

IMPLEMENTASI PENTERJEMAH KODE ISYARAT TANGAN MENGGUNAKAN ANALISIS DETEKSI TEPI PADA ARM 11 OK6410B

Heri Setiawan¹, Iwan Setyawan², Saptadi Nugroho³

Program Studi Teknik Elektro, Fakultas Teknik Elektro dan Komputer,

Universitas Kristen Satya Wacana, Salatiga

¹heri33s@gmail.com, ²iwan.setyawan@ieee.org,

³saptadi_nugroho@yahoo.com

INTISARI

Penerapan penterjemah kode isyarat tangan pada ARM 11 OK6410B masih jarang dilakukan karena pada umumnya dibutuhkan prosesor yang memiliki frekuensi tinggi untuk mengolah citra digital. Analisis Deteksi Tepi merupakan metode yang umum di dalam pengolahan citra digital untuk mendeteksi perbedaan intensitas cahaya di dalam citra digital. Di dalam paper ini penulis membuat implementasi metode analisis deteksi tepi yang telah dioptimalisasi pada ARM 11 OK6410B untuk menterjemahkan kode isyarat tangan. Optimalisasi metode analisis deteksi tepi dilakukan dengan mempercepat perhitungan konvolusi citra. Tingkat pengenalan metode analisis deteksi tepi yang diimplementasikan pada ARM 11 OK6410B adalah sebesar 52,67% dengan waktu pemrosesan 0,27 detik.

Kata kunci : Kode Isyarat Tangan, Analisis Deteksi Tepi, ARM 11

1. PENDAHULUAN

Penelitian tentang metode penterjemah kode isyarat tangan masih merupakan bidang yang berkembang sampai saat ini dan bertujuan untuk memperoleh pengenalan isyarat tangan yang akurat [1]. Namun penerapan penterjemah kode isyarat tangan pada *embedded system* masih jarang dilakukan. Implementasi penterjemah kode isyarat tangan dengan menggunakan analisis deteksi tepi pada *embedded system* ini dapat digunakan untuk mengontrol robot. Di dalam paper ini,

waktu pemrosesan citra dengan menggunakan metode analisis deteksi tepi ini dioptimalkan karena OK6410B bekerja pada frekuensi prosesor yang lebih rendah daripada komputer dan menggunakan tipe data *floating point* yang membutuhkan waktu perhitungan yang relatif lama. Optimalisasi analisis deteksi tepi dilakukan dengan cara mengurangi perulangan dan perhitungan.

2. DASAR TEORI

Deteksi tepi Canny yang dikembangkan oleh John F. Canny [2] memiliki kelebihan hanya menandai nilai maksimal pada tepian gambar sehingga dihasilkan tepian piksel tunggal. Beberapa tahap pemrosesan deteksi tepi Canny adalah *smoothing*, operasi operator Sobel, menandai nilai tepian maksimal dan penelusuran histerisis. *Smoothing* bertujuan mengurangi noise berupa penambahan nilai yang tidak diinginkan akibat penerimaan sensor yang kurang baik [3]. Operasi operator Sobel mendeteksi perbedaan nilai antar piksel yang berdekatan menggunakan Sobel kernel. Nilai tepian maksimal ditandai untuk mendapatkan nilai tepian yang penting saja. Penelusuran histerisis menandai nilai tepian yang terhubung dengan tepian kuat.

OK6410B merupakan *single board computer* berbasis mikrokontroler ARM11 Samsung S3C6410 yang memiliki frekuensi 553MHz. *Board* OK6410B memiliki ukuran yang relatif kecil yaitu $81\text{mm} \times 105\text{mm}$ [4].

