

IMPLEMENTASI METODE INTERPOLASI LINEAR UNTUK PEMBESARAN RESOLUSI CITRA

Kristoko Dwi Hartomo

*Program Studi Teknik Informatika, Fakultas Teknologi Informasi,
Universitas Kristen Satya Wacana
Jalan Diponegoro No. 52-60 Salatiga Jawa Tengah
E-mail : goibmohib@yahoo.com*

ABSTRACT

Processing an image is a process and image analysis which entangling many visual perceptions and in course have some characteristics of input data and output information in binding digital image. Process of changing measure for an image through magnification of resolution or measure often needed to showing the detail, for demonstration at a physic appliance, and for making of document. At the process on enlarging image resolution, method that would be used in this paper is linear interpolation. Linear Interpolation Method is mathematics algorithm which could be applied to appraise the middle price point through a diametrical line, two successive input point. The system in this paper could process image with algorithm of linear interpolation and become a new photo image with image resolution of pixel larger than the original image.

Keywords: image, resolution, linear, interpolation, pixel

1. Pendahuluan

Pengolahan citra merupakan salah satu bidang yang berada di dalam grafika komputer. Pengolahan citra itu sendiri merupakan proses pengolahan dan analisis citra yang banyak melibatkan persepsi visual dan dalam prosesnya mempunyai ciri-ciri data masukan dan informasi keluaran berbentuk berkas citra digital. Pengolahan citra tersebut saat ini telah banyak digantikan oleh komputer sebagai alat pengolah gambar secara digital yang handal. Pengolah gambar secara digital dapat dengan mudah memanipulasi gambar yang diperoleh melalui suatu alat seperti *scanner*, *digital cam* dan lain sebagainya.

Proses perubahan ukuran suatu citra melalui pembesaran ukuran atau resolusi sering dibutuhkan untuk keperluan memperlihatkan detil, untuk peragaan pada alat peraga, maupun untuk keperluan pembuatan dokumen. Salah satu fasilitas penting yang dimiliki program pengolah citra yang telah populer seperti *photoshop*, *corel*, *photopaint* dan lain-lain adalah sebagai fasilitas skala. Dengan fasilitas skala, ukuran dan resolusi dari citra dapat diubah-ubah sesuai keperluan untuk melihat detil dari dokumen.

Sebuah gambar terdiri dari kumpulan titik-titik piksel yang membentuk *array* (larik) dua dimensi, dalam proses pembesaran seringkali kita harus menaksir harga-harga tengahan diantara titik-titik data yang telah tepat. Metode interpolasi

linier merupakan algoritma matematika yang dapat diterapkan untuk menaksir titik harga tengahan melalui suatu garis lurus pada setiap dua titik masukan yang berurutan. Pada proses memperbesar resolusi citra, teknik tersebut akan diimplementasikan dalam makalah ini.

2. LANDASAN TEORI

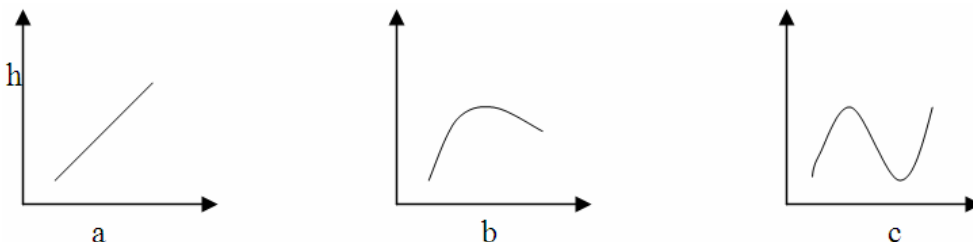
Pembesaran resolusi citra merupakan bagian dari proses skala citra yang merupakan sebuah contoh dari proses resampling suatu citra.

2.1 Interpolasi

Ada beberapa rumus interpolasi, diantaranya interpolasi diferensi terbagi *Newton*, interpolasi *lagrange* dan interpolasi *spline*. Interpolasi diferensi terbagi *newton* terbagi lagi menjadi interpolasi linier dan interpolasi kwadratik pada makalah ini metode interpolasi linear yang akan diterapkan untuk pembesaran citra.

Interpolasi linier merupakan polinomial tingkat pertama dan melalui suatu garis lurus pada setiap dua titik masukan yang berurutan. Dua titik masukan tersebut digunakan untuk menaksir harga-harga tengahan diantara titik-titik data yang telah tepat. Metode yang paling sering digunakan untuk maksud ini adalah interpolasi polinomial.

$$f_1(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n \quad (2.1)$$



Gambar 1. Contoh polinomial interpolasi : a. Orde pertama (linier) menghubungkan 2 titik; b. Orde kedua (kwadratik atau parabola) yang menghubungkan 3 titik; c. Orde ketiga (kubik) yang menghubungkan 4 titik.

