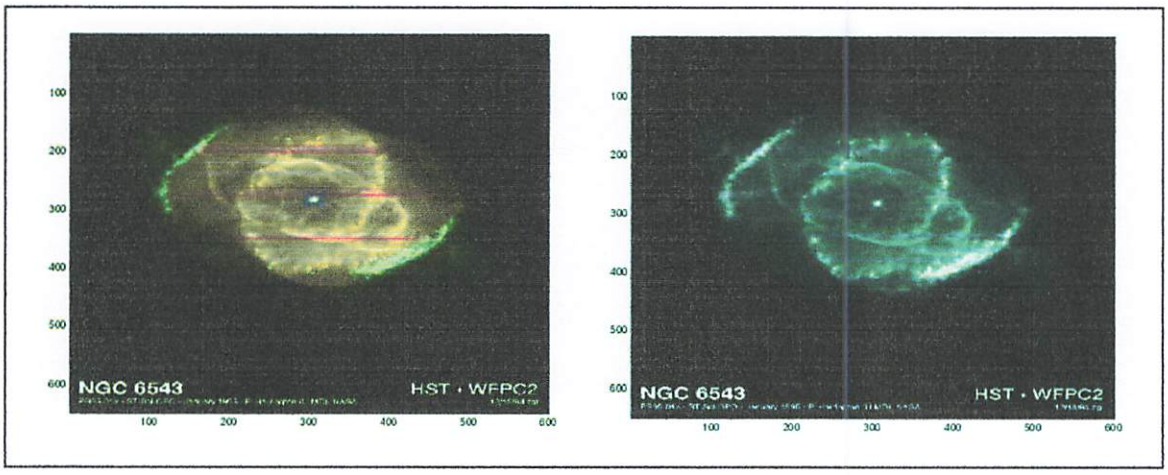
   1. *Image enhancement*, berupa proses perbaikan citra dengan meningkatkan kualitas citra baik kontras maupun *brightness*.
   2. *Image restoration*, proses memperbaiki model citra, biasannya berhubungan dengan bentuk citra yang sesuai.
   3. *Color image processing*, berhubungan dengan citra berwarna baik itu *image enhancement*, *image restoration* atau yang lain.
   4. *Wavelet* dan *Multiresolution Processing*, proses yang menyatakan citra dalam beberapa resolusi.
   5. *Image compression*, merupakan proses yang digunakan untuk mengubah ukuran data pada citra.
   6. *Morphological processing*, proses untuk memperoleh informasi yang menyatakan deskripsi dari suatu bentuk pada citra.
   7. *Segmentation*, proses untuk membedakan atau memisahkan objek-objek yang ada dalam suatu citra, seperti memisahkan objek dengan *background*nya.
   8. *Object recognition*, proses untuk mengetahui ada tidaknya objek dalam citra.

2.1.7. Dasar-dasar pengolahan citra digital

2.1.7.1. Mengubah Citra Berwarna menjadi Gray-Scale

Proses awal yang banyak dilakukan dalam *image processing* adalah mengubah citra berwarna menjadi citra *gray-scale*. Digunakan untuk menyederhanakan model citra. Citra berwarna terdiri dari 3 layer matrik yaitu R-layer, G-layer, dan B-layer. Untuk melakukan proses-proses selanjutnya menggunakan tiga layer, berarti dilakukan tiga perhitungan yang sama. Konsep itu disederhanakan, dengan mengubah 3 layer di atas menjadi 1 layer matrik *grayscale*. Hasilnya : citra *grayscale*. Dalam citra ini tidak ada lagi warna, yang ada adalah derajat keabuan. Mengubah citra berwarna yang mempunyai nilai matrik Ri,Gi, dan Bi menjadi citra grayscale dengan nilai Ko, maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai r,g dan b seperti gambar 2.2 :

$$Ko = \frac{Ri+Gi+Bi}{3}$$



Gambar 2.2. konversi gambar berwarna ke grayscale dengan matlab

2.1.7.2. Deteksi Gerakan

Gerakan adalah suatu pusat perhatian yang digunakan manusia ataupun hewan untuk mengenali suatu obyek dari suatu latar yang tidak teratur[7]. Dalam aplikasi pencitraan, gerakan muncul dari perpindahan tempat antara sistem pendeteksi dan lingkungan yang sedang dilihat, seperti aplikasi robotika, navigasi otomatis, dan analisis lingkungan dinamis.

Deteksi gerakan secara sederhana dapat dilakukan dengan mencari beda antara dua buah citra yang berurutan pada hasil pencitraan menggunakan kamera video digital. Operator yang digunakan adalah pengurangan. Operasi pengurangan pada bagian yang tidak bergerak dalam citra akan menghasilkan nilai nol atau mendekati nol, sedangkan bagian yang bergerak akan memberikan nilai yang tidak nol. Dengan mengevaluasi nilai selisih tersebut, dapat diketahui apakah terdapat objek yang bergerak.

2.1.7.3. Sum Of Absolute Difference

Sum of Absolute Difference (SAD) digunakan pada banyak hal, algoritma yang sangat sederhana untuk korelasi diantara blok citra[8]. Algoritma ini bekerja dengan mengambil nilai absolut dari perbedaan dari setiap pixel pada blok citra asli dengan pixel yang berkorespondensi pada blok yang sedang digunakan sebagai perbandingan. Perbedaan ini kemudian dijumlahkan untuk membuat persamaan matrix sederhana. Sum of Absolute Difference dapat digunakan untuk berbagai tujuan, seperti pengenalan objek, generasi dari peta perbedaan untuk citra stereo, dan perkiraan gerakan untuk kompresi video. Algoritma ini berdasarkan pada teknik perbedaan citra. Secara matematis direpresentasikan dengan persamaan (2.4) :

$$D(t) = \frac{1}{N}\sum\left|I(t_i) - I(t_j)\right| \tag{2.4}$$

Dimana N adalah nomor pixel pada citra yang digunakan sebagai faktor skala, $I(t_i)$ adalah citra I pada waktu i, $I(t_j)$ adalah citra pada waktu j dan $D(t)$ adalah normalisasi sum of absolute difference pada waktu tersebut. Pada waktu ideal dimana tidak terdapat gerakan persamaan yang didapat persamaan (2.5) adalah :

$$I(t_i) = I(t_j) \tag{2.5}$$

Dan $D(t) = 0$ atau $D(t) \approx 0$. Ketika noise (derau) tampil pada citra, maka persamaan yang didapat persamaan (2.6) adalah :

$$I(t_i) = I(t_j) + n(p) \tag{2.6}$$

Dimana $n(p)$ adalah sinyal noise (derau).

2.1.7.4. Two Dimensional Cross Correlation

Cross Correlation adalah metode standar dalam memperkirakan derajat dimana 2 seri citra terkorelasi[10]. Mempertimbangkan 2 seri citra x(i) dan y(i) dimana i=0,1,2...N-1. Maka *cross correlation* r dari *delay* d didefinisikan persamaan (2.7) :

$$r = \frac{\sum_{1} [(x(i) - mx) * (y(i-d) - my)]}{\sqrt{\sum_{1} (x(i) - mx)^2} \sqrt{\sum_{1} (y(i-d) - my)^2}} \qquad (2.7)$$

Cara kerja dari algoritma 2 *dimensional cross correlation* yaitu langkah pertama, dua citra di bagi menjadi 4 sub citra dengan ukuran yang sama. Hal ini dilakukan untuk meningkatkan sensitivitas dari kalkulasi dimana lebih mudah membandingkan antara sub citra daripada membandingkan antara dua citra dengan ukuran penuh. *Two dimensional cross correlation* dikalkulasikan antara setiap sub citra dengan bagian yang berkorespondensi dengan citra lainnya.

Proses ini menghasilkan empat nilai yaitu antara -1 sampai dengan 1 tergantung pada perbedaan dari dua citra yang berkorelasi. Karena tujuan dari pembagian ini adalah untuk memperoleh sensitivitas yang lebih maka nilai minimum dari correlation akan digunakan sebagai referensi pada *threshold* (ambang).

Pada kasus normal, gerakan dapat dideteksi dengan mudah ketika nilai minimum cross correlation digunakan untuk mengatur *threshold* (ambang). Bagaimanapun, deteksi gagal ketika citra berisi variasi global seperti perubahan iluminasi atau ketika kamera bergerak.

2.2. Matlab 7.04

Bahasa pemrograman sebagai media untuk berinteraksi antara manusia dengan komputer dewasa ini dibuat agar semakin mudah dan cepat. Sebagai contoh, dapat dilihat dari perkembangan bahasa pemrograman Pascal, yang terus memunculkan varian baru hingga akhirnya menjadi Delphi, demikian pula dengan Basic dengan Visual Basic-nya serta C dengan C++Builder-nya. Pada akhirnya semua bahasa pemrograman

akan semakin memanjakan pemakainya dengan penambahan fungsi-fungsi baru yang sangat mudah digunakan bahkan oleh tingkat pemula sekalipun.

Matlab muncul di dunia bahasa pemrograman yang cenderung dikuasai oleh bahasa yang telah mapan. Logikanya, sebagai pemain baru tentu saja Matlab akan sukar mendapat hati dari pemakai. Namun Matlab hadir tidak dengan fungsi dan karakteristik yang ditawarkan bahasa pemrograman lain yang biasanya hampir seragam. Matlab dikembangkan sebagai bahasa pemrograman sekaligus alat visualisasi, yang menawarkan banyak kemampuan untuk menyelesaikan berbagai kasus yang berhubungan langsung dengan disiplin keilmuan Matematika, seperti bidang rekayasa teknik, fisika, statistika, komputasi dan modeling. Matlab dibangun dari bahasa induknya yaitu bahasa C, namun tidak dapat dikatakan sebagai varian dari C, karema dalam sintak maupun cara kerjanya sama sekali berbeda dengan C. Namun dengan hubungan langsungnya terhadap bahasa C, Matlab mempunyai kelebihan-kelebihan bahasa C bahkan mampu berjalan pada semua *platform* Sistem Operasi tanpa mengubah sintak sama sekali.

Matlab dikembangkan oleh MathWorks, yang pada awalnya dibuat untuk memberikan kemudahan mengakses data matrik pada program LINPACK dan EISPACK. Selanjutnya menjadi sebuah aplikasi untuk komputasi matrik. Dari sejak awal dipergunakan, Matlab memperoleh masukan ribuan pemakai. Kehadiran Matlab memberikan jawaban sekaligus tantangan. Matlab menyediakan beberapa pilihan untuk dipelajari, mempelajari metoda visualisasi saja, pemrograman saja atau kedua-duanya. Kemudahan yang lain, karena bahasa pemrograman yang lain memang tidak menawarkan kemudahan serupa. Selain itu Matlab juga memberikan keuntungan bagi programmer-developer program yaitu untuk menjadi program pembanding yang sangat handal, hal tersebut dapat dilakukan karena kekayaannya akan fungsi matematika, fisika, statistik dan visualisasi.

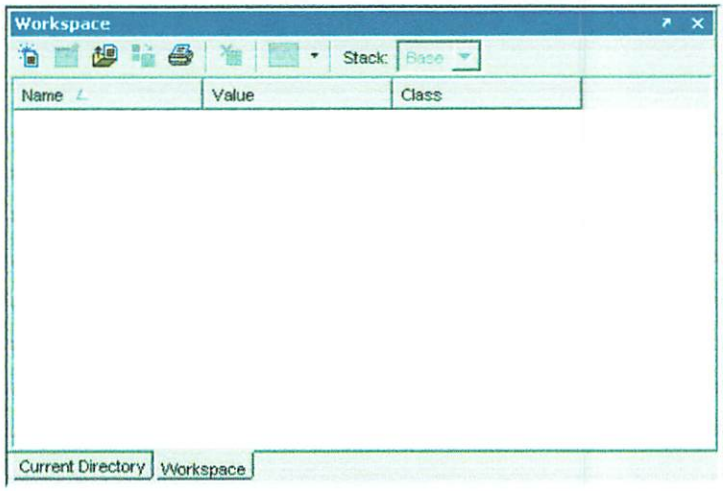
2.2.1. Pengenalan Matlab 7.0.4

Sebagaimana bahasa pemrograman lainnya, Matlab juga menyediakan lingkungan kerja terpadu yang sangat mendukung dalam pembangunan aplikasi. Tampilan jendela Matlab dapat dibagi menjadi beberapa bagian, yaitu :

1. Jendela Utama

2. Workspace

Tampilan *workspace* dalam Matlab dapat dilihat pada gambar 2.3 :

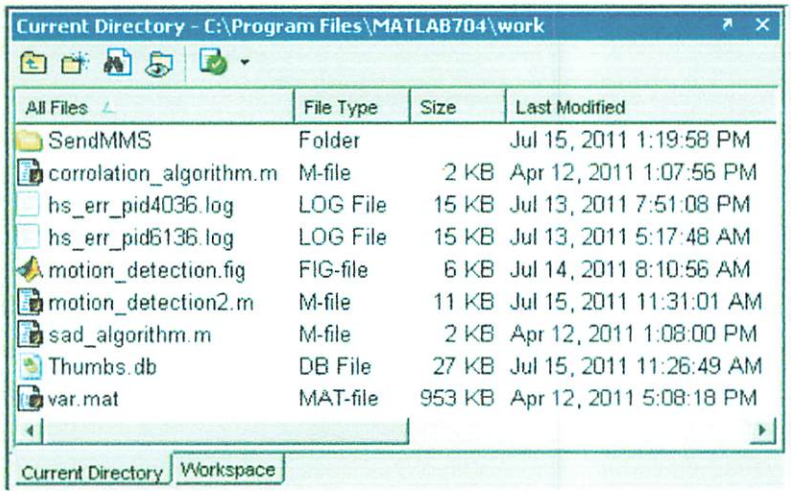


Gambar 2.3 Tampilan *Workspace*

Fungsi *workspace* adalah sebagai navigator bagi pemakai dalam penyediaan informasi mengenai variabel yang sedang aktif dalam workspace pada saat pemakaian. *Workspace* adalah suatu lingkungan abstrak yang menyimpan seluruh variabel dan perintah yang pernah digunakan selama penggunaan Matlab berlangsung.

3. Current directory

Tampilan *current directory* dalam Matlab dapat dilihat pada gambar 2.4 :

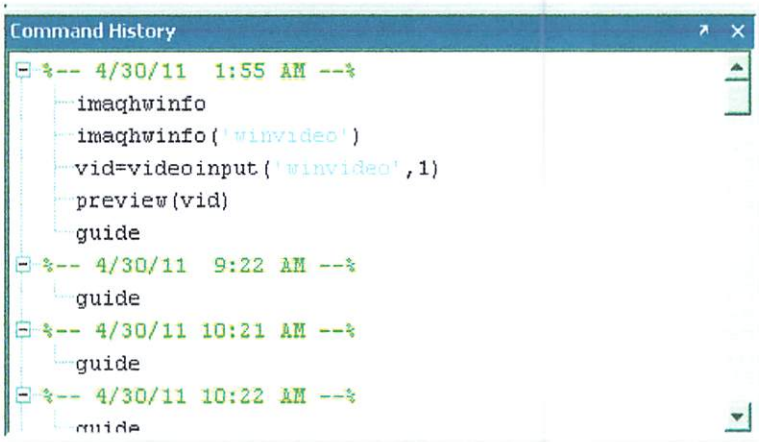


Gambar 2.4 Tampilan *Current Directory*

Fungsi *current directory* adalah memilih direktori yang aktif dan akan digunakan selama penggunaan Matlab berlangsung.

4. Command History

Tampilan *command history* dalam Matlab dapat dilihat pada gambar 2.5 :

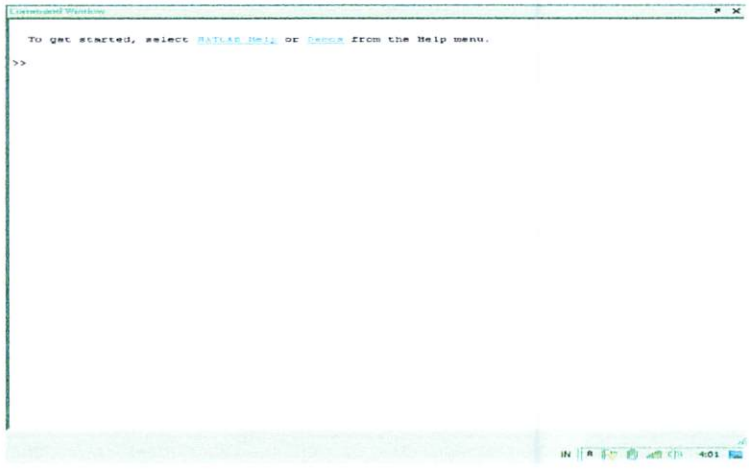


Gambar 2.5 Tampilan *Command History*

Fungsi *command history* adalah menyimpan perintah-perintah yang pernah ditulis pada *command window*.

5. Command window

Tampilan *command window* dalam Matlab dapat dilihat pada gambar 2.6 :

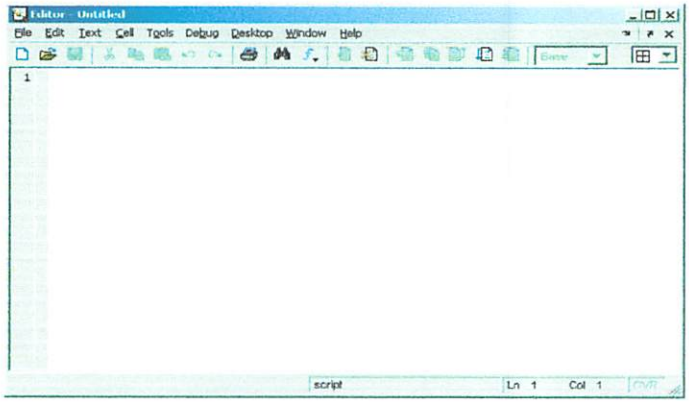


Gambar 2.6 Tampilan *Command Window*

Jendela ini berfungsi sebagai penerima perintah dari pemakai untuk menjalankan seluruh fungsi yang disediakan oleh Matlab. Pada dasarnya jendela ini adalah inti dari pemrograman Matlab yang menjadi media pengguna berinteraksi dengan Matlab.

