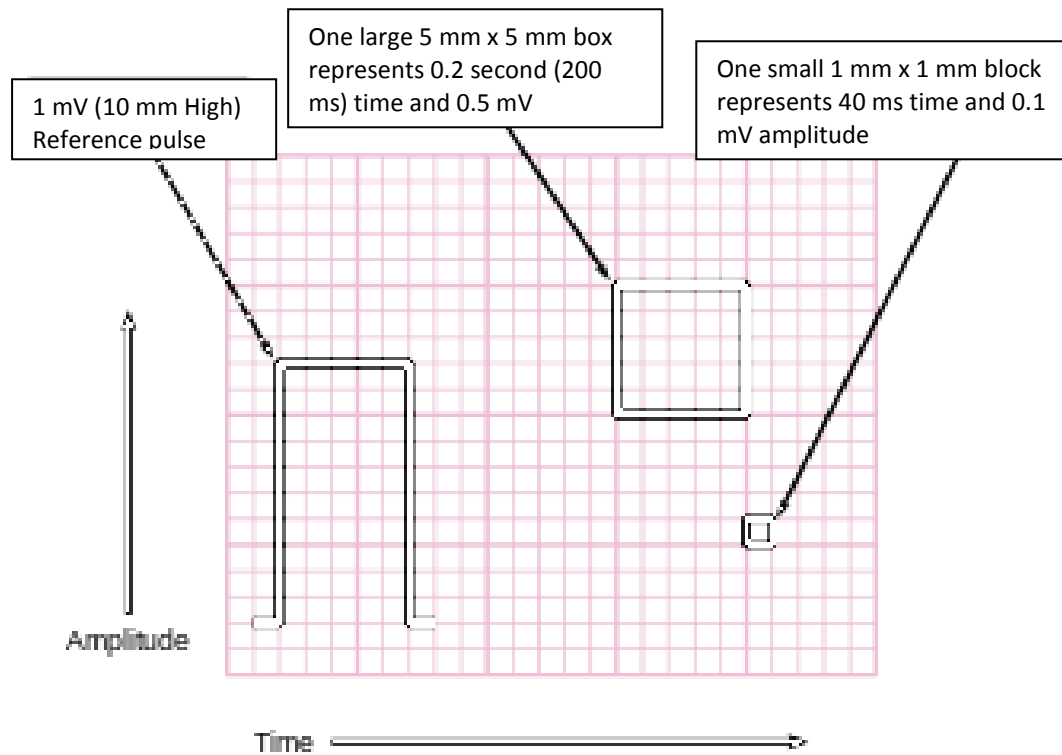


## II. TINJAUAN PUSTAKA

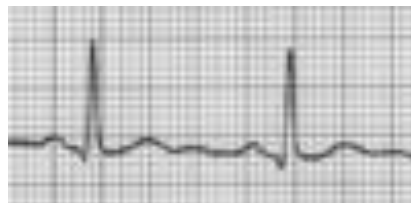
Dalam bab ini akan dijelaskan beberapa informasi penting terkait dengan penelitian yang dilakukan. Mulai dari penjelasan terkait EKG dan pengolahan citra yang melingkupi tentang bagaimana membaca *image* dan *pixel image*, *cropping*, *resize*, membuat *image* dalam skala keabuan (*grayscale*), binerisasi, melakukan “pembersihan” *gridlines*, *contrast*, dan *brightness*, serta operasi perhitungan terkait dengan konversi gambar 2 dimensi menjadi gambar sinyal 1 dimensi.

### 2.1. EKG

EKG merupakan singkatan dari elektrokardiografi, merupakan salah satu cabang ilmu kedokteran yang mempelajari tentang aktivitas listrik jantung manusia. EKG merupakan sebuah alat kesehatan yang mampu menganalisa dan mendiagnosis kelainan kerja jantung.



Gambar 2.1. Kertas perekam EKG.



Gambar 2.2. Gelombang sinyal EKG.

Sebuah elektrokardiograf khusus berjalan di atas kertas dengan kecepatan 25 mm/s, meskipun kecepatan yang di atas daripada itu sering digunakan. Setiap kotak kecil kertas EKG berukuran 1 mm<sup>2</sup>. Dengan kecepatan 25 mm/s, 1 kotak kecil kertas EKG sama dengan 0,04 s (40 ms). 5 kotak kecil menyusun 1 kotak besar, yang sama dengan 0,20 s (200 ms). Karena itu, ada 5 kotak besar per detik. 12 sadapan EKG berkualitas diagnostik dikalibrasikan sebesar 10 mm/mV, jadi 1

mm sama dengan 0,1 mV. Sinyal "kalibrasi" harus dimasukkan dalam tiap rekaman. Sinyal standar 1 mV harus menggerakkan jarum 1 cm secara vertikal, yakni 2 kotak besar di kertas EKG.

Hasilnya adalah rekaman aktivitas kegiatan jantung berupa sinyal yang tergambar pada kertas yang bergridlines. Dari kertas EKG inilah dokter dapat mendiagnosis kesehatan jantung manusia.

## **2.2. Citra (*Image*)**

Definisi dari Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek dalam bentuk 2 dimensi (2D) yang mempunyai panjang dan lebar. Citra disebut juga dengan gambar atau *image*. Beberapa definisi dari citra adalah :

- Citra (*Image*) merupakan gambar pada bidang 2 (dua) dimensi.
- Citra (ditinjau dari sudut pandang matematis) merupakan fungsi yang kontinyu dari intensitas cahaya pada bidang 2 (dua) dimensi.
- Citra merupakan sumber cahaya menerangi objek dan dipantulkan kembali dan di tangkap oleh alat-alat optik misalnya mata manusia, kamera, *scanner*, sensor, satelit dan sebagainya, yang kemudian bayangan pantul yang disebut juga dengan citra dapat terekam.

