# IMAGE QUILTING DALAM PERBESARAN CITRA TEKSTUR

**Diah Arifah P.**

## ABSTRAK

*Tekstur dapat membuat objek-objek terlihat realistis, misalnya pada latar belakang suatu game. Objek yang berukuran yang besar akan memerlukan tekstur yang besar pula. Cara yang biasa digunakan oleh user untuk memperbesar citra yaitu dengan tiling dan stretching. Namun keduanya masih memiliki berbagai kekurangan. Kekurangan tersebut diperbaiki dengan menggunakan algoritma Image Quilting yang membentuk suatu tekstur baru dengan menggabungkan blok-blok yang diambil dari tekstur aslinya dan memiliki keuntungan yaitu tekstur yang dihasilkan memiliki kualitas yang hampir mirip dengan tekstur aslinya, serta prosesnya yang tidak membutuhkan waktu yang terlalu lama. Teknik ini memberi hasil yang cukup memuaskan dalam penerapannya pada citra dengan berbagai jenis tekstur yang berbeda. Sedangkan untuk kualitas citra yang dihasilkan aplikasi ini adalah cukup baik, terutama untuk citra yang memiliki pola tekstur yang jelas.*

Kata kunci : *Image Qualting, tekstur, perbesaran citra*

## PENDAHULUAN

Suatu objek dapat ditambahkan tekstur, sehingga tampilan tampak lebih indah Tekstur ini sebagai corak dari sebuah citra, misalnya corak dari kayu, granit, marmer ataupun pada ubin. Tekstur dapat membuat objek-objek terlihat realistis, misalnya pada pembuatan *game*, Objek dalam *game* tersebut bisa terlihat lebih nyata, jika memakai tekstur sebagai latar belakang dari *game* tersebut. Beberapa objek memiliki ukuran yang besar sehingga diperlukan juga tekstur yang sesuai dengan ukuran objek.

Dalam memperbesar tekstur diperlukan cara yaitu dengan *tiling* dan *stretching*. Pada metode *Tilling* akan memperbanyak suatu citra dengan cara mengulang citra yang ada secara otomatis sampai suatu ukuran yang diinginkan terpenuhi. Kekurangannya hasil tekstur terlihat lebih janggal karena terlihat semacam ada garis-garis memotong setiap perulangan citra sehingga hasilnya seperti terputus-putus. Pada metode *stretching* akan memperbesar atau memperkecil citra dengan cara menarik citra sampai besarnya sesuai dengan ukuran yang diinginkan. Kekurangannya hasil tekstur akan terlihat kurang jelas atau pecah-pecah jika teksturnya diperbesar.

Kedua metode tersebut dapat diperbaiki dengan menggunakan metode sintesa tekstur dengan menggunakan algoritma *Image Quilting*. Dengan algoritma ini akan membentuk suatu tekstur baru dengan menggabungkan blok-blok yang diambil dari tekstur aslinya sehingga dapat memperbesar semua jenis tekstur (dari *stochastic* sampai tekstur

Sekolah Tinggi Informatika dan Komputer Indonesia (STIKI) Malang
Jl. Raya Tidar 100 Malang
Email : diah@stiki.ac.id

*regular*) dengan waktu proses yang lebih cepat.

## DASAR TEORI

Tekstur menunjukkan sifat-sifat atau karakteristik yang dimiliki oleh suatu daerah yang cukup besar sehingga secara alami sifat-sifat tadi dapat berulang dalam daerah tersebut. Tekstur dalam hal ini kurang lebih berupa keteraturan pola-pola tertentu secara berulang-ulang dengan interval jarak dan arah tertentu.yang terbentuk dari susunan piksel-piksel dalam citra digital. Suatu permukaan dikatakan mempunyai suatu informasi tekstur, bila luasannya diperbesar tanpa mengubah skala, maka sifat-sifat permukaan hasil perluasan mempunyai kemiripan dengan permukaan asalnya.

