

PROGRAM PENGHITUNG JUMLAH ORANG LEWAT MENGGUNAKAN WEBCAM

Sudianto Lande

Universitas Kristen Indonesia Paulus
Perintis Kemerdekaan Km 13 Makassar 90243 Sulawesi Selatan

Resmana Lim, Kartika Gunadi & Chandra K.

Fakultas Teknologi Industri, Jurusan Teknik Informatika, Universitas Kristen Petra
Email: resmana@petra.ac.id, kgunadi@petra.ac.id

ABSTRAK: Data jumlah pengunjung suatu tempat umum sangat penting. Data jumlah pengunjung biasanya didapat secara manual. Saat ini kamera video telah diterapkan untuk kepentingan keamanan. Karena itu dibuatlah program penghitung jumlah pengunjung dengan metode Normalized Sum-Squared Differences (NSSD) yang mengambil selisih jumlah pixel frame dan background dan dikuadratkan, dinormalisasi dengan luasan detection window. Nilai NSSD yang didapat diseleksi dengan proses thresholding untuk mendeteksi keberadaan orang pada detection window. Penelitian ini dibuat dengan menggunakan Borland Delphi 5.0, dengan tambahan komponen TVideo. Program ini secara keseluruhan menunjukkan keberhasilan lebih dari 90%. Keberhasilan dari program ini sangat dipengaruhi oleh penentuan nilai threshold yang tepat.

Kata kunci: penghitungan orang, sensor kamera, NSSD, *Image processing*.

ABSTRACT: *The amount of public places's visitor data is very important. Usually we get it manually. AT the moment, video camera has been used for security. Therefor, the people counter software has been made using Normalized Sum-squared difference (NSSD) method that take differences the sum of frame fixel and background, squared it then normalized by detection window area. The NSSD values that have been count then thresholded to detect the people occurance in detection window. This project is made using Borland Delphi 5.0 with Tvideo component. Corect people counting percentation of more than 90% was obtained. The succesness of this program depends on the right thresholding values.*

Keywords: *NSSD, Image processing, Grayscale, Threshold.*

1. PENDAHULUAN

Pemanfaatan sensor kamera pada instrumentasi telah cukup meluas mulai dari penghitungan obyek mikroskopis, pengukuran kecepatan obyek bergerak, dan lain-lain. Bahkan pemanfaatan sensor kamera stereo telah pula dimanfaatkan untuk pengukuran jarak obyek bergerak, yang pada pemanfaatannya digunakan sebagai alat bantu kemudi mobil otomatis. Dalam paper ini akan dibahas tentang pemanfaatan sensor kamera untuk penghitungan obyek bergerak dalam hal ini adalah orang berjalan.

Metode yang digunakan adalah *Normalized Sum-Squared Differences* (NSSD) [1]. Diawali dengan pengambilan citra background, dan menentukan detection window pada jalur obyek yang akan diamati. Hal yang sama dilakukan juga untuk setiap frame video yang ditampilkan. Detection window berupa ROI (Region of Interest) pada background dan input frame harus pada posisi yang sama sehingga luasan area yang diamati sama persis. Nilai pixel pada frame dan background diambil selisihnya dan dikuadratkan. Hasil tersebut dibagi dengan luasan detection window untuk dinormalisasi. Proses filtering perlu dilakukan

untuk mengurangi noise yang terjadi sehingga bilamana dilakukan *thresholding*, sistem dapat bekerja dengan baik.

Proyek sejenis telah banyak diteliti orang dengan berbagai algoritma [2,3,4]. Algoritma NSSD yang digunakan pada paper ini memiliki kelebihan dalam kesederhanaannya sehingga dapat diimplementasikan secara real time dengan platform PC.

Sistem diimplementasikan pada platform O.S Windows menggunakan software Borland Delphi 5.0 dengan tambahan komponen Tvideo. Dari hasil pengujian, diperoleh hasil bahwa sistem dapat menghitung obyek bergerak dengan tingkat keberhasilan $\pm 80\%$.

