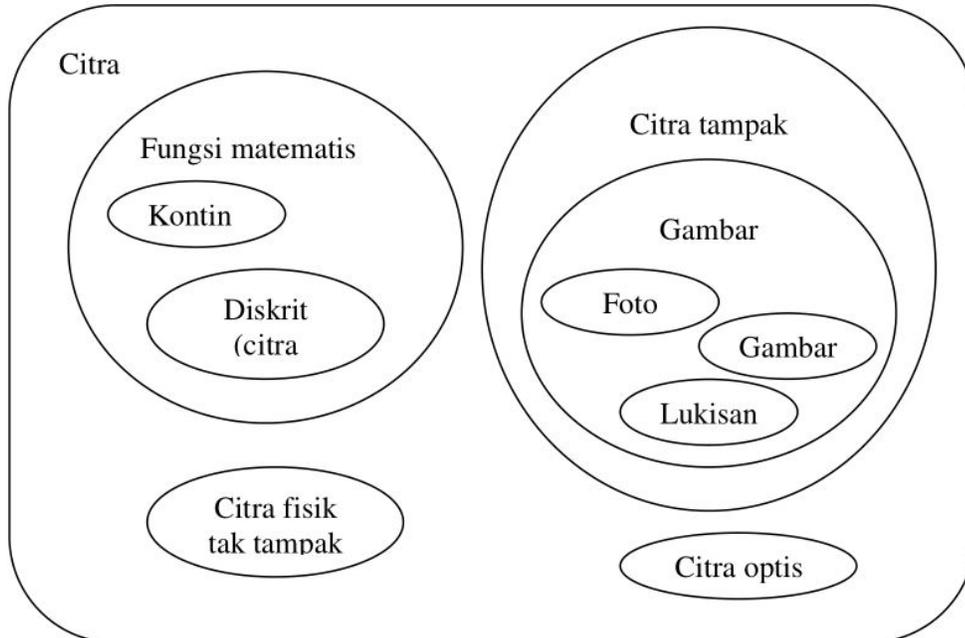


**BAB II**  
**LANDASAN TEORI**

**2.1 Citra (Gambar Diam)**

**2.1.1 Definisi Citra**

Definisi citra menurut Webster adalah suatu representasi, kemiripan, atau imitasi dari suatu obyek atau benda. Selain itu juga didalam sebuah citra juga terdapat Kompresi Citra adalah aplikasi kompresi data yang dilakukan terhadap citra digital dengan tujuan untuk mengurangi redundansi dari data-data yang terdapat dalam citra sehingga dapat disimpan atau ditransmisikan secara efisien. Citra dapat dikelompokkan menjadi citra tampak dan citra tak tampak, sebagaimana disajikan pada gambar 1 di bawah ini :



Gambar 2.1 Pengelompokan jenis-jenis citra (Castleman, 1996)

Contoh citra tampak dalam kehidupan sehari-hari adalah foto keluarga, gambar yang nampak pada layar monitor dan televisi, serta hologram (citra optis). Sedangkan contoh citra tak tampak adalah data gambar dalam *file* (citra *digital*) dan citra yang merepresentasikan menjadi fungsi matematis. Di samping itu ada juga citra fisik tak tampak, misalnya citra distribusi panas di kulit manusia serta peta densitas dalam suatu material. Untuk dapat dilihat mata manusia, citra tak tampak ini harus diubah menjadi citra tampak, misalnya dengan menampilkannya di monitor, dicetak di atas kertas, dan sebagainya.

### **2.1.2 Komponen Citra Digital**

Setiap citra *digital* memiliki beberapa karakteristik, antara lain ukuran citra, resolusi, dan format nilainya. Umumnya citra *digital* berbentuk persegi panjang yang memiliki lebar dan tinggi tertentu. Ukuran ini biasanya dinyatakan dalam banyaknya titik atau *pixel*, sehingga ukuran citra selalu bernilai bulat.

