

### 3. Analisis

Pada bab ini diuraikan mengenai proses analisis kedua parameter sebagai salah satu cara penentuan kualitas gambar masukan pada aplikasi JTenun serta metode yang akan digunakan untuk mendeteksi kedua parameter tersebut.

#### 3.1 Tahapan Analisis

Pada tahapan analisis, dilakukan berbagai pengamatan dan analisis terhadap hasil studi *literature* dan eksplorasi. Tahapan analisis yang dilakukan adalah sebagai berikut.

1. Analisis algoritma FFT dan *Laplacian Operator* dalam mendeteksi *blur*.
2. Analisis algoritma PCA dalam mendeteksi *noise*.
3. Analisis kaitan *blur* dan *noise* dengan karakteristik Ulos.
4. Kesimpulan analisis.

#### 3.2 Analisis Algoritma Fast Fourier Transform dan Laplacian Operator

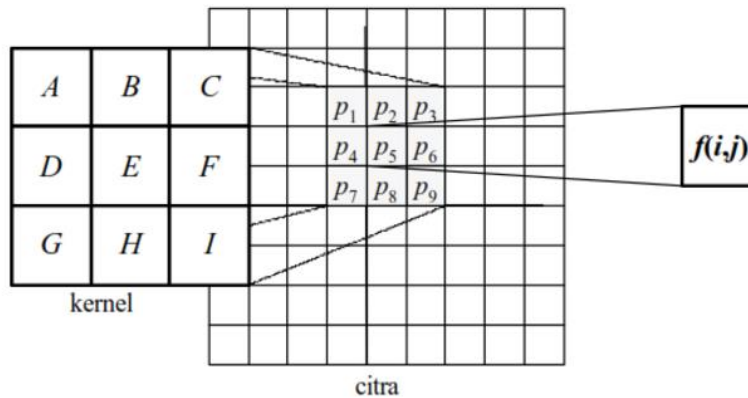
Parameter *blur* akan dihitung dengan menggunakan dua algoritma yang memiliki peranan berbeda yaitu, algoritma *Fast Fourier Transform* dan *laplacian operator*. Awalnya *laplacian operator* yang merupakan operator turunan kedua akan melakukan pendeteksian tepi dengan tujuan untuk memperlihatkan detail gambar. Berikut alur dari pendeteksian yang dilakukan oleh *laplacian* pada gambar masukan:

1. *Input* gambar ke dalam program kemudian gambar atau citra masukan akan diubah dan dibaca dalam bentuk matriks. Pembacaan ini dilakukan berdasarkan warna piksel pada gambar masukan.
2. Selanjutnya, citra masukan akan diubah kedalam gambar berskala keabuan (*grayscale*).
3. Ketika suatu gambar masukan diubah kedalam bentuk matriks, maka terdapat ukuran matriks gambar yang berbeda-beda untuk setiap gambar. Untuk gambar dengan warna RGB maka bentuk matriks akan diubah dalam bentuk matriks biasa (piksel). Setelah itu matriks yang dihasilkan oleh gambar akan dikonvolusi dengan *kernel* 3 x 3, seperti matriks di Gambar 6:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Gambar 1 kernel 3x3

4. Dalam ranah diskrit *kernel*, konvolusi dinyatakan dalam bentuk matriks (umumnya 3 x 3, namun ada juga yang berukuran 2 x 2 atau 2 x 1 atau 1 x 2). Ukuran matriks ini biasanya lebih kecil dari ukuran citra masukan. Setiap elemen matriks disebut koefisien konvolusi.
5. Berikut ilustrasi konvolusi ditunjukkan pada Gambar 7 di bawah ini:



Gambar 2 Ilustrasi kernel

6. Operasi konvolusi dilakukan dengan menggeser *kernel* konvolusi piksel per piksel. Hasil konvolusi disimpan di dalam matriks yang baru. Namun piksel-piksel pinggir diabaikan yang artinya tidak dilakukan konvolusi, maka piksel-piksel pinggir nilainya tetap sama seperti citra asal. Konvolusi dalam pengolahan citra dapat dilakukan untuk banyak hal. Pada Tugas Akhir ini, peneliti mengkonvolusi citra masukan untuk mendeteksi tepi dalam citra.
7. Kemudian, proses ini akan dilanjutkan dengan perhitungan FFT. Pada FFT akan dilakukan perhitungan varians dan standar deviasi kuadrat. Standar deviasi adalah nilai statistik yang dimanfaatkan untuk menentukan bagaimana sebaran data pada citra masukan.

8. Untuk menghitung nilai varians, pertama-tama harus diketahui nilai rata-rata dari data yang ada. Berikut diberikan contoh matriks gambar masukan pada Gambar 8:

	0	1	2	3	4
3	180	179	182	191	192
4	177	174	171	177	183
5	177	169	167	164	165
6	173	169	175	172	162
7	169	174	175	165	159
8	173	178	177	179	175
9	178	179	182	180	177
10	181	179	176	177	175
11	181	186	184	179	170
12	172	165	168	167	156
13	168	157	156	157	145
14	160	158	154	151	147

Gambar 3 Contoh matriks gambar masukan Matriks pada Gambar 8 dengan ukuran 700 x 533 dan dimulai dari indeks 0. Dari matriks inilah akan dilakukan perhitungan varians dan standar

deviasi. Sebelum itu, harus diketahui jumlah rata-rata pada keseluruhan matriks. Setelah rata-rata pada matriks diketahui maka akan dihitung variansnya. Rumus varians dapat dilihat pada Gambar 9:

$$= \frac{\sum (x - \bar{x})^2}{n - 1}$$

Gambar 4 Rumus Varians

Keterangan:

