

Penerapan Formula Mean dan Varian untuk Pengolahan Citra Menggunakan Perangkat FPGA

Aqwam Rosadi Kardian and Sunny Arief Sudiro
STMIK Jakarta STI&K
Jakarta, Indonesia

{aqwam,sunny}@jak-stik.ac.id

Sarifuddin Madenda and Lingga Hermanto
Universitas Gunadarma
Jakarta, Indonesia

{sarif,lingga}@staff.gunadarma.ac.id

Abstract-Proses pengolahan citra membutuhkan rumus matematika yang diterapkan berdasarkan rumus pada data citra. Perangkat lunak mudah untuk melakukan hal itu dan mengakses data dari memori. Sebaliknya untuk implementasi perangkat keras dengan banyak kendala. Penelitian ini mengusulkan penerapan formula Mean dan Varian ke dalam Perangkat FGPA. Selain untuk perhitungan Optimal Mean dan Varian dengan hanya butuh satu komponen (dalam satu akumulator) dan satu pembagi menggunakan register geser kanan (Shift Right Register), untuk ukuran 8x8 citra diperlukan 64 clock putaran untuk menyelesaikan perhitungan mean dan varian.

Kata kunci—mean dan varian, histogram, FPGA, counter, citra

I. PENDAHULUAN

Pengolahan citra digital adalah proses untuk memanipulasi dan menganalisis citra dengan komputer. pengolahan citra dapat dibagi menjadi dua jenis kegiatan :1. Meningkatkan kualitas citra untuk menafsirkan lebih mudah oleh mata manusia, kegiatan ini diketahui sebagai peningkatan citra 2. Informasi Pengolahan dalam citra secara otomatis untuk mengidentifikasi objek. Aplikasi kedua memiliki korelasi yang erat dengan pengenalan pola yang biasanya dapat mengenali objek dengan mengekstraksi informasi penting dari citra (*fitur image*).

Beberapa pengolahan citra menggunakan rumus statistik dasar seperti : histogram, mean, varian, dan lainnya. Pada lingkungan sistem tertanam formula harus diterapkan dalam berbasis perangkat keras, seperti FPGA misalnya, suatu penelitian untuk perhitungan histogram untuk citra 8x8 perlu 256 bit selain itu, 320 siklus jam dan 256 jam untuk antarmuka dengan memori 2 Kb. [1]

Dalam penelitian lain untuk histogram dan perhitungan dasar mean pada perangkat FPGA diusulkan [2]. Artikel ini mengusulkan metode yang perlu mengoperasikan penjumlahan 512, 64 siklus, 1 pembagi (menggunakan Register Geser Kanan), tidak perlu untuk memori, tetapi menggunakan 256 register 8 bit untuk ukuran citra 8x8.

Dalam penelitian lain untuk perhitungan histogram telah diusulkan [3]. Pendekatan ini menggunakan RAM subcel dan

FSM untuk mengendalikan, pendekatan ini menggunakan 850 unsur logika FPGA dan dapat bekerja di lebih dari 100 MHz pada jam frekuensi dan mampu memproses 260 bingkai per detik

Artikel ini mengusulkan dengan metode yang lebih efisien untuk perhitungan mean dan varians dengan analisis statistik tekstur citra dan implementasi ke dalam perangkat FPGA. Sedangkan tujuannya untuk menghasikan formulasi dan algoritma untuk menghitung nilai mean dan varians yang lebih efisien dari sisi waktu proses dan jumlah sumberdaya yang digunakan, menghasikan metode transformasi algoritma penghitungan nilai mean dan varian kedalam bentuk desain rangkaian elektronika digital dan menghasikan sebuah deskripsi komponen elektronik berupa IP core untuk menghitung fitur tekstur mean dan varians.

Bagian pertama menggambarkan keadaan rumus penelitian yang sama. Bagian kedua adalah tentang rumus statistik dasar untuk pengolahan citra. Bagian ketiga algoritma yang diusulkan dengan Matlab. Bagian keempat adalah tentang efisien algoritma dasar penerapan pada perangkat FPGA. Terakhir menyimpulkan dari artikel ini.