Informasi tekstur dapat digunakan untuk membedakan sifat-sifat permukaan suatu benda dalam citra yang berhubungan dengan kasar dan halus, juga sifat-sifat spesifik dari kekasaran dan kehalusan permukaan. Syarat terbentuknya tekstur setidaknya ada dua, yaitu :

1. Adanya pola-pola primitif yang terdiri dari satu atau lebih piksel. Bentuk-bentuk pola primitif ini dapat berupa titik, garis lurus, garis lengkung, luasan, dan lain-lain yang merupakan elemen dasar dari sebuah bentuk.
2. Pola-pola primitif tadi muncul berulang-ulang dengan interval jarak dan arah tertentu sehingga dapat diprediksi atau ditemukan karakteristik pengulanggannya.

Dengan adanya dua syarat ini sebenarnya tekstur suatu permukaan dapat dibentuk melalui aturan-aturan yang berlaku, atau sebaliknya, yaitu suatu permukaan dapat dianalisis teksturnya dengan aturan-aturan yang sama untuk diperbandingkan satu sama lain, atau untuk keperluan *interprestasi digital*[1].

Menurut spektrumnya, tekstur digolongkan menjadi 2 macam, yaitu tekstur *stochastic* dan tekstur *regular*.

1. Tekstur *stochastic*

Tekstur ini susah untuk diidentifikasikan secara primitif, karena terdiri dari aturan yang bermacam-macam dan bermacam-macam warna. Banyak tekstur kelihatan seperti tekstur *stochastic* ketika dipandang dari jauh. Contoh dari suatu tekstur *stochastic* misalnya pada pasir, granit, kulit kayu atau batu-batuan.

2. Tekstur *Regular*

Tekstur ini terdiri dari sekumpulan aturan primitif (misalnya hanya terdiri dari garis, lingkaran, kotak atau sebuah model tekstur sederhana saja) dan sebuah jarak perulangan sederhana. Contoh dari tekstur

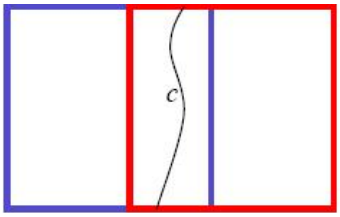
regular adalah suatu dinding tembok, suatu ubin lantai dan papan catur[2] dapat dilihat pada gambar 1.


Gambar 1. Tekstur *Stochastic* ke Tekstur *Regular*

*Image Quilting* merupakan metode sintesa tekstur untuk membentuk suatu tekstur baru dengan menggabungkan blok-blok yang diambil dari tekstur aslinya.

**Minimum *Error Boundary Cut***

Minimum *error boundary cut* digunakan untuk memperkecil perbedaan antara blok yang bersebelahan. Gagasan di belakang *minimum error boundary cut* adalah sederhana. Mengasumsikan kita diberi dua *overlap* blok dalam garis merah dan biru seperti ditunjukkan di Gambar 2.




Gambar 2. Dua *overlap* Blok dalam garis merah dan biru.

Sasarannya adalah untuk mencari suatu kurva *c* antar suatu keluarga dari semua kurva yang mungkin di dalam *overlap region*, seperti piksel blok yang bergaris biru dan piksel blok yang bergaris merah adalah semirip mungkin atas piksel yang dihubungkan oleh kurva ini. Piksel di sebelah kiri kurva akan masuk blok yang bergaris biru dan piksel di sebelah kanan dari kurva akan masuk blok yang bergaris merah.

**Proses *Image Quilting***

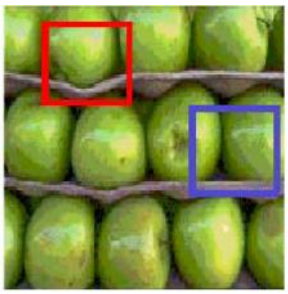
Tahapan dalam algoritma *Image Quilting* adalah sebagai berikut :

- **Proses Inisialisasi**

Dalam proses inisialisasi membuka *file* citra tekstur asal yang akan diproses dengan algoritma *image quilting*.