$X$  = nilai matriks per piksel

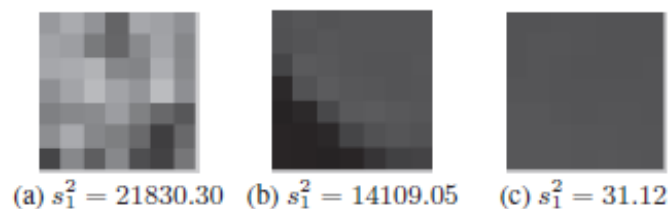
$\bar{X}$  = nilai rata-rata piksel

$n$  = jumlah data

Perhitungan varians pada gambar masukan akan menggunakan rumus varians seperti pada Gambar 9, dimana perhitungan ini didasarkan pada matriks keluaran. Sedangkan perhitungan standar deviasi adalah nilai kuadrat dari hasil perhitungan varians. Penelitian ini memiliki asumsi bahwa sebuah citra masukan yang memiliki jumlah persebaran data yang rendah pada titik-titik frekuensi yang luas maka akan disebut dengan daerah *high frequency* yang hasilnya adalah gambar *blur*, begitu juga sebaliknya. Citra masukan dengan persebaran data yang kecil pada daerah *high frequency* akan dianggap sebuah gambar yang *blur*. Semakin tinggi nilai *output* dari *laplacian* maka gambar akan dianggap sebagai gambar yang tidak *blur* dan sebaliknya semakin rendah nilai *output* maka gambar akan dianggap *blur*.

### 3.3 Analisis Algoritma PCA

Analisis Algoritma PCA (Komponen Utama) banyak digunakan untuk memproyeksikan atau mengubah suatu kumpulan data berukuran besar menjadi bentuk sajian data dengan ukuran yang lebih kecil. PCA digunakan menghitung nilai *noise* pada suatu gambar dari pemilihan WTP (*weak textured patch*). Zhu dan Milanfar mendemostrasikan bahwa struktur dari gambar dapat diukur dengan efektif melalui matriks kovarian [31]. Nilai *noise* pada gambar dapat diukur apabila *weak texture patch* pada gambar dipilih terlebih dahulu. *Patch* pada gambar akan direpresentasikan sebagai matriks yang merepresentasikan operator turunan *vertical* dan *horizontal*. Dalam pemilihan *patch* ada asumsi bahwa *weak textured patch* merupakan “*patch* atau potongan gambar yang memiliki tekstur relatif *smooth* (lembut)”.



Gambar 5 Patches dengan nilai eigen

Dapat dilihat dari Gambar 10, bahwa *weak textured patches* adalah gambar yang paling kanan karena memiliki nilai *eigen* yang paling kecil. Jika berbicara mengenai *threshold*, maka *threshold* akan diberikan secara manual oleh *programmer* misalnya, 0.99. Berikut penjelasan mengenai pemilihan *patch* dan penghitungan nilai *noise* dari gambar:

1. Sebuah gambar masukan akan diproses untuk perhitungan nilai *noise*. Gambar Ulos yang sudah dimasukkan akan diuji coba dengan nilai *threshold* pada rentang keabuan antara 0-255 (*grayscale*).
2. Dilanjutkan dengan segmentasi ukuran citra menjadi beberapa bagian.
3. Selanjutnya, proses ekstraksi ciri dilakukan setelah mendapatkan citra *grayscale*.
4. Kemudian akan dilakukan proses perhitungan nilai *noise* dari matriks kovarian, dimana telah dihasilkan oleh semua *patch* pada gambar masukan. Tahapan berikutnya adalah menghitung nilai *eigen* (E) dari matriks kovarian, setelah itu dilakukan pengurutan nilai *eigen* kemudian dicari *patches* dengan nilai *eigen* paling kecil. Tahap ini dilakukan sebanyak tiga kali iterasi agar didapatkan nilai *noise* yang stabil, namun hal ini bisa ditentukan jumlahnya pada parameter fungsi. Tahap terakhir identifikasi *noise* pada gambar didapat dari gambar masukan. Untuk penghitungan nilai *eigen* melibatkan perhitungan aljabar linear, dapat dilihat dari penelitian Xinhao Liu, 2012 [22].

### **3.4 Analisis kaitan parameter blur dan noise dengan karakteristik motif Ulos**

Beberapa motif Ulos memiliki karakteristik atau corak yang timbul seperti Ulos Sadum. Hal ini menyebabkan corak pada gambar Ulos terdeteksi sebagai *noise* dan akan mempengaruhi program pendeteksi *noise* yang akan meningkatkan nilai *noise* pada gambar.

### **3.5 Kesimpulan Analisis**

Berdasarkan hasil analisis yang dilakukan, terdapat beberapa hal yang menjadi kesimpulan dan pertimbangan dalam tahap implementasi yang akan dilakukan:

1. Proses evaluasi gambar akan dilakukan dengan tiga eksperimen yaitu memeriksa dan menghitung nilai dari 1 (satu) parameter terlebih dahulu yaitu evaluasi gambar untuk

parameter *blur* dan evaluasi gambar untuk parameter *noise*, dan yang ketiga dilakukan secara sekuensial yaitu sebagai berikut, pertama dilakukan pemeriksaan dan penghitungan nilai dari parameter *blur* kemudian sebagian hasil dari evaluasi *blur* akan dilakukan pengecekan parameter *noise*.

2. Parameter *blur* dideteksi dengan algoritma *laplacian operator* dan FFT.
3. Parameter *noise* dideteksi dan dihitung dengan algoritma PCA (*Principal Component Analysis*).