Format citra *digital* ada bermacam-macam. Karena sebenarnya citra mempresentasikan informasi tertentu, sedangkan informasi tersebut dapat dinyatakan secara bervariasi, maka citra yang mewakilinya dapat muncul dalam berbagai format. Citra yang mempresentasikan informasi yang hanya bersifat biner untuk membedakan 2 keadaan tentu tidak sama citra dengan informasi yang lebih kompleks sehingga memerlukan lebih banyak keadaan yang diwakilinya. Pada citra *digital* semua informasi disimpan dalam angka, sedangkan penampilan angka tersebut biasanya dikaitkan dengan warna.

Citra *digital* tersusun atas titik-titik yang biasanya berbentuk persegi panjang atau bujursangkar (pada beberapa system pencitraan, piksel-piksel penyusun citra ada pula yang berbentuk segi enam) yang secara beraturan membentuk barisan-barisan dan kolom-kolom.

Setiap titik memiliki koordinat sesuai dengan posisinya dalam citra. Koordinat ini biasanya dinyatakan dalam bilangan bulat positif, yang dapat dimulai dari 0 atau 1 bergantung pada system yang digunakan. Setiap titik juga memiliki nilai berupa angka digital yang mempresentasikan informasi yang diwakili titik tersebut. Format nilai piksel sama dengan format citra keseluruhan. Pada kebanyakan system pencitraan, nilai ini biasanya berupa bilangan bulat positif juga.

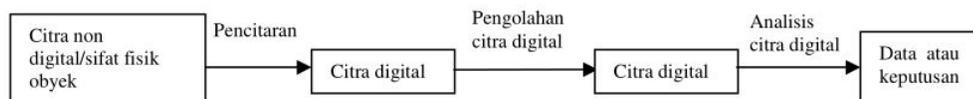
### **2.1.3 Representasi Citra**

Komputer dapat mengolah isyarat-isyarat elektronik digital yang merupakan kumpulan sinyal biner (bernilai dua: 0 dan 1). Untuk itu, citra digital harus mempunyai format tertentu yang sesuai sehingga dapat mempresentasikan obyek pencitraan dalam bentuk kombinasi data biner.

Pada kebanyakan kasus, terutama untuk keperluan penampilan secara visual, nilai data digital tersebut mempresentasikan warna dari citra yang diolah, dengan demikian format data citra digital berhubungan erat dengan warna. Format citra digital yang banyak dipakai adalah citra biner, skala keabuan, warna, dan warna berindeks.

### 2.1.4 Pengolahan Citra

Kegiatan untuk mengubah informasi citra fisik non digital menjadi digital disebut sebagai pencitraan (*imaging*). Citra digital dapat diolah dengan komputer karena berbentuk data numeris. Suatu citra digital melalui pengolahan citra *digital* (*digital image processing*) menghasilkan citra digital yang baru; termasuk di dalamnya adalah perbaikan citra (*image restoration*) dan peningkatan kualitas citra (*image enhancement*). Sedangkan analisis citra *digital* (*digital image analysis*) menghasilkan suatu keputusan atau suatu data; termasuk di dalamnya adalah pengenalan pola (*pattern recognition*), sebagaimana disajikan pada Gambar 2.2



Gambar 2.2 Urutan Pengolahan Citra Digital

Setiap tampilan citra digital memiliki beberapa karakteristik, antara lain ukuran citra, resolusi, dan format nilainya. Umumnya citra *digital* berbentuk persegi panjang yang memiliki lebar dan tinggi tertentu. Ukuran ini biasanya dinyatakan dalam banyaknya titik atau piksel (berasal dari kata *picture element*) yang memiliki koordinat sesuai dengan posisinya dalam citra. Koordinat ini biasanya dinyatakan dalam bilangan bulat positif, yang dapat dimulai dari 0 atau bergantung pada sistem yang digunakan. Setiap titik juga memiliki nilai berupa angka digital yang merepresentasikan informasi yang diwakili titik tersebut. Format nilai piksel sama dengan format citra

keseluruhan. Pada citra *digital* semua informasi disimpan dalam bentuk angka, sedangkan penampilan angka tersebut biasanya dikaitkan dengan warna.