6. Matlab Editor

Tampilan Matlab *Editor* dapat dilihat pada gambar 2.7:

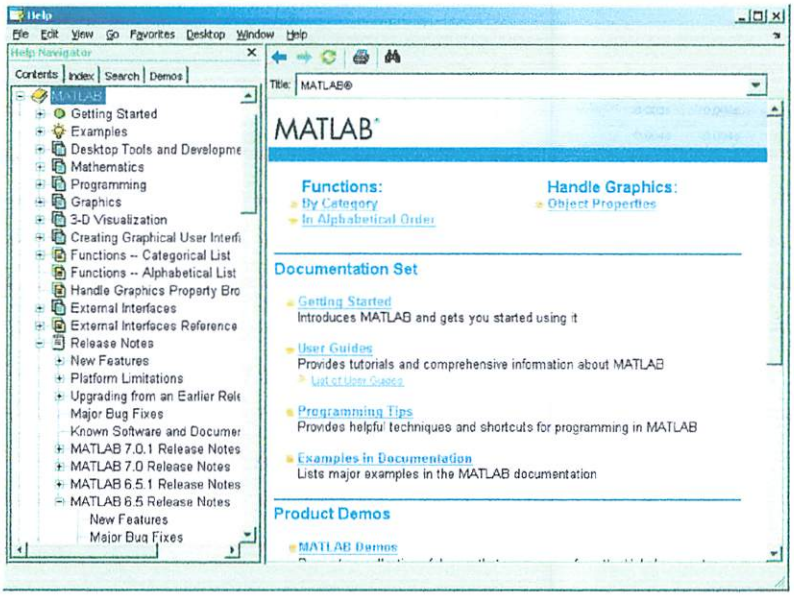


Gambar 2.7 Tampilan Matlab *Editor*

Fungsi Matlab *editor* adalah tempat membuat *script* program pada Matlab. Untuk memunculkan Matlab *Editor*, kita menggunakan perintah File – New – M-file atau dengan mengetikkan >>edit pada command window.

7. Help

Tampilan *Help* dalam Matlab dapat dilihat pada gambar 2.8 :



Gambar 2.8 Tampilan *Help*

2.2.2. Membuat Aplikasi Matlab 7.0.4

Dalam melakukan pekerjaan pemrograman menggunakan bahasa Matlab, ada beberapa cara yang digunakan yaitu sebagai berikut :

1. Pada Command Window

Untuk membuat program, hanya diperlukan untuk mengetikkan program pada prompt Matlab dalam Command Window, misalnya :

>> pjg = 5;

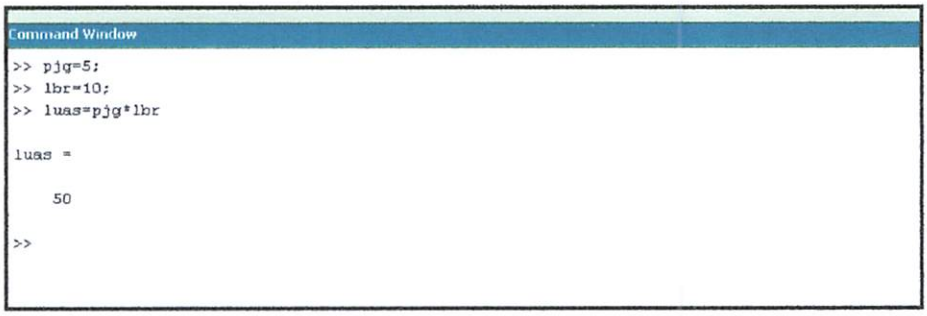
Tekan tombol enter, lalu ketikkan :

>> lbr = 10;

Tekan enter, lalu ketikkan :

>> luas = pjg * lbr

Untuk skrip terakhir tidak diberikan tanda titik koma (;) sehingga hasil dari perhitungan diatas dapat langsung dilihat pada gambar 2.9



Gambar 2.9 Hasil Perhitungan pada Command Window

2. Menggunakan Matlab Editor

Pada command window ketikkan:

```
>> edit
```

Tekan enter, selanjutnya muncul Matlab Editor dan ketikkan program dibawah ini:

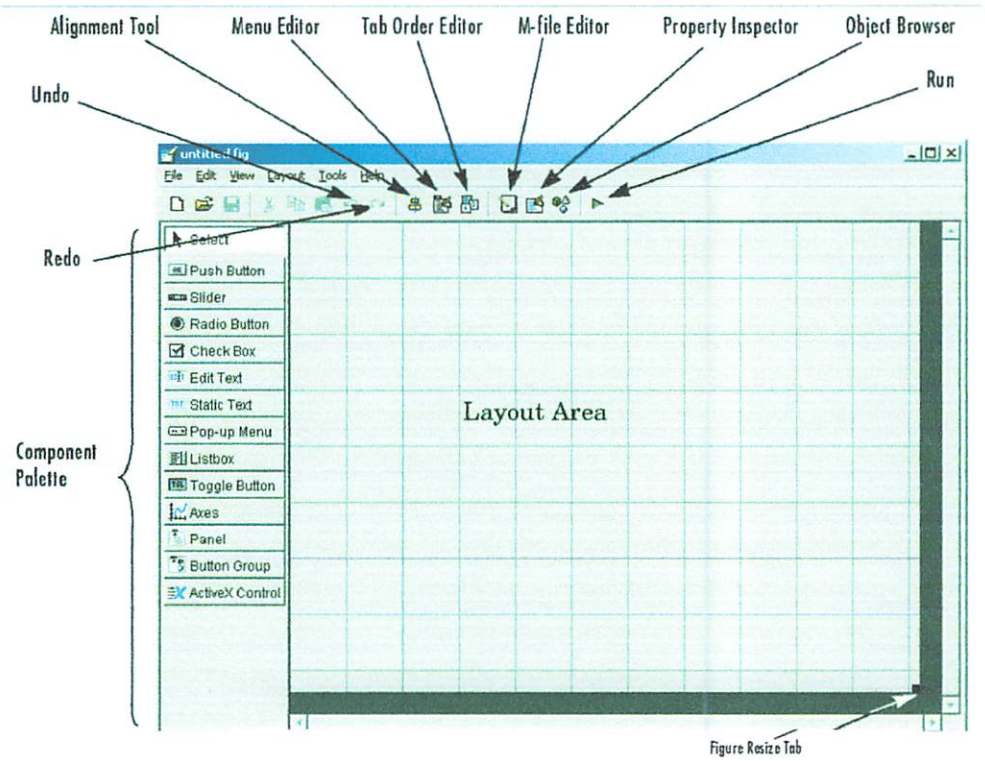
```
% ------------------------------------------------
% Matlab Programming
% oleh : Yozie
% ------------------------------------------------
pjg=10;
lbr=5;
luas=pjg*lbr;
disp(['Luas->' num2str(luas)]);
%------------------------------------------------
```

2.2.3. Bekerja dengan GUI

GUI merupakan tampilan grafis yang memudahkan user berinteraksi dengan perintah teks. Dengan GUI, program yang dibuat menjadi lebih *user friendly*, sehingga user mudah menjalankan suatu aplikasi program.

Untuk membuka lembar kerja GUI dalam Matlab, kita menggunakan perintah File – New – GUI atau dengan mengetikkan `>>guide` pada command window.

Tampilan lembar kerja GUI dalam Matlab dapat dilihat pada gambar 2.10 :



Gambar 2.10 Tampilan Lembar Kerja GUI Matlab [13]

Pada lembar kerja GUI terdapat component pallete dimana komponen-komponen tersebut yang akan digunakan untuk membuat tampilan Matlab yang lebih user friendly, yaitu :

1. Push Button

   Push button merupakan tombol yang jika diklik akan menghasilkan suatu tindakan.

2. Slider

Slider menerima masukan berupa angka pada suatu range tertentu dimana pengguna menggeser kontrol pada slider.

3. Radio Button

   Radio Button merupakan kontrol yang digunakan untuk memilih satu pilihan dari beberapa pilihan yang ditampilkan.

4. Check Box

   Check box merupakan kontrol yang digunakan untuk memilih antara satu atau lebih pilihan dari beberapa pilihan yang ditampilkan.

5. Edit Text

   Edit text merupakan kontrol untuk meng-inputkan atau memodifikasi teks.

6. Static Text

   Static text merupakan kontrol untuk membuat teks label.

7. Pop Up Menu

   Pop up menu merupakan kontrol yang digunakan untuk membuka tampilan daftar pilihan yang telah didefinisikan dengan mengklik tanda panah yang terdapat pada pop up menu.

8. List Box
9. List Box merupakan kontrol yang digunakan untuk menampilkan semua daftar item. Kemudian, pengguna memilih satu diantara item-item yang ada.
10. Toggle Button

    Toggle button hampir sama dengan push button, hanya push button diklik, tombol akan kembali ke posisi semula. Sebaliknya jika toggle button diklik, tombol tidak kembali ke posisi semula kecuali diklik kembali.

11. Axes

    Axes digunakan untuk menampilkan grafik atau gambar.

12. Panel

    Panel merupakan kotak yang digunakan untuk menandai atau mengelompokkan daerah tertentu pada figure.

13. Button group

    Button group hampir sama dengan panel, tetapi button group lebih digunakan untuk mengelompokkan radio button dan toggle button.

14. Active X Control

Digunakan jika Matlab berjalan pada Sistem Operasi Microsoft Windows, dengan menggunakan komponen ini Matlab dapat terhubung dengan komponen Microsoft Windows.

2.3. *Multimedia Messaging Service (MMS)*

Multimedia Messaging Service adalah sebuah metode pengiriman pesan yang mampu mengirimkan data dalam bentuk gambar, suara. maupun teks sekaligus.Dalam hal ini gambar, suara maupun teks harus dibuat dahulu dalam suatu program. MMS hanya dapat beroperasi di dalam jaringan General Packet Radio Service (GPRS). Saat ini layanan MMS memungkinkan untuk melakukan pengiriman pesan MMS ini antar onperator. Akan tetapi masih belum mendukung layanan internasional rooming. Content yang diperbolehkan dalam pengiriman MMS adalah dalam bentuk teks, gambar, klip suara, atau kombinasi dari ketiganya. Dasar dari implementasi MMS adalah penggabungan dari penggunaan teknologi Wireless Application Protocol (WAP) dan teknologi Short Messaging Service (SMS).

2.3.1. Proses Sederhana MMS

Berikut ini adalah pronses kerja pengiriman MMS kepada handphone yang dituju :

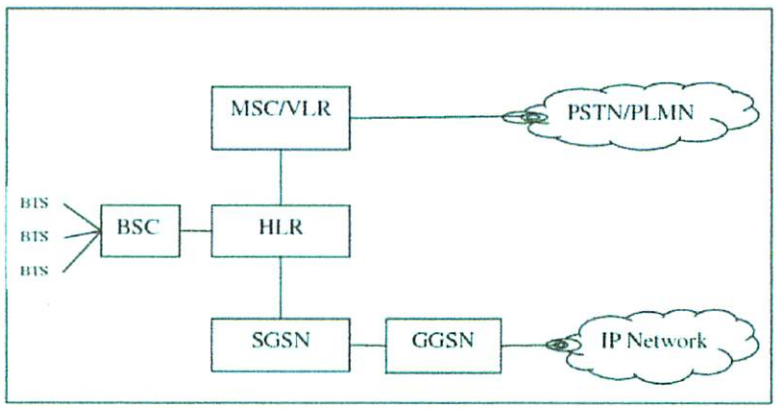
1. Pertama-tama dimulai dari content file yang memuat isi dari suatu pesan yang bertipe MMS. Pesan ini dapat dibuat melalui suatu program aplikasi atau dibuat oleh user yang mengirimkan message melalui handphone. Ketika message dikirim dari handphone maka handphone tersebut terhubung pada MMSC (Multimedia Messaging Service Center) melalui WAP dan mengirimkan isi dari message tersebut kepada MMSC untuk selanjutnya MMSC akan bertanggungjawab atas pengiriman selanjutnya.
2. Isi dari message MMS tersebut di-*publish* sehingga dapat diakses melalui URL.
3. Sebuah MMS nofication message (speciai binary message yang dikirimkan melalui SMS) dikirimkan kepada handphone penerima. MMS nofication message yang terdiri dari header information tentang MMS message dan pointer URL. Sehingga penerima dapat membuka isi dari MMS message yang dikirim

4. Penerima membuka WAP session untuk dapat membuka isi dari MMS message yang dikirmkan.

Jadi penerima yang tidak mempunyai handphone yang mendukung MMS tetap dapat membaca pesan yang dikirim melalui URL yang tersedia karena sebelumnya menerima SMS yang berisi header information dan URL tempat message di-*publish.*

2.4. *General Packet Radio Service (GPRS)*

Jaringan GSM mempunyai dua macam network yaitu *circuit switched network* yang berguna untuk pengiriman voice serta *packed switched network* yang berguna untuk pengiriman data. Fasilitas GPRS diimplementasikan melalui jaringan GSM yang telah ada. Untuk itu ditambahkan dua elemen network baru ke dalam jaringan GSM. yaitu *Gateway GPRS Support Node* (GGSN) dan *Serving GPRS Support Node* (SGSN) dapat dilihat pada gambar 2.11



Gambar 2.11 Struktur GPRS Network

Seperti yang terlihat pada gambar 2.12 Base Transceiver Station (BTS) berguna untuk menyediakan service bagi pengguna handphone. BTS dikontrol oleh Base Station Controier (BSC). BSC berfungsi untuk mengatur dan menjembatani antara BTS dengan Mobiie Switching Center (MSC) atau dengan SGSN. Secara garis besar fungsi dari SGSN dan MSC adalah sama, yang membedakan adalah jika MSC mengatur lalu lintas percakapan suara dan mengatur sistem pembayaran untuk percakapan yang dilakukan, sedangkan SGSN mengatur lalu lintas pengiriman data dan mengatur sistem pembayaran untuk penggunaan GPRS. Sedangkan GGSN berguna untuk menjembatani antara SGSN dengan external network. Biasanya GGSN difungsikan menjadi WAP gateway. Sedangkan Home Location Register (HLR) adalah database yang berisi data

2.5 NetBeans IDE 6.7.1

The NetBeans IDE adalah sebuah lingkungan pengembangan - sebuah kakas untuk pemrogram menulis, mengompilasi, mencari kesalahan dan menyebarkan program. Netbeans IDE ditulis dalam Java - namun dapat mendukung bahasa pemrograman lain. Terdapat banyak modul untuk memperluas Netbeans IDE. Netbeans IDE adalah sebuah produk bebas dengan tanpa batasan bagaimana digunakan.

NetBeans dimulai pada tahun 1996 sebagai Xelfi (kata bermain pada Delphi), Java IDE proyek mahasiswa di bawah bimbingan Fakultas Matematika dan Fisika di Charles University di Praha . Pada tahun 1997 Staněk Romawi membentuk perusahaan sekitar proyek tersebut dan menghasilkan versi komersial NetBeans IDE hingga kemudian dibeli oleh Sun Microsystems pada tahun 1999. Sun open-source IDE NetBeans pada bulan Juni tahun berikutnya. Sejak itu, komunitas NetBeans terus berkembang.

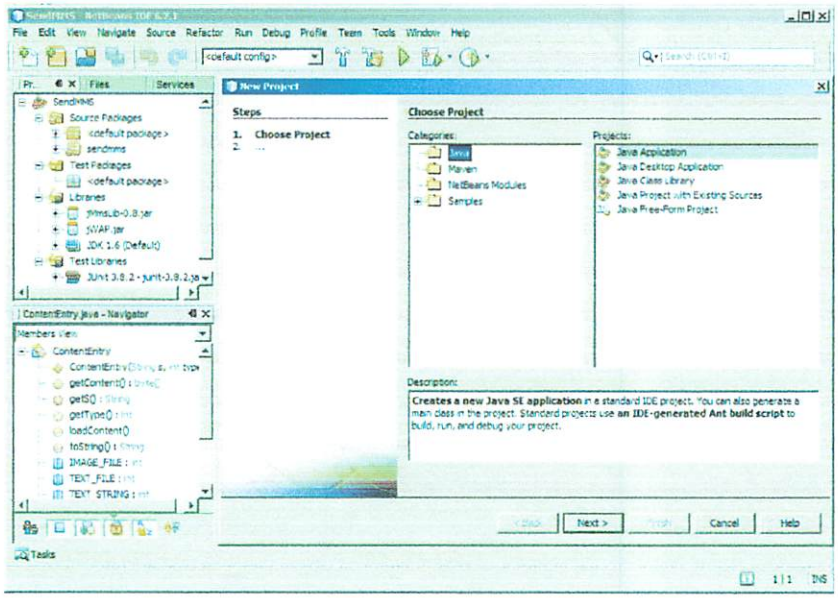
NetBeans mengacu pada kedua platform kerangka untuk aplikasi desktop Java, dan sebuah lingkungan pengembangan terpadu (IDE) untuk pengembangan dengan Java , JavaScript , PHP , Python , Ruby , Groovy , C , C + + , Scala , Clojure , dan lain-lain. NetBeans IDE ditulis dalam Java dan dapat berjalan di manapun JVM diinstal, termasuk Windows, Mac OS, Linux, dan Solaris. Sebuah JDK diperlukan untuk pengembangan fungsionalitas Jawa, tetapi tidak diperlukan untuk pembangunan di bahasa pemrograman lain. Platform NetBeans memungkinkan aplikasi untuk dikembangkan dari satu set modular komponen software yang disebut modul. Aplikasi berbasis platform NetBeans (termasuk IDE NetBeans) dapat diperpanjang oleh pengembang pihak ketiga .

Aplikasi dapat menginstal modul secara dinamis. Setiap aplikasi dapat memasukkan modul Update Center untuk mengijinkan pengguna aplikasi untuk download digital-ditandatangani upgrade dan fitur-fitur baru secara langsung ke dalam aplikasi yang berjalan. Menginstal Ulang upgrade atau rilis baru tidak memaksa pengguna untuk men-download keseluruhan aplikasi lagi.

Platform ini menawarkan layanan dapat digunakan kembali umum untuk aplikasi desktop, mengijinkan pengembang untuk fokus pada logika khusus untuk aplikasi mereka. Di antara fitur platform adalah:

- User antarmuka manajemen (misalnya menu dan toolbar)
- Pengaturan pengguna manajemen
- Manajemen penyimpanan (tabungan dan memuat segala jenis data)

- Window manajemen
- kerangka Wizard (mendukung dialog langkah-demi-langkah)
- NetBeans Visual Perpustakaan
- Integrated Development Tools



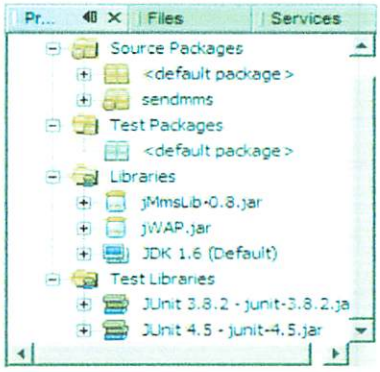
Gambar 2.12 Tampilan Interface NetBeans IDE 6.7.1

2.5.1 Pengenalan NetBeans IDE 6.7.1

Sebagaimana bahasa pemrograman lainnya, NetBeans juga menyediakan lingkungan kerja terpadu yang sangat mendukung dalam pembangunan aplikasi. Tampilan jendela Netbeans dapat dibagi menjadi beberapa bagian, yaitu :

1. Jendela Utama
2. Project

   Tampilan *Project* dalam Netbeans dapat dilihat pada gambar 2.13 :

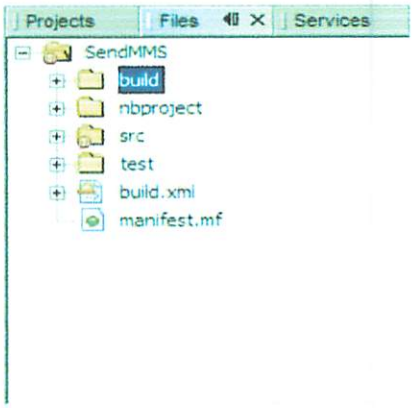


Gambar 2.13 Tampilan *Project*

Fungsi *Project* adalah memberitahukan project – project yang akan digunakan atau project aktif dan akan digunakan selama penggunaan NetBeans berlangsung.