Untuk $n + 1$ titik-titik data, terdapat satu dan hanya satu polinomial orde ke- n atau kurang yang melewati semua titik. Misalnya hanya terdapat satu garis lurus (yakni polinomial orde pertama) yang menghubungkan kedua titik seperti gambar 1.a di atas.

Dengan cara yang sama hanya satu parabola yang menghubungkan kumpulan dari 3 buah titik. Interpolasi polinomial ini kemudian memberikan sebuah formula untuk menghitung nilai-nilai antara.

2.1.1 Interpolasi Linier

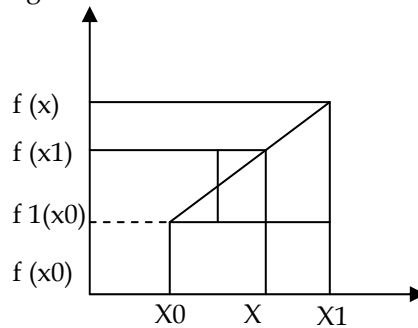
Bentuk interpolasi yang paling mudah adalah menghubungkan dua buah titik data dengan sebuah garis lurus. Teknik ini dan sebagainya interpolasi linier dan dilukiskan secara grafis pada gambar 2 dengan menggunakan segitiga sebangun.

$$\frac{f(x) - f(x_0)}{x - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \quad (2.2)$$

Kemudian dapat diatur kembali supaya memenuhi sebuah formula interpolasi linier.

$$f(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) \quad (2.3)$$

Notasi $f_1(x)$ menandakan bahwa rumus tersebut adalah sebuah polinomial interpolasi orde pertama. Perhatikan bahwa disamping menyatakan skope dari gambar yang menghubungkan titik.



Gambar 2. Luas segitiga sebangun menunjukkan area yang digunakan untuk menurunkan formula interpolasi linear.

Umumnya semakin kecil interval diantara titik-titik data, aproksimasinya semakin baik. Karakteristik ini ditunjukkan dalam contoh yang berikut :

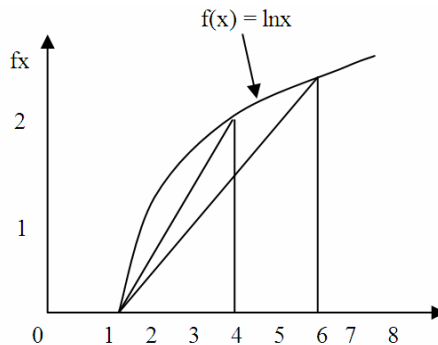
Dalam contoh di bawah ini kita akan menaksir logaritma asli dari Z ($\ln Z$) dengan menggunakan interpolasi linear. Langkah pertama dengan melakukan komputasi interpolasi antara $\ln 1 = 0$ dan $\ln 6 = 1,7917595$ sebagai berikut :

$$f_i(2) = 0 + \frac{1,7917595 - 0}{6 - 1}(2 - 1) = 0,35835190$$

Yang menunjukkan suatu persen kesalahan $e_t = 48,3\%$. Dengan menggunakan material yang lebih kecil dari $x_0 = 1$ sampai $x_1 = 4$, maka diperoleh perhitungan sebagai berikut :

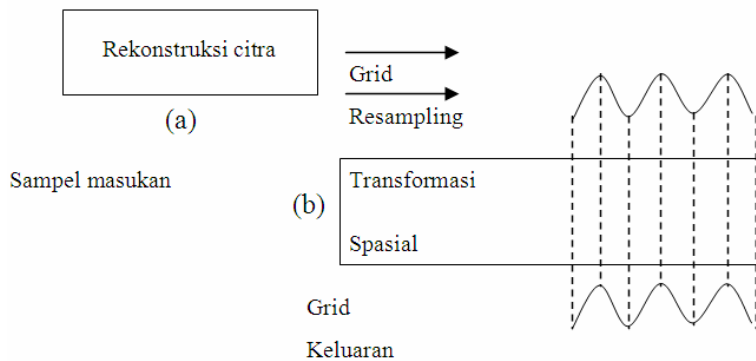
$$f(2) = 0 + \frac{1,3862944 - 0}{4 - 1}(2 - 1) = 0,469209813$$

Jadi dengan memakai interval yang lebih pendek mengurangi kesalahan relatif persen menjadi $e_t = 33,3\%$. Kedua interpolasi beserta fungsi yang sesungguhnya terlihat pada gambar 3 sebagai berikut :



Gambar 3. Dua interpolasi linier untuk menaksir $\ln 2$, yang menunjukkan interval yang semakin kecil memberikan suatu taksiran yang lebih baik.