- **Proses Mendapatkan Kandidat Blok dan Nilai *Error*nya**

Dalam proses ini diambil blok-blok citra pada tekstur asli, hal ini dapat ditunjukkan pada gambar 3.



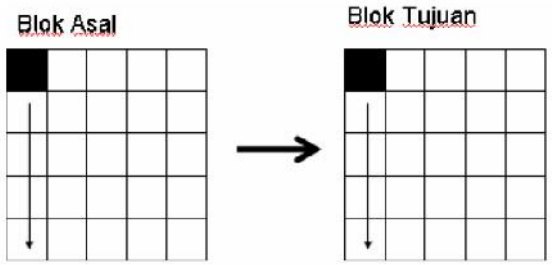
Gambar 3. Blok-blok citra untuk kandidat blok

Tujuan pengambilan blok-blok citra adalah sebagai kandidat atau calon blok citra yang akan disusun untuk membentuk tekstur baru. Semakin besar jumlah kandidat blok, maka *sampel* kemungkinan blok yang akan

dipilih sebagai blok penyusun berikutnya akan semakin banyak, sehingga hasilnya akan semakin halus tetapi proses yang dijalankan akan memakan waktu yang lebih lama.

Langkah selanjutnya adalah memilih blok-blok yang cocok untuk berdampingan. Cara untuk memilih blok yang akan dipasangkan adalah dengan membandingkan nilai *error* tiap blok untuk dicari yang paling kecil. Semakin kecil nilai *error*nya maka bisa dikatakan kedua blok tersebut memiliki kemiripan.

Nilai *error* tiap blok dihitung dengan cara menjumlahkan nilai *error* tiap piksel yang terdapat pada blok tersebut. Dengan cara dengan melibatkan komponen dasar piksel berdasar warna R (*Red*), G (*Green*), B (*Blue*) pada tiap-tiap piksel, hal ini ditunjukkan pada gambar 4.

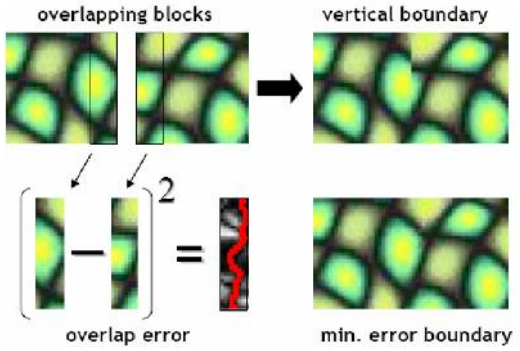



Gambar 4. Menghitung nilai *error* tiap blok

Piksel pada kiri atas pada blok citra asal dihitung dengan piksel kiri atas pada blok citra tujuan sampai piksel pada kiri bawah pada blok citra asal dihitung dengan piksel kiri bawah pada blok citra tujuan, kemudian piksel sebelahnya dengan cara yang sama sampai seluruh piksel didapat nilai *error*nya.

- **Proses Menghitung *Minimum Boundary Cut* Vertikal dan Horisontal**

Proses ini dilakukan untuk memperhalus hasil tekstur agar tidak tampak kesan terkotak-kotak atau terbagi blok-blok. *Minimum error boundary cut* dapat diartikan sebagai kurva yang memotong permukaan dua blok yang saling ber-*overlap*. Cara mendapatkan kurva ini adalah dengan menghitung nilai error antar piksel pada permukaan *overlap*. Proses perhitungan ini dapat dilihat pada gambar 5.