3. Files

Tampilan *Files* dalam NetBeans dapat dilihat pada gambar 2.14:

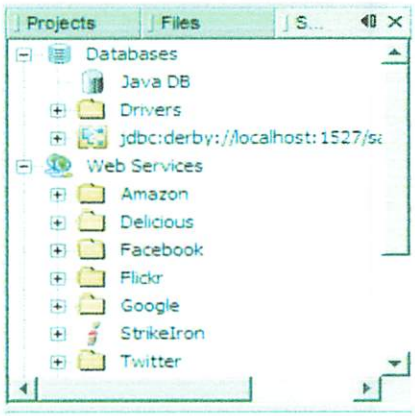


Gambar 2.14 Tampilan *Files*

Fungsi *Files* adalah menujukan semua *files* yang digunakan atau dipilih di dalam komputer

4. Service

Tampilan *service* dalam NetBeans dapat dilihat pada gambar 2.15:

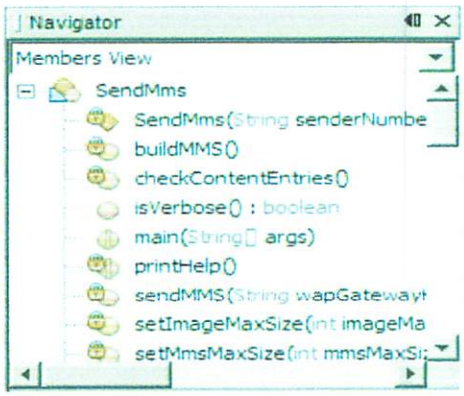


Gambar 2.15 Tampilan *Service*

Fungsi *service* adalah menujukan macam – macam *service* yang ditawarkan oleh Netbeans

5. Navigator

Tampilan *navigator* dalam Netbeans dapat dilihat pada gambar 2.16:

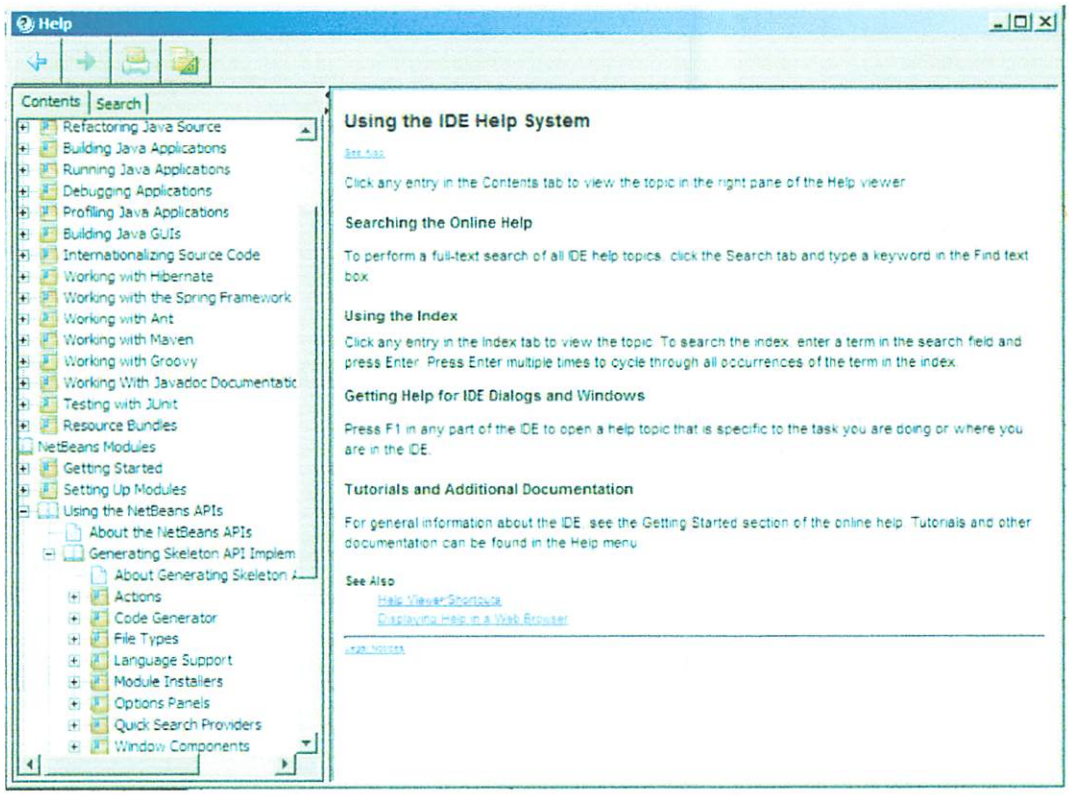


Gambar 2.16 Tampilan *Navigator*

Fungsi *navigator* adalah menujukan atau me*navigator*i *project* yang sedang kita buka

6. Help

Tampilan *help* dalam Netbeans dapat dilihat pada gambar 2.17 :



Gambar 2.17 Tampilan *Help*

BAB III
ANALISA DAN PERANCANGAN SISTEM

3.1. Analisa Sistem

Pada bab ini akan dibahas tentang desain sistem pemantauan ruangan menggunakan *motion detection* yang akan dibangun, mengenai bagian-bagian dari sistem tersebut dan juga cara-cara serta metode yang digunakan.

Cara kerja dari sistem pemantauan ruangan berdasarkan *motion detection* menggunakan *webcam* adalah :

a) Dari *webcam* program menampilkan citra yang di*capture* sesuai dengan area pemantauan, serta mendeteksi adanya gerakan.
b) Jika terjadi pergerakan, maka program *motion detection* akan memberikan sebuah peringatan berupa alarm (jika diinginkan) melalui suara PC/ Laptop yang digunakan.
c) Menyimpan hasil ke dalam *hard disk.*

3.2. Rancangan Program *Motion Detection*

Program *motion detection* yang akan dibangun, menggunakan bahasa Matlab 7.0.4. Dari bahasa pemrograman ini akan memanfaatkan *toolbox Image Acquisition* yang telah tersedia. *Toolbox* ini memiliki fungsi-fungsi yang digunakan untuk pengolahan citra, diantaranya yang digunakan untuk program ini adalah untuk mengkoneksikan perangkat keras citra seperti *frame grabber*, *webcam*, *firewire* dan lain-lain ke Matlab 7.0.4, melakukan *preview* (peninjauan) secara langsung dari perangkat citra, mengkonfigurasi fungsi *callback* yang dapat mengatur perangkat keras citra, serta membawa data citra ke Matlab.

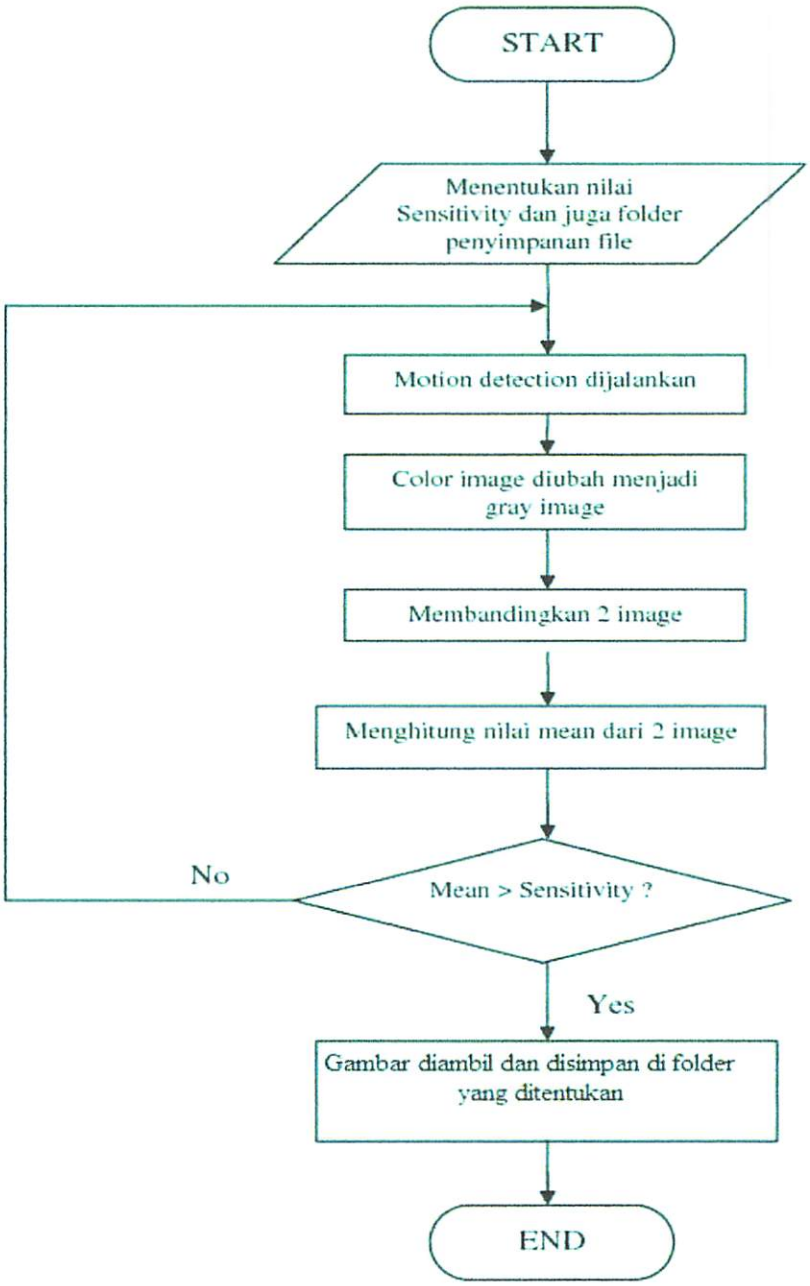
Berikut adalah urutan rencana pembuatan program *motion detection* :

1. *Webcam* menangkap citra yang berupa *color image* (citra berwarna). Setelah itu ditampilkan di layar.
2. Citra yang ditangkap berupa *color image* (citra berwarna) diubah menjadi *gray scale*. Hal ini dilakukan agar dapat mempermudah proses pembandingan citra.
3. Dari citra *gray*, dilakukan proses perhitungan berdasarkan algoritma yang dipilih sebelumnya yaitu *Sum of Absulute Difference* atau *Two Dimensional Cross Correlation.*

4. Melakukan perhitungan antara dua citra yang berurutan apakah terdapat perbedaan antara dua citra tersebut atau tidak yang kemudian disebut dengan *variance value*.
5. Nilai *variance* dapat dijadikan perbandingan dengan nilai *threshold* (ambang), jika terdapat selisih, maka *webcam* akan mengambil citra dan menampilkan ke program serta menyimpannya ke *memory* komputer/ laptop. Dengan mengevaluasi nilai selisih dari dua citra dan membandingkannya dengan nilai *threshold* maka dapat diketahui apakah pada citra terdapat objek yang bergerak. Pada saat yang ideal dimana hanya terdapat 2 citra yang dibandingkan bagian yang tidak bergerak akan menghasilkan nilai = 0 atau nilai $\approx$ 0 pada saat komponen lain berpengaruh terhadap nilai citra seperti *noise* (derau). Sedangkan pada bagian yang bergerak menghasilkan nilai $\neq$ 0. Dalam hal ini nilai *threshold* yang ditentukan berpengaruh terhadap kesensitifan pendeteksian gerakan. Sehingga jika nilai *threshold* berada pada nilai tertentu dapat mengakibatkan pergerakan kecil dapat terdeteksi atau tidak terdeteksi sama sekali.
6. Dengan demikian jika program dijalankan, maka program akan menghitung nilai *variance value* dari dua citra dan membandingkannya dengan nilai *threshold* atau *sensitivity* yang telah ditentukan. Hal tersebut berjalan terus menerus secara kontinyu. Sehingga jika terjadi yang mengakibatkan nilai *variance* lebih besar daripada nilai *sensitivity* yang telah ditentukan, maka *webcam* akan melakukan pengambilan citra lalu menyimpannya ke dalam *hard disk*. Semua langkah tersebut berjalan secara otomatis. Citra yang diambil akan tersimpan berupa image.

3.2.1 Diagram Alir Program Utama

Secara umum sistem pemantauan ruangan dengan menggunakan deteksi gerakan yang dirancang dapat dilihat pada diagram alir pada gambar 3.1 :



Gambar 3.1 Diagram alir proses *motion detection*

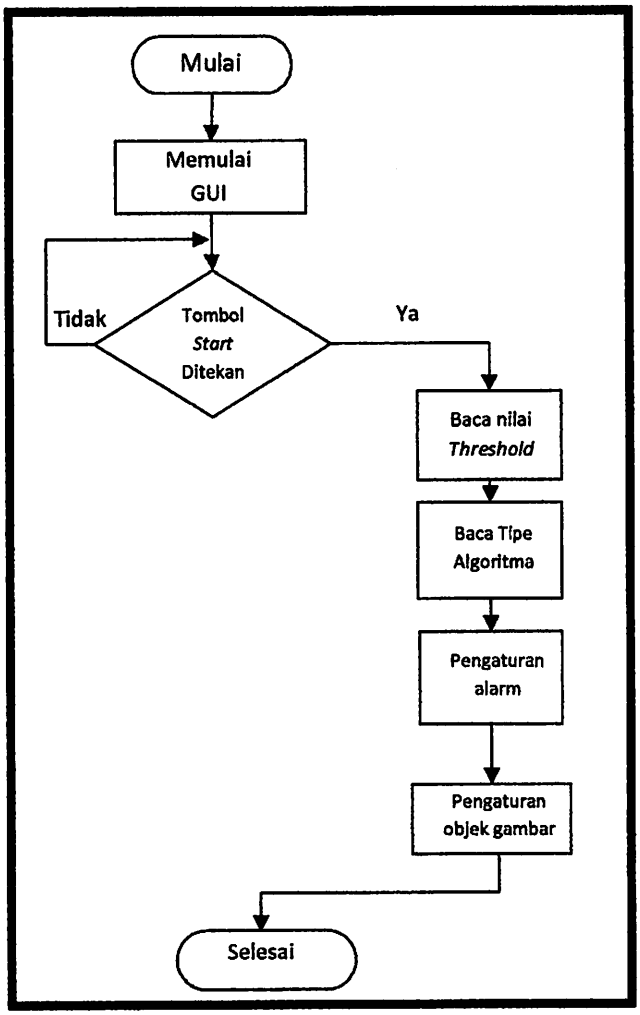
Dari diagram alir diatas dapat dilihat proses dari program sistem pemantauan ruangan menggunakan *motion detection*. Langkah pertama adalah memulai inisialisasi dari parameter program dan pengaturan alat seperti penempatan *webcam*. Kemudian, ketika program dijalankan nilai dari *flag* akan mengindikasikan apakah tombol *stop* ditekan atau tidak. Jika tombol *stop* tidak ditekan maka program akan tetap membaca citra kemudian memprosesnya dengan menggunakan salah satu dari dua algoritma yang telah dipilih operator sebelumnya. Jika gerakan terdeteksi maka program akan merespon dengan mengambil serangkaian tindakan sebagai respon dari deteksi gerakan dan

kemudian kembali membaca citra berikutnya, namun jika tidak terdeteksi gerakan maka program secara langsung akan membaca citra berikutnya dan membadingkan dengan citra berikutnya.

Jika tombol *stop* ditekan maka nilai dari *flag* adalah 0 (nol) dan program akan berhenti, kemudian *memory* PC/*notebook* dibersihkan dan hasil yang diperlukan akan direkam. Nilai *flag* yang berupa nol akan mengakhiri program dan operator dapat mengumpulkan hasil dari program yang berdasarkan pada deteksi gerakan yang sudah dijalankan.

3.2.2 Pengaturan dan Inisialisasi

Pada proses pengaturan dan inisialisasi, terdapat beberapa rincian pada gambar 3.2:



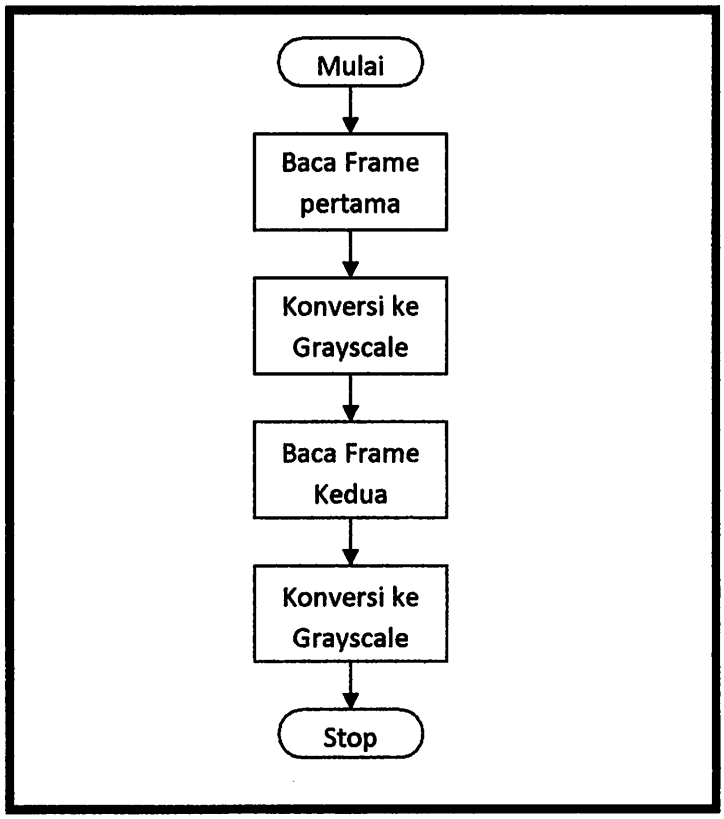
Gambar 3.2 Diagram alir pengaturan dan inisialisasi

Diagram alir diatas mendeskripsikan proses pengaturan dan inisialisasi. Proses ini termasuk pengeksekusian program dalam bentuk *graphical user interface* (GUI) dimana

pada GUI tersebut dilakukan pemilihan algoritma dan nilai *threshold* (nilai *sensitivity*). Serta pengaturan alarm apakah diaktifkan atau tidak jika terdeteksi gerakan dan pengaturan objek video yang akan ditangkap. Objek video merupakan bagian dari proses *image acquisition* namun harus dijalankan di awal program.

