BAB II

LANDASAN TEORI

2.1 Perkembangan Penggunaan Intenet

Pengguna internet yang semakin banyak mendorong kemajuan dalam banyak bidang, berdasarkan data survey dari APJII (2017 dan 2018) (Asosiasi Penyedia Jasa Internet Indonesia) pada 2017 pengguna internet di Indonesia sebanyak 143,26 juta jiwa, pada 2018 pengguna internet bertambah menjadi 171,17 juta jiwa dengan 93 persen pengguna aktif menggunakan *smart phone*. Di masa inise makin lebar penggunaan internet diaplikasikan, tidak hanya untuk mengakses konten hiburan dan bertukar pesan, tetapi juga merambah ke bisnis seperti makanan, perbankan, penyedia informasi, jual beli *online*, transportasi *online,* dan sebagainya. Sekarang pihak konsumen, bisnis, dan produsen harus *update* dengan kemajuan teknologi yang ada.

2.2 Citra digital

Citra digital adalah citra yang dihasilkan melalui proses digitalisasi terhadap citra analog (Ginting, 2010). Citra dapat dikatakan sebagai citra digital bila terekam dan tersimpan dalam format file digital.

Menurut Salomon dan Motta (2010) citra digital adalah sebuah matriks berbentuk persegi panjang yang terdiri atas titik – titik yang disebut elemen citra (piksel), yang tersusun dalam M baris dan N kolom. Resolusi pada citra digital merupakan perkalian M dan N.

$$f(x) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0, N-1) \\ f(1,0) & f(1,1) & \cdots & f(1, N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1, N-1) \end{bmatrix} \cdots (2.1)$$

Citra digital terbagi atas citra biner, citra *grayscale*, dan citra *full color*. Citra biner terbentuk dari dua warna yaitu hitam dan putih. Satu satuan warna atau yang disebut piksel merepresentasikan sebuah titik dalam sebuah citra. Dalam citra biner, nilai piksel yang ada adalah 0 untuk hitam dan 1 untuk putih. Menurut Van (Van, 2009) citra *grayscale* atau disebut monokrom tersusun dari nilai antara 0 sampai 255. Tiap piksel direpresentasi dalam 1 *byte*. Sedangkan citra *full color*, tiap piksel direpresentasi dengan tiga warna yang berbeda, yaitu merah, hijau, dan biru. Tiap warna tersusun dari nilai 0 sampai 255 dan ketiga warna tersebut terwujud dalam 3 *byte*.

2.3 *Discrete Cosine Transform (DCT)*

Discrete Cosine Transform (DCT) atau disebut dengan Transformasi Cosinus Diskrit adalah model transformasi *fourier* yang mengacu pada fungsi diskrit dengan mengambil bagian kosinus dari eksponensial kompleks, dan hasilnya juga diskrit. DCT akan mengubah detil warna dari gambar asli, namun karena keterbatasan indra manusia, perubahan yang terjadi tidak begitu terlihat. Operasi dasar yang ditampilkan dalam transformasi ini adalah mengambil suatu *signal* dan mentransformasikannya dari representasi satu tipe *file* ke tipe *file* yang lain (Noviardhi, 2008). Ada dua macam persamaan yang bisa digunakan yaitu DCT 1 dimensi yang digunakan untuk menghitung data vektor dan DCT 2 dimensi untuk menghitung data matriks. Persamaan umum untuk DCT 1 dimensi adalah sebagai berikut

$$DCT(i) = \frac{2}{N} C(i) \sum_{x=0}^{N-1} pixel(x) \cos\left(\frac{\pi(2x+1)i}{2N}\right) \quad \ldots (2.2)$$

Dengan

DCT(i) = nilai DCT indeks ke-i

N = ukuran vektor

Pixel(x) = nilai *pixel* pada indeks ke-x

C(i) = 1 jika x>0

C(i)= $\frac{1}{\sqrt{2}}$ jika x=0

Sedangkan persamaan untuk invers DCT 1 dimensi adalah sebgai berikut.

$$Pixel(x) = \frac{2}{N}\sum_{i=0}^{N-1} C(i)DCT(i)\cos\left[\frac{\pi(2x+1)i}{2N}\right] \quad \text{... (2.3)}$$

Dengan

DCT(i) = nilai DCT indeks ke-i

N = ukuran vektor

Pixel(x) = nilai *pixel* pada indeks ke-x

C(i) = 1 jika $i > 0$

C(i)= $\frac{1}{\sqrt{2}}$ jika $i = 0$

Persamaan DCT dan *invers* DCT (IDCT) di atas disebut persamaan DCT 1 dimensi dan IDCT 1 dimensi karena kedua persamaan tersebut digunakan untuk mentransformasikan nilai *pixel* suatu citra berukuran 1xN. Contoh ilustrasi metode DCT 1 dimesi sebagai berikut.