Gambar 5. Proses *Minimum Error Bounding Cut*

Langkah-langkah yang dilakukan dalam menghitung *minimum boundary cut* vertikal adalah :

1. Menghitung nilai *error* tiap piksel dengan rumus $E_{ij} = e_{ij} + \min(E_{i-1,j-1}$ ,

$E_{i-1,j}$ , $E_{i-1,j+1}$ ), dimana $e_{ij}$ adalah *error* antara piksel asal dengan piksel tujuan

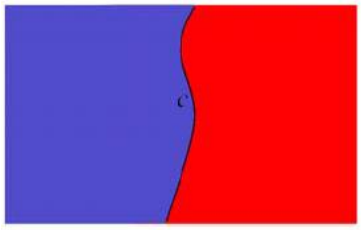
2. Membuat kurva *minimum boundary cut vertical* dengan cara membandingkan piksel yang terletak pada satu baris, lalu berlanjut pada baris-baris berikutnya sampai seluruh baris terhitung.

Langkah-langkah yang dilakukan dalam menghitung *minimum boundary cut* horisontal adalah :

1. Menghitung nilai *error* tiap piksel dengan rumus $E_{ij} = e_{ij} + \min(E_{i-1,j-1}$ , $E_{i-1,j}$ , $E_{i-1,j+1}$ ), dimana $e_{ij}$ adalah *error* antara piksel asal dengan piksel tujuan
2. Membuat kurva *minimum boundary cut vertical* dengan cara membandingkan piksel yang terletak pada satu kolom, lalu berlanjut pada kolom-kolom berikutnya sampai seluruh kolom terhitung. Nilai *error* piksel yang dibandingkan adalah nilai antara tiga piksel yang terletak pada satu baris sebelah atas baris sebelumnya, baris yg sama dengan baris sebelumnya, dan satu baris sebelah bawah baris sebelumnya.

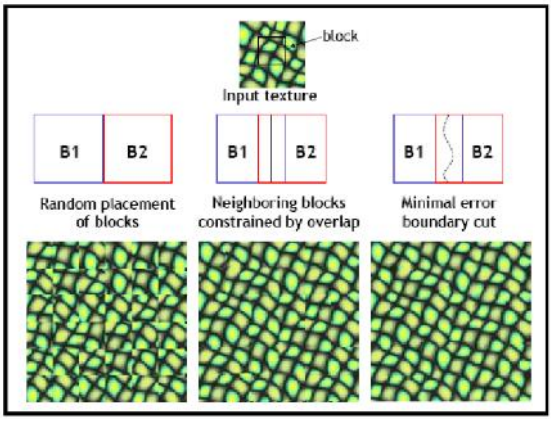
- **Proses Mengcopy Blok dari Citra asal ke Citra tujuan**

Proses penyusunan blok ke dalam citra tekstur yang baru berdasarkan nilai *overlap* yang sudah ditentukan dan nilai *minimum error boundary cut* yang telah dihitung. Apabila koordinat piksel kedua blok yang ber*overlap* terletak pada kurva maka kedua piksel tersebut dicampur dan di-*copy* pada citra tekstur baru, sedangkan apabila koordinat piksel tidak termasuk sebagai penyusun kurva, maka piksel tersebut langsung di-*copy* pada citra tekstur baru, dapat dilihat pada gambar 6.



Gambar 6. Daerah yang warna pikselnya dicampur

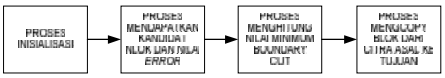
Proses *Image Quilting* secara keseluruhan dapat dilihat pada gambar 7.