Selain dalam bentuk 2 dimensi ada juga citra dalam bentuk 1 dimensi, yaitu sebuah gambar/citra yang hanya memiliki panjang, tidak memiliki lebar atau ruang lingkup. Contoh dari gambar 1 dimensi adalah sebuah garis lurus. Garis memiliki panjang tetapi tidak memiliki lebar maupun ruang lingkup.

Citra sebagai keluaran dari sistem perekam data dapat bersifat :

- Optik atau *photo*
- Analog berupa sinyal video seperti tampilan gambar pada monitor televisi
- Digital dapat disimpan langsung pada media perekam magnetik

Citra juga dapat dikelompokkan menjadi dua bagian yaitu :

- Citra tampak (*photo*, gambar, lukisan, apa yang tampak dilayar monitor/ televisi, hologram dll).
- Citra tidak tampak (data pada photo/ gambar pada file, citra yang direpresentasikan dalam fungsi matematis).

Ada dua jenis citra yang kita kenal, yaitu citra analog dan citra digital.

#### 1. Citra digital

Citra digital adalah sebuah representasi numerik dengan nilai-nilai diskrit yang bisa terbaca dan diolah oleh komputer. Pada citra digital sebenarnya terdiri dari kumpulan kotak-kotak yang terbentuk dari perpotongan baris dan kolom di mana setiap kotak berisi angka-angka yang merepresentasikan warna pada komputer. Kotak-kotak berisi angka tersebut dinamakan *Picture element* atau sering kita sebut dengan *Pixel*. Hanya citra digital yang dapat diolah menggunakan komputer. Untuk citra non digital sebelum diolah harus dirubah terlebih dahulu kedalam bentuk citra digital. Citra digital merupakan gambar yang dihasilkan dari gambar analog 2 dimensi yang kontinyu menjadi gambar diskrit melalui proses sampling.

Sampling adalah proses untuk menentukan warna pada *pixel* tertentu pada citra dari sebuah gambar kontinyu. Proses ini bisaanya dengan mencari

nilai rata-rata dari gambar analog yang kemudian dibulatkan. Sampling menyatakan banyaknya *pixel* (blok) untuk mendefinisikan suatu gambar, sedangkan kuantisasi menunjukkan banyaknya derajat nilai pada setiap *pixel*.

Pada umumnya citra digital berbentuk persegi panjang, dan dimensi ukurannya dinyatakan sebagai tinggi x lebar atau panjang x lebar atau baris x kolom. Citra digital merupakan representasi dari kumpulan baris dan kolom yang memiliki nilai-nilai yang merepresentasikan warna. Suatu citra dapat didefinisikan sebagai fungsi  $f(x,y)$ , berukuran M baris dan N kolom, dimana  $x$  dan  $y$  adalah koordinat spasial dari citra, sedangkan  $f(x,y)$  merupakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai  $x$ ,  $y$ , dan  $f$  secara keseluruhan memiliki nilai yang berhingga (finite) dan bernilai diskrit, maka citra tersebut disebut dengan citra digital.<sup>[8]</sup>

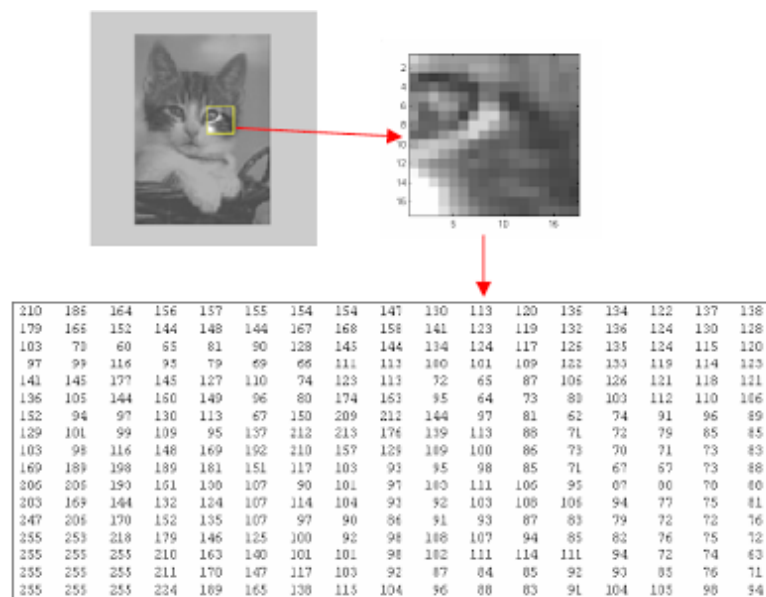
Citra digital merupakan representatif dari citra yang diambil oleh mesin dengan bentuk pendekatan berdasarkan sampling dan kuantisasi. Sampling menyatakan besarnya kotak-kotak yang disusun dalam baris dan kolom atau dengan kata lain sampling pada citra menyatakan besar kecilnya ukuran *pixel* (titik) pada citra, dan kuantisasi menyatakan besarnya nilai tingkat kecerahan yang dinyatakan dalam nilai tingkat keabuan (*grayscale*) sesuai dengan jumlah bit biner yang digunakan oleh mesin atau dengan kata lain kuantisasi pada citra menyatakan jumlah warna yang ada pada citra.<sup>[1]</sup>

Citra digital dapat dimodelkan sebagai suatu matriks dimana indeks baris dan kolomnya menyatakan suatu titik pada citra tersebut dan elemen

matriksnya menyatakan tingkat keabuan pada titik tersebut. Berikut ini adalah pemodelan citra digital dalam bentuk matriks berukuran  $N \times M$ .

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

Nilai pada suatu irisan antara baris dan kolom (pada posisi  $x, y$ ) disebut dengan *picture element*, *image element*, *pels*, atau *pixel*. Istilah *pixel* merupakan istilah yang sering digunakan dalam citra digital.<sup>[8]</sup> Berkaitan dengan pemodelan citra digital dalam bentuk matriks, Basuki (2005) memberikan gambaran sebagai berikut.<sup>[1]</sup>



Gambar 2.3. Matriks nilai *pixel image*.