Terdapat sebuah vektor 1 dimensi [1 2 3 4 5 6 7 8]. Vektor ini akan dikenai persamaan (2.2) dengan :

i = 0, 1, 2…N

x = 0, 1, 2 ….N-1

N= 8

pixel(x) = 1,2,3…8

C(i) = 1, jika x>0, C(i) = 1/√2, jika x=0 (koefisien yang didapat dari DCT(i))

pixel(x) = nilai *pixel* ke-x

setelah dilakukan penghitungan didapatkan vektor DCT(i) adalah [12,727 -6.442 0 -0.673 0 -0.201 0 -0.051] Hasil ini kemudian akan dikenai fungsi *inverse* dari DCT. Setelah dikembalikan, nilainya menjadi [1 2 3 4 5 6 7 8]. Dapat dilihat bahwa hasil *invers* masih terlihat sama degan data asli.

Persamaan DCT 2 dimensi ditampilkan dalam matriks NxN. Sehingga menghasilkan keluaran berupa matriks NxN. Persamaan DCT 2 berguna dalam transformasi matrik dari gambar 2 dimensi.

Persamaan dari DCT 2 dimensi adalah sebagai berikut.

$$DCT(i,j) = \frac{2}{N} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} pixel\,(x,y) \cos\left[\frac{\pi(2x+1)i}{2N}\right] \cos\left[\frac{\pi(2y+1)i}{2N}\right] \quad \ldots (2.4)$$

Dengan

DCT(i,j) = nilai DCT indeks ke-(i,j)

N = ukuran matriks

pixel(x,y) = nilai *pixel* pada indeks ke-(x,y)

C(i),C(j) = 1 jika i,,j > 0 (koefisien yang didapat dari DCT(i, j))

C(i),C(j) = $\frac{1}{\sqrt{2}}$ jika i,,j = 0 (koefisien yang didapat dari DCT(i,j))

Jika matriks berukuran MxN, persamaan di atas dapat dituliskan sebagai berikut.

$$DCT(i,j)$$
$$= \sqrt{\frac{2}{M}}\sqrt{\frac{2}{N}}C(i)C(j)\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} pixel\,(x,y)\cos\left[\frac{\pi(2x+1)i}{2M}\right]\cos\left[\frac{\pi(2y+1)j}{2N}\right] \ldots (2.5)$$

Dengan

DCT(i,j) = nilai DCT indeks ke-(i,j)

N = ukuran baris matriks

M = ukuran kolom matriks

pixel(x,y) = nilai *pixel* pada indeks ke-(x,y)

C(i),C(j) = 1 jika i,,j > 0 (koefisien yang didapat dari DCT(i, j))

C(i),C(j) = $\frac{1}{\sqrt{2}}$ jika i,,j = 0 (koefisien yang didapat dari DCT(i, j))

Sedangkan persamaan untuk *invers* DCT (IDCT) dengan matriks NxN sebagai berikut.

$$Pixel(x, y) = \frac{2}{N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} C(i)C(j)DCT\,(i, j)\cos\left[\frac{\pi(2x+1)i}{2N}\right]\cos\left[\frac{\pi(2y+1)j}{2N}\right] \quad \ldots (2.6)$$

Dengan

DCT(*i*,*j*) = nilai DCT indeks ke-(*i*,*j*)

N = ukuran baris dan kolom matriks

pixel(x,y) = nilai *pixel* pada indeks ke-(*x*,*y*)

C(*i*),C(*j*) = 1 jika *i*,,*j* > 0 (koefisien yang didapat dari DCT(i, j))

C(*i*),C(*j*) = $\frac{1}{\sqrt{2}}$ jika *i*,,*j* = 0 (koefisien yang didapat dari DCT(i, j))

Jika matriks berukuran MxN akan menggunakan persamaan invers sebagai berikut.

$$Pixel(x, y) = \sqrt{\frac{2}{M}}\sqrt{\frac{2}{N}}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} C(i)C(j)DCT\,(i, j)\cos\left[\frac{\pi(2x+1)i}{2M}\right]\cos\left[\frac{\pi(2y+1)j}{2N}\right] \ldots (2.7)$$

Dengan

DCT(i, j) = nilai DCT pada indeks ke-(i, j)

N,M = ukuran baris dan kolom matriks

pixel(x,y) = nilai *pixel* pada indeks ke-(x,y)

C(*i*),C(*j*) = 1 jika *i*,,*j* > 0 (koefisien yang didapat dari DCT(i, j))

C(*i*),C(*j*) = $\frac{1}{\sqrt{2}}$ jika *i*,,*j* = 0 (koefisien yang didapat dari DCT(i, j))

Invers DCT berfungsi untuk mengembalikan data citra seperti semula.