3. PERANCANGAN SISTEM

Tujuan deteksi tepi adalah mengolah citra *grayscale* menjadi tepian dari citra masukan dalam bentuk citra biner. Perubahan intensitas cahaya dalam citra dideteksi dengan menentukan nilai gradient citra dengan menggunakan operator *Sobel* yang disederhanakan proses konvolusinya dengan menghilangkan perkalian dengan elemen 0 pada matriks. Selain 0, matriks hanya berisi nilai 1 dan 2. Dengan membuat matriks $D=2*I$, perhitungan G_x dan G_y untuk piksel koordinat y_0, x_0 dapat dijadikan persamaan sebagai berikut.

$$\begin{aligned} G_x(y_0, x_0) = & -I(y_0 - 1, x_0 - 1) - D(y_0, x_0 - 1) - I(y_0 + 1, x_0 - 1) \\ & + I(y_0 - 1, x_0 + 1) + D(y_0, x_0 + 1) + I(y_0 + 1, x_0 + 1) \end{aligned} \quad (1)$$

$$\begin{aligned} G_Y(y_0, x_0) = & -I(y_0 - 1, x_0 - 1) - D(y_0 - 1, x_0) - I(y_0 - 1, x_0 + 1) \\ & + I(y_0 + 1, x_0 - 1) + D(y_0 + 1, x_0) + I(y_0 + 1, x_0 + 1) \end{aligned} \quad (2)$$

Dengan persamaan di atas, perhitungan G_X dan G_Y menjadi lebih sedikit daripada perhitungan konvolusi biasa.

Setelah itu nilai maksimum lokal dicari untuk masing-masing hasil nilai gradien horisontal dan vertikal dengan cara membandingkan nilai piksel suatu titik dengan nilai piksel tetangganya. Nilai yang tersisa dari pencarian nilai lokal maksimal ditandai dengan *thresholding*. Deteksi tepi *Canny* menggunakan dua nilai untuk *thresholding*, yaitu nilai *threshold* rendah dan nilai *threshold* tinggi yang ditentukan dengan percobaan. Jika nilai suatu piksel lebih besar daripada nilai *thresholding* tinggi berarti dapat ditandai sebagai nilai tepian kuat, maka nilai piksel tersebut dianggap sebagai “pasti tepian citra”.

Pendeteksian jari dilakukan dengan cara menelusuri tepian citra hasil deteksi tepi *Canny* yang dimulai dari titik mulai ditemukan tepian citra. Bentuk lengkungan jari dideteksi berdasarkan perubahan nilai arah penelusuran garis tepian. Nilai perubahan arah pada lengkungan dapat dihitung dengan persamaan berikut.