## 2. Citra Analog

Citra analog adalah citra bersifat kontinu, seperti gambar pada televisi, foto sinar-x, foto yang tercetak pada kertas foto, lukisan, hasil *ct-scan*, gambar-gambar yang terekam pada pita kaset, dan lain sebagainya. Citra analog tidak dapat direpresentasikan pada komputer, sehingga tidak dapat diproses komputer secara langsung. Maka dari itu, agar sebuah citra analog bisa diproses oleh komputer, maka citra tersebut harus terlebih dahulu dikonversi ke dalam bentuk citra digital. Citra analog dihasilkan oleh alat-alat analog, video kamera analog, kamera foto analog, *CT scan*, sensor *Ultra Sonic* pada *USG*, *printout EKG*, dan lain sebagainya.<sup>[1]</sup>

### 2.3. Jenis Citra Digital

Berdasarkan jenisnya, citra digital dapat dibagi menjadi 3<sup>[4]</sup>, yaitu:

#### 1. Citra Biner (Monokrom)

Memiliki 2 buah warna, yaitu hitam dan putih. Warna hitam bernilai 1 dan warna putih bernilai 0. Untuk menyimpan kedua warna ini dibutuhkan 1 bit di memori. Contoh dari susunan *pixel* pada citra monokrom adalah sebagai berikut:

						=	1	1	1	1	1
						=	0	0	1	0	0
						=	0	0	1	0	0
						=	0	0	1	0	0
						=	0	0	1	0	0

Gambar 2.4. Citra biner dan nilainya.

Contoh gambar citra biner:



Gambar 2.5. Citra biner luffi2.jpg.

## 2. Citra *Grayscale* (skala keabuan)



Gambar 2.6. Skala keabuan.

Citra *grayscale* mempunyai kemungkinan warna hitam untuk nilai minimal dan warna putih untuk nilai maksimal. Banyaknya warna tergantung pada jumlah bit yang disediakan di memori untuk menampung kebutuhan warna tersebut. Semakin besar jumlah bit warna yang disediakan di memori, maka semakin halus gradasi warna yang terbentuk. Contoh: skala keabuan 2 bit, jumlah kemungkinan  $2^2 = 4$  warna. Jadi, kemungkinan warna 0 (minimal) sampai 4 (maksimal). Contoh gambar berskala keabuan :





Gambar 2.7. Citra keabuan luffi1.jpg.

Untuk memudahkan proses perhitungan dan pengolahan citra, diperlukan perubahan bentuk *image* dari format RGB ke dalam bentuk *grayscale*. Akan sangat merepotkan jikalau dalam kasus ini, harus mengolah citra dengan 3 *element* penstruktur *red*, *green*, dan *blue*, sebab itu berarti harus mengolah 3 nilai *pixel*. Untuk memudahkan proses perhitungan, maka dilakukan pencarian nilai rata-rata *pixel* RGB, yaitu dengan menjumlahkan matriks R, matriks G, matriks B kemudian dibagi 3. Sehingga diperoleh nilai rata-rata yang disebut skala keabuan. Disebut skala keabuan karena jikalau direpresentasikan ke dalam gambar, akan terbentuk suatu *image* berwarna abu-abu.

Secara umum citra berwarna dapat dikonversi ke citra berskala keabuan melalui rumus :

$$I = a \times R + b \times G + c \times B, a+b+c = 1 \quad (2.2)$$

Sebagai contoh :

$$R = 50$$

$$G = 70$$

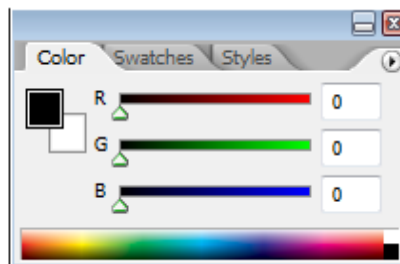
$$B = 60$$

Jika diasumsikan nilai  $a$ ,  $b$ ,  $c$  pada persamaan di atas dibuat sama, akan diperoleh hasil seperti berikut :

$$I = (50+70+60)/3 = 60 \quad (2.3)$$

Nilai skala keabuan berkisar antara 0 – 255. Nilai 0 merepresentasikan warna hitam dan nilai 255 merepresentasikan putih. <sup>[1]</sup>

### 3. Citra Warna (*true colour*)



Gambar 2.8. Skala RGB.

Setiap *pixel* pada citra warna mewakili warna yang merupakan kombinasi tiga warna dasar, yaitu merah, hijau, dan biru (RGB = *Red, Green, Blue*). Setiap warna dasar menggunakan penyimpanan 8 bit = 1 *byte* (nilai maksimum 255 warna), jadi satu *pixel* pada citra warna diwakili oleh 3 *byte*.

Tabel nilai yang merepresentasikan warna RGB dapat dilihat di bawah ini <sup>[4]</sup>:

Tabel 2.1. Warna dan nilai penyusun warna.

Warna	R	G	B
Merah	255	0	0
Hijau	0	255	0
Biru	0	0	255
Hitam	0	0	0
Putih	255	255	255
Kuning	0	255	255

Contoh representasi nilai dalam bentuk gambar dapat dilihat pada gambar 2.9 di bawah ini.



Gambar 2.9. Citra berwarna dan representasi warnanya. setiap *pixel* dinyatakan dengan nilai R, G, dan B.

## 2.4. PENGOLAHAN CITRA DIGITAL

Dalam subbab ini, akan dijelaskan beberapa operasi dan jenis pengolahan citra yang akan digunakan dalam penelitian. Meskipun sebuah citra kaya akan informasi, namun sering kali citra yang kita miliki mengalami penurunan mutu (*degradasi*), misalnya mengandung cacat atau derau (*noise*), warnanya terlalu kontras, atau kurangnya tajam, kabur (*blurring*), dan sebagainya. Citra yang semacam ini lebih sulit diinterpretasikan karena informasi yang disampaikan oleh citra tersebut menjadi berkurang.<sup>[8]</sup>

Agar citra yang mengalami gangguan mudah diinterpretasikan (baik oleh mesin maupun manusia), maka citra tersebut perlu dimanipulasi agar informasi pada citra kembali utuh bahkan lebih baik. Proses manipulasi citra menggunakan komputer ini disebut pengolahan citra.