3.2.3 *Image Acquisition*

Pada proses *image acquisition*, secara lebih terperinci yaitu seperti pada diagram alir pada gambar 3.3 :



Gambar 3.3 Diagram Alir Proses *Image Acquisition*

Setelah pengaturan, maka proses selanjutnya adalah proses *image acquisition* pada gambar 3.3 diatas. Proses ini membaca citra dari *webcam* dan menyimpannya ke dalam format yang sesuai untuk deteksi gerakan.

Untuk program yang akan dibangun fungsi *videoinput* digunakan untuk menginisialisasi objek *video* yang terhubung ke PC *camera* secara langsung. Kemudian fungsi *preview* digunakan untuk menampilkan *video* secara langsung dari monitor. Fungsi *getsnapshot* juga digunakan pada program ini untuk membaca citra dari kamera dan menempatkannya di *workspace* MATLAB. Pendekatan ini dipilih karena mampu mencapai pengambilan citra lebih cepat dengan kecepatan mencapai 5 frame per detik

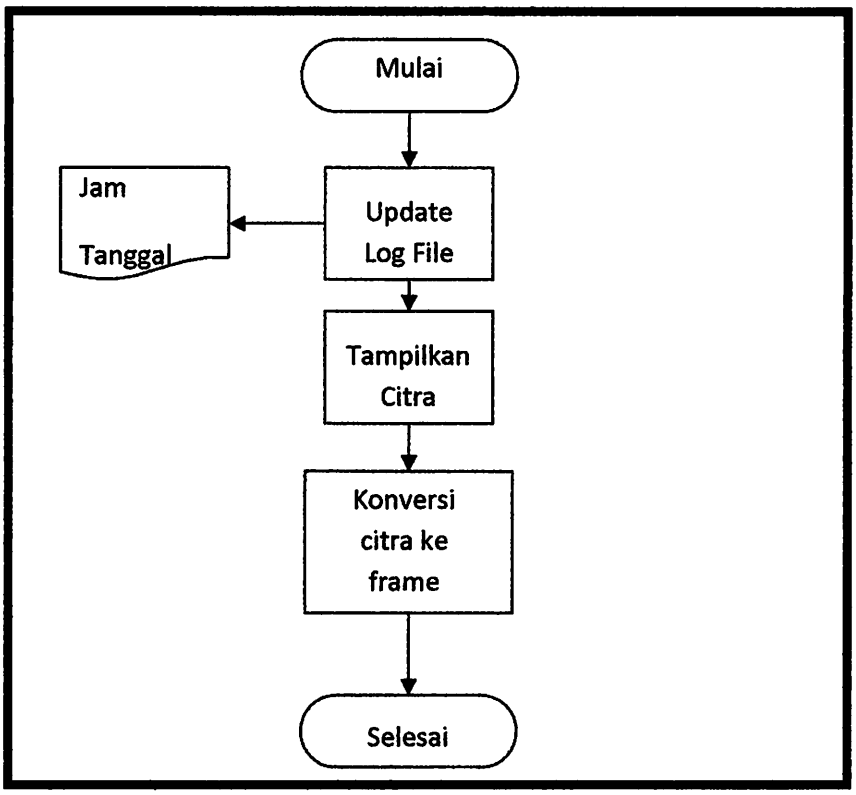
tergantung pada kompleksitas algoritma dan kecepatan prosesor PC. Serta, masalah sinkronisasi dapat diatasi karena proses pengambilan citra dan pemrosesan citra menggunakan program yang sama.

Semua citra yang dibaca kemudian dikonversi menjadi citra dua dimensi monokrom. Hal ini dilakukan untuk mempermudah penghitungan *motion detection* (deteksi gerak).

3.2.4 Respon terhadap *Motion Detection*

Nilai *variance* yang sudah dihitung pada proses sebelumnya baik lebih rendah atau lebih tinggi dari nilai threshold akan disimpan dalam sebuah *array*, dimana akan digunakan sebagai data pada grafik perbandingan antara nomor *frame* dengan *variance value*. Grafik ini membantu dalam membandingkan *variance value* dengan *threshold* sebagai acuan dalam memilih nilai *threshold* yang optimal.

Ketika *variance value* kurang dari *threshold* maka citra tidak akan diproses dan hanya nilai *variance* yang akan disimpan. Namun jika nilai *variance* lebih besar dari *threshold* maka beberapa tindakan akan merespon. Respon dari motion detection dijelaskan pada gambar 3.4:



Gambar 3.4 Respon terhadap *motion detection*

Seperti yang digambarkan pada diagram alir diatas, beberapa respon aktif ketika gerakan terdeteksi. Pertama, alarm akan dieksekusi (jika diaktifkan). Kemudian file log akan diciptakan dan diisi dengan informasi yang berupa jam dan tanggal terjadinya gerakan dan nomor frame dimana gerakan terdeteksi. Selanjutnya file citra yang berupa *still image* akan dikonversi ke dalam format *frame* yang akan ditambahkan ke struktur film.

3.2.5 Proses *Break and Clear*

Setelah algoritma deteksi gerak melakukan kalkulasi pada citra, program melakukan pengecekan apakah tombol stop pada GUI ditekan. Jika ditekan maka nilai *flag* akan berubah dari satu menjadi nol dan program akan berhenti, kemudian mengakhiri perulangan dan kembali ke kontrol tombol pada GUI serta objek video akan diputus. Proses ini dapat memutuskan hubungan perangkat dengan PC sehingga *memory space* yang digunakan dapat berkurang.

3.2.6 *Data record*

Pada saat program diakhiri maka proses pengumpulan data dimulai dengan mengolah data pada variabel dan array pada *memory* yang akan disimpan pada *hard disk*. Pendekatan ini digunakan untuk memisahkan waktu nyata antara proses pencitraan dengan proses hasil. Hal ini dapat bermanfaat untuk memanggil data ketika dibutuhkan. Variabel yang disimpan dari *memory* ke *hard disk* adalah *variance value* dan struktur film yang berisi frame-frame yang berisi gerakan terdeteksi.

3.3. *Desain Graphical User Interface*

GUI didesain untuk memfasilitasi sistem operasi program yang interaktif. GUI dapat digunakan untuk pengaturan awal program, menjalankan program, mengakhiri program dan menampilkan hasil dari kalkulasi dalam program. Pada saat pengaturan awal program operator dapat memilih algoritma deteksi gerakan mana yang akan digunakan serta menentukan sensitivitas deteksi.

Ketika tombol *toggle start/stop* ditekan program akan dijalankan dan algoritma yang terpilih dipanggil yaitu file SAD.m atau 2Dcorr.m untuk melakukan kalkulasi terhadap masukan berupa citra dari *webcam* yang terhubung dengan PC / *notebook* sampai tombol *toggle start/stop* ditekan lagi dimana akan mengakhiri perhitungan dan

mengembalikan kontrol ke *graphical user interface* (GUI). Hasilnya dapat dilihat melalui file *log* yang berisi informasi nomor frame dan waktu terdeteksi gerakan, film yang berisi frame dengan gerakan terdeteksi dan grafik plot yang berisi perbandingan nomor frame dengan *variance value* mulai program dijalankan sampai program selesai dijalankan.

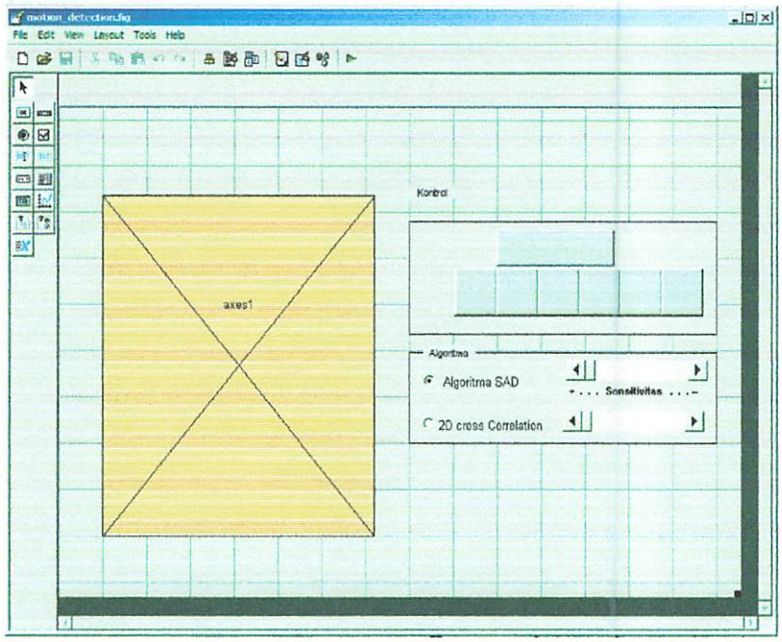
Di dalam GUI program *motion detection*, terdapat beberapa komponen yang digunakan, yaitu *button, radio button, axes, slider* dan *toggle button.* Berikut ini perencanaan tombol-tombol yang akan digunakan dalam program *motion detection* :

- *Start* : untuk memulai program melakukan deteksi gerakan.
- *Stop* : untuk berhenti melakukan deteksi gerakan.
- *View Log* : untuk melihat isi dari file log.txt
- *Video* : untuk membuka citra yang sudah ditangkap oleh program.
- *Alarm* : digunakan untuk mengaktifkan/menonaktifkan alarm jika terdeteksi gerakan.
- *View Plot* : untuk melihat grafik antara variance value dengan nilai threshold pada citra yang terdeteksi gerakan.
- *Help* : untuk melihat deskripsi singkat dari program.
- *Exit* : untuk keluar dari program.
- *Axes* : digunakan untuk melihat citra yang terdeteksi adanya gerakan serta melihat hasil dari pengambilan citra, melihat grafik.
- *SAD* : untuk mengaktifkan algoritma Sum of Absolute Difference
- *Slider* : digunakan untuk menentukan sensitivitas yang ditentukan oleh operator dimana nilai minimal dan maksimal sudah ditentukan sebelumnya sehingga dapat memaksimalkan algoritma.
- *2D cross corr* : untuk mengaktifkan algoritma *2 dimensional cross correlation.*

Dari deteksi gerakan tersebut akan menghasilkan dua *file* yaitu log.txt dan film.avi, yaitu dengan keterangan sebagai berikut:

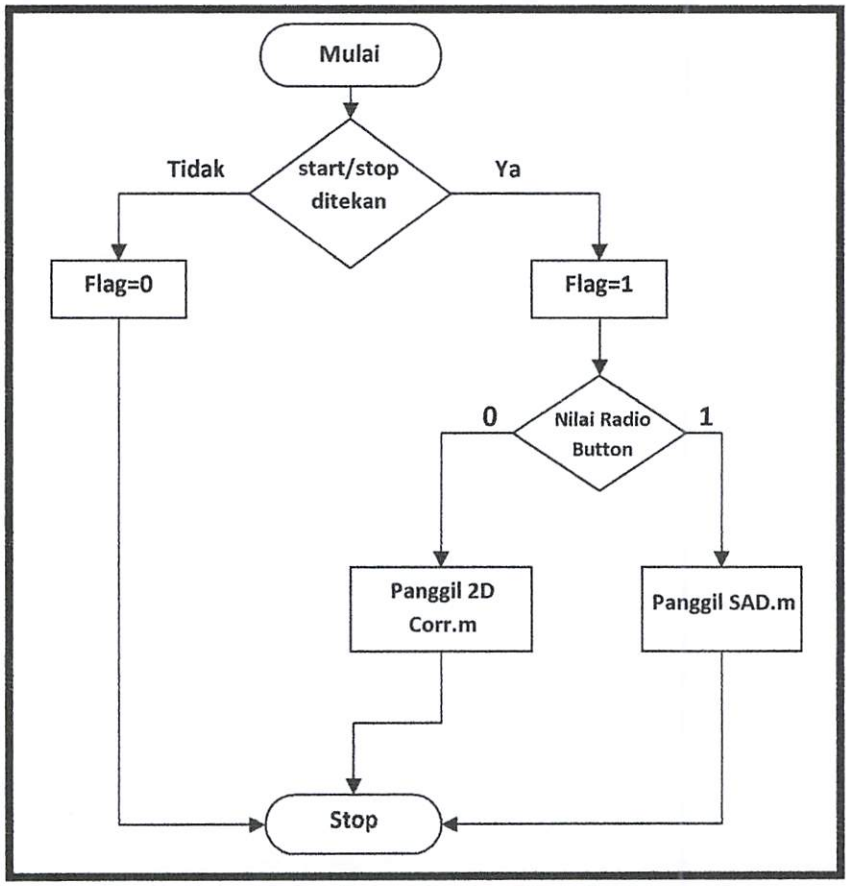
- Log.txt : menyimpan keterangan berupa nomor frame serta tanggal dan jam ketika gerakan terdeteksi oleh program.

Berikut ini adalah rencana tampilan dan letak ikon-ikon yang dibuat pada gambar 3.5 :



Gambar 3.5 Rencana tampilan program *motion detection*

Gambar 3.6 dibawah ini menunjukkan alur program pada saat tombol *toggle start/stop* ditekan.



Gambar 3.6 Diagram alir tombol *start/stop*

Ketika tombol *start/stop* pertama ditekan, nilai *flag* menjadi 1 kemudian nilai dari *radio button* diperiksa apakah 1 atau 0 untuk menentukan program utama mana yang akan dipanggil (SAD.m atau 2dcorr.m). File log berisi informasi tentang waktu jam dan tanggal serta nomor frame dimana gerakan terdeteksi akan terbuka ketika tombol *view* log ditekan. Program akan ditutup dan berhenti jika tombol *exit* ditekan.

3.3.1 Pengujian Berdasarkan Algoritma yang Dipilih

Dalam pengujian ini dilakukan suatu aktivitas pendeteksian gerakan pada suatu ruangan yang sama dan dengan kondisi yang sama. Dimana pengujian tersebut dilakukan dengan memonitoring secara kontinyu suatu ruangan dengan memilih diantara dua algoritma yang digunakan untuk membantu penghitungan algoritma motion detection sehingga didapatkan suatu informasi yang relevan mengenai kondisi atau sistem maupun aplikasi yang berjalan seperti pada tabel 3.1

Tabel 3.1 Pengujian algoritma SAD dengan sensitivitas tinggi

No	CAHAYA		JENIS GERAKAN			HASIL DETEKSI	
	TERANG	REDUP	CEPAT	NORMAL	LAMBAT	ALARM	PENYIMPANAN
1	√		√			√	√
2	√			√		√	√
3	√				√	√	√
4		√	√			√	√
5		√		√		√	√
6		√			√	√	√

Tabel 3.1 menunjukkan bahwa pada pengujian algoritma sum of absolute difference dengan sensitivitas yang tinggi dapat disimpulkan bahwa pada saat cahaya terang maupun redup, dengan gerakan cepat normal atau lambat dapat terdeteksi dengan baik dan menghasilkan keluaran berupa alarm berbunyi (alarm diaktifkan) dan penyimpanan video. Sedangkan pengujian yang dilakukan pada algoritma SAD dengan sensitivitas rendah, dapat direpresentasikan pada tabel 3.2:

Tabel 3.2 Pengujian algoritma SAD dengan sensitivitas rendah

No	CAHAYA		JENIS GERAKAN			HASIL DETEKSI	
	TERANG	REDUP	CEPAT	NORMAL	LAMBAT	ALARM	PENYIMPANAN
1	√		√			-	-
2	√			√		√	√
3	√				√	-	-
4		√	√			√	√
5		√		√		√	√
6		√			√	-	-

pada pengujian dengan sensitifitas rendah, algoritma sum of absolute difference bekerja kurang maksimal sehingga pada saat cahaya terang dengan gerakan yang cepat dan pada saat gerakan lambat, tidak dapat terdeteksi karena nilai sensitifitas rendah menyebabkan perbandingan variance value tidak lebih besar dibandingkan dengan nilai threshold.

Untuk pengujian dengan menggunakan algoritma 2d cross correlation dapat direpresentasikan pada tabel 3.3 dibawah ini :

Tabel 3.3 Pengujian algoritma 2D Cross Correlation dengan sensitivitas tinggi

NO	CAHAYA		JENIS GERAKAN			HASIL DETEKSI	
	TERANG	REDUP	CEPAT	NORMAL	LAMBAT	ALARM	PENYIMPANAN
1	√		√			-	-
2	√			√		-	-
3	√				√	-	-
4		√	√			√	√
5		√		√		√	√
6		√			√	-	-

Pada tabel diatas terlihat bahwa gerakan dapat terdeteksi pada saat tertentu, yaitu pada saat cahaya redup dengan jenis gerakan cepat dan normal. Sehingga algoritma ini sesuai digunakan untuk kondisi tersebut.

Sedangkan pengujian dengan sensitivitas rendah dapat dilihat pada tabel 3.4:

Tabel 3.4 Pengujian algoritma 2D Cross Correlation dengan sensitivitas rendah

NO	CAHAYA		JENIS GERAKAN			HASIL DETEKSI	
	TERANG	REDUP	CEPAT	NORMAL	LAMBAT	ALARM	PENYIMPANAN
1	√		√			-	-
2	√			√		√	√
3	√				√	-	-
4		√	√			√	√
5		√		√		√	√
6		√			√	-	-

Pada saat cahaya yang berada pada ruangan yang dipantau terang, gerakan yang terdeteksi adalah gerakan yang normal, sedangkan gerakan cepat dan lambat tidak terdeteksi. Sedangkan pada saat cahaya pada ruangan yang dipantau redup, maka gerakan lambat tidak dapat terdeteksi. Dan pada gerakan cepat dan normal gerakan dapat terdeteksi dengan baik.

Sehingga pada algoritma 2D cross correlation dengan sensitivitas rendah sesuai untuk mendeteksi gerakan cepat dan normal pada saat cahaya yang berada dalam ruangan redup, dan pada saat gerakan normal dengan cahaya terang pada ruangan yang dipantau.

3.4 Rencana Program Pembuatan dan Pengiriman MMS

Pembuatan dan pengiriman MMS dibuat dengan menggunakan java language dengan menggunakan Netbeans yang menggunakan Nokia MMS java library serta JWAP protocol stack dan Ozenky MMS Messenger. Untuk dapat mengirim MMS diperlukan koneksi GPRS yang merupakan fasilitas yang terdapat dalam operator GSM.

Untuk memperoleh fasilitas tersebut pengguna wajib mendaftar dahulu kepada operator yang bersangkutan. Proses pendaftaran dilakukan dengan mengirimkan SMS kepada operator GSM yang digunakan oleh pcngguna. Kemudian operator akan mengirimkan pesan yang berisi tentang setting konfigurasi GPRS maupun MMS. Dengan melakukan setting tersebut maka proses pengirimaan dan penerimaan MMS akan dapat dilakukan.

3.4.l. Pembuatan Pesan MMS

Hal yang pertama yang harus dilakukan sebelum mengirimkan MMS adalah membuat sebuah pesan MMS. Untuk membuat pesan MMS yang baru diperlukan Nokia MMS library. Pada Nokia MMS library terdapat beberapa inisialisasi antara lain:

❖ MMS *version*

Menunjukkan versi dari MMS yang akan digunakan. Versi MMS yang digunakan berdasarkan dari Nokia MMS library yang digunakan.