2.2 Sampling dan Proses Pembesaran



Gambar 4. Proses Transformasi Spasial

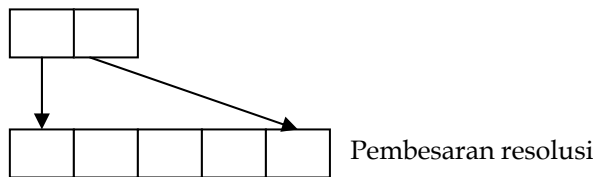
Secara keseluruhan proses registrasi dan resampling citra ini dapat dibagi menjadi 2 tahap, yaitu proses rekonstruksi citra atau sering disebut juga sebagai proses interpolasi citra, dan proses resamplingnya sendiri. Proses rekonstruksi citra dilakukan dengan melakukan proses *konvolusi* sinyal masukan diskrit menurut suatu fungsi interpolasi yang kontinu. Jadi citra masukan yang bersifat diskrit-diskrit ini lebih dahulu diubah menjadi kontinu diskrit, dengan demikian dapat dilakukan proses resampling citra menurut posisi dimana saja. Tujuan dari sampling ini adalah untuk menggambarkan (*represent*) sebuah citra yang kontinu,

dengan sebuah *array* (larik bilangan) [1]. Proses ini dapat digambarkan dengan diagram gambar 4.

Sebelum citra masukan yang berupa citra digital dapat diolah dengan proses pembesaran, diperlukan proses sampling terhadap titik-titik piksel gambar asli. Titik-titik hasil sampling ini kemudian disimpan dalam suatu larik yang memiliki struktur data record yang berguna sebagai titik kontrol dalam proses interpolasi. Struktur record bisa berguna karena dalam setiap elemen data record bisa memiliki tipe larik yang berbeda. Data record tersebut akan menyimpan informasi posisi titik dan sekaligus intensitas warna RGB yang dimiliki titik tersebut. Proses pembesaran setiap warna piksel dilakukan dengan berdasarkan barisnya baru kemudian menurut kolomnya. Sebelum gambar masukan disampling dilakukan dulu proses yang disebut rekonstruksi dengan memetakan intensitas warna RGB yang ada pada citra asli ke warna citra pada NewBitmap, untuk piksel dengan posisi 0 dan piksel terakhir (height -1) atau (Width -1) akan dipetakan ke posisi yang sama pada NewBitmap, hal ini penting agar tidak terjadi piksel sisa yang tidak berguna atau menyebabkan kerusakan gambar. Sedangkan untuk piksel $n=2$ s/d height -2 atau $n=2$ s/d width -2 dilakukan pemetaan langsung dengan proses mengalihkan setiap titik piksel dengan sebuah rumus sebagai berikut :

$$dx = (\text{new height} - 1) / (\text{orig height} - 2)$$

Setiap piksel dikalikan dengan rumus diatas sampel $fx [I] = \text{round}(I * dx)$. Kemudian hasilnya disimpan dalam sebuah struktur data *array*, yang akan menjadi titik kontrol dalam melakukan proses interpolasi linier untuk menaksir harga-harga warna piksel yang masih kosong. Perlu diingat bahwa semakin kecil interval dari titik kontrol akan semakin baik aproksimasinya, maka variabel *scale* dibaca dengan menghitung rata-ratanya. Untuk memperjelas gambaran dari proses pembesaran citra, dapat dilihat pada gambar 5.



Gambar 5. Proses pembesaran resolusi citra

2.3 Proses Interpolasi

Setelah pemetaan intensitas warna piksel dari gambar asli ke newbitmap maka piksel-piksel yang ada dalam newbitmap (piksel yang sudah terisi warnanya) ini akan menjadi titik kontrol dalam proses interpolasi. Interpolasi dilakukan dengan memasukkan setiap titik sample ke dalam rumusan interpolasi linear sebagai berikut :

$$f(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0)$$

Dimana :

- f(x) : intensitas warna RGB titik yang dicari
- f(x₀) : intensitas warna RGB titik sebelah kiri
- f(x₁) : intensitas warna RGB titik sebelah kanan
- x₀ : Posisi titik sebelah kiri
- x₁ : Posisi titik sebelah kanan
- x : Titik yang dicari

2.4 Mode Warna RGB

Monitor komputer menggunakan tiga buah warna primer yaitu : merah (*red*), hijau (*green*) dan biru (*blue*) yang disebut dengan istilah mode warna RGB¹. Pada mode warna RGB setiap titik pada layar berisi angka yang bukan menunjukkan intensitas warna dari titik tersebut, melainkan menunjukkan intensitas yang dipilih pada suatu tabel. Jadi pada setiap titik, dapat dipilih salah satu warna dari tabel.