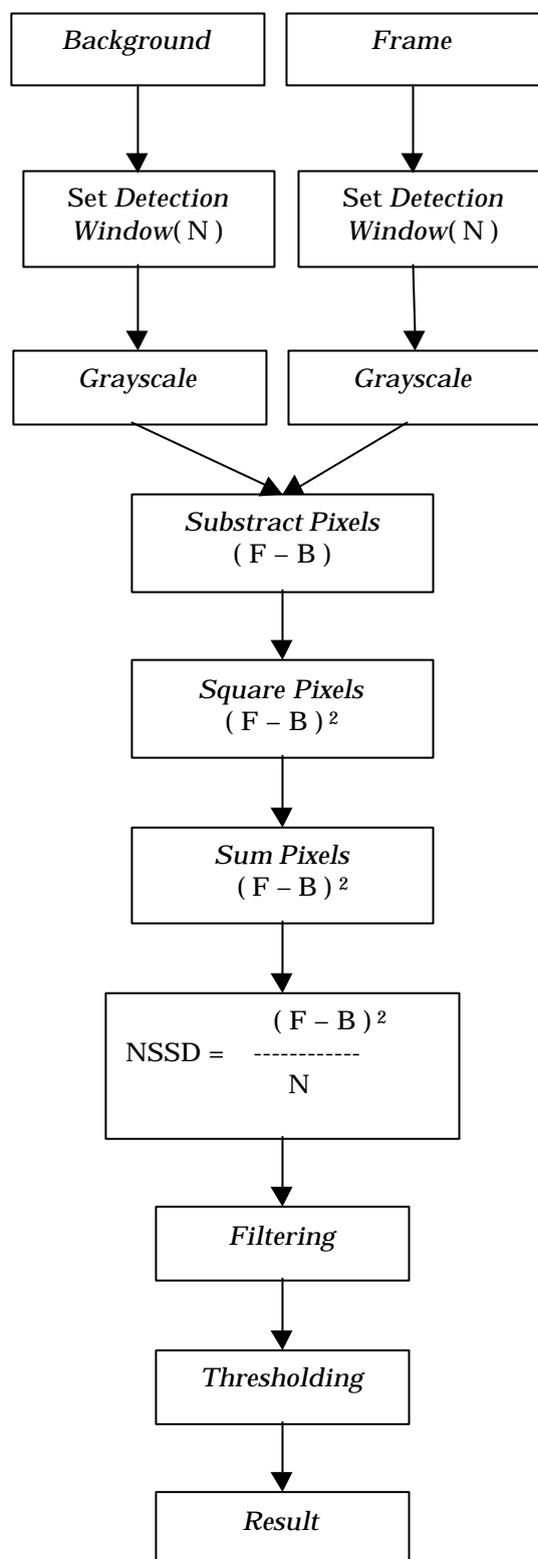
Berikutnya paper ini diorganisasi sebagai berikut: pada bab 2 akan dibahas sistem secara keseluruhan dan dilanjutkan dengan presentasi hasil-hasil pengujian serta diskusi dan kesimpulan.

2. DESKRIPSI SISTEM

Secara blok diagram, sistem yang dibuat adalah seperti gambar 1. Sistem ini dibagi menjadi empat bagian, yaitu pemrosesan background image dan frame, perhitungan NSSD, proses filtering, serta Thresholding.

Metode yang digunakan di sini adalah *Normalized Sum-squared Difference* (NSSD). Langkah pertama yang harus dilakukan adalah pengambilan background dan kemudian mengubahnya menjadi grayscale dan menentukan detection window pada jalur orang lewat yang akan diamati. Hal yang sama dilakukan juga untuk setiap frame video yang ditampilkan. Detection window pada background dan frame harus pada posisi yang sama sehingga luasan area yang diamati sama persis. Jumlah nilai pixel pada frame dan background diambil selisihnya dan dikuadratkan. Kemudian hasilnya dibagi dengan luasan detection window dan dinormalisasi. Selain itu perlu ditambahkan proses anti-flicking di mana proses ini bertujuan untuk meredam flicking yang mungkin terjadi

pada saat perekaman gambar, sehingga apabila dilakukan thresholding, sistem dapat bekerja dengan lebih tepat.