Di bawah ini diilustrasikan bagaimana DCT diterapkan dalam sebuah matriks 2 dimensi. Terdapat sebuah matriks 2 dimensi sebagai berikut.

$$\begin{bmatrix} 1 & 3 & 5 \\ 5 & 2 & 4 \\ 2 & 4 & 6 \end{bmatrix}$$

Matriks ini akan dikenai operasi DCT 2 dimensi dengan persamaan 2.4 dengan

i = 0, 1, 2 ; j = 0, 1, 2

x = 0, 1, 2 ; j = 0, 1, 2

pixel (x,y) = 1, 3, 5…6

N = 3

C(*i*),C(*j*) = 1 jika *i*,,*j* > 0

C(*i*),C(*j*) = $\frac{1}{\sqrt{2}}$ jika *i*,,*j* = 0

Dari persamaan di atas didapat matriks DCT(i, j) yaitu

$$\begin{bmatrix} 10.67 & -2.85 & 1.16 \\ -1.22 & 0 & 0 \\ -0.25 & -2.88 & -1.67 \end{bmatrix}$$. Kemudian setelah dilakukan *invers* DCT menghasilkan matriks yang mirip dengan matriks asli yaitu $$\begin{bmatrix} 1 & 3,01 & 4,99 \\ 5 & 2,01 & 4 \\ 2 & 4 & 5,98 \end{bmatrix}$$.

Tabel penghitungan DCT terdapat pada Tabel 2.1

	XoYo	XoY1	XoY2	X1Yo	X1Y1	X1Y2	X2Y0	X2Y1	X2Y2	DCT
DCT(0,0)	0.33	1.00	1.67	1.67	0.67	1.33	0.67	1.33	2.00	10.67
DCT(0,1)	0.4	0.0	-2.0	2.0	0.0	-1.6	0.8	0.0	-2.4	-2.85
DCT(0,2)	0.2	-1.4	1.2	1.2	-0.9	0.9	0.5	-1.9	1.4	1.16
DCT(1,0)	0.4	1.2	2.0	0.0	0.0	0.0	-0.8	-1.6	-2.4	-1.22
DCT(1,1)	0.5	0.0	-2.5	0.0	0.0	-0.0	-1.0	-0.0	3.0	0.00
DCT(1,2)	0.3	-1.7	1.4	0.0	-0.0	0.0	-0.6	2.3	-1.7	0.00
DCT(2,0)	0.2	0.7	1.2	-2.4	-0.9	-1.9	0.5	0.9	1.4	-0.25
DCT(2,1)	0.3	0.0	-1.4	-2.9	-0.0	2.3	0.6	0.0	-1.7	-2.88
DCT(2,2)	0.2	-1.0	0.8	-1.7	1.3	-1.3	0.3	-1.3	1.0	-1.67

Tabel 2.1 Tabel penghitungan DCT 2 dimensi

Tabel penghitungan *invers* DCT terdapat pada Tabel 2.2

	DCT(0,0)	DCT(0,1)	DCT(0,2)	DCT(1,0)	DCT(1,1)	DCT(1,2	DCT(2,0)	DCT(2,1)	DCT(2,2)	DCT
Pixel(0,0)	3.56	-1.16	0.27	-0.50	-0.00	0.00	-0.06	-0.83	-0.28	1.00
Pixel(0,1)	3.6	-0.0	-0.5	-0.5	-0.0	-0.0	-0.1	-0.0	0.6	3.01
Pixel(0,2)	3.6	1.2	0.3	-0.5	0.0	0.0	-0.1	0.8	-0.3	4.99
Pixel(1,0)	3.6	-1.2	0.3	-0.0	-0.0	0.0	0.1	1.7	0.6	5.00
Pixel(1,1)	3.6	-0.0	-0.5	-0.0	-0.0	-0.0	0.1	0.0	-1.1	2.01
Pixel(1,2)	3.6	1.2	0.3	-0.0	0.0	0.0	0.1	-1.7	0.6	4.00
Pixel(2,0)	3.6	-1.2	0.3	0.5	0.0	-0.0	-0.1	-0.8	-0.3	2.00
Pixel(2,1)	3.6	-0.0	-0.5	0.5	0.0	0.0	-0.1	-0.0	0.6	4.00
Pixel(2,2)	3.6	1.2	0.3	0.5	-0.0	-0.0	-0.1	0.8	-0.3	5.98

Tabel 2.2 Tabel Penghitungan *Invers* DCT 2 Dimensi

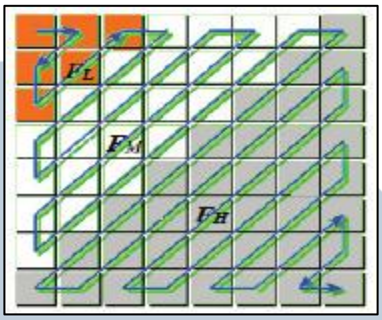
2.4 Kuantisasi

Kuantisasi bertujuan untuk menentukan kualitas kompresi dengan menghilangkan beberapa informasi dari nilai piksel yang telah ditransformasi. Informasi yang telah dihilangkan adalah yang dianggap tidak penting. Dalam hal ini adalah nilai piksel yang paling mendekati nol (0) setelah dilakukan transformasi, ataupun pembulatan nilai decimal dari nilai yang didapat. Kuantisasi pada uji coba dilakukan setelah mendapat matriks DCT yang didapat dari persamaan DCT.