Adapun masing-masing warna dari tabel memiliki tiga buah kombinasi angka yaitu R, G dan B yang menentukan proporsi warna merah (*Red*), hijau (*Green*) dan biru (*Blue*) dari warna tersebut. R, G dan B masing-masing memiliki range antara 0 hingga 255 sehingga jumlah warna ditabel yang dapat dipilih untuk mengisi warna pada sebuah piksel adalah $256 \times 256 \times 256 = 16,7$ juta warna.

Mode warna RGB yang berlaku untuk *grayscale* nilai R, G dan B adalah sama. Jika nilai R adalah 1 maka nilai G dan B adalah 1 juga, demikian seterusnya untuk lebih jelasnya nilai R, G dan B dapat dilihat pada tabel 1.

Tabel 1. Nomor warna skala abu-abu

No.	Red	Green	Blue
0	0	0	0
1	1	1	1
2	2	2	2
...
...
254	254	254	254
255	255	255	255

¹ Hearn, Donald., Baker, Pauline M., 1996, *Computer Graphics C Version 2nd Edition*, Prentice Hall Inc, New Jersey, Hal: 572

3. ANALISIS DAN PERANCANGAN

3.1 Analisis

Pembesaran citra dilakukan di dalam obyek bitmap dan resolusi akan bertambah setelah proses pembesaran dilakukan pada gambar masukan. Proses penambahan piksel/titik baru ini dilakukan dengan menyisipkan titik-titik tersebut diantara titik-titik yang harganya telah tetap yang dipetakan langsung dari citra asli. Piksel yang ada dalam citra masukan tersebut berisi informasi warna/intensitas warna RGB yang penting untuk menaksir intensitas warna dari piksel yang masih kosong pada citra obyek bitmap. Proses interpolasi adalah suatu proses untuk menentukan harga suatu fungsi pada titik-titik posisi antara suatu sampel dengan sampel tetangganya.

3.2 Perancangan

3.2.1 Perancangan Proses

Proses yang berlangsung untuk makalah ini dapat digambarkan dalam algoritma yang menjelaskan secara berurutan langkah demi langkah program ini. Program diimplementasikan menggunakan bahasa pemrograman Borland Delphi.

3.2.1.1 Algoritma Program

Secara keseluruhan program yang dibuat mempunyai algoritma atau urutan langkah-langkah pemecahan masalah sebagai berikut :

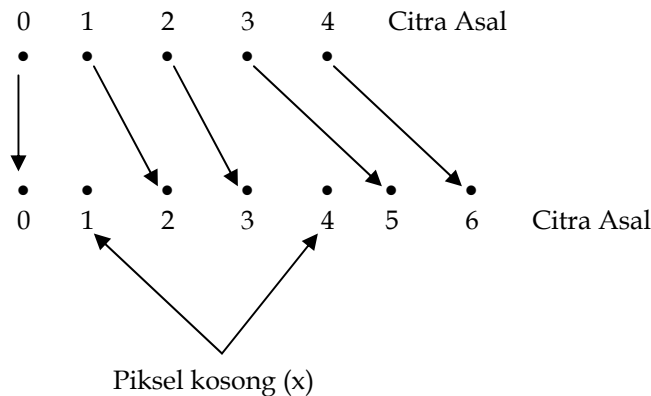
1. Membuka file grafik ke layar monitor yang merupakan data masukan.
2. Membaca ukuran tinggi dan lebar dari gambar masukan
OrigWidth : `integral.picture.bitmap.width`
OrigHeight : `integral.picture.bitmap.height`
3. Melakukan pembesaran terhadap gambar asli di dalam obyek bitmap.
Newwidth : `Pound (OrigWidth * scale)`
Newheight : `Pound (OrigHeight * scale)`
Newbitmap : `Tbitmap.Create`
NewBitmap.Width : `NewWidth`
NewBitmap.Height : `NewHeight`
4. Membaca intensitas warna R, G, B setiap piksel dari gambar asli berdasarkan baris dan kolomnya.
LRedSampel.Fxn => menampung intensitas warna merah
LGreenSampel.Fxn => menampung intensitas warna hijau
LBlueSampel.Fxn => menampung intensitas warna biru
Warna LRedSampel, LGreenSampel, LBlueSampel merupakan variabel record TInterpolationSample yang mempunyai elemen Fxn yang berupa array berdimensi 1 bertipe byte.
5. Memetakan warna setiap piksel gambar asli ke gambar baru (Newitmap).
Proses pemetaan informasi warna dari gambar asli ke gambar baru NewBitmap dilakukan dengan 2 tahap.

- o Memetakan piksel yang pertama (I=0) gambar asli ke piksel pertama gambar baru dan piksel terakhir (I = NewWidth -1) ke piksel terakhir gambar baru, agar tidak terjadi piksel sisa/tidak terisi informasi warna.