Gambar 1. Blok Diagram Sistem

2.1 Pemrosesan Background Image dan Frame

Langkah pertama yang dilakukan adalah mengambil gambar *background* yang akan dipakai. Pengambilan *background* dilakukan dengan cara menangkap *image* dari *video preview*. *Background* tersebut selalu di *update* pada saat akan melakukan proses *recording*. Dengan demikian diharapkan *background* yang didapat mewakili *frame* kosong yang paling sesuai dengan kondisi terakhir sebelum dilakukan *recording*.

Langkah kedua adalah menentukan *detection windows* pada *background* dan *frame*. *Detection windows* pada *background* dan *frame* input diletakkan pada posisi yang sama. Hal ini dilakukan untuk membatasi daerah yang akan diproses.

Setelah menentukan *detection windows*, langkah berikutnya adalah mengubah *frame* yang semula dalam format RGB menjadi *grayscale*. Hal ini dilakukan agar setiap *pixel* hanya mempunyai satu macam nilai *pixel*. Setiap *detection windows*, baik pada *background* maupun pada *frame*, dihitung jumlah nilai *pixel*-nya dengan menggunakan *procedure* penjumlahan.

2.2 Perhitungan NSSD

Dari setiap nilai *pixel grayscale* yang didapat, hasilnya dikurangkan sehingga perbedaan jumlah nilai *pixel* inilah yang dikenali sebagai keberadaan suatu objek. Untuk menghindari nilai negatif dari operasi tersebut, hasil yang diperoleh dipangkatduakan. Hasilnya dibagi dengan luasan *detection windows*. Dari proses tersebut didapat nilai NSSD. Metode NSSD dapat dirumuskan dalam persamaan 1.

$$NSSD = \frac{\sum_{I_C} (I_C(i) - I_B(i))^2}{N} \quad (1)$$

Keterangan:

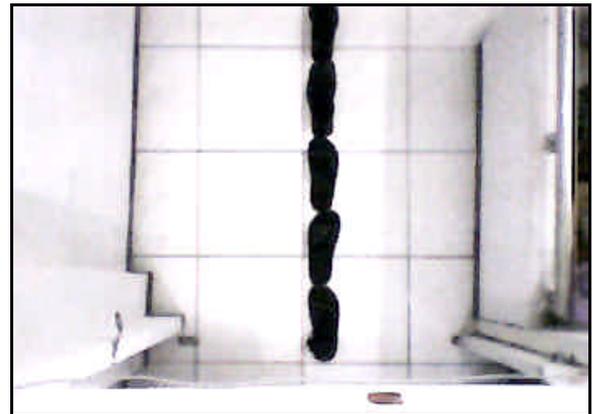
$I_C(i)$ = intensitas *frame* pada *pixel* (i)

$I_B(i)$ = intensitas *background* pada *pixel* (i)

Q = *detection region*/ROI (*Region of Interest*)

N = jumlah *pixel* pada area Q

Dalam perhitungan NSSD, yang digunakan adalah selisih dari jumlah nilai *pixel* pada *background image* dan pada *frame*. Selisih jumlah nilai *pixel*, memungkinkan untuk menghasilkan nilai negatif sehingga hasil selisih tersebut dikuadratkan untuk menghindari hasil negatif. Nilai negatif terjadi bilamana warna dari obyek pada *frame* lebih gelap daripada *background image*. Sebab warna gelap memiliki nilai yang kecil.