II. DASAR PENGOLAHAN CITRA

A. Histogram Citra

Histogram Citra, adalah diagram yang menggambarkan kemunculan frekuensi pada setiap nilai intensitas dari unsur seluruh citra pixel. [4],[5]. Nilai yang lebih tinggi dari histogram menunjukkan jumlah piksel dengan yang nilai intensitas tinggi dan sebaliknya. Histogram dapat menunjukkan kecerahan dan kontras gambar. Histogram banyak digunakan dalam analisis citra tekstur karena kesederhanaan algoritma. Rumus matematika histogram adalah :

$$(H_i) = \sum_{n=0}^N \sum_{m=0}^M 1, \quad \text{if } i = f(n, m) \quad (1)$$

$H(i)$ adalah histogram dari citra dengan ukuran NxM dan i adalah nilai intensitas pixel dalam citra. Gambar 1. menunjukkan beberapa diagram histogram dari beberapa citra dengan tekstur masing-masing. Foto-foto tersebut diperoleh dari <http://www.freeimages.com/search/texture>

B. Formula Mean

Dari nilai histogram dapat menghitung nilai 'Mean' dari citra. Rumus matematika untuk 'Mean' (μ) adalah : [4]

$$\mu = \sum_{i=0}^{L-1} i \cdot p(i) \quad (2)$$

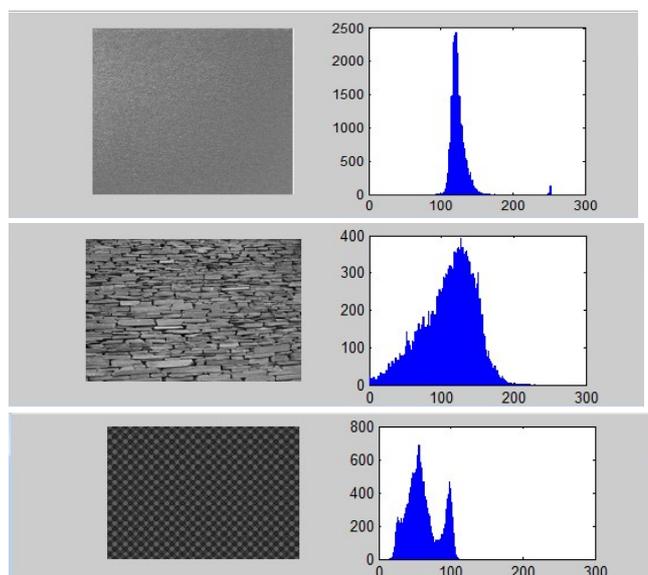
Pada rumus i adalah tingkat keabuan dari intensitas pixel dalam citra dan $p(i)$ adalah nilai kemungkinan dari kejadian i . L adalah nilai terbesar abu-abu tingkat i . Formula ini akan menghasilkan kecerahan rata-rata suatu objek.

C. Formula Varians

Fitur ketiga berupa deviasi standar (*variance*). Persamaan matematis standar deviasi.

$$\sigma^2 = \sum_{i=1}^{L-1} (i - m)^2 p(i) \quad (3)$$

σ^2 adalah dinamakan varians atau momen orde dua ternormalisasi karena $p(i)$ merupakan fungsi peluang. Fitur ini memberikan ukuran kekontrasan.



Gambar-1 : Beberapa contoh diagram histogram.

III. USULAN ALGORITMA

Mean dan varians adalah orde pertama metode analisis statistik yang digunakan untuk segmentasi dan ciri dari proses ekstraksi citra. Dimana Mean menggambarkan Nilai rata-rata (mean) yang merupakan intensitas dari seluruh piksel dalam citra, maka mengacu pada persamaan (2) dan (3) di mana mean dan varians yang dihitung dengan menggunakan hasil perhitungan histogram, diusulkan persamaan 4 seperti berikut :

$$\sigma^2 = \sum_{i=0}^{255} (i - \mu)^2 p(i) \quad (4)$$

Pseudocode perhitungan mean dan varian berdasarkan persamaan 4 ditunjukkan pada Pseudocode 1 ..

Pseudo code 1. Formula Mean dan Varian (1st versi pertama)

```
(1) h=0
(2) htsum = 0
(3) Vsum = 0
(4) For i ← 1 to M
(5)     For j ← 1 to N
(6)         h(I(i,j)) = h(I(i,j))+ 1
(7)         htsum = htsum + I(i,j)
(8)         Vsum= Vsum + I(i,j)2
(9)     End for
(10) End for
(11)  $\mu_{\text{optimal}} = \text{htsum}/M*N$ 
(12)  $\sigma_{\text{optimal}} = (\text{Vsum}/M*N) - \mu_{\text{optimal}}^2$ 
```