$$\Delta\omega = d / m \quad (3)$$

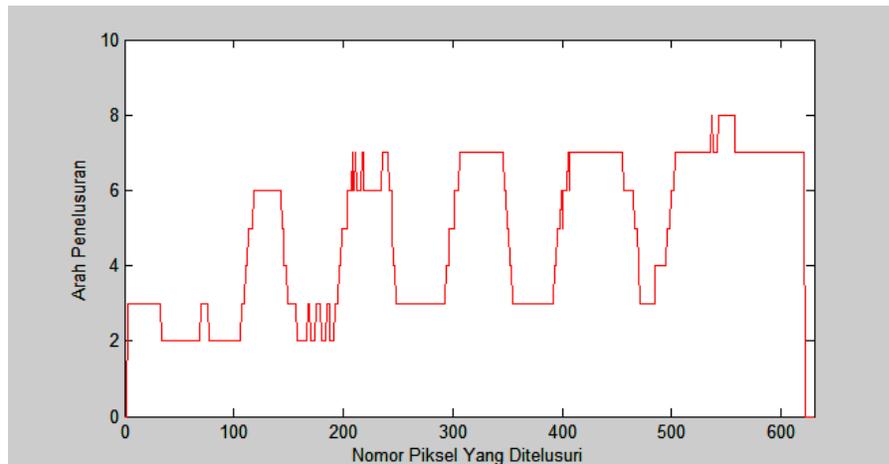
dengan

$\Delta\omega$ adalah perubahan nilai arah pada lengkungan,

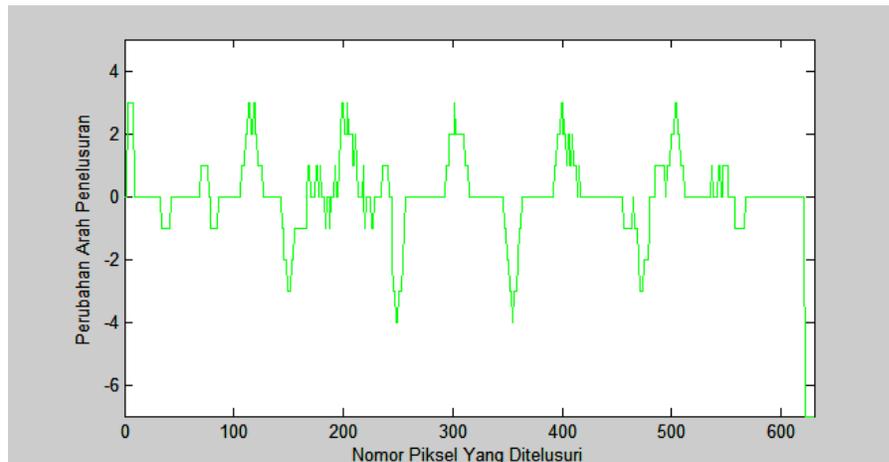
d adalah nilai akumulasi perubahan arah penelusuran,

m adalah nilai selisih urutan piksel pada penelusuran sudut.

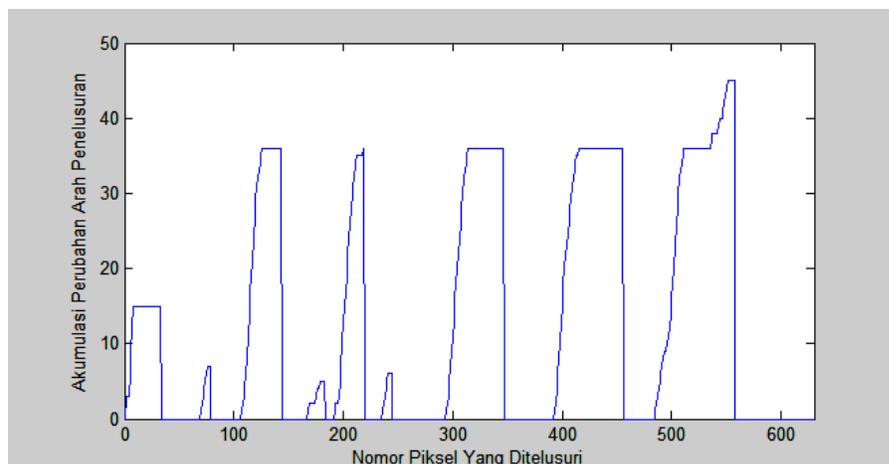
Berikut ini adalah contoh hasil penelusuran garis pada sebuah jalur tepian citra tangan digambarkan dalam bentuk grafik.



Gambar 1. Grafik Arah Penelusuran



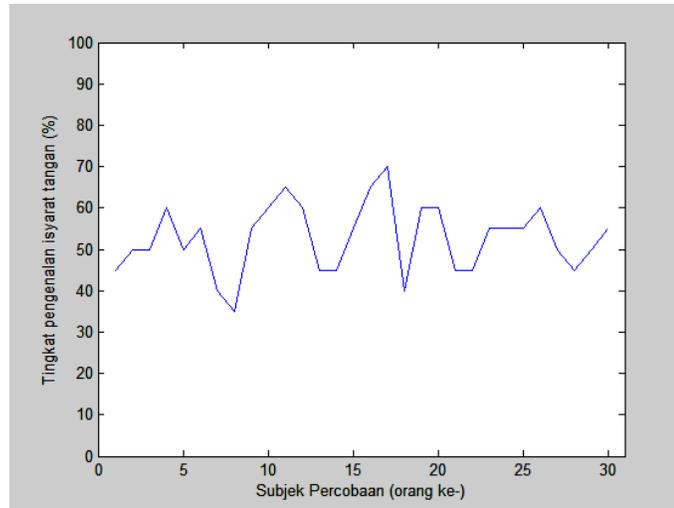
Gambar 2. Grafik Perubahan Arah Penelusuran



