



Hak cipta dan penggunaan kembali:

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

Copyright and reuse:

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

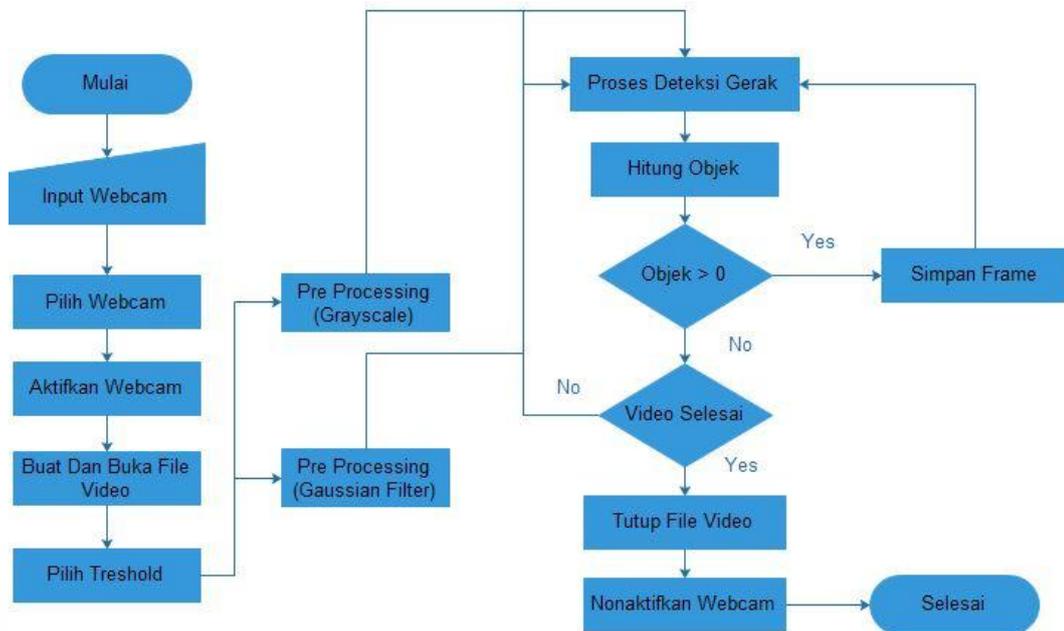
BAB III

METODOLOGI PENELITIAN

Pada bab ini akan membahas langkah-langkah perancangan sistem aplikasi dan implementasi dari aplikasi deteksi pergerakan menggunakan *webcam* dalam bahasa pemrograman Visual Studio. Langkah pertama yang dilakukan oleh sistem aplikasi ini adalah dengan memulai atau menjalankan aplikasi, kemudian webcam akan aktif dan akan menetapkan warna latar belakang sehingga jika terjadi perubahan warna, maka akan ditetapkan sebagai objek bergerak. Hal selanjutnya yaitu aplikasi melakukan proses deteksi objek, jika ada objek yang bergerak maka aplikasi menghitung jumlah objek tersebut dan jika objek nilainya lebih dari 0 maka akan disimpan menjadi *file* gambar dan video. Ketika tidak ada objek yang terdeteksi maka secara otomatis *webcam* akan mati dan tidak menyimpan *frame*, gambar ataupun video. Tahap terakhir yaitu selesai sebagai penanda bahwa rekaman sudah dihentikan dan menghasilkan sebuah *file* video yang berisikan rekaman objek yang bergerak saja.

3.1 Rancangan Sistem

Diagram alur (*flowchart*) yang dibuat pada gambar 3.1 menjelaskan proses atau cara kerja dari aplikasi sehingga dapat memudahkan pemahaman pada sistem.



Gambar 3.1 Diagram Alur Sistem Aplikasi Tab Implementasi

Proses atau cara kerja aplikasi yaitu dari mulai, kemudian dilanjutkan dengan memasukkan tipe *webcam* yang baru jika ada. Setelah itu dapat memilih tipe *webcam* apa yang ingin dipakai, maka secara otomatis *webcam* akan menyala atau aktif. Kemudian ketika *button* rekam ditekan maka secara otomatis sistem akan membuat *file* video yang berisikan objek bergerak saja. Jika kita ingin melihat hasil rekaman video atau ingin melakukan percobaan terhadap semua video **avi*, maka cukup menekan *button* AVI File dan pilih video **avi* mana yang ingin dipakai sebagai percobaan. Setelah itu pilih *threshold* atau nilai panjang dan lebar piksel. Nilai *threshold* sendiri berjumlah 10-100. Semakin kecil *threshold* yang kita pilih, maka sensitivitas aplikasi akan semakin tinggi dalam menangkap objek-objek yang bergerak. Tahap selanjutnya yaitu *pre processing Gaussian Filter* yang berguna untuk meminimalisir noise sehingga hasil gambar menjadi