❖ *To*

Inisialisasi yang menunjukan kemana MMS akan dikirim. Berikut ini address type dalam MMS yang menggunakan handphone :

- PLMN (Pablic Land Mobile Network) : Address type untuk pengiriman MMS melalui handphone.
  Contoh : +35899000066/TYPE:PLMN

❖ *From*

Menunjukan nomor handphone yang mengirimkan pesan MMS. Dalam sistem alarm ini yang dipakai adalah nomor handphone yang terhubung dengan komputer.
Contoh : +35899000066/TYPE:PLMN

❖ *Subject*

Menunjukan judul dari pesan MMS yang akan dikirimkan.

❖ *Content type*

Menunjukan content type MMS yang akan dibuat. Ada dua macam content type yang dapat dipilih yaitu :

- *Multipart mixed* : *content type* yang digunakan untuk MMS yang tidak memerlukan presentasi.
- *Multipart related* : *content type* yang digunakan untuk MMS yang memerlukan presentasi.

Dalam sistem pemberitahuan dengan menggunakan motion detection ini, *content type* yang digunakan adalah multipart mixed. Karena yang dibutuhkan untuk sistem pemberitahuan ini hanyalah pesan MMS biasa sehingga tidak memerlukan presentasi.

3.4.2. Pengisian pesan MMS

Setelah melakukan inisialisasi awal maka yang perlu dilakukan sekarang adalah mengisi MMS message dengan content yang diinginkan. Ada beberapa macam content yang dapat dimasukan yaitu :

- Ct_image_JPE : *Content* yang berupa *image* yang mempunyai ekstensi *.JPEG!JPG.
- Ct_image_GIF : *Content* yang berupa *image* yang mempunyai ekstensi *.GIF.
- Ct_image_WBMP : *Content* yang berupa *image* yang mempunyai ekstensi *.BMP.
- Ct_image_TIFF : *Content* yang berupa *image* yang mempunyai ekstensi *.TIFF.
- Ct_image_PNG : *Content* yang berupa *image* yang mempunyai ekstcnsi *.PNG.
- Ct_text_HT`ML : *Content* yang berupa teks yang mempunyai format HTML.
- Ct_text_plain : *Content* yang berupa teks biasa.
- Ct_text_WML : *Content* yang berupa teks yang mempunyai format WML.

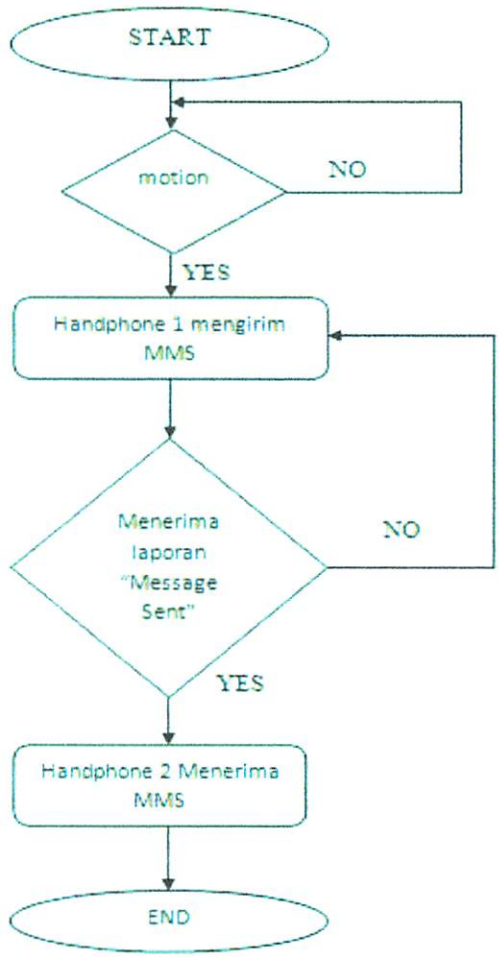
Dalam sistem peringatan keamanan rumah dengan menggunakan *motion detection* ini, *content* yang dipergunakan adalah "ct_image_JPEG". Hal ini disebabkan karena gambar *content* JPEG memiliki ukuran *filesize* yang paling kecil dibandingkan dengan *content* type yang lainnya dan besarnya *filesize* menentukan prosentasi keberhasilan proses pengiriman MMS.

3.4.3. Pengiriman Pesan MMS

Setelah MMS *message* selesai dibuat maka yang perlu dilakukan selanjutnya adalah mengirimkan pesan MMS tersebut. Berikut ini langkah - langkah proses pengiriman MMS :

- Program pengiriman MMS akan mengadakan koneksi ke WAP *gateway.*
- Setelah koneksi ke WAP *gateway* tercapai maka program pengiriman MMS akan mengirimkan MMS sebagai *content*
- Program pengiriman MMS akan memutuskan koneksi ke WAP *gateway.*
- WAP *gateway* akan mengirimkan pesan MMS tersebut ke MMSC (*Multimedia Messaging Service Center*).
- Setelah MMSC menerima pesan MMS tersebut, maka MMSC akan mengirimkan sinyal kepada pengirim. Handphone pengirim akan tercantum "Message Sent".
- MMSC akan menggunakan WAP PUSH untuk mengirimkan pesan MMS tersebut ke penerima bahwa ada pesan MMS yang baru.
- Dengan mengasumsikan bahwa *handphone* penerima sudah di *setting* untuk penerimaan pesan MMS, maka *handphone* penerima akan mengadakan sebuah koneksi ke WAP *gateway.*
- Penerima mengirimkan sinyal kepada WAP *gateway* untuk men*download* pesan MMS tesebut.
- WAP *gateway* mendownload pesan MMS dari MMSC.
- Pesan MMS dikirim kepada penerima sebagai *content* melalui koneksi WAP yang sama.
- Setelah pesan MMS selesai di download maka penerima memutuskan koneksi ke WAP gateway.
- MMSC memberitahukan kepada pengirim MMS bahwa pesan MMS tersebut sudah terkirim. Pada *handphone* pengirim akan tercantum "*Message Delivered*"

Diagram alir pengiriman *MMS* dapat dilihat pada gambar 3.7 berikut ini :



Gambar 3.7 Diagram alir pengiriman *MMS*

BAB IV
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Implementasi Sistem *Motion Detection*

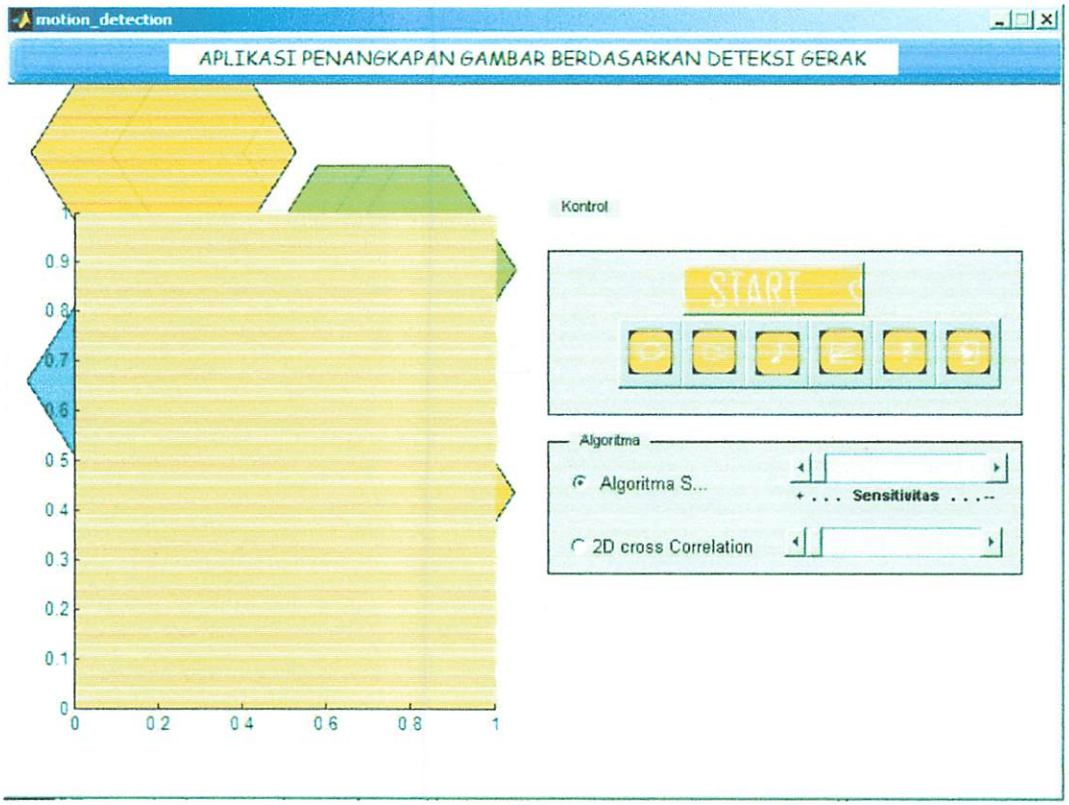
Implementasi dilakukan dengan menerapkan hasil desain yang telah dibuat kedalam bahasa pemrograman (*Coding*) Matlab 7.0.4, sehingga prosedur-prosedur yang telah dibuat dapat dimengerti oleh mesin dan menghasilkan keluaran seperti apa yang diharapkan. Berikut ini adalah perlengkapan yang digunakan dalam implementasi sistem :

Tabel 4.1 Spesifikasi Perlengkapan Implementasi

NO	Perlengkapan	Spesifikasi	Keterangan
1	*Software*	Sistem Operasi	Windows XP
		Bahasa Pemrograman	• Matlab 7.04 • NetBean IDE 6.7.1
2	*Notebook*	*Processor*	AMD Turion X2 2,2Ghz
		Memori	2,00 GB DDR2
		Hard disk	250 GB
4	Webcam Cristal Eye	Resolusi	1,3 Mega Pixels
		View Angle	58^0
		Focus Range	10cm to infinity
		Sensor Citra	640 x 480
		Lensa	F=2.4, f=4.9
		Light Source	60 Hz

4.1.1 Tampilan Program *Motion Detection*

Pada program *motion detection* ini, akan terdapat beberapa komponen yang digunakan untuk mempermudah pengguna menggunakan program *motion detection* gambar 4.1 berikut ini .



Gambar 4.1 Tampilan GUI Program Motion Detection

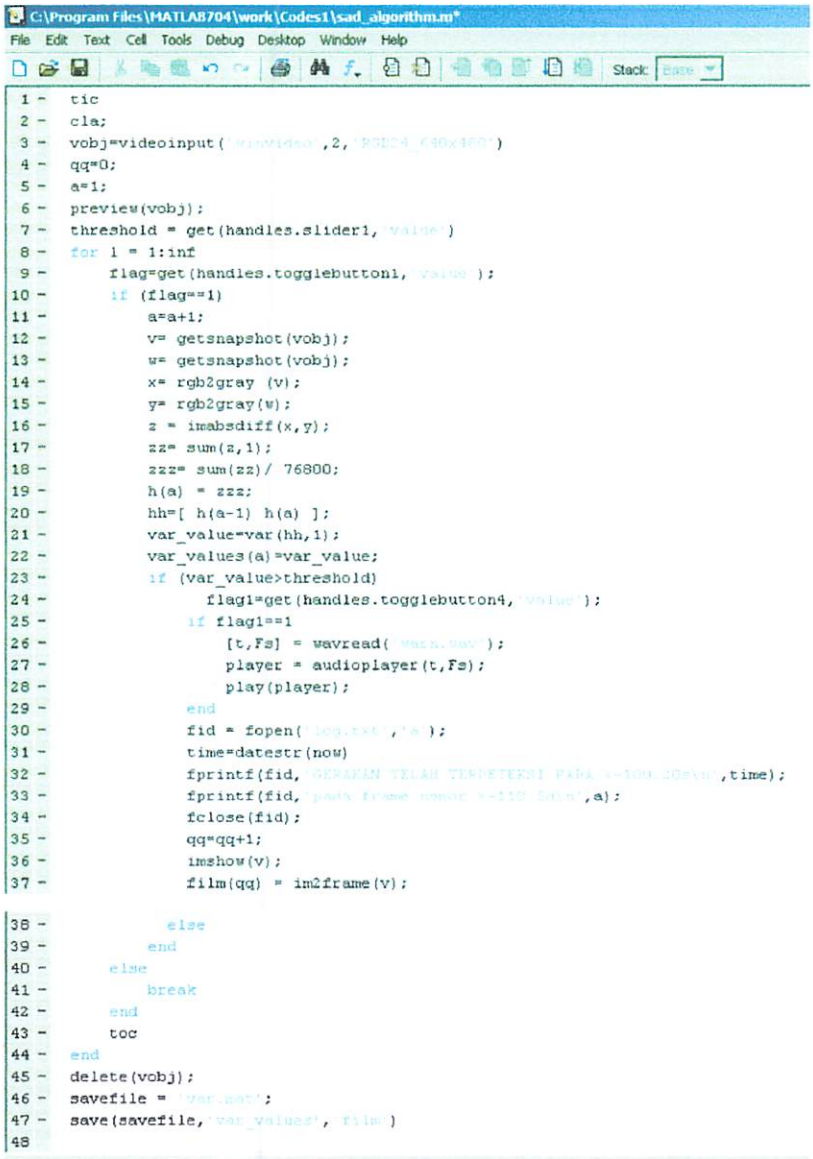
Pada program motion detection ini terdapat tiga file .m yaitu motion_detection.m yang berisi callback dari GUI dan sebagai interaksi komponen-komponen yang digunakan pada GUI program *motion detection*, serta corrolation_algorithm.m dan sad_algorithm.m dimana salah satu dari file tersebut akan dipanggil jika algoritma tersebut dipilih oleh operator pada GUI program motion detection. Berikut ini adalah listing program dari corrolation_algorithm.m dimana pada file ini berisi algoritma 2D cross correlation dapat dilihat pada gambar 4.2 berikut ini:

```
tic
cla;
vobj=videoinput( 'winvideo',2, 'RGB24_640x480' )
qq=0;
a=1;
preview(vobj);
threshold = get(handles.slider2,'value')
for 1 = 1:inf
    flag=get(handles.togglebutton1,'value');
    if (flag==1)
        a=a+1;
        v= getsnapshot(vobj);
        w= getsnapshot(vobj);
        x= rgb2gray (v);
        x1=x(1:144,1:176);
        x2=x(1:144,177:352);
        x3=x(145:288,177:352);
        x4=x(145:288,1:176);
        y= rgb2gray(w);
        y1=y(1:144,1:176);
        y2=y(1:144,177:352);
        y3=y(145:288,177:352);
        y4=y(145:288,1:176);
        %mesure of 2 dimensions cross correlation for each picture
        z= corr2(x,y);
        z1= corr2(x1,y1);
        z2= corr2(x2,y2);
        z3= corr2(x3,y3);
        z4= corr2(x4,y4);
        %put values of correlation into arrays

        h(a) = z;
        h1(a)=z1;
        h2(a)=z2;
        h3(a)=z3;
        h4(a)=z4;
        %measure the minimum value for correlation
        hh=[z1,z2,z3,z4];
        zz=min(hh);
        hmin(a)=zz;
        kk=[hmin(a-1),hmin(a) ];
        var_value=var(kk,1);
        var_values(a)=var_value;
        if (var_value > threshold)
            flag1=get(handles.togglebutton4,'value');
            if flag1==1
                [t,Fs] = wavread( 'warn.wav');
                player = audioplayer(t,Fs);
                play(player);
            end
            fid = fopen( 'log.txt' ,'a');
            time=datestr(now)
            fprintf(fid,'GERAKAN TELAH TERDETEKSI PADA %-100.30s\n',time);
            fprintf(fid, 'pada frame Nomor %-110.5d\n',a);
            fclose(fid);
            qq=qq+1;
            imshow(v);
            film(qq) = im2frame(v);
        else
        end
```

Gambar 4.2 Listing file *corrolation_algorithm.m*

Sedangkan listing program untuk algoritma sad dapat dilihat pada gambar 4.3:



```
C:\Program Files\MATLAB704\work\Codes1\sad_algorithm.m*
File  Edit  Text  Cell  Tools  Debug  Desktop  Window  Help
Stack: Base

 1 -  tic
 2 -  cla;
 3 -  vobj=videoinput( winvideo ,2, 'RGB24_640x480')
 4 -  qq=0;
 5 -  a=1;
 6 -  preview(vobj);
 7 -  threshold = get(handles.slider1, 'value')
 8 -  for i = 1:inf
 9 -      flag=get(handles.togglebutton1, 'value');
10 -      if (flag==1)
11 -          a=a+1;
12 -          v= getsnapshot(vobj);
13 -          w= getsnapshot(vobj);
14 -          x= rgb2gray (v);
15 -          y= rgb2gray(w);
16 -          z = imabsdiff(x,y);
17 -          zz= sum(z,1);
18 -          zzz= sum(zz)/ 76800;
19 -          h(a) = zzz;
20 -          hh=[ h(a-1) h(a) ];
21 -          var_value=var(hh,1);
22 -          var_values(a)=var_value;
23 -          if (var_value>threshold)
24 -                flag1=get(handles.togglebutton4, 'value');
25 -              if flag1==1
26 -                 [t,Fs] = wavread( 'mark.wav');
27 -                 player = audioplayer(t,Fs);
28 -                 play(player);
29 -              end
30 -              fid = fopen( 'log.txt' ,'a' );
31 -              time=datestr(now)
32 -              fprintf(fid,'GERAKAN TELAH TERDETEKSI PADA =>%s\r\n',time);
33 -              fprintf(fid,'pada frame nomor =>%d\r\n',a);
34 -              fclose(fid);
35 -              qq=qq+1;
36 -              imshow(v);
37 -              film(qq) = im2frame(v);

38 -            else
39 -          end
40 -      else
41 -          break
42 -      end
43 -      toc
44 -  end
45 -  delete(vobj);
46 -  savefile = 'var.mat';
47 -  save(savefile, 'var_values', 'film')
48
```

Gambar 4.3 Listing file *SAD.m*

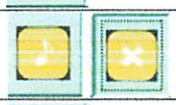
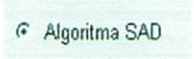
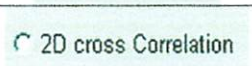
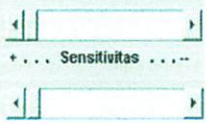
Cara kerja program *motion detection* ini, pertama-tama adalah *webcam* melakukan proses monitoring dan menampilkannya pada layar. Selanjutnya image yang berupa *color image(rgb)* diubah menjadi *grayscale*. Dan kemudian perbandingan antara kedua image dihitung berdasarkan pada algoritma yang dipilih dan hasil perhitungan ditampung pada variabel *variance value* jika *variance value* lebih besar dari nilai *threshold* yang telah ditetapkan, maka dianggap telah terjadi pergerakan. Gerakan tersebut akan disimpan menjadi beberapa gambar dan disatukan menjadi video. Jumlah gambar yang akan disimpan sesuai dengan pendeteksian gerakan yang terjadi. Dalam hal ini program *motion detection* akan terus menerus melakukan pendeteksian gerak

sampai *toggle button start/stop* ditekan. Citra yang terdeteksi akan disimpan dalam bentuk image yang kemudian dikonversikan menjadi frame dan jika dipanggil akan disatukan menjadi file video berformat .avi. Kemudian tanggal, dan jam akan disimpan ke log.txt.