Gambar 3. Grafik Akumulasi Perubahan Arah Penelusuran

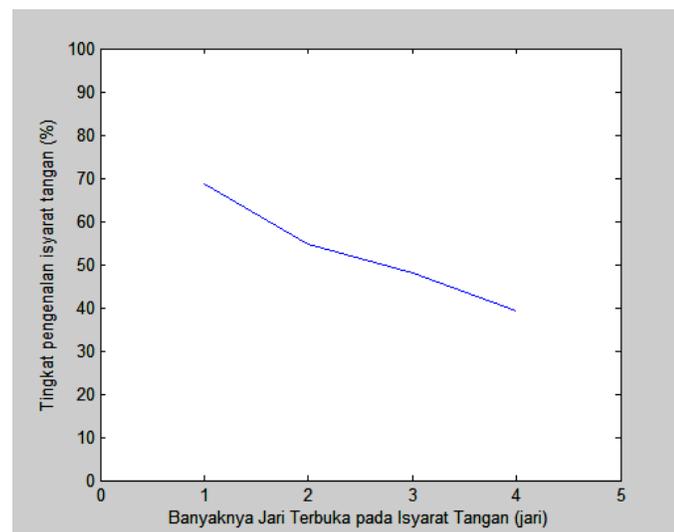
4. PENGUJIAN

Di dalam pengujian ini tingkat keberhasilan penterjemah kode isyarat tangan diuji dengan memberikan isyarat tangan awal. Pengujian dilakukan dengan latar belakan polos. Setelah isyarat tangan awal terdeteksi, isyarat tangan perintah diberikan pada batas terdeteksinya isyarat tangan awal. Berikut adalah grafik hasil pengenalan isyarat tangan untuk metode analisis deteksi tepi.



Gambar 4. Grafik Tingkat Pengenalan Isyarat Tangan Metode Analisis Deteksi Tepi

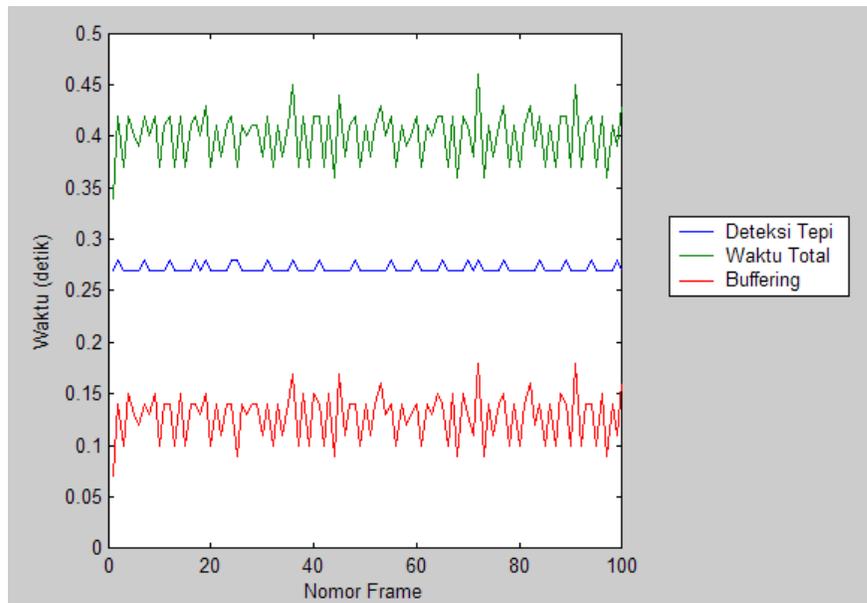
Dari hasil pengujian didapat rata-rata hasil pengenalan isyarat tangan untuk metode Analisis Deteksi Tepi sebesar 52,67%.



Gambar 5. Grafik Tingkat Pengenalan Isyarat Tangan Terhadap Isyarat Tangan

Dari hasil pengujian, semakin banyak jari tangan terbuka semakin rendah tingkat pengenalan isyarat perintah karena jalur tepian obyek tangan yang dideteksi panjang sehingga bisa menyebabkan kemungkinan adanya kesalahan pendeteksian.