### **2.1.5 Operasi Pengolahan Citra**

Pengolahan citra pada dasarnya dilakukan dengan cara memodifikasi setiap titik dalam citra tersebut sesuai dengan keperluan. Secara garis besar, modifikasi tersebut dikelompokkan menjadi :

1. Operasi titik, dimana setiap titik diolah secara tidak gayut terhadap titik-titik yang lain.
2. Operasi global, dimana karakteristik global (biasanya berupa sifat statistik) dari citra digunakan untuk memodifikasi nilai setiap titik.
3. Operasi temporal/berbasis bingkai, dimana sebuah citra diolah dengan cara dikombinasikan dengan citra lain.
4. Operasi geometri, dimana bentuk, ukuran, atau orientasi citra dimodifikasi secara geometris.
5. Operasi banyak titik bertetangga , dimana data dari titik yang bersebelahan (bertetangga) dengan titik yang ditinjau ikut berperan dalam mengubah nilai.
6. Operasi morfologi, yaitu operasi yang berdasarkan segmen atau bagian dalam citra yang menjadi perhatian.

Batasan yang dipakai dalam pengolahan citra yang difokuskan pada format citra skala keabuan 8 bit dengan warna hitam pekat untuk nilai

minimum (0) dan warna putih cemerlang untuk nilai maksimal (255), serta citra warna true color.

#### **2.1.5.1 Deteksi gerakan**

Deteksi gerakan secara sederhana dapat dilakukan dengan mencari beda antara 2 buah citra yang berurutan pada hasil pencitraan menggunakan kamera *video digital*. Untuk mengetahui beda antara 2 buah citra tersebut maka operasi yang digunakan adalah pengurangan. Operasi pengurangan pada bagian yang tidak bergerak dalam citra akan menghasilkan nilai *RGB* (*red,green,blue*) per piksel sama dengan nol, sedangkan bagian yang bergerak dalam citra memberikan nilai *RGB* (*red,green,blue*) per piksel tidak sama dengan nol. Dengan mengevaluasi selisih nilai *RGB* per piksel, dapat diketahui apakah pada citra terdapat obyek yang bergerak.

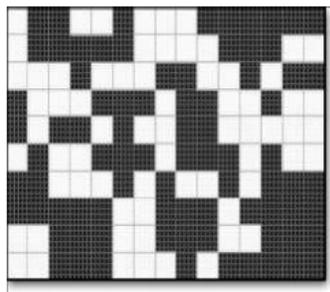
Nilai *RGB* per piksel pada koordinat *x,y* citra pertama dikurangkan dengan nilai piksel koordinat *x,y* citra kedua. Seperti dijelaskan di atas jika hasil pengurangan bernilai nol maka dinyatakan tidak ada gerakan, begitu pula sebaliknya. Untuk mengantisipasi nilai kurang dari nol maka hasil dari pengurangan ditambahkan suatu konstanta.

Obyek yang ditangkap dibuat bercahaya untuk mempermudah pendeteksian gerakan. Dengan obyek yang bercahaya perbedaan citra dari sisi nilai *RGB* per piksel akan lebih jelas dan pendeteksian gerakan jadi lebih mudah.

### 2.1.6 *Bitmap*

Citra *bitmap* adalah susunan bit-bit warna untuk tiap pixel yang membentuk pola tertentu. Pola-pola warna ini menyajikan informasi yang dapat dipahami sesuai dengan persepsi indera penglihatan manusia. Format file ini merupakan format grafis yang fleksibel untuk platform *Windows* sehingga dapat dibaca oleh program grafis manapun. Format ini mampu menyimpan informasi dengan kualitas tingkat 1 bit samapi 24 bit. (<http://slametriyanto.web.id>). Sedangkan pengertian *Bitmap* yang lain adalah sekumpulan titik-titik yang disebut pixel. Pixel menyimpan informasi tentang warna. Titik - titik ini saling berderet berhimpit membentuk baris dan kolom sehingga membentuk gambar yang utuh. Sebuah bitmap dapat mempunyai resolusi dan kedalaman bit yang berbeda. ( Cybulski,2004 ).