Gambar 7. Proses *Image Quilting*

## METODE

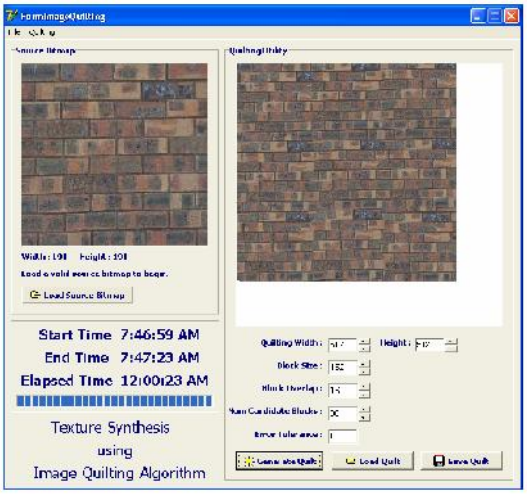
Berikut ini merupakan tahapan penelitian seperti ditunjukkan pada gambar 8.



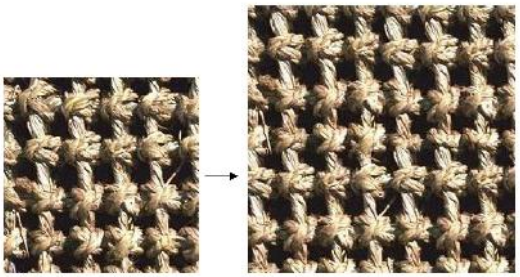

Gambar 8. Diagram blok tahapan penelitian

## HASIL PENELITIAN

Hasil penelitian dapat dilihat pada tampilan program simulasi

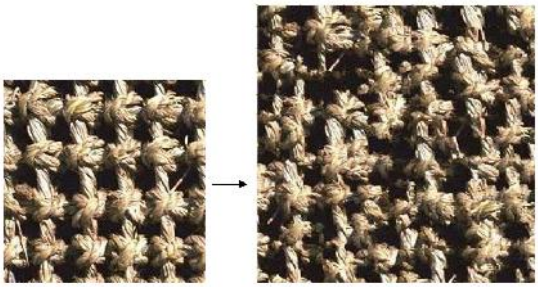



Gambar 9. Layout setelah proses



Gambar 9. Citra Anyaman Tali dengan Nilai Blok Besar (Ukuran Blok = 152)

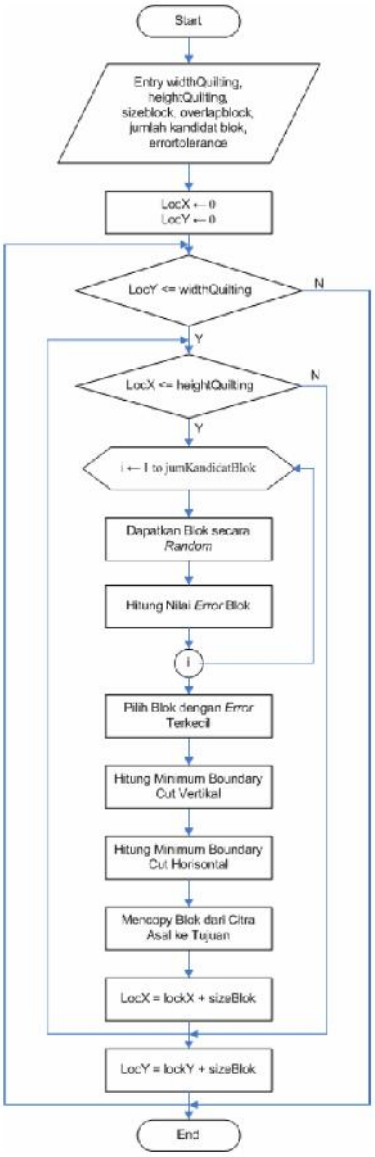
Dari gambar 9 dapat dilihat bahwa dengan lebar blok berukuran besar, citra masukan dan citra hasil masih terlihat mirip. Jika lebar blok terlalu kecil maka akan didapat citra hasil yang tidak mirip dengan citra asli. Hal ini disebabkan karena ukuran blok tidak cukup menangkap pola tekstur dari citra yang asli sehingga citra hasil kehilangan pola struktur, hal ini dapat dilihat pada gambar 10.