Pengolahan citra digital adalah salah satu bentuk pemrosesan informasi dengan inputan berupa citra (*image*) dan keluaran yang juga berupa citra atau dapat juga bagian dari citra tersebut. Tujuan dari pemrosesan ini adalah memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin komputer. Operasi-operasi pada pengolahan citra digital secara umum dapat diklasifikasikan sebagai berikut:

- Perbaikan kualitas citra (*image enhancement*), contohnya perbaikan kontras gelap/terang, penajaman (*sharpening*), dan perbaikan tepian objek (*edge enhancement*)
- Restorasi citra (*image restoration*), contohnya penghilangan kesamaran (*deblurring*)

- Pemampatan citra (*image compression*)
- Segmentasi citra (*image segmentation*)
- Pengorakan citra (*image analysis*), contohnya pendeteksian tepi objek (*edge enhancement*) dan ekstraksi batas (*boundary*)
- Rekonstruksi citra (*image reconstruction*)

Secara khusus, beberapa pengolahan citra digital yang digunakan dalam penelitian ini yaitu :

### **1. Pembacaan *Image***

Pembacaan *image* yang akan dibahas adalah pembacaan *image* menggunakan komputer. Gambar digital pertama kali diproduksi oleh kabel sistem transmisi Bartlane, pada tahun 1920. Kemudian dikembangkan oleh 2 penemu dari Inggris, Harry G. Bratholomew dan Maynard D. Mcfarlane. Prosesnya sendiri terdiri dari “serangkaian negatif pada seng piring yang terbuka untuk jangka waktu yang lama, sehingga menghasilkan kepadatan yang berbeda-beda.”<sup>[12]</sup> Kabel Bartlane sistem transmisi gambar yang dihasilkan pada kedua transmitter dan receiver yang mengakhiri data menekan kartu atau tape yang diciptakan sebagai gambar.<sup>[11]</sup> Pada saat itu, diperoleh potongan waktu pengiriman data dari 1 minggu menjadi hanya kurang dari 3 jam. Foto tersebut dikirim dalam bentuk kode digital, selanjutnya diubah kembali oleh suatu printer *telegraph* pada sisi penerima. Masalah yang muncul pada saat itu berkisar pada teknik transmisi data secara digital serta teknik reproduksi pada sisi penerima untuk mendapatkan satu resolusi gambar yang baik.

Perkembangan secara pesat baru tercatat pada sekitar tahun 1960. Pada saat itu, pertama kali ditemukannya peralatan yang menghasilkan gambar digital secara *real time* yaitu radiografi digital *fluoroscopic*.<sup>[12]</sup> Pada saat itu, teknologi komputer juga telah dianggap memenuhi suatu kecepatan proses serta kapasitas memori yang dibutuhkan oleh berbagai algoritma pengolahan citra. Hingga sekarang, kemajuan teknologi menjadikan kebutuhan akan digitalisasi gambar hampir berlaku untuk setiap orang yang hidup di Bumi ini.

Secara umum, ada beberapa alasan mengapa diperlukannya komputasi gambar, yaitu :

a. Media Penyimpanan

Gambar dalam bentuk *hardcopy* memerlukan media penyimpanan yang luas dan membutuhkan waktu tenaga dan biaya lebih banyak daripada penyimpanan gambar dalam bentuk *softcopy*. Komputer dengan kapasitas memori 500 Gb dapat menampung lebih dari 500.000 gambar dengan asumsi setiap gambar memiliki ukuran data 1 Mb. Bandingkan jika semua gambar tersebut berupa *hardcopy*, maka akan diperlukan ruang penyimpanan yang sangat luas dan rentan terjadi kerusakan.

b. Backup data

Salah satu alasan manusia untuk melakukan digitalisasi gambar adalah untuk mem-*backup* data. *Backup* data dapat dijadikan bukti penguat kepemilikan dan bukti-bukti penting lainnya, apabila terjadi kehilangan data atau gambar.

c. Pengolahan gambar

Manipulasi gambar sudah menjadi kebutuhan hidup di zaman sekarang. Hampir semua gambar yang berada di layar monitor komputer merupakan hasil dari pengolahan citra/gambar. Di dalam semua khazanah keilmuan, pengolahan citra memiliki peran yang besar. Sebagai contoh, dalam bidang kedokteran, pengolahan citra digunakan untuk mendeteksi jenis penyakit organ dalam, seperti jantung (EKG), scanning otak, pencarian tulang yang patah (*Rontgent*), pengecekan janin (USG), dan lainnya. Dalam bidang pertanian, pengolahan citra juga memiliki peran-peran penting seperti dalam pendeteksian hama dan penyakit, kesuburan tanah, dan lainnya.

d. Pengiriman data

Pengiriman data menjadi kebutuhan yang tidak kalah penting bagi kehidupan manusia. Saat ini, dengan digitalisasi gambar, kita dapat melakukan pengiriman gambar apapun, kepada siapapun, dan kapanpun dengan cepat dan murah, dengan ketentuan terdapat akses jaringan komunikasi dan alamat yang tepat. Sehingga orang-orang dapat saling berbagi gambar untuk menunjukkan kebahagiaan, saling bertukar kabar, dan sebagainya.

Pembacaan *image* digital memerlukan alat pengambil gambar digital sebelum data gambar dimasukan ke dalam komputer. Secara urutan, digitalisasi gambar adalah sebagai berikut :

- Media pengambilan gambar

Diperlukan untuk mengambil objek gambar secara langsung. Setelah gambar diambil, kemudian disimpan dalam format yang bisa dibaca oleh komputer dan media pengolahan citra lainnya. Contoh : kamera digital, scanner, kamera handphone, dan lainnya.