Citra *bitmap* didefinisikan sebagai fungsi  $f(x,y)$  dengan  $x$  dan  $y$  adalah koordinat bidang. Besaran  $f$  untuk tiap koordinat  $(x,y)$  disebut intensitas atau derajat keabuan citra pada titik tersebut.



(a) *bitmap* 15×10 *pixel*

1	0	0	1	1	0	1	1	1	0	0	0	0	0	0	0
1	0	0	0	0	0	1	1	1	1	0	0	0	1	1	
1	1	1	0	1	1	1	0	0	1	1	0	1	0	0	
0	1	1	1	0	0	0	1	0	0	1	1	0	1	1	
0	1	0	0	1	0	1	1	0	0	1	1	1	1	1	
1	0	1	1	0	0	0	1	0	0	0	1	0	1	1	
0	0	1	1	1	0	1	0	1	1	0	1	0	0	0	
0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	
1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	
1	1	0	0	0	1	1	1	0	1	0	0	0	0	0	

(b) Matriks *bitmap*

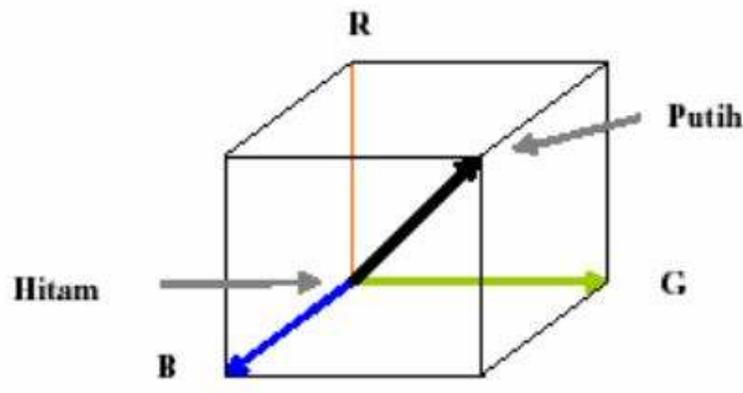
Gambar 2.3 *Bitmap* dengan nilai matriksnya

Dari definisi di atas yang diperjelas oleh gambar 1.1, bitmap dimodelkan dalam bentuk matriks. Nilai pixel atau entri-entri dari matriks ini mewakili warna yang ditampilkan dimana ordo matriks merupakan dimensi panjang dan lebar dari *bitmap*.

Nilai-nilai warna ditentukan berdasarkan intensitas cahaya yang masuk. Dalam komputer, derajat intensitas cahaya diwakili oleh bilangan cacah. Nilai 0 menerangkan tidak adanya cahaya sedangkan nilai yang lain menerangkan adanya cahaya dengan intensitas tertentu. Nilai-nilai ini bisa didapatkan melalui fungsi-fungsi yang disediakan oleh bahasa pemrograman berdasarkan input berupa lokasi entri-entri matriks yang hendak dicari.

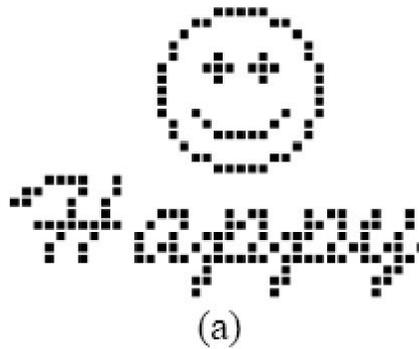
#### 2.1.6.1 Pixel

*Pixel* adalah bagian terkecil dari sebuah gambar. *Pixel* kependekandari *Picture Element*. *Pixel* menggambarkan posisi koordinat dan mempunyai intensitas yang dapat dinyatakan dengan bilangan. Intensitas ini menunjukkan warna gambar, dalam bentuk RGB (Red, Green, Blue).