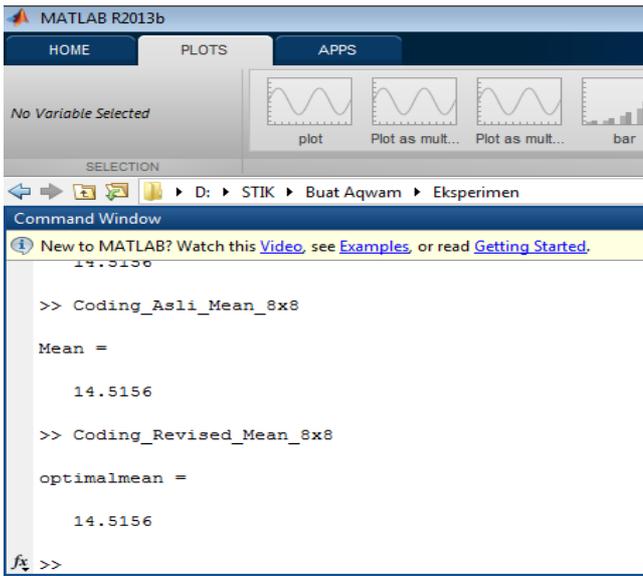
Pseudo code 2. Formula Mean dan Varian(versi revisi)

```
(1) [M,N,L]=size(Im);
(2) figure(1),imshow(uint8(Im));
(3) h=0;
(4) htsum=0;
(5) Vsum=0;
(6) for i=1:M;
(7)     for j=1:N;
(8)         htsum=htsum+Im(i,j,1);
(9)         Vsum=Vsum+Im(i,j,1)2;
(10)     end;
(11) end;
(12) optimalmean=htsum/(M*N)
(13) optimalvariansi=(Vsum/M*N)-optimalmean2 ;
```

Pseudo code 2 adalah algoritma direvisi dengan beberapa efisiensi, mengurangi lingkaran dan operasi matematika

Hasil perhitungan rata-rata menggunakan algoritma dasar dan algoritma optimasi untuk data yang sama (Im) adalah sama, seperti yang ditunjukkan pada Gambar 2. Nilai-nilai yang dihasilkan dari evaluasi menggunakan Matlab akan digunakan untuk perbandingan dengan nilai-nilai yang diproses menggunakan komponen FPGA. Kompleksitas waktu dari algoritma berkurang dengan menurunnya jumlah operasi aritmatika.

```
Im= [ 6 6 6 6 10 10 8 8
      6 6 48 48 10 10 8 8
      6 6 48 48 10 10 8 8
      6 6 48 48 13 13 10 10
      6 6 6 52 13 13 10 10
      5 5 5 52 13 13 10 10
      5 5 5 52 4 4 12 12
      5 5 5 52 4 4 12 12];
```

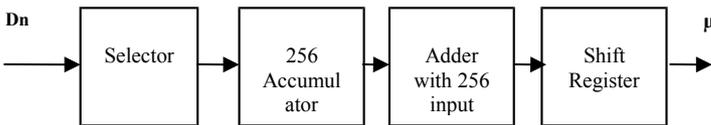


Gambar-2 : Hasil simulasi pseudocode

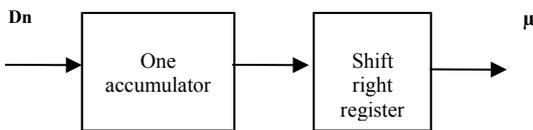
IV. HARDWARE IMPLEMENTATION

A. Disain Komponen

Gambar 3 adalah desain pertama berdasarkan kode pseudo pertama, desain ini perlu pemilih untuk memilih nilai intensitas pixel dari citra antara 0 sampai 255 dan di kirim ke masing-masing accumulator (256 accumulator dengan 256 komponen tambahan). Semua nilai accumulator tersebut dikirimkan ke 256 input tambahan. Selain input (atau 256 tambahan lain) sehingga untuk desain ini kita memerlukan input tambahan 512. Terakhir untuk mendapatkan nilai Mean kita bagi dengan jumlah pixel yaitu $8 \times 8 = 64$. Kami menggunakan Register Geser Kanan 6 bit untuk melakukan operasi pembagian ini.



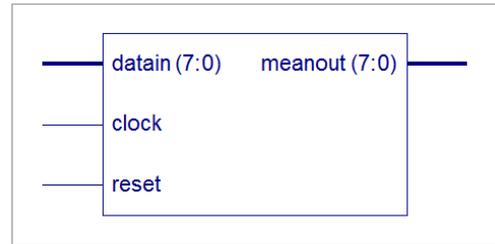
Gambar-3 : Component design for first pseudo code[2]



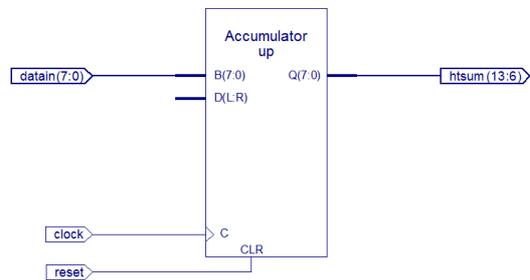
Gambar-4 : New Component design for 2nd pseudo code.