Gambar 10. Citra Anyaman Tali dengan Nilai Blok Kecil(Ukuran Blok = 24)

## PEMBAHASAN

Secara keseluruhan dari diagram alir algoritma *image quilting* ditunjukkan pada gambar 11.




Gambar 11. Aliran data algoritma *image quilting*

- Proses Inisialisasi

*User* memilih citra bertekstur dengan file ekstensi *.bmp dan memasukkan nilai dari enam variabel, yaitu : ukuran lebar citra hasil, ukuran tinggi citra hasil, ukuran lebar*tinggi blok citra, batas *overlap* antar blok, jumlah kandidat blok, toleransi *error*.

- Proses mendapatkan kandidat blok dan nilai *error*nya

Dalam proses ini akan memilih blok yang akan dipasangkan dengan cara membandingkan nilai *error* tiap blok. Cara mendapatkan nilai *error* tiap blok adalah melakukan perulangan sebanyak lebar*tinggi citra blok untuk mendapatkan nilai *error* antar piksel citra asal dan citra tujuan dan menghitung total *error* yang didapatkan.

Untuk mendapatkan nilai *error* tiap piksel dengan cara hitung nilai *error* masing-masing komponen warna R(*Red*), G(*Green*), B(*Blue*) kemudian jumlahkan nilai *error* dari masing-masing komponen warna tersebut.

- Proses Menghitung *Minimum Boundary Cut* Vertikal dan Horisontal

Algoritma untuk menghitung *Minimum Boundary Cut* Vertikal adalah :

1. Buat array matrik 2 dimensi dengan ukuran lebar * tinggi
2. Lakukan perulangan sebanyak ukuran tinggi:
   a. Dapatkan nilai *error* dengan menghitung piksel asal dengan piksel tujuan dimana piksel-piksel tersebut terletak di sebelah paling kiri dari citra blok.

b. Simpan pada array matrik yang menyatakan posisi piksel sebelah kiri dari citra blok.

3. Lakukan perulangan sebanyak ukuran lebar

**Untuk baris piksel paling atas :**

a. Dapatkan nilai Eij dengan rumus sebagai berikut :

   $Eij = e_{ij} + \min(E_{i-1,j-1}, E_{i-1,j}, E_{i-1,j+1})$

   Mendapatkan nilai *error* dengan menghitung piksel asal dengan piksel tujuan dimana piksel-piksel tersebut terletak di sebelah paling atas dari citra blok

b. Dapatkan Ei,0 dengan menggunakan rumus:

   $Eij = e_{ij} + \min(E_{i-0}, 0+1, E_{i-1}, 0)$

c. Simpan pada array matrik yang menyatakan posisi piksel sebelah atas dari citra blok

**Untuk baris piksel paling bawah :**

a. Dapatkan nilai Eij untuk mendapatkan nilai *error* dengan menghitung piksel asal dengan piksel tujuan dimana piksel- piksel tersebut terletak di sebelah paling bawah dari citra blok

b. Simpan pada array matrik yang menyatakan posisi piksel sebelah atas dari citra blok.

**Untuk baris piksel tengah :**

a. Dapatkan nilai Eij untuk mendapatkan nilai *error* dengan menghitung piksel asal dengan piksel tujuan dimana piksel- piksel tersebut terletak di sebelah tengah(antara atas dan bawah) dari citra blok.

b. Ei,j dengan menggunakan rumus:

   Ei,j = eij + min(Ei-1, j-1 , Ei-1,j , Ei-1,j+1 )

c. Simpan pada array matrik yang menyatakan posisi piksel sebelah atas dari citra blok

d. Buat array kurva sebesar ukuran lebar.

e. Lakukan perulangan sebanyak ukuran tinggi.

   ▪Cari nilai *error* paling minimal pada antara piksel sebelah paling kanan.

f. Simpan nilai tersebut sebagai nilai awal kurva.

g. Lakukan perulangan sebanyak ukuran lebar-1 dari posisi kolom piksel kanan ke kiri.