- Media koneksi

Media koneksi diperlukan untuk melakukan pengiriman gambar digital dari media pengambilan gambar ke dalam komputer. Contoh media koneksi : USB *flashdisk*, kabel USB, *bluetooth*, *Infrared*, dan lain sebagainya.

- Media komputer

Media komputer diperlukan untuk berbagai kegiatan dalam rangka mempermudah urusan manusia. Secara khusus, dalam pembacaan *image*, komputer memiliki peran sentral sebagai media penyimpanan, penampilan, dan pengolahan citra. Meskipun, saat ini banyak dikembangkan media aplikasi pengolahan citra dalam format *handphone* atau *smartphone*, tetapi komputer (PC dan laptop) memiliki keunggulan yang tidak tergantikan, yaitu semua pengembangan aplikasi pengolahan citra dilakukan oleh komputer.

Pembacaan file gambar pada komputer sangat berkaitan dengan format penyimpanan file gambar. Format penyimpanan file gambar secara umum dibagi menjadi dua, yaitu format untuk citra bitmap dan format untuk citra *vector*.



- Format file citra Bitmap

Citra bitmap menyimpan data kode citra secara digital dan lengkap (cara penyimpanannya per *pixel*). Citra bitmap direpresentasikan dalam bentuk matrik nilai *pixel* citra baik dalam bentuk nilai integer, biner atau sistem bilangan lainnya. Kelebihan dari format ini adalah manipulasi warna lebih mudah dilakukan. Kemudian format ini mampu menunjukkan kehalusan gradasi bayangan dan warna dari sebuah gambar. Dengan kelebihan seperti dijelaskan di atas, maka format ini sangat sesuai untuk menyimpan gambar dengan perpaduan gradasi warna yang rumit, seperti foto dan lukisan digital.

Kelemahan dari format ini adalah kualitas yang menurun ketika gambar diperbesar. Ketika sebuah gambar dengan format ini diperbesar maka akan tampak pecah. Kemudian kelemahan yang lain adalah lebih sulit dalam mengubah bentuk objek gambar.

JPEG adalah jenis format penyimpanan yang paling banyak digunakan. Hal ini dikarenakan format ini mampu menyimpan data gambar dengan tingkat gradasi warna yang tinggi. Selain JPEG, format yang sering digunakan dalam penyimpanan gambar adalah PNG (ping). Hal ini karena dalam dapat melakukan pemampatan data tanpa menghilangkan kualitas gambar.

Format gambar bitmap bisaanya di dapat dari penyimpanan scanner, kamera digital, dan lainnya. Beberapa format yang bisaanya digunakan dalam pengolahan citra secara umum dapat dilihat pada table 2.1. di bawah ini :

Table 2.2. Format penyimpanan image bitmap pada komputer.

Nama format	Ekstensi	Keterangan
<i>Microsoft Words Bitmap Format</i>	.bmp	Format umum yang dikembangkan oleh Microsoft untuk menyimpan citra bitmap
<i>Compuserve Graphics Interface Format</i>	.gif	<i>Graphics Interface Format</i> merupakan format yang memungkinkan pemampatan data hingga 50%. Cocok untuk citra yang memiliki area yang cukup besar dengan warna yang sama.
<i>Aldus Tagged Image File Format</i>	.tif, .tiff	<i>Tagged Image File Format</i> merupakan format citra yang mula-mula dibuat oleh Aldus. Kemudian, dikembangkan oleh <i>Microsoft</i> dan terakhir oleh <i>Adobe</i> .
<i>Word Perfect Graphics Format</i>	.wpg	Format <i>vector</i> yang juga mendukung citra bitmap
<i>GEM Image Format</i>	.img	Format bitmap yang dikembangkan untuk riset digital di lingkungan GEM
<i>Zsoft Pengolahan Citra Painbrush Format</i>	.pcx	Dirancang untuk menyimpan citra layar dan merupakan format bitmap yang didukung luas
<i>Microsoft Paint Bitmap Format</i>	.msp	Secara fungsional mirip dengan IMG dan PCX, tapi kurang populer
<i>AT &amp; T Targa format</i>	.tga	Format untuk 16-bit dan 24-bit citra warna penuh, diciptakan untuk system <i>truevision</i>
<i>Apple Mac Paint Format</i>	.pntg	Format asli dari <i>Macintosh Macpaint</i> program
<i>Sun Microsystem Raster Format</i>	.ras	Format bitmap asli yang digunakan pada pada sun SPARCS
<i>X Windows X-11 Bitmap Format</i>	.xbm	Format umum untuk menyimpan citra bitmap yang dikembangkan untuk X Windows

<i>Joint Photographic Expert Group</i>	.jpg, .jpeg	<i>Joint Photographics Expert Group</i> adalah format citra yang dirancang agar bisa memampatkan data dengan rasio 1:16
<i>Portable Network Graphics</i>	.png	<i>Portable Network Graphics</i> bisaa dibaca 'ping'. Asal mulanya dikembangkan sebagai pengganti format GIF karena adanya penerapan lisensi GIF. Mendukung pemampatan data tanpa menghilangkan informasi aslinya.

- Format file citra *vector*

Citra *vector* dihasilkan tidak berdasarkan *pixel*, tetapi perhitungan matematis, yaitu data tersimpan dalam bentuk *vector* posisi, di mana yang tersimpan hanya informasi *vector* posisi dalam bentuk sebuah fungsi.

Secara umum kelebihan dan kekuarangan file *vector* berbanding terbalik dengan format penyimpanan bitmap. Kelebihan format ini adalah tidak mengubah kualitas gambar ketika diperbesar ataupun diperkecil. Sedangkan kelemahannya adalah kesulitan dalam memanipulasi warna.