Gambar asli	Proses	Gambar baru
Piksel (I=0)	Dipetakan	Piksel (I=0)
Piksel (I=NewWidth -1)	Dipetakan	Piksel (I=NewWidth -2)
Piksel (I=NewHeight -1)	Dipetakan	Piksel (I=NewHeight -2)

- o Memetakan piksel-piksel diantara piksel pertama dan terakhir ke posisi yang telah ditentukan melalui perhitungan melalui perhitungan dengan rumus :
 Line => Pound (I * dx)
 dimana dx = (NewWidth -2 / origWidth -2)
 Column => Round (I * dy)
 dimana dy = (NewHeight -2 / origHeight -2)
6. Menaksir harga-harga titik piksel pada gambar baru (newbitmap) yang masih kosong, dengan rumus interpolasi.
 Berdasarkan contoh di bawah dx = 5/3 atau 1,66666667 dengan rumus Round (I * dx) :

$$\begin{aligned} \text{LRedSample.Xn}(i) &= \text{Round}(1 \times 1,66666667) \Rightarrow 2 \\ \text{LRedSample.Xn}(i) &= \text{Round}(2 \times 1,66666667) \Rightarrow 3 \\ \text{LRedSample.Xn}(i) &= \text{Round}(3 \times 1,66666667) \Rightarrow 5 \end{aligned}$$



Gambar 6. Proses pemetaan piksel citra asli ke citra baru

$$\text{Intepolasi linear} = \frac{\text{Sample.Fxn}[\text{leftpoint}] + (\text{Sample.fxn}[\text{rightpoint}] - \text{Sample.Fxn}[\text{leftpoint}] * (x - \text{Sample.xn}[\text{leftpoint}]) / (\text{Sample.xn}[\text{rightpoint}] - \text{Sample.xn}[\text{leftpoint}])}{}$$

Sample.fxn[leftpoint] = intensitas warna titik sebelah kiri yaitu (I-1)
 Sample.fxn[rightpoint] = intensitas warna titik sebelah kanan yaitu (I-1)
 Sample.xn[leftpoint] = posisi titik sebelah kiri yaitu (I-1)
 Sample.xn[rightpoint] = posisi titik sebelah kiri yaitu (I-1)
 X = titik yang dicari harganya

Memetakan harga (informasi warna) hasil perhitungan intrepolasi linear pada titik yang dicari {memetakan warna pada newbitmap berdasarkan baris}

newbitmap.canvas.pixels[A,B]=RGB(ResultRed[A],resultgreen[A],resultblue [A]) {memetakan warna pada newbitmap berdasarkan kolom} - sama [A] ganti [B].

7. Citra hasil proses pembesaran dapat ditampilkan dilayar monitor, disimpan dalam file format bitmap atau dicetak ke printer.

3.2.2 Perancangan Program

3.2.2.1 Struktur Data

Struktur data merupakan bagian yang sangat penting dalam pengolahan citra. Didalamnya akan menentukan bagaimana data yang berupa piksel-piksel tersebut akan diakses dan disimpan dalam memori komputer. Setiap kali melakukan pendeklarasian sebuah variabel, diperlukan penentuan tipe data yang akan disimpannya. Tipe data yang tepat sangat berguna untuk menghemat penggunaan memori komputer, yang berarti berhubungan juga dengan kecepatan program saat dieksekusi.

Piksel-piksel yang ada pada gambar diakses dan ditampung pada *array* yang telah ditentukan ukuran maupun dimensinya. *Array* merupakan salah satu fasilitas yang dapat menyimpan data secara berurutan dalam sebuah nama variabel-variabel. Dalam *array* ini data disimpan berdasarkan indeks untuk memudahkan pencariannya. Pada variabel biasa jika dimasukkan data baru maka data lama akan dihapus, sedangkan pada *array* data baru tidak akan menghapus data lama tapi diletakkan sesudahnya.

Record digunakan untuk menyimpan serangkaian data dalam suatu variabel. Perbedaannya dengan *array* adalah tipe data setiap elemen pada sebuah record tidak perlu sama. Record umumnya digunakan untuk menyimpan satu informasi yang saling berkaitan (semacam *database*). Gambar yang ditampilkan pada monitor merupakan suatu jenis matrik 2 dimensi yang masing-masing elemennya menunjukkan koordinat [x,y]. namun di dalam program pembesaran citra ini, struktur data *array* yang digunakan hanya berdimensi satu, untuk mengakses dan menampung piksel, berdasarkan kolom dan baris.