   ▪Cari nilai error piksel paling minimal antara piksel dengan posisi y-1, y, y+1 pada kolom sebelah kiri piksel awal, y adalah posisi baris piksel awal.

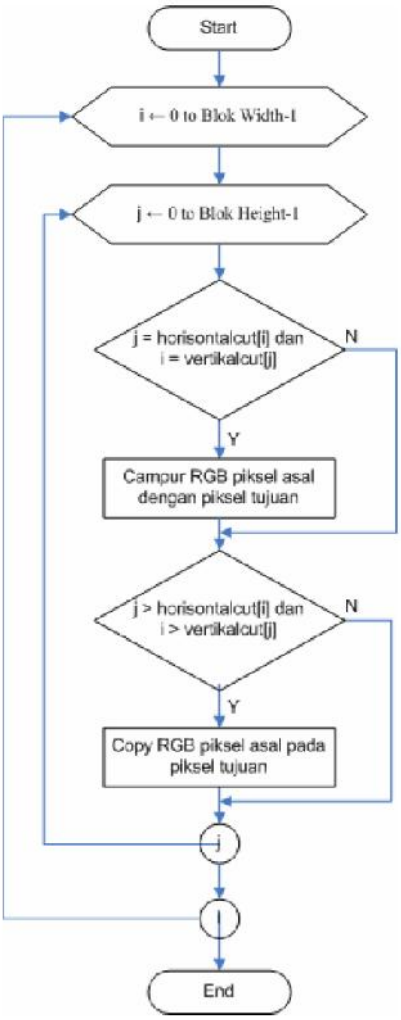
Proses untuk menghitung *minimum boundary cut* vertikal dan horisontal gambar 12 dan gambar 13


Gambar 12. Proses minimum *error boundary cut vertical*


Gambar 13. Proses minimum *error boundary cut horisontal*

- Proses Mengcopy Blok Citra asal ke Citra tujuan

Proses penyusunan blok citra asal ke dalam citra tekstur yang baru ditunjukkan pada gambar 14.




Gambar 14. Proses meng*copy* blok citra asal ke citra tujuan

## KESIMPULAN DAN SARAN

Dari hasil pengujian secara keseluruhan dapat disimpulkan :

[1] Algoritma *Image Quilting* secara umum menghasilkan hasil yang mirip dengan tekstur asalnya pada semua jenis tekstur. Terutama untuk tekstur dengan jenis *stochastic*, Misalnya seperti pada citra tekstur citra tanah dan

tekstur lain yang polanya tidak dikenali dengan jelas.

[2] Semakin besar jumlah kandidat blok maka hasil sintesa yang dihasilkan semakin mirip dengan tekstur asalnya, tetapi waktu proses yang dibutuhkan juga semakin lama.

[3] *Image Quilting* akan menghasilkan hasil tidak mirip untuk citra jenis *iregular* dengan tekstur citra seperti citra biji-bijian yang berbeda warnanya. Hal ini disebabkan karena berbeda pola tekstur tiap objek biji memiliki warna yang berbeda dan pola tekstur objek biji dari biji sulit untuk dicari yang cocok.

**DAFTAR RUJUKAN**


[1] Ahmad, Usman. 2005, **Pengolahan Citra Digital dan Teknik Pemrogramannya**, Penerbit Graha Ilmu, Yogyakarta.

[2] Dutta, Arup. 2005, **Texture Synthesis and Face Merging Using An Image Quilting Algorithm**.

[3] www.homepages.cae.wisc.edu/~arup/cs766/CS%20766%20Final%20Project%20Report.pdf

[4] http://en.wikipedia.org/wiki/Texture_synthesis-html/

[5] http://www.cs.ualberta.ca/~ghali/courses/611/courseNotes/p341-efros.pdf[5]