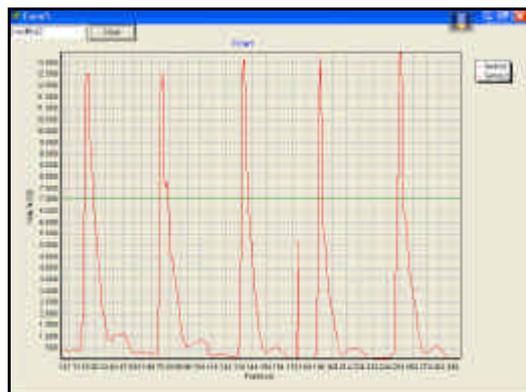


Gambar 2a. Image input frame-Ic



Gambar 2b. Background image-Ib

Perhitungan NSSD dilakukan pada setiap *frame* video yang ditampilkan, sehingga jumlah data yang didapat sesuai dengan jumlah *frame* yang ditampilkan. Misalnya file video dengan 25 fps maka data yang diperoleh dalam 1 detik sebanyak 25 data. Setiap nilai NSSD yang didapat dibuat grafiknya seperti gambar 3.



Gambar 3. Grafik NSSD sebelum filtering

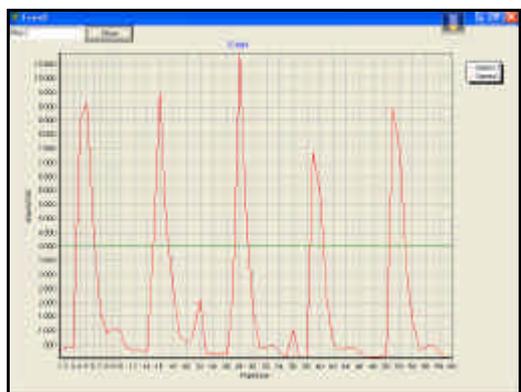
Data yang diambil pada grafik tersebut mewakili lima buah kendaraan. Setiap puncak dari grafik mewakili sebuah kendaraan. Dengan kondisi grafik yang demikian akan sulit untuk menentukan nilai thresholdnya. Untuk itu perlu dilakukan proses filtering.

2.3 Filtering

Proses filtering dilakukan untuk mengurangi perbedaan nilai data yang mencolok. Dengan menggunakan *Low-Pass Filter*, data yang memiliki nilai besar akan diredam sehingga perbedaan antar data tidak terlalu besar. *LowPass Filter* ini menggunakan cara merata-rata setiap data NSSD yang diperoleh. Untuk merata-rata 5 data, data NSSD ke-(n) dirata-rata dengan semua data (n-5).

$$\text{Fil}(1) = (\text{NSSD}(1) + \text{NSSD}(2) + \text{NSSD}(3) + \text{NSSD}(4) + \text{NSSD}(5)) / 5 \quad (2)$$

Dengan melakukan proses filtering, data yang diperoleh akan menghasilkan grafik seperti gambar 4.



Gambar 4. Grafik NSSD Setelah Filtering

Dapat dilihat bahwa dengan hasil filtering penentuan nilai threshold akan lebih mudah.

2.5 Thresholding

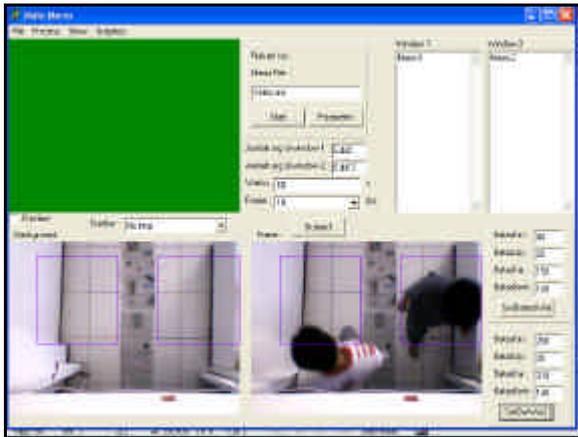
Dari proses filtering, pada gambar 4 dapat dilihat bahwa terdapat lima puncak grafik yang mewakili lima orang. Dengan menentukan nilai threshold yang mendeteksi adanya lima orang, maka nilai tersebut diambil sebagai nilai threshold.