Gambar 2.4 Koordinat RGB

Dalam gambar hitam putih, setiap pixel dapat direpresentasikan dalam satu bit, berisi 1 bila warnanya hitam dan berisi 0 bila warnanya putih. Ketika komputer akan menggambar mulailah komputer mencari angka-angka yang mengartikan informasi dari gambar. Setiap komputer menemukan angka 0 komputer akan menggambar warna putih, ketika menemukan angka 1 komputer akan menggambar sebuah pixel dengan warna hitam.



```

000000000000000000000011110000000000000000
0000000000000000000001100001100000000000000
000000000000000000000100000000010000000000000
00000000000000000000010000000000010000000000000
00000000000000000000010001000100100000000000000
000000000000000000000100011101110010000000000000
000000000000000000000100001000100010000000000000
000000000000000000000100000000000010000000000000
000000000000000000000100000000000001000000000000
000000000000000000000100100000010010000000000000
000000000000000000000100011110001000000000000000
000000000000000000000100000000010000000000000000
0000000000000000000001100001100000000000000000000
0000000000000000000001111000000000000000000000000
000111100100000000000000000000000000000000000000000
01100010010000000000000000000000000000000000000000000
11000100100000000000000000000000000000000000000000000
0000100100001110001011000101100100100100
00111111110010010001101000110101100100100
00001001000100100111001011100101001001000
0001001000011011011111011111101101101100
000000000000000000000100000010000000011000
0000000000000000000001100000110000000110000
0000000000000000000001000000100000000100000

```

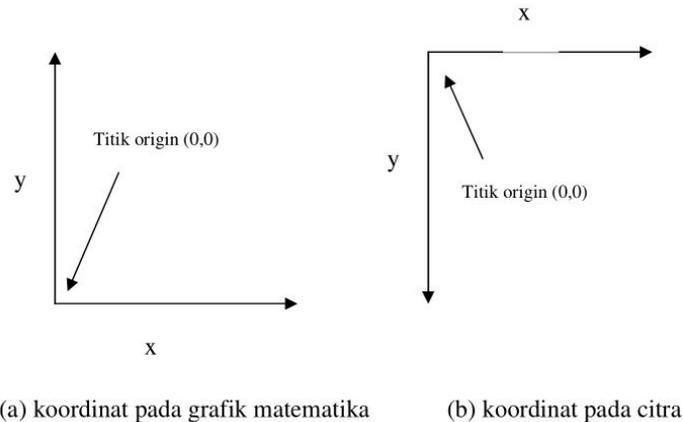
(b)

Gambar 2.5 *Bitmap* hitam putih dan representasi biner

(a) *Bitmap* hitam putih

(b) Representasi biner

Menurut Usman Ahmad (2005:14) sebuah *pixel* adalah sampel dari pemandangan yang mengandung intensitas citra yang dinyatakan dalam bilangan bulat. Sebuah citra adalah kumpulan *pixel-pixel* yang disusun dalam larik dua dimensi. Indeks baris dan kolom (x,y) dari sebuah pixel dinyatakan dalam bilangan bulat. *Pixel* (0,0) terletak pada sudut kiri atas pada citra, indeks x bergerak ke kanan dan indeks y bergerak ke bawah. Konvensi ini dipakai merujuk pada cara penulisan larik yang digunakan dalam pemrograman komputer. Letak titik origin pada koordinat grafik citra dan koordinat pada grafik matematika terdapat perbedaan. Hal yang berlawanan untuk arah vertikal berlaku pada kenyataan dan juga pada sistem grafik dalam matematika yang sudah lebih dulu dikenal. Gambar berikut memperlihatkan perbedaan kedua sistem ini.



Gambar 2.6 Perbedaan letak titik origin pada koordinat grafik dan pada citra

### 2.1.6.2 Kedalaman Bit Dalam Gambar

Kedalaman bit dalam gambar adalah jumlah bit yang digunakan untuk menyimpan informasi tentang sebuah pixel dalam gambar. Semakin tinggi kedalamannya, semakin banyak warna yang dapat dihasilkan dalam sebuah gambar. Contohnya pada kedalaman bit terendah ,1 bit sebuah gambar hanya dapat menghasilkan 2 warna, hitam dan putih. Ini karena hanya ada dua kombinasi angka dalam satu buah bit, yaitu 0 dan 1. Empat bit warna dapat menghasilkan 16 Warna karena mereka ada 16 kombinasi yang berbeda dalam empat bit, yaitu : 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111. Pada warna Titik dengan kedalaman 8 bit, kita dapatkan 256 warna yang dapat dihasilkan.