Gambar 5 dan 6, adalah desain baru berdasarkan desain pada gambar 4, untuk menghitung nilai Mean sebenarnya tidak harus menghitung histogram, dan bisa langsung menggunakan salah satu Penjumlahan (accumulator) untuk mendapatkan nilai nilai total elemen pixel (htsum) di kode pseudo 2. Pada pendekatan ini memberi pengurangan operasi

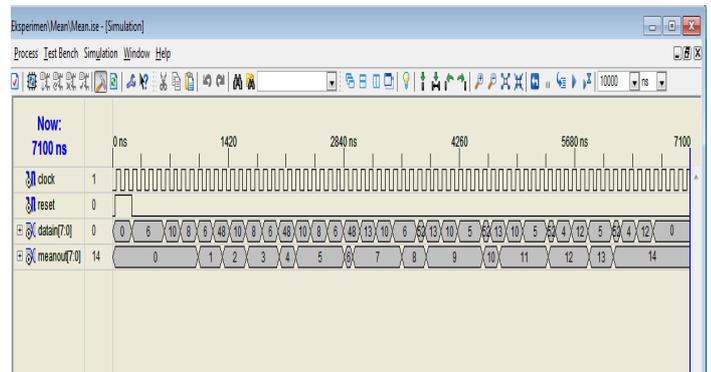
matematika (komponen), dengan desain ini hanya perlu satu tambahan dalam satu accumulator dan satu Register Geser (Shift Register), dengan waktu yang sama pengolahan (64 siklus jam) untuk ukuran citra 8x8 dan hasil yang sama dari nilai Mean dan Varians.



Gambar-5 : Entity Diagram of efficient component



Gambar-6 : RTL Schematic Diagram of efficient component.



Gambar-7 : Hasil simulasi praktek untuk komponen yang diusulkan

B. Simulasi

Gambar 6 menunjukkan unjuk kerja berdasarkan hasil simulasi praktek. Angka ini menunjukkan hasilnya untuk data yang sama yang 14, pada akhir ke 64 clock putaran, dibandingkan dengan Matlab hasil evaluasi yang adalah 14,5 perbedaannya adalah 0,5 atau kesalahan 0,005%. Kecepatan proses adalah paralel dengan masukan data yang datang (data in).

V. KESIMPULAN

Penerapan efisien formula mean dan varians yang diusulkan ke dalam komponen perangkat keras menggunakan perangkat FPGA, merupakan suatu bentuk satu tambahan

penggunaan komponen satu Register Geser Kanan (Shift Right Register) dengan 64 clock putaran untuk menghitung nilai mean dengan nilai ukuran citra 8x8. Desain ini membutuhkan 14 keping flip-flops dan 14 dari 4 LUTs masukan. Perbedaan (sebagai kesalahan) dari nilai mean dengan hasil Matlab adalah 0,5 atau 0,005% karena disebabkan masalah pada floating point dalam FPGA.

PENGHARGAAN

Ucapan terima kasih kepada, Yayasan Pendidikan Gunadarma dan Yayasan Ilmu Komputer untuk mendukung penelitian ini sebagai penelitian Doktor di Program Doktor Universitas Gunadarma, Jakarta Indonesia .

DAFTAR PUSTAKA

- [1] Nitin Sachdeva and Tarun Sachdeva, “ An FPGA Based Real-time Histogram Equalization Circuit for Image Enhancement”, International Journal of Electronics and Communication Technology (IJECT) Vol 1 Issue 1 December 2010.
- [2] Atit Pertiwi, “Optimalisasi Metode Implementasi Algoritma Histogram, Mean Dan Variansi Ke Dalam IC-FPGA Untuk Aplikasi Analisis Tekstur Citra Real Time”, Disertasi, Teknologi Informasi, Universitas Gunadarma, 2015.
- [3] Luca Maggiani, Claudio Salvadori, Matteo Petracca, Paolo Pagano, Roberto Saletti, “ Reconfigurable Architecture for Computing Histograms in Real-Time Tailored to FPGA-based Smart Camera”, Industrial Electronics (ISIE), IEEE 23rd International Symposium on , Istambul Turkey, June 2014.
- [4] Wendy L. Martinez and Angel R. Martinez, “Computational Statistics Handbook with MATLAB”, Chapman & Hall/CRC, USA, 2002.
- [5] R.E. Woods , S.L. Eddins , dan R.C. Gonzales, “Digital Image Processing Using MATLAB . Pearson Educatio, 2005.