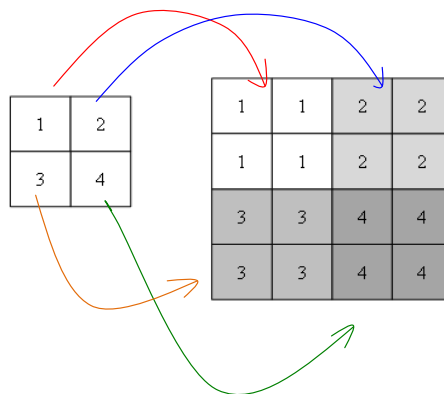
Format ini bisaa dipakai dalam aplikasi-aplikasi citra *vector*, seperti *CorelDRAW*, *Adobe photoshop*, *AutoCAD*, *Macromedia freehand*, dan lain-lain.

Bentuk-bentuk format file citra *vector* yaitu, *AutoCAD Drawing Format* (DWG), *AutoCAD Drawing Exchange Format* (DXF), *Microstation Drawing Format* (DGN), dan *Scalable Vector Graphics* (SVG).

## 2. *Resize Image*

Dalam bahasa yang paling sederhana *resize* berarti mengubah resolusi atau ukuran horizontal dan vertikal suatu citra (gambar). *Resize* dapat membatasi dan atau meninggikan serta melebarkan citra yang sudah ditampilkan. Pasokan maksimal, terdapat pada *pixel*, untuk satu atau kedua dimensi. Seperti tercantum dalam preferensi , aspek rasio (*Aspek rasio adalah rasio panjang dan lebar sebuah foto. Jika panjang dan lebar foto sama, maka aspek rasionya 1:1* ) gambar itu dipertahankan . Jika membatasi kedua tinggi dan lebar, beberapa aplikasi pengolahan citra akan mempertahankan aspek rasio dengan menggunakan maksimum yang ditetapkan untuk satu dimensi dan sesuatu yang kurang maksimal untuk dimensi lainnya. Hal ini diperlukan untuk meminimalisir besar *image*, sehingga process dapat berjalan dengan baik dan cepat.

Pada dasarnya *resize image* hanya ada dua, yaitu memperbesar dan memperkecil citra. Dalam perbesaran citra, maka akan dilakukan dengan membuat setiap *pixel* menjadi beberapa *pixel*. Gambar 5.13 memberikan contoh cara memperbesar citra.



Gambar 2.10. Cara memperbesar citra.

Pada contoh di atas pembesaran pada arah vertikal dan horizontal sebesar 2 kali. Ada juga perbesaran citra yang berbeda antara vertical dan horizontalnya. Jadi tergantung algoritma yang digunakan untuk mengatur perbesaran citra tersebut.

Secara prinsip, pengecilan citra berarti mengurangi jumlah *pixel*. Algoritma yang digunakan untuk mewujudkan perbesaran citra dapat digunakan untuk melakukan pengecilan citra dengan mengganti nilai pembanding menjadi nilai pecahan seperti  $1/2$ ,  $1/4$ ,  $1/8$ , dan seterusnya.

### 3. *Crop*

*Cropping* diperlukan untuk memilih gambar secara efisien dan fleksibel (tergantung kebutuhan dari *user*). Sehingga dapat dilakukan pemilihan gambar yang benar-benar diinginkan dan dibutuhkan *user*. *Crop* digunakan untuk mengambil sebagian persegi panjang dari suatu gambar pada aplikasi pengolahan citra, fungsi yang digunakan bervariasi, ada yang disimbolkan dengan gambar, ada juga melalui perintah pada *workspace*. Apapun itu, perintah atau fungsi *crop* menerima dua argumen utama :

1. Gambar yang akan dipotong
2. Koordinat persegi panjang yang mendefinisikan area yang di *crop*

Penggunaan fungsi *crop* tanpa menentukan nilai persegi panjang wilayah yang di *crop*, dapat menentukan besar persegi panjang wilayah yang akan di *crop* secara interaktif. Dalam hal ini, kursor berubah menjadi bidik ketika berada di atas gambar. Posisi bidik melalui sudut wilayah gambar yang akan di *crop* kemudian secara bersamaan tekan dan tahan tombol kiri mouse. Apabila ditarik garis bidik di atas gambar yang Anda tentukan wilayah persegi

panjangnya, fungsi *crop* akan menarik persegi panjang di sekitar area yang dipilih. Kemudian daerah (gambar) baru yang telah daerah yang dipilih akan ditampilkan setelah kursor dilepaskan. Fungsi *crop* biasanya ditampilkan dalam jendela gambar dan menunggu untuk dilakukan penggambaran/ penarikan bidik berbentuk persegi panjang pada gambar. Dalam gambar, persegi panjang yang dipilih akan ditampilkan dalam warna yang berbeda.

#### 4. Meningkatkan kecerahan

Merupakan bentuk pengolahan citra digital yang lebih sering dikenal sebagai *brightness*. Operasi ini dilakukan untuk membuat gambar menjadi lebih terang.

Secara matematis, peningkatan kecerahan dilakukan dengan cara menambahkan suatu konstanta terhadap nilai seluruh *pixel*. Misalkan,  $f(y, x)$  menyatakan nilai *pixel* pada citra berskala keabuan pada koordinat  $(y, x)$ . Maka, citra baru diperoleh seperti persamaan dibawah ini:

$$G(y, x) = f(y, x) + \beta \quad (2.4)$$

Artinya, citra  $G$  telah mengalami peningkatan nilai kecerahan di semua *pixel*-nya sebesar  $\beta$  terhadap citra asli  $f(y, x)$ . apabila  $\beta$  berupa bilangan negatif, kecarahan akan menurun menjadi lebih gelap. Pada prinsipnya persamaan di atas dapat diimplementasikan tidak hanya untuk bentuk citra digital *grayscale* tetapi juga untuk citra berwarna.

#### 5. Contrast

*Contrast* dalam suatu citra menyatakan distribusi warna terang dan warna gelap. Suatu citra berskala keabuan dikatakan memiliki kontras rendah

apabila distribusi warna cenderung pada jangkauan aras keabuan yang sempit. Sebaliknya, citra mempunyai kontras tinggi apabila jangkauan aras keabuan terdistribusi secara melebar. Kontras dapat diukur berdasarkan perbedaan antar nilai intensitas tertinggi dengan intensitas terendah yang menyusun *pixel-pixel* dalam citra.