2.6 Proses Penghitungan Orang

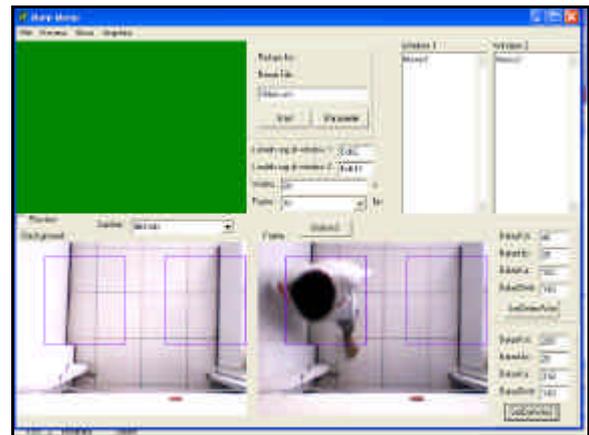
Pada program penghitungan orang yang lewat ini diberi batasan hanya dilakukan di pintu masuk dan di pintu keluar bagi pengunjung perpustakaan. Demikian pula untuk intensitas cahaya pada saat penghitungan dilakukan. Intensitas cahaya akan mempengaruhi warna dari orang yang akan dihitung. Untuk itu pengujian penghitungan orang dilakukan pada ruang tertutup dimana intensitas cahaya cenderung tetap.

Proses penghitungan dilakukan apabila orang melewati *detection windows*, baik *detection windows 1* maupun *detection windows 2*. Masing-masing *detection windows* mewakili dua jalur pada arah yang berbeda. Jumlah nilai *pixel* yang terbaca di *detection windows* pada *frame* akan dikurangkan dengan jumlah nilai *pixel detection windows* pada *background*. Selisih dari operasi ini dikuadratkan untuk menghindari nilai negatif dan juga dari selisih ini objek dengan warna terang masih dapat terdeteksi meskipun nilainya kecil. Untuk mendapatkan nilai normalisasi, hasil tersebut dibagi dengan luasan *detection window*.

Pengujian dengan *file video *.avi* tidak memungkinkan dikarenakan penangkapan gambar dilakukan secara *real-time*, di mana jika menggunakan *file *.avi*, maka akan menghabiskan kapasitas *hard disk* yang tersedia.