Deklarasinya dalam delphi :

Type

DataintArray = array[0...1024] of integer;

DatabyteArray = array[0...1024] of byte;

DatboolArray = array[0...1024] of boolean;

TInterpolationSample = record

```
Xn : dataintarray;  
fxn :databytearray;  
end;
```

Tipe data seperti di atas mampu menampung ukuran gambar sebesar 1 layar monitor penuh pada resolusi 640 x 480 sesuai dengan batasan masalah pada makalah ini. Masing-masing piksel pada gambar tersebut mampu ditampung sesuai pada koordinatnya. Penyimpanan pellete warna dari citra tidak diperlukan karena pada lingkungan windows saat gambar dibuka, pallete warna akan langsung ditambahkan ke pallete sistem windows dan akan langsung dipakai jika gambar yang bersangkutan ditampilkan.

3.2.2.2 Perancangan Proses Pembesaran

Awal dari proses pembesaran gambar program yang dibuat adalah membaca resolusi file grafik sebagai masukan yang berupa bitmap obyek Timage yang telah disediakan Delphi. Untuk mengetahui ukuran/resolusi gambarnya, dilakukan dengan properti width dan height. Sebelum masukan dalam proses pembesaran citra, masukan skala dari pembesarannya yang nilainya akan dicatat dalam variabel scale.

Nilai tiap piksel, pada gambar dapat diketahui dari properti pixels yang dimiliki oleh Timage. Properti pixel ini bertipe larik dua dimensi, sehingga seluruh nilai piksel tersimpan sesuai koordinatnya. Untuk memudahkan proses pengolahan gambar, nilai tiap piksel dari gambar disimpan dalam variabel rekaman (record). Didalam variabel record ini terdapat dua elemen sebagai penampung data pixel dan variabel larik yang bertipe byte sebagai penampung intensitas warna Red, Green, Blue masing-masing piksel.

Walaupun demikian kedua variabel tersebut menyimpan informasi yang saling berkaitan. Penggunaan tipe rekaman/record yang termasuk salah satu data terstruktur, dimaksudkan agar prosesnya sesuai dengan proses yang berlaku dalam interpolasi linear yaitu interpolasi dilakukan terhadap setiap data titik masukan.

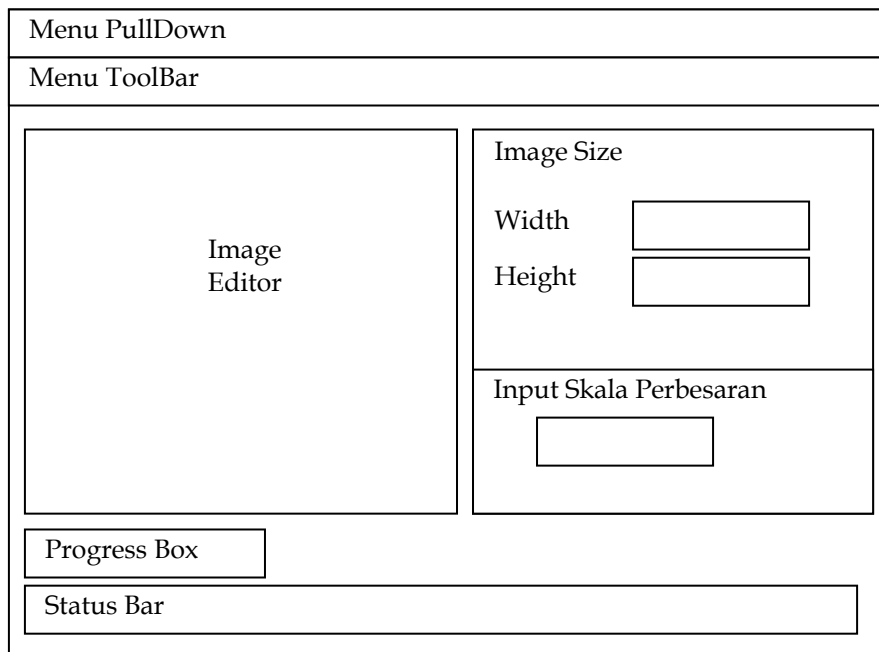
Pembesaran citra dilakukan dengan mengalikan skala terhadap lebar dan tinggi dari citra, seperti ($\text{newheight} = \text{oldheight} * \text{scale}$) dan ($\text{newwidth} = \text{oldwidth} * \text{scale}$) sehingga ukuran dari newbitmap menjadi lebih besar dari citra asli. Langkah selanjutnya setiap piksel yang menyimpan informasi warna R, G, B dari gambar asli dipetakan atau diset ke newbitmap. Sebagai contoh seperti yang ditunjukkan pada gambar 6. Citra asli terdiri dari lima titik data (piksel), diperbesar menjadi 2 kalinya. Setelah newbitmap dibuat maka akan terdapat titik-titik piksel yang sudah memiliki warna. Variabel skala tersebut akan dihitung dalam square, hal ini dimaksudkan agar interval antar pikselnya semakin kecil, sehingga aproksimasinya semakin baik [2]. Proses pemetakan titik asli ke titik baru dalam newbitmap dengan membaca kembali array dari setiap baris dan kolom. Untuk titik 0 citra asli langsung dipetakan ke titik 0 titik baru pada newbitmap, demikian juga titik terkahir citra asli ke titik terakhir citra baru, dengan demikian piksel pertama dan piksel terakhir terisi data. Dengan bantuan sebuah variabel $dx = (\text{newwidth} - 2 / \text{origwidth} - 2)$ dan $dy = (\text{newheight} - 2 / \text{orig height} - 2)$ maka