4.1.2 Komponen yang Digunakan pada Program *Motion Detection*

Dalam program *motion detection* ini terdapat beberapa komponen yang telah disediakan oleh Matlab 7.0.4. Berikut ini adalah komponen-komponen yang ada pada program *motion detection* dapat dilihat pada tabel 4.2 berikut ini:

Tabel 4.2 Komponen pada Program *Motion Detection*

No.	Komponen	Gambar
1	Toggle button start	
2	Push button log	
3	Push button video	
4	Toggle button alarm	
5	Push button plot	
6	Push button help	
7	Push button exit	
8	Radion button SAD	
9	Radio button 2D Cross Corr	
10	Slider sensitivitas	

Cara kerja dari komponen-komponen diatas adalah :

- Toggle Button Start/ Stop

  Digunakan untuk memulai menjalankan program, jika ditekan satu kali maka tombol akan berfungsi seperti diatas, namun jika ditekan lagi maka tombol akan berubah tampilan menjadi Stop dan akan menghentikan program. Berikut ini adalah script dari komponen toggle button start:

- Push Button Log

```
% --- Executes on button press in togglebutton1.
function togglebutton1_Callback(hObject, eventdata, handles)
% hObject    handle to togglebutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of togglebutton1
button_state = get(hObject,'Value');
if button_state == get(hObject,'Max')
   icon5=imread('stop.png');
set(handles.togglebutton1,'CData',icon5)
   % toggle button is pressed
   delete 'log.txt'
   delete 'film.avi'
   delete 'var.mat'
   radio = get(handles.radiobutton1,'value')
if (radio==1)
   sad_algorithm
else
  corrolation_algorithm
end

elseif button_state == get(hObject,'Min')
   % toggle button is not pressed
   icon5=imread('start.png');
set(handles.togglebutton1,'CData',icon5)
   set(handles.togglebutton1, 'Value', 0);
end
```

Digunakan untuk membuka file log.txt yang menyimpan informasi berupa nomor frame, tanggal dan waktu terdeteksi gerakan. Berikut adalah script untuk push button Log :

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
open 'log.txt';
```

- Push Button Video

  Digunakan untuk memanggil citra yang berisi gerakan yang sudah dideteksi sebelumnya.

```
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 %axes('position',[.03  .25  .8  .6])
 load('var')
movie(film,1,2)
movie2avi(film,'film.avi','compression','None','fps',5);%,'fps',5)
```

- Toggle Button Alarm

  Digunakan untuk mengaktifkan alarm, jika ditekan satu kali maka tombol akan berfungsi memanggil file audio pada saat terdeteksi gerakan. Jika ditekan lagi maka tombol akan mematikan proses pemanggilan file audio (alarm tidak aktif). Berikut adalah script untuk toggle button alarm:

```
% --- Executes on button press in togglebutton4.
function togglebutton4_Callback(hObject, eventdata, handles)
% hObject    handle to togglebutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of togglebutton4
button_state = get(hObject,'Value');
if button_state == get(hObject,'Max')
icon6=imread('close.png');
set(handles.togglebutton4,'CData',icon6)

elseif button_state == get(hObject,'Min')
icon6=imread('music.png');
set(handles.togglebutton4,'CData',icon6)

end
```

- Push Button Plot

Digunakan untuk memanggil plot yang berisi grafik informasi tentang nomor frame dengan variance value. Berikut adalah script yang digunakan pada push button plot :

```
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
axes(handles.axes1);
cla;
load('var')
      semilogy(var_values);
       grid on;
       xlabel('Nomor Frame');
       ylabel('VARIANCE VALUE');
       title('Deteksi gerakan menggunakan variance');
```

- Push Button Help

  Digunakan untuk memanggil file yang berisi panduan singkat mengenai program GUI motion detection. Berikut adalah script pada push button help :

```
% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
I=imread('helpteks.png');

figure,imshow(I),title('HELP');
```

- Push Button Exit

  Digunakan untuk keluar dari program *motion detection.*

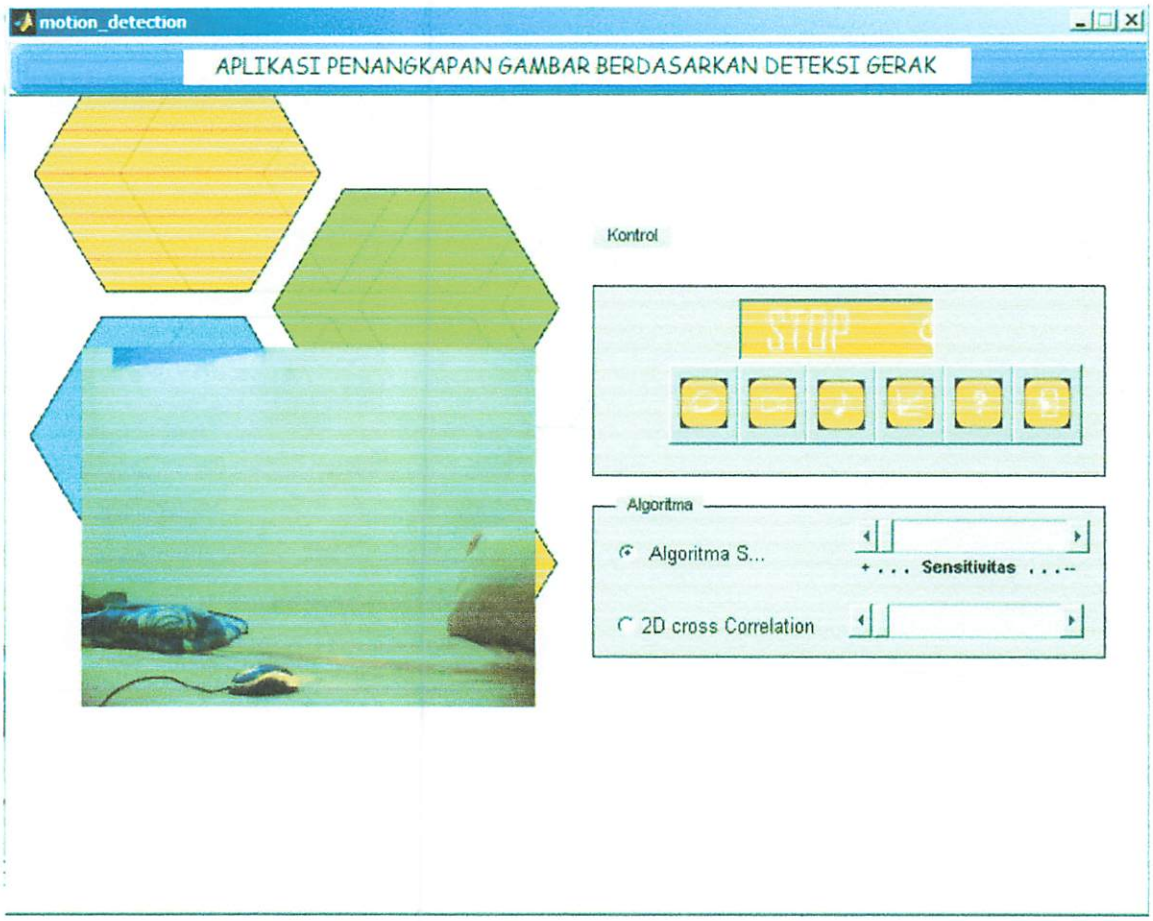
  Berikut adalah script pada push button exit:

```
% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

delete 'var.mat'
close
```

4.1.3 Hasil Implementasi

Berikut ini adalah hasil implementasi program *motion detection* yang diimplementasikan pada ruangan. Seperti yang sudah dijelaskan pada bab tiga, hal pertama yang harus dilakukan pada pengoperasian program *motion detection* adalah menentukan algoritma yang akan digunakan serta nilai sensitivitas yang digunakan, kemudian menekan tombol toggle start, berikut adalah tampilan pada saat program pertama kali dijalankan seperti gambar 4.4 berikut ini.

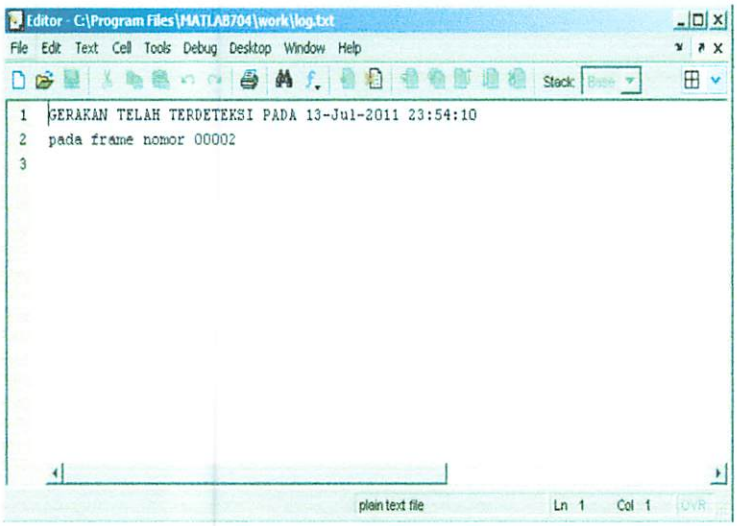


Gambar 4.4 Tampilan pada Saat Program Dijalankan

Dari tombol start yang ditekan program akan mulai menghitung, dan akan keluar tampilan preview video seperti gambar 4.5 berikut ini:

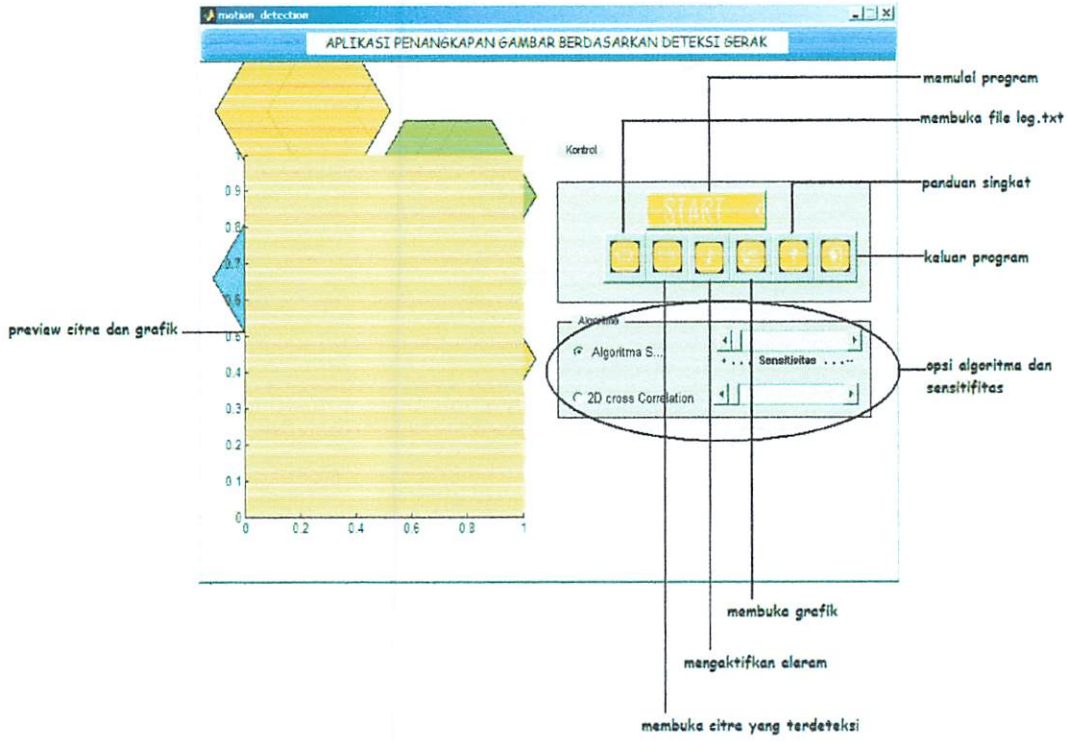
Gambar 4.5 Tampilan Preview Video

Dan pada saat push button log ditekan maka akan memanggil file log.txt dan ditampilkan pada editor Matlab, yaitu seperti gambar 4.6 berikut ini :



Gambar 4.6 Tampilan *File* log.txt

Jika push button help ditekan maka akan keluar jendela figure yang berisi tentang penjelasan singkat keterangan program yang dapat membantu operator untu memahami program, seperti gambar 4.7 berikut ini :



Gambar 4.7 Tampilan *Figure Help*

Jika tombol push button video ditekan maka akan muncul tampilan yang berupa kumpulan citra yang terdeteksi gerakan pada komponen axes, yaitu :

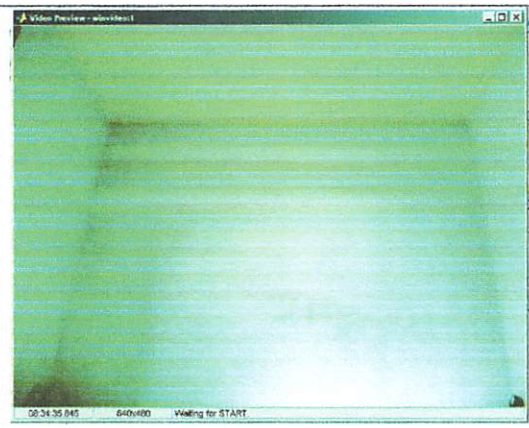
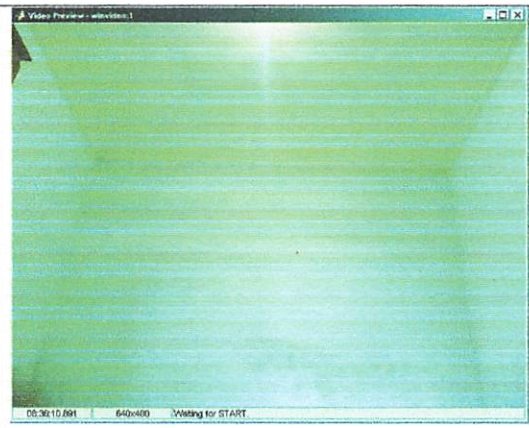
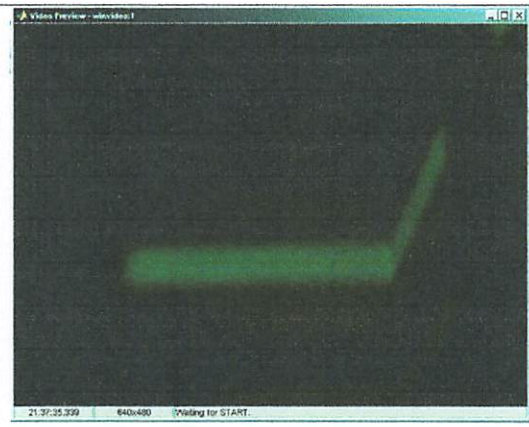
4.2. Pengujian Sistem *Motion Detection*

Pengujian ini dilakukan untuk mengetahui kinerja program motion detection beserta algoritma yang digunakan serta pengujian pada suatu kondisi yang dapat berdampan pada pendeteksian objek pada suatu ruangan yang dipantau.

4.2.1 Pengujian Program Menggunakan Algoritma *Sum Of Absolute Difference*

Dalam pengujian ini dilakukan suatu aktivitas pendeteksian gerakan pada suatu ruangan dengan beberapa kondisi. Dimana pengujian tersebut dilakukan dengan memonitoring secara kontinyu suatu ruangan sehingga didapatkan suatu informasi yang relevan mengenai kondisi atau sistem maupun aplikasi yang berjalan. Pengujian ini dilakukan dengan 4 kondisi ruangan dilihat dari pencahayaan. 4 kondisi ruangan dapat dilihat pada tabel 4.3 berikut ini:

Tabel 4.3 Kondisi ruangan berdasarkan pada pencahayaan

KONDISI	CITRA
SIANG TANPA LAMPU	
SIANG DENGAN LAMPU	
MALAM TANPA LAMPU	

MALAM DENGAN LAMPU	

Hasil pengujian algoritma ini dapat dilihat pada tabel 4.4 berikut ini:

Tabel 4.4 Pengujian Algoritma *Sum of Absolute Difference* berdasarkan pencahayaan

No.	Kondisi	*Sensitivity*	Hasil Deteksi	
			Alarm	Penyimpanan
1	Siang dengan lampu	1	√	√
2	Siang tanpa lampu	1	√	√
3	Malam dengan lampu	1	√	√
4	Malam tanpa lampu	1	-	-
5	Siang dengan lampu	1000	√	√
6	Siang tanpa lampu	1000	√	√
7	Malam dengan lampu	1000	√	√
8	Malam tanpa lampu	1000	-	-

Tabel 4.4 menunjukkan bahwa pada pengujian algoritma *sum of absolute difference* dengan sensitivitas 1 dan 1000 dapat disimpulkan bahwa pada kondisi siang dengan dan tanpa lampu, serta malam dengan lampu gerakan terdeteksi dengan baik sehingga penyimpanan video dan alarm dapat berjalan dengan baik. Sedangkan pada pengujian malam tanpa lampu dan malam dengan lampu dengan sensitivitas 1000 gerakan tidak terdeteksi, sehingga pada algoritma ini pencahayaan berpengaruh terhadap pendeteksian gerakan.

Berikut ini adalah beberapa hasil dari pendeteksian gerakan menggunakan algoritma *sum of absolute difference* pada malam hari dengan lampu:

Frame 1

Frame 2

Frame 3

Gambar 4.8 Hasil deteksi pada kondisi siang tanpa lampu menggunakan SAD

Pada pengujian selanjutnya, dilakukan pengujian algoritma sum of absolute difference berdasarkan pada kecepatan gerakan. Pembagian lama gerakan ini diperoleh dengan cara menghitung jumlah frame dibagi dengan frame per detik dari kecepatan penyimpanan video dimana *frame rate* video adalah 30 fps, dapat dilihat pada tabel 4.5 berikut ini:

Tabel 4.5 Kondisi pengujian berdasarkan kecepatan gerakan

KONDISI GERAKAN	LAMA GERAKAN
Cepat	5/30=0,167 detik
Normal	30/30=1 detik
Lambat	126/30=4,2 detik

Hasil dari pengujian algoritma berdasarkan pada lama gerakan direpresentasikan dapat dilihat pada tabel 4.6 berikut ini :

Tabel 4.6 Pengujian algoritma *Sum of Absolute Difference* berdasarkan kecepatan gerakan

No.	Kondisi gerakan	Hasil Deteksi	
		ALARM	PENYIMPANAN
1	Normal	√	√
2	Lambat	-	-
3	Cepat	√	√

Dari hasil pengujian diatas, gerakan yang tidak terdeteksi adalah gerakan lambat dengan lama gerakan 4,2 detik. Sedangkan gerakan normal dan cepat terdeteksi dengan baik, sehingga algoritma ini sesuai untuk mendeteksi gerakan normal dan cepat.

4.2.2 Pengujian Program Menggunakan Algoritma 2 Dimensional Cross Correlation

Untuk pengujian dengan menggunakan algoritma 2D *cross correlation* dapat direpresentasikan pada tabel 4.7 dibawah ini :

Tabel 4.7 Pengujian Algoritma *2D Cross Correlation* dengan Sensitivitas Tinggi

No.	Kondisi	*Sensitivity*	Hasil Deteksi	
			Alarm	Penyimpanan
1	Siang dengan lampu	0,0001	√	√
2	Siang tanpa lampu	0,0001	√	√

3	Malam dengan lampu	0,0001	√	√
4	Malam tanpa lampu	0,0001	-	-
5	Siang dengan lampu	0,01	√	√
6	Siang tanpa lampu	0,01	√	√
7	Malam dengan lampu	0,01	√	√
8	Malam tanpa lampu	0,01	-	-

Pada tabel 4.7 diatas terlihat bahwa gerakan dapat terdeteksi pada saat tertentu, yaitu pada kondisi siang tanpa lampu, malam dengan lampu, malam tanpa lampu pada sensitivitas 0,0001. Serta pada kondisi siang tanpa lampu, malam dengan lampu dan malam tanpa lampu dengan sensitivitas 0,01. Pada saat kondisi ruangan tanpa lampu, program mendeteksi adanya gerakan, hal ini dikarenakan adanya gangguan *noise* yang dideteksi program sebagai gerakan. Sehingga diperlukan pengaturan sensitivitas yang sesuai.

Berikut ini adalah beberapa hasil dari pendeteksian gerakan menggunakan algoritma *two domensional cross correlation* pada malam hari dengan lampu:

Frame 1

Frame 2

Frame 3

Gambar 4.9 Hasil deteksi pada kondisi siang dengan lampu menggunakan *2D cross correlation*

Pada pengujian selanjutnya, dilakukan pengujian algoritma two dimensional cross correlation berdasarkan pada kecepatan gerakan. Pembagian lama gerakan ini diperoleh dengan cara menghitung jumlah frame dibagi dengan frame per detik dari kecepatan penyimpanan video dimana kecepatan video adalah 30 fps, dapat dilihat pada tabel 4.8 berikut ini:

Tabel 4.8 Kondisi pengujian berdasarkan kecepatan gerakan

KONDISI GERAKAN	LAMA GERAKAN
Cepat	4/30=0,13 detik
Normal	8/30=0,267 detik
Lambat	78/30=2,6 detik

Hasil dari pengujian algoritma berdasarkan pada kecepatan gerakan direpresentasikan pada tabel 4.9 berikut :

Tabel 4.9 Pengujian algoritma *2D Cross Correlation* berdasarkan kecepatan gerakan

No.	Kondisi gerakan	Hasil Deteksi	
		ALARM	PENYIMPANAN
1	Normal	√	√

2	Lambat	√	√
3	Cepat	√	√

Dari hasil pengujian diatas, gerakan yang terdeteksi adalah semua gerakan baik gerakan normal dengan lama gerakan 0,267 detik, lambat dengan lama gerakan 2,6 detik dan cepat dengan lama gerakan 0,13 detik.

4.2.3 Pengujian Perbedaan Waktu Antar 2 Algoritma

Untuk pengujian berdasarkan perbedaan waktu antar 2 Algoritma dapat direpresentasikan pada tabel 4.10 dibawah ini :

Tabel 4.10 Pengujian berdasarkan perbedaan waktu antar 2 Algoritma

No.	Algoritma yang digunakan	Waktu yang dibutuhkan untuk menangkap deteksi gerak	Waktu yang dibutuhkan untuk mengirimkan MMS
1.	*Sum of Absolute Difference*	0,256 detik	± 30 detik
2.	*2 Dimensional Cross Correlation*	0.176 detik	± 30 detik

Dari hasil pengujian di atas dapat disimpulkan bahwa dalam perhitungan waktu yang dibutuhkan untuk menangkap deteksi gerak Algoritma *2 Dimensional Cross Correlation* lebih cepat menangkap deteksi gerak dibandingkan dengan Algortma *Sum of Absolute Difference,* sedangkan dalam perhitungan waktu yang dibutuhkan untuk mengirimkan MMS keduanya sama yakni ± 30 detik,lama tidaknya waktu yang dibutuhkan untuk mengirimkan MMS tergantung dari kualitas jaringan yang digunakan untuk mengirim MMS tersebut.

4.3 Pengujian Sistem MMS

4.3.1 *Class* Dan *Interface* yang Digunakan Pada Program Pengiriman MMS

Ada dua macam package yang di gunakan pada program pengiriman MMS yaitu :

- MMS java library : Digunakan untuk membuat kerangka MMS. Melakukan proses encode, decode, dan pengisian content pada kerangka MMS.
- JWAP protocol stack : Digunakan untuk pengiriman MMS ke WAP gateway.

4.3.1.1 MMS *Java Library*

MMS java Library mempunyai tujuh buah class yaitu :

- MMAddress : Bertujuan untuk merepresentasikan generic address dari pengirim dan penerima MMS.
- MMContent : Bertujuan untuk merepresentasikan generic entry dari suatu pesan MMS
- MMDecoder : Bertujuan untuk melakukan proses decode pada pesan MMS yang dibuat.
- MMEncoder : Bertujuan untuk melakukan proses encode pada pesan MMS yang dibuat.
- MMMessage : Bertujuan untuk merepresentasikan suatu MMS.
- MMResponse : Berisi method untuk mendapatkan respon dari MMSC.
- MMSender : Berisi method untuk mengirimkan MMS kepada MMSC. Selain mempunyai interface dan class, juga mempunyai exception yaitu :
- MMDecoderException : Bertujuan untuk mengetahui error yang terjadi pada waktu proses decode suatu pesan MMS.
- MMEncoderException : Bertujuan untuk mengetahui error yang terjadi pada waktu proses encode suatu pesan MMS.
- MMSenderExeeption : Bertujuan untuk mengetahui error yang terjadi pada waktu proses pengiriman suatu pesan MMS.

4.3.1.2 *JWAP Protocol Stack*

Dalam *JWAP prorocol stack* terdiri dari tujuh buah package yaitu

- net.sourceforge.jwap
- net.sourceforge.jwap.util
- net.sourceforge.jwap.wsp
- net.sourceforge.jwap.wsp.pdu
- net.sourceforge.jwap.wtp
- net.sourceforge.jwap.wtp.pdu

4.4. Cara Kerja Program Pengiriman MMS

Cara kerja program pengiriman MMS meliputi empat tahap yaitu menentukan nomor tujuan MMS,pembuatan *temporary file*,konfirmasi pengiriman pesan MMS, dan

connect to server untuk upload MMS. Berikut adalah penjelasan dari keempat tahap tersebut :

4.4.1. Menentukan Nomor Tujuan MMS

Tahap ini digunakan untuk menentukan nomor tujuan MMS ,yang akan dikirimkan yang digunakan dalam program pengiriman MMS. Lisring program menentukan nomor tujuan MMS, dapat dilihat pada gambar 4.10 berikut :