Pada kedalaman 24 bit, tiap pixel dapat dinyatakan dengan :

- bit 0 sampai dengan 7 untuk warna merah.
- bit 7 sampai dengan 15 untuk warna hijau
- bit 16 sampai dengan 24 untuk warna biru

Kemungkinan kombinasi warna yang ada adalah =  $256^3 + 256^2 + 256^1 = 16.843.008$ , dimana nilai 0 menyatakan warna hitam sedangkan nilai 16 843 008 menyatakan warna putih.

## 2.2 Perangkat Keras (Hardware)

### 2.2.1 Definisi Perangkat Keras

Pengertian dari perangkat keras adalah peralatan yang secara fisik terlihat dan bisa dijamah, perangkat keras terdiri dari seperangkat computer pribadi, I/O device, dan memori.

### 2.2.2 *Raspberry Pi*

*Raspberry Pi*, sering juga disingkat dengan nama *Raspi*, adalah komputer papan tunggal (*Single Board Circuit /SBC*) yang memiliki ukuran sebesar kartu kredit. *Raspberry Pi* bisa digunakan untuk berbagai keperluan, seperti *spreadsheet*, *game*, bahkan bisa digunakan sebagai *media player* karena kemampuannya dalam memutar video high definition. *Raspberry Pi* dikembangkan oleh yayasan nirlaba, *Raspberry Pi Foundation* yang digawangi sejumlah *developer* dan ahli komputer dari Universitas Cambridge, Inggris.

Ide dibalik komputer mungil ini diawali dari keinginan untuk mencetak generasi baru *programmer*, pada 2006 lalu. Seperti disebutkan dalam situs resmi *Raspberry Pi Foundation*, waktu itu Eben Upton, Rob Mullins, Jack Lang, dan Alan Mycroft, dari Laboratorium Komputer Universitas *Cambridge* memiliki kekhawatiran melihat kian turunnya keahlian dan jumlah siswa yang hendak belajar ilmu komputer. Sesuatu telah mengubah cara bagaimana anak-anak berinteraksi dengan komputer. Beberapa masalah telah diketahui seperti kolonisasi dari kurikulum *ICT* dengan pelajaran seperti menggunakan *Word*, *Excel*, atau menulis halaman *web*, yang merupakan akhir dari boomingnya *dot-com*, dan kemunculan dari *Personal Computer* dan konsol *game* untuk menggantikan *Amigas*, *BBC Micros*, *Spectrum ZX* dan mesin *Commodore 64* yang digunakan oleh generasi sebelumnya untuk belajar memprogram. Parahnya lagi, generasi sekarang mungkin sudah melupakan bagaimana komputer bekerja, mereka lebih asik dengan

*smartphone* dan menghabiskan waktu mereka dengan *selfie* dan *hangout* di media sosial.

Tidak banyak orang dengan kelompok kecil dapat memecahkan masalah di atas berkaitan dengan kurikulum sekolah yang tidak memadai, atau pembengkakan anggaran. Namun menurut founder dari *Raspberry Pi Foundation* kelompok kecilnya mencoba untuk melakukan sesuatu dengan situasi tersebut dimana komputer telah menjadi sangat mahal dan eksperimen pemrograman sembunyi-sembunyi pada komputer mahal tersebut mesti dilarang oleh orang tua sehingga sulit untuk menemukan platform yang seperti komputer rumahan tua, yang bisa booting ke lingkungan pemrograman. Dari 2006 hingga 2008, Eben dkk mendesain beberapa versi dari apa yang dikenal orang sekarang dengan *Raspberry Pi*, seperti terlihat pada gambar berikut :



Gambar 2.7 Purwarupa Raspberry Pi dari Veroboard tahun 2006

Mereka lantas mendirikan yayasan *Raspberry Pi* bersama dengan Pete Lomas dan David Braben pada 2009. Tiga tahun kemudian, *Raspberry Pi*

*Model B* memasuki produksi massal. Dalam peluncuran pertamanya pada akhir Februari 2012 dalam beberapa jam saja sudah terjual 100.000 unit.