piksel proses pemetakan piksel dapat dilakukan, dx dikalikan dengan piksel baris (larik baris) sedangkan dy dikalikan dengan piksel kolom (larik kolom).

Warna yang dipetakan ke piksel baru adalah intensitas R, G, B setiap piksel. Untuk menghitung warna dan piksel-piksel yang masih kosong dilakukan dengan proses interpolasi linear berdasarkan dua titik sebelah kiri dan sebelah kanan piksel yang dicari yaitu titik x. Langkah yang terakhir adalah memetakan intensitas warna masing-masing piksel hasil proses interpolasi dari rumus di atas ke dalam bitmap.

3.2.3 Perancangan Tampilan

Interface dibuat dalam mode VGA Hi dengan resolusi 800 x 600 piksel. Tampilan dibuat dalam bentuk dekstop yang dibagi menjadi beberapa bagian, gambaran mengenai perancangan tampilan dari form utama dapat dilihat seperti pada gambar 7 sebagai berikut :



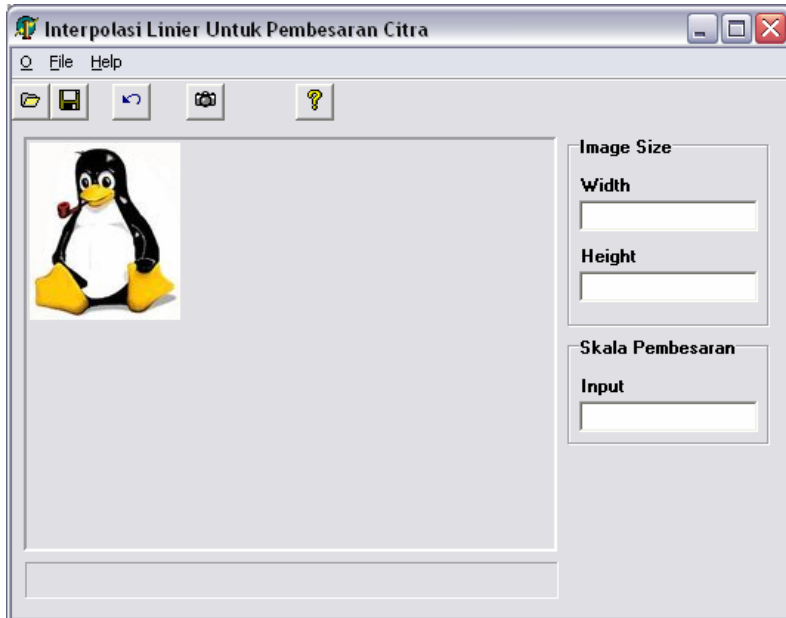
Gambar 7. Perancangan tampilan Menu Utama