Berikut ini adalah grafik waktu pemrosesan citra dengan menggunakan metode analisis deteksi tepi.



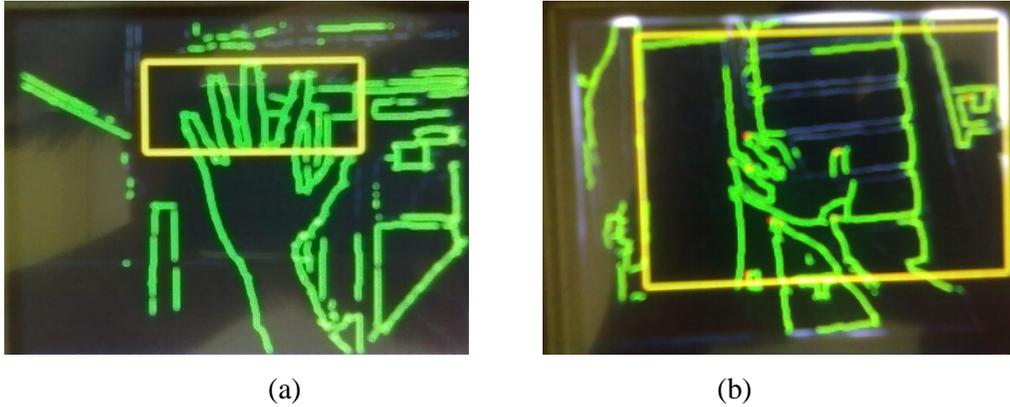
Gambar 6. Grafik Waktu Pemrosesan Citra Metode Analisis Deteksi Tepi

Dari hasil pengujian didapat rata-rata waktu pemrosesan citra 0,27 detik, rata-rata waktu pengambilan citra 0,4 detik, dan rata-rata waktu tunggu sebesar 0,13 detik. Pengujian isyarat tangan dengan menggunakan analisis deteksi tepi juga dilakukan pada latar belakang yang bervariasi seperti pada Gambar 7 berikut ini.



Gambar 7. (a) Tempat Pengujian Pertama (b) Tempat Pengujian Kedua

Hasil pengujian dapat dilihat pada Gambar 8. Dari hasil pengujian isyarat tangan dapat dilihat bahwa isyarat tangan tidak dapat dideteksi dengan stabil sehingga tingkat pengenalan isyarat tangan pada latar belakang bervariasi adalah 0%.



Gambar 8. (a) Hasil Deteksi Pada Tempat Pengujian Pertama
(b) Hasil Deteksi Pada Tempat Pengujian Kedua

5. KESIMPULAN

Pada sistem pengenalan isyarat tangan dengan menggunakan metode analisa deteksi tepi diperoleh tingkat pengenalan isyarat tangan sebesar 52,67%. Isyarat tangan tidak dapat terdeteksi dengan baik pada latar belakang yang bervariasi. Rata-rata waktu pemrosesan citra untuk metode Analisis Deteksi Tepi adalah 0,27 detik. Dari hasil pengujian tersebut, penterjemah kode isyarat tangan dengan menggunakan analisa deteksi tepi bisa diterapkan pada ARM 11 tetapi sulit diterapkan pada kontrol robot karena tingkat pengenalan isyarat tangan yang cukup rendah.

DAFTAR PUSTAKA

- [1] R. Lionnie, I. K. Timotius, and I. Setyawan, "An analysis of edge detection as a feature extractor in a hand gesture recognition system based on nearest neighbor," *Electrical Engineering and Informatics (ICEEI), 2011 International Conference on. IEEE*, pp.1–4, 2011.
- [2] J. Canny, "A computational approach to edge detection". *Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-8(6):679–698*, Nov. 1986.

- [3] R. C. Gonzalez, R. E. Woods, “Digital Image Processing Second Edition”, New Jersey: Prentice Hall, 2002.
- [4] -----, "OK6410-B Users Manual", Witech Co., Ltd, h. 5, 2010.