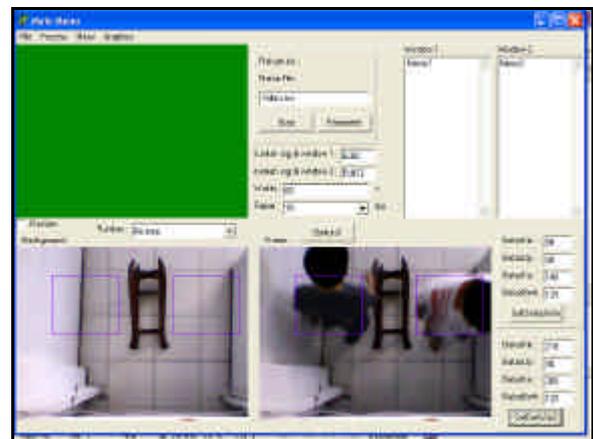
Gambar 5. Tampilan Sistem



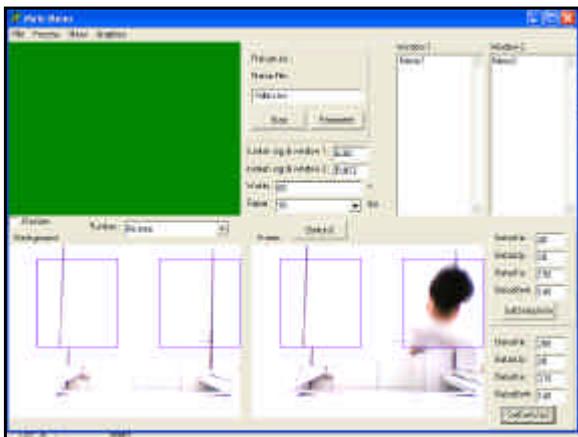
Gambar 7. Background normal

3. HASIL PENGUJIAN

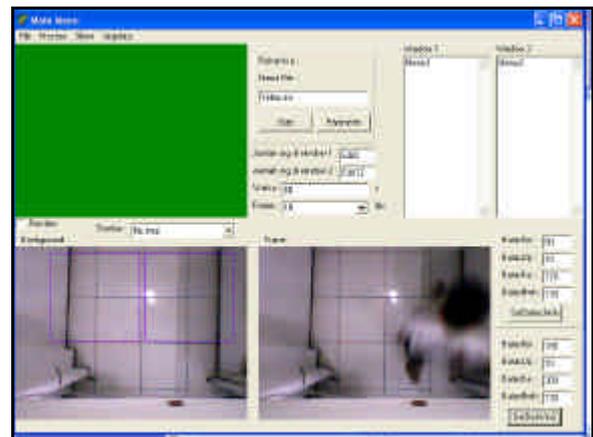
Pada penelitian ini, sumber gambar dapat diambil dari file .avi, dan kemudian dikonversikan ke dalam format .bmp, dan juga dapat dilakukan secara langsung dari kamera. Untuk penghitungan secara langsung, setiap frame yang ditangkap disimpan ke dalam file, kemudian diproses dan dihitung, kemudian file yang disimpan dihapus kembali. Hal ini dilakukan agar menghemat kapasitas hard disk yang tersedia. Pengujian sistem dilakukan dengan menguji performa sistem terhadap berbagai macam kondisi video.



Gambar 8. Background agak gelap 2 orang



Gambar 6. Background terang



Gambar 9. Obyek yang bergerak cepat

Hasil pengujian penghitungan orang dapat dilihat pada tabel 1.

Tabel 1. Hasil Pengujian

No.	Kondisi	Jumlah orang Sebenarnya	Jumlah orang terhitung	Tingkat Keberhasilan (%)
1	Terang	82	82	100,00
2	Normal	70	65	92,86
3	Agak gelap	96	84	87,50
4	Obyek bergerak cepat	88	102	86,27
Rata-rata:				91,66

4. KESIMPULAN

Secara keseluruhan sistem bekerja cukup baik, hal ini dapat dibuktikan dengan tingkat keberhasilan lebih dari 90% pada penghitungan jumlah orang lewat dengan berbagai variasi iluminasi. Penentuan nilai threshold secara manual dapat dikembangkan lebih lanjut agar bisa dilakukan secara otomatis. Variasi iluminasi cukup berpengaruh, agar perhitungan lebih teliti seharusnya citra background tidak selalu tetap untuk seluruh perhitungan. Citra background seyogyanya diambil secara adaptif sehingga dapat meminimalisasi variasi iluminasi.

DAFTAR PUSTAKA

1. Jae-Won, Kim, Choi, Kang-Sun, Choi, Byeong-Doo, Ko, Sung-Jea. *Real-time Vision-based People Counting System for the Security Door*. PDF. http://www.kmutt.ac.th/itc2002/CD/pdf/18_07_45/TP2_OC/4.pdf, 2002.
2. Danny B. Yang, Hector H. Gonzalez-Banos, Leonidas J. Guibas. *Counting People in Crowds with a Real-Time Network of Simple Image Sensors*. PDF. [\[ings/CollaborativeProcessing/dbyang_paper.pdf\]\(ings/CollaborativeProcessing/dbyang_paper.pdf\), 2003.](http://graphics.stanford.edu/courses/cs428-03-pring/apers/read-

</div>
<div data-bbox=)

3. N. Paragios, G. Tziritas. *Detection and Location of Moving Objects Using Deterministic Relaxation Algorithms*. PDF. http://www.csd.ucl.ac.uk/~tziritas/papers/icpr_96_change.pdf, 2002
4. Piroddi, Roberta, Vlachos, Theodore. *Multiple-Feature Spatiotemporal Segmentation of Moving Sequences using a Rule-based Approach*. http://www.bmva.ac.uk/bmvc/2002/papers/163/full_163.pdf, 2002