lebih lembut. *Pre processing Grayscale* berguna untuk memberi warna keabuan pada gambar sehingga hasil gambar *grayscale* berbeda dengan gambar aslinya. Setelah itu aplikasi akan memroses objek yang bergerak berdasarkan perubahan warna atau nilai pada latar belakang. Jika ada warna atau nilai yang berubah, maka dinyatakan sebagai objek yang bergerak dan aplikasi akan menghitung objek tersebut. Jika objek yang terdeteksi nilainya lebih dari 0, maka *frame* akan disimpan secara terus menerus. Apakah rekaman video sudah selesai ? Jika belum, maka aplikasi akan terus memroses dan menghitung objek yang bergerak, dan jika sudah, maka video akan ditutup dan secara otomatis *webcam* akan mati dan selesai.

3.2 Implementasi Aplikasi Deteksi Objek Menggunakan Webcam

Pada subbab ini akan dijelaskan pembuatan aplikasi menggunakan Microsoft Visual Studio 2008. Langkah-langkah pada pembuatan aplikasi ini, yaitu:

1. pengenalan *webcam* pada aplikasi,
2. meng-aktifkan dan menonaktifkan *webcam*,
3. *Pre Processing Gaussian*,
4. *Pre Processing Grayscale*,
5. deteksi gerakan,
6. menghitung objek,
7. membuat file video,

8. merekam hasil dari deteksi gerak, dan
9. menutup file video.

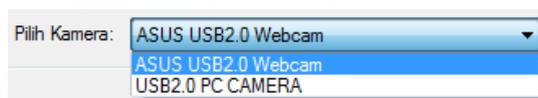
3.2.1 Pengenalan Webcam

Proses ini mengumpulkan *webcam* yang telah terpasang pada komputer/laptop. Diperlukan *label* dan *combo box* untuk proses ini. *Combo box* digunakan untuk mengumpulkan nama-nama dari *webcam*. Gambar 3.2 adalah tampilan dari *combo box* yang digunakan pada proses pengenalan *webcam*.



Gambar 3.2 Combo Box

Combo Box tersebut menggunakan *library* AForge .Net, yaitu Video dan *DirectShow*.



Gambar 3.3 Combo Box Setelah Diberi Kode

3.2.2 Mengaktifkan Webcam

Setelah memilih *webcam* yang sudah dikenali oleh aplikasi, proses selanjutnya adalah mengaktifkan *webcam* dengan *button* dan menampilkannya pada *video source player*. Gambar 3.4 merupakan *button*.



Gambar 3.4 Button Untuk Mengaktifkan Webcam

Hasil dari mengaktifkan *webcam* tersebut akan ditampilkan pada video *source player* seperti Gambar 3.5.



Gambar 3.5 Video Source Player Sebelum dan Sesudah Diaktifkan

3.2.3 Menonaktifkan *Webcam*

Apabila *webcam* sudah tidak digunakan atau aplikasi sudah selesai digunakan, untuk menonaktifkan *webcam* tersebut menggunakan *button*. Gambar 3.6 merupakan *button* untuk menonaktifkan *webcam*.



Gambar 3.6 Button untuk Menonaktifkan *Webcam*

Hasil dari menonaktifkan *webcam* tersebut akan memberhentikan tampilan pada *video source player* seperti Gambar 3.7.



Gambar 3.7 Video Source Player Di Nonaktifkan

3.2.4 Deteksi Pergerakan

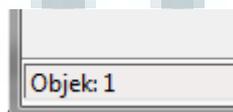
Pada proses deteksi pergerakan dan menghitung objek menggunakan *library* dari Aforge .Net, yaitu *Vision Motion*. Deteksi pergerakan akan diproses pada *video source player*. Pendeteksi gerak menggunakan metode *Simple Background Modeling Detector*.



Gambar 3.8 Deteksi Gerak

3.2.5 Menghitung Objek yang Terdeteksi Bergerak

Pada saat pendeklarasian variabel pendeteksi, metode *Simple Background Modeling Detector* dipanggil untuk menjadi pendeteksi gerak, sementara untuk memproses hasil dari deteksi gerak digunakan metode *Blob Counting Objects Processing* yang berfungsi untuk menghitung objek bergerak. Hasil dari menghitung objek akan ditampilkan pada *label*, seperti Gambar 3.9.



Gambar 3.9 Hitung Objek