Dalam pengolahan citra, pengaturan kontras dilakukan untuk mendapatkan hasil citra yang tegas. Secara matematis agar distribusi nilai intensitas *pixel* berubah, perlu dilakukannya peregangan kontras. Dengan rumus matematis sebagai berikut

$$G(y, x) = \alpha f(y, x) \quad (2.5)$$

Berdasarkan rumus di atas, kontras akan naik apabila nilai  $\alpha > 1$ , dan kontras akan turun apabila nilai  $\alpha < 1$ .

## 6. Batas ambang (*Threshold*)

Dalam beberapa kasus pengolahan citra, bisaanya citra warna akan dikonversikan ke dalam bentuk citra *grayscale* dikarenakan untuk menyederhanakan operasi pengolahan citra. Dalam kasus lain, ada juga pengolahan citra yang memerlukan konversi dari citra *grayscale* menjadi citra biner, dikarenakan ada beberapa operasi yang hanya bisa dijalankan menggunakan jeniis citra ini. Ketika pengolahan citra memerlukan konversi dari citra *grayscale* menjadi biner, maka diperlukanlah nilai batas ambang (*thresh*). Nilai *thresh* inilah yang menjadi batas nilai konversi *pixel* menjadi 1 dan 0. Secara sederhana, dapat dijelaskan sebagai berikut, semisal ada sebuah citra warna dengan nilai 3 elemen *pixel* RGB kemudian dikonversi menjadi citra *grayscale*, maka, pembagian rerata nilai *pixel* dari 3 elemen RGB inilah

yang kemudian direpresentasikan dalam sebuah citra *grayscale* yang hanya memiliki 1 elemen dan bernilai antara 0 sampai 255. Ketika citra *grayscale* ini dikonversi ke biner menggunakan batas ambang, maka nilai *pixel* citra yang > dari batas ambang ataupun < dari batas ambang dapat dikonversikan menjadi nilai 0 dan 1 tergantung dengan kebutuhan. Sehingga akan terbentuklah nilai-nilai *pixel* biner (0 dan 1) yang ketika direpresentasikan ke dalam bentuk gambar hanya memiliki warna hitam dan putih.

## **2.5. Rekayasa Perangkat Lunak**

Rekayasa perangkat lunak merupakan sebuah disiplin ilmu dalam bidang teknologi informasi, yang mencakup tentang bagaimana membuat perangkat lunak, pemeliharaan perangkat yang dibuat, manajemen organisasi pengembangan dan manajemen kualitasnya<sup>[7]</sup>. Teknologi software saat ini sudah semakin berkembang pesat. Saat ini sudah tidak zaman lagi seorang programmer bekerja sendirian. Software saat ini digunakan untuk berbagai macam keperluan dalam pengolahan produksi, manajerial, pencarian, modifikasi, menampilkan, dan mentransfer informasi kemanapun di seluruh dunia<sup>[7]</sup>. Sehingga programmer dituntut untuk bekerja secara terorganisir dan memahami keilmuan dalam merancang, membuat, mengembangkan, mengorganisasi, dan pemeliharaan software atau sistem perangkat lunak.

Ada beberapa metode pengembangan perangkat lunak antara lain metode sekuensial linier, *prototype*, RAD, *Fourth Generation Techniques* (4GT), spiral, metode Proses Perangkat Lunak Evolusioner (pertambahan, spiral, konkruen), dan metode formal<sup>[7]</sup>.



## 1. Metode sekuensial linear

Metode sekuensial linear sering disebut juga sebagai *waterfall method* (metode air terjun). Ini dikarenakan karakteristik metode yang digunakan dalam pengembangan perangkat lunak bersifat sistematis dan sekuensial. Di mana output dari setiap step digunakan sebagai input step berikutnya. Metode ini merupakan metode yang paling tua dan paling sering digunakan. Metode ini dimulai dari analisis, kemudian desain, kode, pengujian, dan pemeliharaan. Model ini melingkupi<sup>[7]</sup> :

### a. Rekayasa dan pemodelan system/informasi

Perangkat lunak adalah bagian dari sistem yang lebih besar, pekerjaan dimulai dari pembentukan kebutuhan-kebutuhan untuk seluruh elemen sistem dan kemudian memilah mana yang akan digunakan untuk pengembangan perangkat lunak. Hal ini menjadi penting, karena perangkat lunak harus berkomunikasi dengan *hardware*, orang dan basis data.

### b. Analisis kebutuhan perangkat lunak

Pengumpulan kebutuhan dengan fokus pada perangkat lunak, yang meliputi : Domain informasi, fungsi yang dibutuhkan, unjuk kerja/performansi dan antarmuka. Hasilnya harus didokumentasi dan *direview* ke pelanggan/ pengguna.

### c. Desain

Ada 4 atribut untuk program yaitu : Struktur Data, Arsitektur perangkat lunak, Prosedur detil dan Karakteristik Antarmuka. Proses desain mengubah kebutuhan-kebutuhan menjadi bentuk karakteristik yang

dimengerti perangkat lunak sebelum dimulai penulisan program. Desain ini harus terdokumentasi dengan baik dan menjadi bagian konfigurasi perangkat lunak.

d. Generasi kode

Penterjemahan perancangan ke dalam bentuk yang dapat dimengerti oleh mesin, yaitu dengan menggunakan bahasa pemrograman.

e. Pengujian

Setelah kode program selesai testing dapat dilakukan. Testing memfokuskan pada logika internal dari perangkat lunak, fungsi eksternal Pemodelan system informasi analisis desain kode tes 3 dan mencari segala kemungkinan kesalahan dan memeriksa apakah sesuai dengan hasil yang diinginkan.

f. Pemeliharaan

Merupakan bagian paling akhir dari siklus pengembangan dan dilakukan setelah perangkat lunak dipergunakan. Pemeliharaan perangkat lunak mengaplikasikan lagi Setiap fase program sebelumnya dan tidak membuat yang baru. Kegiatan :

- *Corrective Maintenance* : Mengoreksi kesalahan pada perangkat lunak, yang baru terdeteksi pada saat perangkat lunak dipergunakan.
- *Adaptive Maintenance* : Penyesuaian dengan lingkungan baru, misalnya sistem operasi atau sebagai tuntutan atas perkembangan sistem komputer, misalnya penambahan *printer driver*
- *Perfektive Maintenance* : Bila perangkat lunak sukses dipergunakan oleh pemakai. Pemeliharaan ditujukan untuk menambah kemampuannya seperti memberikan fungsi-fungsi tambahan, peningkatan kinerja dan sebagainya.