4. IMPLEMENTASI DAN ANALISIS HASIL KERJA

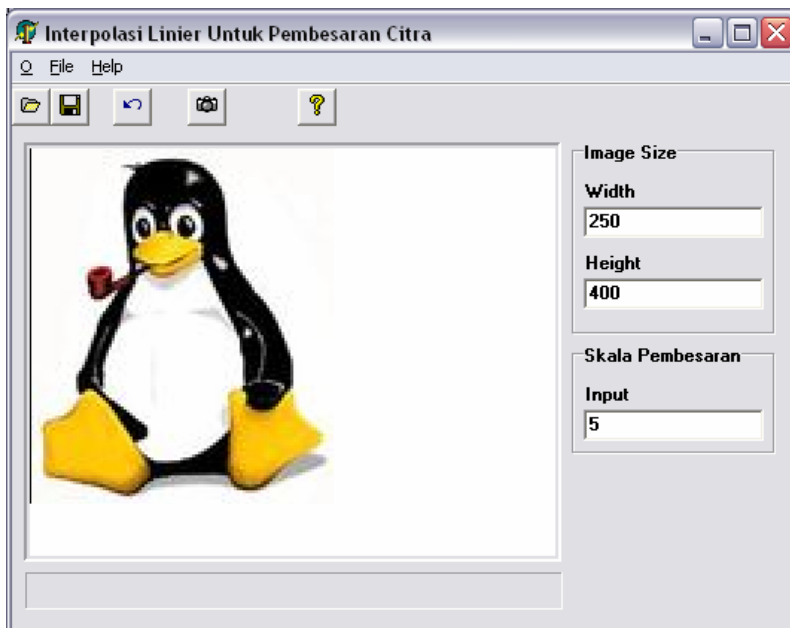
4.1 Implementasi

Untuk melakukan proses pembesaran resolusi citra dengan metode interpolasi linier, klik menu file, klik open, pilih file citra yang akan diperbesar. Tentukan ukuran dan skala pembesaran citra yang diinginkan, untuk melihat hasil

akhir proses pembesaran citra tekan tombol enlarge. Terlihat pada gambar 8 dan 9, citra telah berhasil diperbesar dengan interpolasi linier.



Gambar 8. Tampilan Sistem Sebelum Proses Pembesaran



Gambar 9. Tampilan Sistem Setelah Proses Pembesaran

4.2 Analisis Hasil Kinerja Program

Setelah program dijalankan maka citra akan diproses dengan algoritma interpolasi linier dan menjadi sebuah citra foto baru dengan resolusi citra piksel yang lebih besar dari aslinya.

Proses pembesaran citra dilakukan didalam obyek bitmap dan resolusi akan bertambah setelah proses pembesaran dilakukan pada gambar masukan. Proses penambahan piksel/titik baru ini dilakukan dengan menyisipkan titik-titik tersebut diantara titik-titik yang harganya telah tetap yang dipetakan langsung dari citra asli.

Interpolasi sebenarnya adalah suatu proses untuk menentukan harga suatu fungsi pada titik-titik posisi antara suatu sampel dengan sampel tetangganya. Didalam resampling citra terdapat suatu proses transformasi suatu citra diskrit-diskrit dari suatu sistem kordinat ke sistem kordinat lainnya. Hubungan antara kedua sistem koordinat tersebut dinyatakan dengan suatu fungsi pemetaan dapat diperoleh dengan bantuan titik-titik kontrol dalam proses koreksi geometrinya. Proses ini dikenal juga dengan sebutan proses regresi citra, dimana penggunaan titik-titik kontrol akan mempengaruhi bentuk fungsi pemotretan yang diperoleh pada proses transformasi.

5. SIMPULAN DAN SARAN

5.1 Simpulan

Setelah menyelesaikan penulisan makalah dengan judul Perancangan Dan Implementasi Metode Interpolasi Linear Untuk Pembesaran Resolusi Citra, maka dapat disimpulkan :

- Metode Interpolasi Linier dapat diterapkan untuk memperbesar resolusi citra sehingga ukuran citra lebih besar dari aslinya.
- Metode Interpolasi Linier sangat tepat digunakan untuk memperbesar citra, dengan menyisipkan titik diantara dua titik masukan yang telah tepat posisinya.
- Agar proses pembesaran citra berjalan cepat, diperlukan dukungan *hardware* yang memadai, karena pada proses akan dilakukan pengolahan tiap piksel dari gambar asli dengan metode Interpolasi Linier.
- Data gambar yang berbentuk matrik dua dimensi dapat disimpan secara efektif dalam variabel yang bertipe larik.

5.2 Saran

Perancangan program Perancangan Dan Implementasi Metode Interpolasi Linear Untuk Pembesaran Resolusi Citra masih banyak kekurangan. Untuk lebih menyempurnakan program sebaiknya :

- Program dapat memproses format selain bmp, misalnya gif, pcx, jpeg, tift, dan lain-lain.
- Ditambahkan proses untuk memperkecil citra.

- Untuk hasil yang lebih halus, dengan proses penghalusan yang tidak berlebihan diperlukan polinomial dengan derajat lebih tinggi seperti polinomial kubik.
- Ditambahkan fasilitas rotasi citra, translasi citra, *mirroring*, dan lain-lain.

DAFTAR PUSTAKA

- [1] Pacheco, Xavier., Teixeira, Steve., (2002) *Borland Delphi 6 Developer's Guide*, Sams Publiser, New York
- [2] Rogers, F. David., Adams, Alan, J., (2002) *Mathematical Elements for Computer Graphics 2nd Edition*, Tata McGraw-Hill Publishing Company Limited, New Delhi
- [3] Nalwan Agustinus., (1997) *Pengolahan Gambar Secara Digital*, PT Elex Media Komputindo, Jakarta
- [4] Kadir, Abdul., (1999) *Borland Delphi Lengkap*, Andi Offset, Yogyakarta
- [5] Santosa, Insap., (1999) *Grafika Komputer dan Antarmuka Grafis*, Andi Offset, Yogyakarta