```
static String path;
static String username =
static String hpAdmin  =
static String hpTujuan =

static String password =
static String web =

static String server =                                    +web+
public static class mms {
```

Gambar 4.10 *Listing* Program menentukan nomor tujuan MMS

4.4.2 Pembuatan *Temporary File*

Tahap ini digunakan untuk pembuatan *temporary file*, Lisring program pembuatan *temporary file* dapat dilihat pada gambar 4.11 berikut :

```
private static void copyfile(String srFile, String dtFile){
    try
    {

            File f1 = new File(srFile);
            File f2 = new File(dtFile);
            InputStream in = new FileInputStream(f1);


            OutputStream out = new FileOutputStream(f2);

            byte[] buf = new byte[1024];
            int len;
            while ((len = in.read(buf)) > 0){
                    out.write(buf, 0, len);
            }
            in.close();
            out.close();
            System.out.println("File copied.");
    }catch(FileNotFoundException ex){
            System.out.println(ex.getMessage() + " in the specified directory.");
            System.exit(0);
    }
            catch(IOException e){
            System.out.println(e.getMessage());
    }
}
```

Gambar 4.11 *Listing* Program pembuatan *temporary file* MMS

4.4.3 Konfirmas Pengiriman Pesan MMS

Tahap ini digunakan untuk konfirmasi pengiriman pesan MMS, Lisring program konfirmasi pengiriman pesan MMS dapat dilihat pada gambar 4.12 berikut :

```
    try {
        Runtime rt = Runtime.getRuntime();

        Process pr = rt.exec(app);

        BufferedReader input = new BufferedReader(new InputStreamReader(pr.getInputStream()));
        String line = null;
        while ((line = input.readLine()) != null) {
            System.out.println(line);
        }
        int exitVal = pr.waitFor();
        return "Exited with error code " + exitVal ;
    } catch ( Exception ex) {

        Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
        return null;
    }

}
```

Gambar 4.12 *Listing* Program konfirmasi pengiriman pesan MMS

4.4.4 *Upload* MMS

Tahap ini digunakan untuk connect to server untuk upload MMS, Lisring program connect to server untuk upload MMS dapat dilihat pada gambar 4.13 berikut

```
    public boolean sending(String s){

        try {
            File f = new File(path);
            String urlString = "http://" + username + ":" + password + "@" + server + "/" + f.getName();
            URL url = new URL(urlString);
            URLConnection connection = url.openConnection();
            connection.setDoOutput(true);
            BufferedOutputStream out = new BufferedOutputStream(connection.getOutputStream());
            FileInputStream in = new FileInputStream(f);
            byte[] buffer = new byte[1024];

            int i = 0;
            while ((i = in.read(buffer)) >= 0) {
                out.write(buffer, 0, i);
            }
            out.close();
            in.close();
            System.out.println("File Sesa. : "+s);
            return true;
        }catch (Exception e) {

            System.out.println(e.getMessage());
            return false;
        }


    }
}
```

Gambar 4.13 *Listing* Program connect to server untuk upload MMS

4.5 Hasil Pengujian Pengiriman MMS

Secara keseluruhan, pengujian sistem pada sistem Sistem peringatan dini untuk rumah dengan menggunakan *motion detection* melalui *Multimedia Messaging Service* (MMS) berjalan dengan cukup baik. Berdasarkan pengujian sistem, kualitas koneksi GPRS di Indonesia masih kurang baik, terutama jika melalui lintas operator.

Sedangkan pada pengujian *filesize* diuji hubungan antara besamya data MMS yang dikirimkan dengan prosentase keberhasilan pengiriman MMS. Ada tiga buah variasi *filesize* yang dikirimkan dengan menggunakan kartu selular dari operator yang sama yaitu :

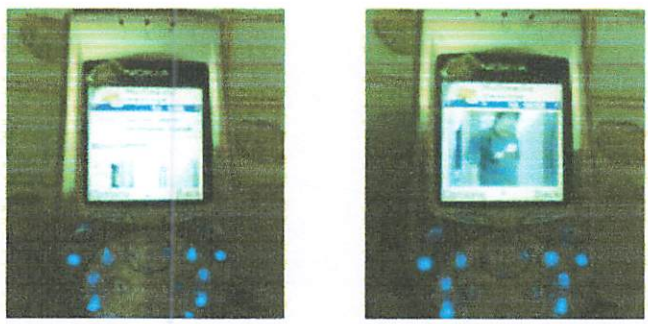
1. Lebih kecil dari 10 KB
2. Antara 10KB – 30 KB
3. Lebih besar dari 30 KB

Tabel untuk hasil pengujian *filesize* dapat dilihat pada tabel 4.11 berikut :

Tabel 4.11 Hasil Pengiriman *MMS* berdasarkan *filesize* Citra

Filesize	Jumlah Pengujian	Hasil
< 10 KB	5 Kali	5 Kali Berhasil
10 KB - 30 KB	5 Kali	5 Kali Berhasil
30 KB – 60 KB	5 Kali	2 Kali Berhasil

Berikut ini Gambar 4.14 Hasil Pengiriman *MMS* yang berhasil



Gambar 4.14 Hasil Pengiriman *MMS*

BAB V
PENUTUP

Dari beberapa uraian yang telah dikemukakan pada bab-bab sebelumnya, dapat diambil kesimpulan dan saran sebagai berikut :

5.1 Kesimpulan

Berdasarkan hasil pengujian yang dilakukan pada proyek akhir ini, maka disimpulkan bahwa :

1. Sistem peringatan dini untuk rumah dengan menggunakan *motion detection* dapat mendeteksi adanya gerakan dengan baik. Jika lngin menangkap gerakan yang cepat maka *Sensitivity* yang ada dapat diturunkan sesual dengan kebutuhan. Dalam hal ini kondisi ruangan sangat rnenentukan nilal *sensitivity* yang tepat. Dari hasil pengujian sistem, dengan nilai *sensitivity*, sistem alarm keamanan rumah sudah dapat mendeteksi adanya pergerakan dengan baik.
2. Sistem peringatan dini untuk rumah dengan menggunakan motion detection melalul *Multimedia Messaging Service* dapat mengirimkan *MMS* dengan tujuan handphone dengan menggunakan fasliltas *GPRS* melalui koneksl dengan *MMS o*perator sebagai access poinnya.
3. Kualitas jaringan lnfrastruktur layanan *GPRS dan MMS* masih kurang, terutama layanan antar operator. Sistem peringatan dini untuk rumah dengan menggunakan *motion detection* melalul *Multimedia Messaging Service* cocok digunakan di Indonesia, namun disarankan dalam penggunaan program pengiriman *MMS* user menggunakan 2 kartu selular dari operator yang sama. Karena keberhasilan dari pengiriman *MMS* dengan menggunakan kartu selular dari operator yang sama tingkat keberhasilannya cukup tinggi.
4. Berhasil tidaknya pengiriman *MMS* juga ditentukan oleh besar kecilnya size data yang dikirim. Dalam pengujian *filesize* yang dilakukan, besar *filesize* yang cocok untuk pengiriman *MMS* adalah kurang dari 30 kb.
5. *Filesize* yang cocok untuk pengiriman MMS yaitu < 30 kb karena *filezise* sangat berpengaruh besar terhadap keberhasilan MMS.

6. Apabila *Filezise* citra yang dikirim lebih dari 30 kb (> 30 kb) maka tingkat keberhasilan MMS juga kecil dan waktu yang di perlukan untuk mengirim MMS juga lebih lama dibandingkan mengirim MMS dengan ukuran *filesize* yang lebih kecil.
7. Kualitas webcam sangat dipengaruhi oleh kualitas pencahayaan. Jika pada malam hari tanpa menggunakan lampu, sistem alarm keamanan rumah tidak dapat bekerja secara maksimal. Namun dengan menggunakan lampu. Sistem alarm sudah dapat bekerja dengan baik.
8. Dalam perhitungan waktu yang dibutuhkan untuk menangkap deteksi gerak Algoritma *2 Dimensional Cross Correlation* lebih cepat menangkap deteksi gerak dibandingkan dengan Algortma *Sum of Absolute Difference.*
9. Dalam perhitungan waktu yang dibutuhkan untuk mengirimkan MMS keduanya sama yakni ± 30 detik,lama tidaknya waktu yang dibutuhkan untuk mengirimkan MMS tergantung dari kualitas jaringan yang digunakan untuk mengirim MMS tersebut.

5.2 Saran

Setelah melakukan evaluasi terhadap sistem secara keseluruhan, penulis berharap skripsi ini dapat dikembangkan lebih lanjut dengan saran-saran pengembangan sebagai berikut :

1. Jika kualitas layanan *MMS dan GPRS* sudah memadai. data yang dikirimkan tidak hanya berupa teks dan gambar saja melainkan clip video dan suara.
2. Sistem peringatan dini untuk rumah dengan menggunakan *motion detection* melalui *Multimedia Messaging Service* dapat digabungkan dengan peralatan lainnya yang mendukung sistem keamanan. Seperti sensor pintu. inframerah, dan lainnya.
3. Sistem peringatan dini untuk rumah dengan menggunakan *motion detection* melalui *Multimedia Messaging Service* dapat dibuat menggunakan bahasa pemrograman lainya, contohnya Visual Basic, atau Delphi, agar Sistem peringatan dini untuk rumah dengan menggunakan *motion detection* melalui *Multimedia Messaging Service* dapat dijalankan di komputer manapun tanpa harus menginstal software dari bahasa pemrograman yang di gunakan.

DAFTAR PUSTAKA

[1] http://forums.netbeans.org/topic34896.html
Diakses pada tanggal 4 Juli 2011, 13.45 WIB

[2] http://www.java2s.com/Open-Source/Java-Document/IDE-Netbeans/mobility/example/mms/MMSSend.java.diagram.htm
Diakses pada tanggal 6 Juli 2011, 20.00 WIB

[3] http://www.flickr.com/photos/7702002@N08/3103830956/lightbox/#/photos/ethanhein/3103830956/
Diakses pada tanggal 12 Juli 2011, 16:24 WIB

[4] http://repository.usu.ac.id/bitstream/123456789/16876/4/Chapter%20II.pdf
Diakses pada tanggal 12 Juli 2011, 14:30 WIB

[5] Winarno, Edi., 2009, Pengolahan Citra, FTI-UNISBANK, Semarang.

[6] http://www.mathworks.com/access/helpdesk/help/techdoc/creating_plots/f2-1667.html
Diakses pada tanggal 12 Juli 2011, 14:40 WIB

[7] Isa, Sani M., Lauro, Manatap Dolok., 2006, *Aplikasi Pendeteksi Gerakan Menggunakan Metode Spatial Domain dengan Pelaporan Otomatis Ke Telepon Genggam*, Seminar Nasional Sistem dan Informatika, Taruamnegara, Jakarta.

[8] http://en.wikipedia.org/wiki/Sum_of_absolute_differences
Diakses pada tanggal 3 Agustus 2011, 13:16 WIB

[9] S. Sulaiman, A. Hussain, N. Tahir, S.A. Samad and M.M. Mustafa.,2008, *Human Silhouette Extraction Using Background Modeling and Subtraction Techniques*, Inform. Technol. J. 7: 155-159.

[10] http://local.wasp.uwa.edu.au/~pbourke/miscellaneous/correlate/index.html
Diakses pada tanggal 3 Agustus 2011, 12:59 WIB

[11] http://stahlberglab.ucdavis.edu/teaching/dip/stahlberg-correlation.pdf
Diakses pada tanggal 12 Agustus 2011, 14:50 WIB

[12] Away, Gunaidi Abdia., 2009, *The Shortcut of MATLAB Programming*, Informatika, Bandung.

L A M P I R A N

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK
PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG
Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

NAMA : YOSIA EKA SATYA SURYA S
NIM : 0612669
JURUSAN : Teknik Elektro S-1
KONSENTRASI : Teknik Komputer dan Informatika
JUDUL : **SISTEM PEMBERTAHUAN DINI UNTUK RUMAH MENGGUNAKAN *MOTION DETECTION* MELALUI *MULTIMEDIA MESSAGING SERVICE* (MMS)**

Dipertahankan dihadapan Tim Pengujian Skripsi jenjang Program Strata Satu (S-1) pada

Hari : Sabtu
Tanggal : 13 Agustus 2011
Dengan Nilai : A (85,5)

PANITIA UJIAN SKRIPSI

KETUA,

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.1018800189

SEKRETARIS,

Dr. Eng. Aryuanto S, ST, MT
NIP.Y.1030800417

ANGGOTA PENGUJI

PENGUJI I,

M.Ibrahim Ashari,ST,MT
NIP.P.1030100358

PENGUJI II,

Ahmad Faisol,ST
NIP.P.1031000431

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
Jl. Karanglo KM.2 Malang

FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata 1 Jurusan Teknik Elektro Konsentrasi Teknik Komputer & Inforamtika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

Nama	:	YOSIA EKA SATYA SURYA S
Nim	:	06.12.669
Jurusan	:	Teknik Elektro S-1
Konsentrasi	:	Teknik Komputer & Informatika
Judul	:	**SISTEM PEMBERTAHUAN DINI UNTUK RUMAH MENGGUNAKAN *MOTION DETECTION* MELALUI *MULTIMEDIA MESSAGING SERVICE* (MMS)**

No.	Penguji	Tanggal	Uraian	Paraf
1	Penguji I	13/08/2011	Tambahkan keterangan pada gambar dan table, pengujian dicoca berulang kali	
2	Penguji II	13/08/2011	Tambahkan pengujian perbedaan waktu antar 2 algoritma	
			Hasil pengujian masukan ke kesimpulan	
			Tambahkan saran untuk pengembangan	
			Tambahkan text pada MMS (jangan hanya gambar)	

Disetujui :

Penguji I

M. Ibrahim Ashari, ST. MT
NIP.P.1030100358

Penguji II

Ahmad Faisol, ST
NIP.P.1031000431

Mengetahui :

Dosen Pembimbing I

Dr. Eng. Aryuanto Soetedjo, ST. MT
NIP.Y.1030800417

INSTITUT TEKNOLOGI NASIONAL
Jl. Raya Karanglo 10 KM
MALANG

FORMULIR BIMBINGAN SKRIPSI

Nama : YOSIA EKA SATYA SURYA
NIM : 06.12.669
Masa Bimbingan : 31 Januari 2011 s/d 31 Juli 2011
Judul Skripsi : SISTEM PERINGATAN DINI UNTUK RUMAH DENGAN MENGGUNAKAN MOTION DETECTION MELALUI MULTIMEDIA MESSAGING SERVICE (MMS)

No	Tanggal	Uraian	Paraf Pembimbing
1	05/7 2011	Pengajuan Laporan Skripsi BAB I	
2	05/7 2011	Pengajuan Laporan Skripsi BAB II	
3	06/7 2011	Pengajuan Laporan Skripsi BAB III	
4	06/7 2011	Pengajuan Laporan Skripsi BAB IV	
5	06/7 2011	Pengajuan Laporan Skripsi BAB V	
6			
7			
8			
9			
10			

Malang,
Dosen Pembimbing

Dr. Aryuanto, ST, MT
NIP. Y. 1030800417

Form S-4b

- **Motion detection.m**