## 2. *Prototype*

Metode *prototype* menggunakan sistem pembuatan contoh produk dalam skala kecil (*prototype*) dari sebuah objek atau rancangan sistem yang akan dibangun tanpa memperhatikan perincian kebutuhan input dan lainnya sebelum membuat sistem dalam skala yang besar. Metode ini berlaku untuk kondisi di mana pengguna hanya mendefinisikan perangkat lunak secara umum tanpa dapat merinci kebutuhan input, pemrosesan, maupun outputnya, dan pengembang tidak begitu yakin akan efisiensi dari algoritma, adaptasi sistem operasi, atau bentuk antarmukanya.

Secara umum metode ini digunakan untuk membuat contoh sistem yang akan dibangun dalam skala kecil yang kemudian oleh pengembang disajikan kepada pelanggan untuk melihat respon atau umpan balik, apakah produk dapat dilanjutkan pengembangannya atau tidak. Metode ini mempunyai keuntungan berupa proses produksi dapat dilakukan lebih cepat dan pengguna dapat berperan aktif dalam proses pengembangan sistem atau produk yang dibuat. Sedangkan kekurangannya adalah dengan proses pengembangan yang terlalu cepat menyebabkan banyak alternatif pemecahan masalah yang dikesampingkan. Kemudian juga tidak selamanya sebuah *prototype* dapat dengan mudah diubah-ubah.

## 3. RAD (*Rapid Application Development*)

RAD merupakan adaptasi dari model sekuensial dengan kecepatan yang tinggi. Syaratnya adalah harus sangat memahami proyek yang dibuat dan membatasi

ruang lingkup proyek dengan ketat dengan siklus pengembangan yang sangat pendek. Cakupan dari model RAD ini antara lain :

a. Pemodelan bisnis (*bussines modeling*)

Memodelkan aliran informasi bisnis dengan menggunakan 3W, What, where, who. Informasi Apa, kemana perginya dan siapa penerima informasi tersebut.

b. Pemodelan data (*data modeling*)

Digunakan untuk menyaring aliran informasi yang merupakan bagian dari fase pemodelan bisnis ke dalam serangkaian objek data yang dibutuhkan dalam menopang bisnis tersebut. Kemudian mendefinisikan karakteristik/atribut masing-masing objek dan hubungan antar objeknya.

c. Pemodelan proses (*proces modeling*)

Mentransformasikan aliran informasi yang didefinisikan dalam pemodelan data untuk mencapai aliran informasi yang diperlukan bagi implementasi sebuah fungsi bisnis. Penggambaran pemrosesan diciptakan untuk menambah, memodifikasi, menghapus atau mendapatkan kembali sebuah objek data.

d. Pembuatan aplikasi (*aplication modeling*)

Menggunakan alat bantu otomatis dalam memfasilitasi konstruksi perangkat lunak. Selain itu menggunakan bahasa pemrograman generasi 3 yang konvensional.

e. Pengujian dan pergantian (*testing and turnover*)

Pengujian yang dilakukan harus menyeluruh, meskipun masing-masing komponen telah diuji karena dipakai kembali.

Kelebihan RAD adalah pada penghematan waktu pembuatan, biaya dan waktu pengujian. Sedangkan kekurangannya ada pada komitmen pengembang dalam menyelesaikan proyek besar secara cepat.

#### 4. *Fourth Generation Techniques* (4GT)

Fourth generation merepresentasikan karakteristik perangkat lunak dengan level tinggi. Cakupan tools yang terdapat pada pengembangan perangkat lunak yang mendukung 4G yaitu bahasa non-prosedural untuk query basis data, report generation, data manipulation, interaksi layar, kemampuan grafik level tinggi, kemampuan spreadsheet.

Kelebihan model ini adalah pengurangan waktu dan peningkatan produktivitas secara besar, sedangkan kekurangannya terletak pada kesulitan penggunaan perangkat bantu (tools) dibandingkan dengan bahasa pemrograman, dan juga kode sumber yang dihasilkan kurang efisien.

#### 5. Proses Perangkat Lunak Evolusioner

Pada model ini ada 2 bagian yang memiliki peranan sangat penting. Pertama adalah model incremental, dan yang kedua adalah model spiral. Model incremental menggabungkan elemen-elemen model sekuensial secara berulang dengan filosofi iterative dari model *prototype*. Setiap kenaikannya menghasilkan versi kerja dengan peningkatan kualitas fungsi. Pada model ini tidak digunakan kode sekali pakai. Model spiral juga menggabungkan sifat iterative dari model *prototype* dengan kontrol sistematis yang ada pada model sekuensial.

## 6. Metode formal

Metode formal merupakan suatu metode yang sangat berpotensi menghasilkan perangkat lunak yang bebas dari cacat. Metode ini menggunakan notasi matematika yang ketat dalam menentukan, mendesain, dan memverifikasi system berbasis computer. Pembuktian matematis digunakan untuk memverifikasi kebenaran system, bukan untuk pengujian empiris. Contoh dari penggunaan model ini adalah perangkat lunak *cleanroom*. Kelemahan dari model ini adalah memakan banyak waktu dan biaya (mahal).