Dalam proses pemampatan data kuantisasi ini penting, karena penentuan kuantisasi juga akan berpengaruh terhadap kualitas citra yang dimampatkan dan besarnya *file* yang telah dimampatkan.

Nilai kuantisasi dapat ditentukan dengan menyatakan suatu nilai sebagai batasan kuantisasi tersebut. Contoh nilai kuantisasi dinyatakan sebagai Q, nilai yang berada dalam jangkauan nilai Q dan –Q tersebut akan dikuantisasi menjadi nol (0). Contoh penerapan nya, nilai [28.6475, 39.2241 2.5042 0.5000 -11.2604 , -0.3450 6.7378 -1.9927 -0.8713 -1.5000 -3.3713 -3.3321 2.2168 0.7604] akan dikuantisasi dengan Q=1. Kemudian nilai yang berada di antara -1 sampai 1 akan dikuantisasi menjadi nol (0) menjadi [28.6475, 39.2241, 0, -11.2604, 0, 6.7378, -1.9927, 0, -1.5000, -3.3713, -3.3321, 2.2168, 0].

Kuantisasi untuk matriks berukuran 2^nx2^n dapat dilakukan dengan cara mengubah matriks menjadi vector secara *zig-zag scanning.*



Gambar 2.3 Urutan *Zig-Zag scanning* (sumber : Saboori, 2014)

Ilustrasi penerapan kuantisasi zig zag adalah sebagai berikut.

Terdapat matriks berukuran 4x4 dengan elemen $\begin{bmatrix} 30 & 19 & 8 & 11 \\ 28 & 10 & 5 & 9 \\ 33 & 4 & 0 & 6 \\ 43 & 15 & 1 & 2 \end{bmatrix}$, kemudian matriks ini akan ditransformasikan menjadi

$$\begin{bmatrix} 56.0000 & 39.2241 & 25.0000 & 3.2359 \\ 3.5042 & -11.2604 & -6.6585 & -3.3713 \\ 8.5000 & 6.7378 & -3.5000 & -3.3321 \\ -1.9927 & -0.8713 & 2.2168 & 0.7604 \end{bmatrix}$$

Matriks hasil transformasi di atas akan dikenai fungsi zig-zag scan dan menjadi vektor dengan urutan [56.000 39.2241 3.5042 8.5000 -11.2604 25.000 3.2359 -6.6585 6.7378 -1.9927 -0.8713 -3.5000 -3.3713 -3.3321 2.2168 0.7604]. Dari hasil zig-zag scan tersebut kuantisasi dilakukan dengan cara mengambil nilai dari belakang, tergantung prosentase yang diinginkan. Contoh, prosentase yang diinginkan adalah Q=25%, lalu diambil empat nilai paling belakang, dan nilai tersebut dikuantisasi menjadi nol (0). Hasilnya adalah urutan [56.000 39.2241 3.5042 8.5000 -11.2604 25.000 3.2359 -6.6585 6.7378 -1.9927 -0.8713 -3.5000 0 0 0 0].

2.5 Pengukuran MSE, PSNR

Dalam proses kompresi citra terdapat standar pengukuran error yaitu sebagai berikut.

- *Mean Square Error* (MSE) yaitu sigma dari jumlah error antara citra hasil kompresi dengan citra asli.

$$MSE = \frac{1}{MN}\sum_{y=1}^{M}\sum_{x=1}^{N}[I(x,y) - I'(x,y)] \qquad \dots(2.8)$$

Dimana : I(x,y) adalah nilai piksel dari citra asli

I′ (x,y) adalah nilai piksel dari citra hasil kompresi

M,N adalah resolusi citra

- *Peak Signal to Noise Ratio* (PSNR)

$$PSNR = 10log10\left(\frac{C^2max}{MSE}\right) \qquad \dots(2.9)$$

Dimana : C^2 max nilai maksimum dalam gambar yaitu nilai maksimum dari nilai piksel adalah 255 dan minimum adalah 1, dan MSE adalah nilai *Mean Square Error*. Semakin rendah nilai MSE akan semakin baik, sedangkan nilai PSNR yang tinggi akan jauh lebih baik.