```
function varargout = motion_detection(varargin)
% MOTION_DETECTION M-file for motion_detection.fig
%      MOTION_DETECTION, by itself, creates a new MOTION_DETECTION or raises the
existing
%      singleton*.
%
%      H = MOTION_DETECTION returns the handle to a new MOTION_DETECTION or the
handle to
%      the existing singleton*.
%
%      MOTION_DETECTION('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in MOTION_DETECTION.M with the given input arguments.
%
%      MOTION_DETECTION('Property','Value',...) creates a new MOTION_DETECTION or
raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before motion_detection_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to motion_detection_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help motion_detection

% Last Modified by GUIDE v2.5 05-Jul-20101 20:35:43

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @motion_detection_OpeningFcn, ...
                   'gui_OutputFcn',  @motion_detection_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
```

```
end

if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
  % --- Executes just before motion_detection is made visible.
function motion_detection_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to motion_detection (see VARARGIN)
 % Choose default command line output for motion_detection
handles.output = hObject;
 % Update handles structure
guidata(hObject, handles);
 % UIWAIT makes motion_detection wait for user response (see UIRESUME)
% uiwait(handles.figure1);
 icon1=imread('door.png');
icon2=imread('charts01.png');
icon3=imread('object_14.png');
icon4=imread('log.png');
icon5=imread('start.png');
icon6=imread('music.png');
icon7=imread('question.png');
icon8=imread('header.png');
set(handles.pushbutton8,'CData',icon8)
set(handles.pushbutton4,'CData',icon1)
set(handles.pushbutton3,'CData',icon2)
set(handles.pushbutton2,'CData',icon3)
set(handles.pushbutton1,'CData',icon4)
set(handles.pushbutton7,'CData',icon7)
set(handles.togglebutton1,'CData',icon5)
set(handles.togglebutton4,'CData',icon6)
 % Create background axes and move them to the background
hback = axes('units','normalized','position',[0 0 1 1]);
uistack(hback,'bottom');
```

```
% Load background image and display it
[back map]=imread('2.jpg');
image(back)
colormap(map)
% Turn the handlevisibility off so that we don't inadvertently plot into
% the axes again. Also, make the axes invisible
set(hback,'handlevisibility','off','visible','off')
 % --- Outputs from this function are returned to the command line.
function varargout = motion_detection_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 % Get default command line output from handles structure
varargout{1} = handles.output;
 % --- Executes on button press in togglebutton1.
function togglebutton1_Callback(hObject, eventdata, handles)
% hObject    handle to togglebutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 % Hint: get(hObject,'Value') returns toggle state of togglebutton1
button_state = get(hObject,'Value');
if button_state == get(hObject,'Max')
   icon5=imread('stop.png');
set(handles.togglebutton1,'CData',icon5)
   % toggle button is pressed
   delete 'log.txt'
   delete 'film.avi'
   delete 'var.mat'
   radio = get(handles.radiobutton1,'value')
if (radio==1)
   sad_algorithm
else
  corrolation_algorithm
end

elseif button_state == get(hObject,'Min')
   % toggle button is not pressed
   icon5=imread('start.png');
set(handles.togglebutton1,'CData',icon5)
```

```
   set(handles.togglebutton1, 'Value', 0);
end
 % --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
open 'log.txt';
 % --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 %axes('position',[.03  .25  .8  .6])
 load('var')
movie(film,1,2)
movie2avi(film,'film.avi','compression','None','fps',5);%,'fps',5)
 % --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
axes(handles.axes1);
cla;
load('var')
      semilogy(var_values);
       grid on;
       xlabel('Nomor Frame');
       ylabel('VARIANCE VALUE');
       title('Deteksi gerakan menggunakan variance');

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
delete 'var.mat'
close
% --- Executes on button press in radiobutton1.
function radiobutton1_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to radiobutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 % Hint: get(hObject,'Value') returns toggle state of radiobutton1
set(handles.radiobutton1, 'Value', 1);
set(handles.radiobutton2, 'Value', 0);

% --- Executes on button press in radiobutton2.
function radiobutton2_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton2
set(handles.radiobutton1, 'Value', 0);
set(handles.radiobutton2, 'Value', 1);

% --- Executes during object creation, after setting all properties.
function slider1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background, change
%       'usewhitebg' to 0 to use default.  See ISPC and COMPUTER.
usewhitebg = 1;
if usewhitebg
   set(hObject,'BackgroundColor',[.9 .9 .9]);
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes on slider movement.
function slider1_Callback(hObject, eventdata, handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
```

```
%       get(hObject,'Min') and get(hObject,'Max') to determine range of slider

%t = get(handles.slider1,'value');
%set(handles.valText1,'String',num2str(t));

t = get(handles.slider1,'value');
set(handles.valText1,'String',num2str(t));

% --- Executes during object creation, after setting all properties.
function slider2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background, change
%       'usewhitebg' to 0 to use default.  See ISPC and COMPUTER.
usewhitebg = 1;
if usewhitebg
   set(hObject,'BackgroundColor',[.9 .9 .9]);
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

% --- Executes on slider movement.
function slider2_Callback(hObject, eventdata, handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%       get(hObject,'Min') and get(hObject,'Max') to determine range of slider

%t = get(handles.slider2,'value');
%set(handles.valText2,'String',num2str(t));

t = get(handles.slider2,'value');
set(handles.valText2,'String',num2str(t));
% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton8 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in togglebutton4.
function togglebutton4_Callback(hObject, eventdata, handles)
% hObject    handle to togglebutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of togglebutton4
button_state = get(hObject,'Value');
if button_state == get(hObject,'Max')
icon6=imread('close.png');
set(handles.togglebutton4,'CData',icon6)

elseif button_state == get(hObject,'Min')
icon6=imread('music.png');
set(handles.togglebutton4,'CData',icon6)
end

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
I=imread('helpteks.png');

figure,imshow(I),title('HELP');
```

- **SAD.m**

```
tic
cla;
vobj=videoinput('winvideo',2,'RGB24_640x480')
qq=0;
a=1;
preview(vobj);
threshold = get(handles.slider1,'value')
for I = 1:inf
   flag=get(handles.togglebutton1,'value');
   if (flag==1)
      a=a+1;
      v= getsnapshot(vobj);
      w= getsnapshot(vobj);
      x= rgb2gray (v);
      y= rgb2gray(w);
      z = imabsdiff(x,y);
      zz= sum(z,1);
      zzz= sum(zz)/ 76800;
      h(a) = zzz;
      hh=[ h(a-1) h(a) ];
      var_value=var(hh,1);
      var_values(a)=var_value;
      if (var_value>threshold)
          flag1=get(handles.togglebutton4,'value');
        if flag1==1
           [t,Fs] = wavread('warn.wav');
           player = audioplayer(t,Fs);
           play(player);
        end
        fid = fopen('log.txt','a');
        time=datestr(now)
        fprintf(fid,'GERAKAN TELAH TERDETEKSI PADA %-100.20s\n',time);
        fprintf(fid,'pada frame nomor %-110.5d\n',a);
        fclose(fid);
        qq=qq+1;
        imshow(v);
        film(qq) = im2frame(v);
       else
```

```
    end
  else
    break
  end
  toc
end
delete(vobj);
savefile = 'var.mat';
save(savefile,'var_values','film')
```

- **2D Corrolation_algorithm.m**

```
tic
cla;
vobj=videoinput('winvideo',2,'RGB24_640x480')
qq=0;
a=1;
preview(vobj);
threshold = get(handles.slider2,'value')
for l = 1:inf
   flag=get(handles.togglebutton1,'value');
   if (flag==1)
      a=a+1;
      v= getsnapshot(vobj);
      w= getsnapshot(vobj);
      x= rgb2gray (v);
      %dividing a 240 by 320 into 4 pictures
      x1=x(1:144,1:176);
      x2=x(1:144,177:352);
      x3=x(145:288,177:352);
      x4=x(145:288,1:176);
      y= rgb2gray(w);
      %dividing a 240 by 320 into 4 pictures
      y1=y(1:144,1:176);
      y2=y(1:144,177:352);
      y3=y(145:288,177:352);
      y4=y(145:288,1:176);
      %mesure of 2 dimensions cross correlation for each picture
      z= corr2(x,y);
      z1= corr2(x1,y1);
      z2= corr2(x2,y2);
      z3= corr2(x3,y3);
      z4= corr2(x4,y4);
      %put values of correlation into arrays
      h(a) = z;
      h1(a)=z1;
      h2(a)=z2;
      h3(a)=z3;
      h4(a)=z4;
      %measure the minimum value for correlation
```

```
        hh=[z1,z2,z3,z4];
        zz=min(hh);
        hmin(a)=zz;
        kk=[hmin(a-1),hmin(a) ];
        var_value=var(kk,1);
        var_values(a)=var_value;
        if (var_value > threshold)
           flag1=get(handles.togglebutton4,'value');
           if flag1==1
              [t,Fs] = wavread('warn.wav');
              player = audioplayer(t,Fs);
              play(player);
           end
           fid = fopen('log.txt','a');
           time=datestr(now)
           fprintf(fid,'GERAKAN TELAH TERDETEKSI PADA %-100.20s\n',time);
           fprintf(fid,'pada frame nomor %-110.5d\n',a);
           fclose(fid);
           qq=qq+1;
           imshow(v);
           film(qq) = im2frame(v);
        else
        end
     else
        break
     end
     toc
end
delete(vobj);
savefile = 'var.mat';
save(savefile,'var_values','film')
```

- **Main.java**

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package mymms;

import java.io.BufferedOutputStream;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.net.URL;
import java.net.URLConnection;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFileChooser;

/**
 *
 *
 */
public class Main {

    /**
     * @param args the command line arguments
     */
    static String path;
    static String username ="b16_8700630";
    static String hpAdmin  ="085755975505";
    static String hpTujuan ="085755179381";

    public static class mms  {
```

```
    public boolean sending(String s){

        try {
            File f = new File(path);
            String urlString = "ftp://" + username + ":" + password  + "@" + server    + "/" +
f.getName();
            URL url = new URL(urlString);
            URLConnection connection = url.openConnection();
            connection.setDoOutput(true);
            BufferedOutputStream out = new
BufferedOutputStream(connection.getOutputStream());
            FileInputStream in = new FileInputStream(f);
            byte[] buffer = new byte[1024];

            int i = 0;
            while ((i = in.read(buffer)) >= 0) {
                out.write(buffer, 0, i);
            }
            out.close();
            in.close();
            System.out.println("File Send. : "+s);
            return  true;
        }catch (Exception e) {

            System.out.println(e.getMessage());
            return  false;
        }


    }
  }

  private static void copyfile(String srFile, String dtFile){
        try
    {

                File f1 = new File(srFile);
                File f2 = new File(dtFile);
                InputStream in = new FileInputStream(f1);
```

```
                OutputStream out = new FileOutputStream(f2);

                byte[] buf = new byte[1024];
                int len;
                while ((len = in.read(buf)) > 0){
                        out.write(buf, 0, len);
                }
                in.close();
                out.close();
                System.out.println("File Copied.");
        }catch(FileNotFoundException ex){
                System.out.println(ex.getMessage() + " in the specified directory.");
                System.exit(0);
        }
                catch(IOException e){
                System.out.println(e.getMessage());
        }
    }
   public static String executes(String app){
      try {
         Runtime rt = Runtime.getRuntime();

         Process pr = rt.exec(app);

         BufferedReader input = new BufferedReader(new
InputStreamReader(pr.getInputStream()));
         String line = null;
         while ((line = input.readLine()) != null) {
            System.out.println(line);
         }
         int exitVal = pr.waitFor();
         return  "Exited with error code " + exitVal ;
      } catch ( Exception ex) {

          Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
          return null;
      }

   }
```

```
    public static void main(String[] args) {

        {
            String s;
            mms m = new mms();
            if  (args.length==0){
               JFileChooser jfc = new JFileChooser();
               jfc.showDialog(jfc, "Choose an image file");
               s = jfc.getSelectedFile().getAbsolutePath();
            }else{
               s =args[0];
            }
            long pass = System.currentTimeMillis();
            path = pass + ".jpg";
            copyfile(s,  path);
//            File f;
//            f = new File(hpAdmin + ".txt");
//            try {
//               f.createNewFile();
//            } catch (IOException ex) {
//               Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
//            }
//            m.sending(hpAdmin + ".txt");
            copyfile("log.txt",  pass+".txt");
            new mms().sending( pass+  ".txt"  );

            if (m.sending( path)) {
                //new File( path).delete();

                new File(pass +  ".txt" ).delete();
                StringBuilder sb = new StringBuilder();
                sb.append("Anda telah menerima mms dari ").append(hpAdmin);
                sb.append(",link http://").append(web).append("/
(PIN:").append(pass).append(")");
                String loc = "gammu-smsd-inject.exe -c smsdrc TEXT " + hpTujuan + " -
text \"" + sb.toString() + "\"";
                executes(loc);

            }
```

- **ContentEntry.java**

```
package sendmms;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.io.RandomAccessFile;
import java.io.UnsupportedEncodingException;

/**
 * Helper class for loading contents.
 *


 *
 */
public class ContentEntry {

	public static final int TEXT_STRING = 0;
	public static final int TEXT_FILE = 1;
	public static final int IMAGE_FILE = 2;

	private String s;
	private int type;
	byte[] content = null;
	String stringContent = null;
	String fileName = null;

	public ContentEntry(String s, int type){
		this.s = s;
		this.type = type;
	}

	public String getS() {
		return s;
	}

	public int getType() {
		return type;
```

```
}

public byte[] getContent(){
        return content;
}

public String toString(){
        switch(type){
                case TEXT_STRING:
                        return "TEXT_STRING: " + stringContent;
                case TEXT_FILE:
                        return "TEXT_FILE: " +  fileName;
                case IMAGE_FILE:
                        return "TEXT_IMAGE: " + fileName;
                default:
                        return "UNKNOWN";

        }
}

public void loadContent() throws SendMmsException{
        switch(type){
        case TEXT_STRING:
                try {
                                stringContent = s;
                                content = s.getBytes("US-ASCII");
                        } catch (UnsupportedEncodingException e) {
                                e.printStackTrace();
                        }
                break;
        case TEXT_FILE:
                FileReader fr;
                fileName = s;
                StringBuffer text = new StringBuffer();
                try {
                        fr = new FileReader(s);
                        char[] buffer = new char[1024];
                        int i;
                        do{
                                i = fr.read(buffer);
```

```
                        if (i>0)text.append(buffer, 0, i);
                        else break;
                    }while(true);
                } catch (FileNotFoundException e) {
                    throw new SendMmsException("File " + s + " not found");
                } catch (IOException e) {
                    e.printStackTrace();
                }

                try {
                        content = text.toString().getBytes("US-ASCII");
                    } catch (UnsupportedEncodingException e1) {
                        e1.printStackTrace();
                    }
            case IMAGE_FILE:
                FileInputStream fin;
                RandomAccessFile f;
                fileName = s;
                try {
                    fin = new FileInputStream(s);
                    f = new RandomAccessFile(s,"r");
                } catch (FileNotFoundException e) {
                    throw new SendMmsException("File " + s + " not found");
                }

                long size=0;
                try {
                    size = f.length();
                } catch (IOException e) {
                    e.printStackTrace();
                }
                if (size > Integer.MAX_VALUE) throw new SendMmsException("File " +
s + " too big");

                content = new byte[(int)size];
                try {
                    fin.read(content);
                } catch (IOException e) {
                    e.printStackTrace();
                }
```