Sejak dirilis pada tahun 2012 *Raspberry Pi* telah memiliki lima model, empat diantaranya digunakan oleh orang umum namun satu untuk tujuan pengembangan. Berikut adalah ulasan dari model-model Raspi yang ada:

### 1. Raspberry Pi Model A



Gambar 2.8 *Raspberry Pi Model A*

Ini adalah perangkat yang paling dasar, dengan satu buah USB port dan 256MB *SDRAM*. Port pada board-nya terdiri dari:

- Full size SD card
- HDMI output port
- Composite video output

- 1 USB port
- 26 pin header GPIO, I2C dll
- 3.5mm audio jack
- 1 Camera interface port (CSI-2)
- 1 LCD display interface port (DSI)
- 1 mircoUSB power connector untuk menyalakan perangkat

Karena tidak terdapat ethernet atau USB port ekstra pada model ini, maka model ini menggunakan konsumsi daya yang lebih rendah dari model B/B+.

## 2. Raspberry Pi Model A+



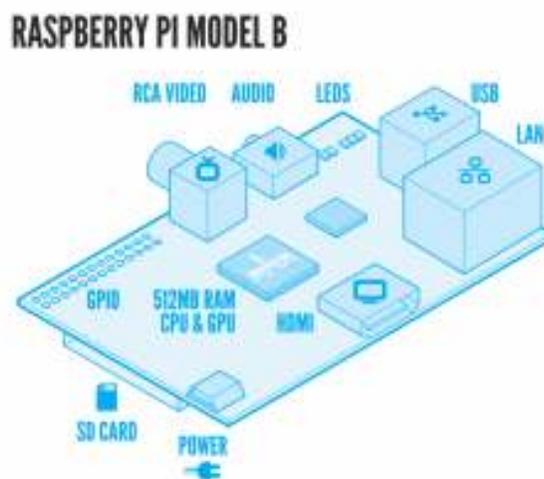
Gambar 2.9 *Raspberry Pi Model A+*

Dirilis pada November 2014, ini adalah varian 'plus' dari model A. Memiliki 40 GPIO pin, satu USB board, tanpa ethernet dan 256MB SDRAM. Juga memiliki form factor yang lebih kecil dari model yang lain dengan panjang 65mm.

### 3. Raspberry Pi Model B



Gambar 2.10 Raspberry Pi Model B



Gambar 2.11 Skema Raspberry Pi Model B

Hingga Juli 2014, ini adalah perangkat yang paling banyak di pasaran. Memiliki dua port USB, dan RAM sebesar 512MB SDRAM. Sebagai catatan, Model B dalam revisi pertama (Raspberry Pi Model B Rev. 1) hanya memiliki RAM sebesar 256MB. Port tambahan yang disertakan dari pendahulunya model A adalah satu buah port ethernet dan satu buah port USB sehingga total memiliki dua buah port USB.

#### 4. Raspberry Pi Model B+



Gambar 2.12 *Raspberry Pi Model B+*

Dirilis pada Juli 2014, model B+ adalah pembaharuan revisi dari model B. Terdapat penambahan jumlah USB port menjadi 4 dan jumlah pin header GPIO menjadi 40. Sebagai tambahan, model ini memiliki sirkuit power supply yang lebih baik yang memungkinkan perangkat USB yang memerlukan daya besar untuk digunakan pada Raspberry dengan mode hot-plugged. Composite video connector yang menonjol besar telah dihilangkan dan digantikan dengan jack audio/video 3.5mm. SD Card full size juga diganti dengan versi yang lebih robust yaitu slot microSD.

Berikut adalah daftar rinci beberapa peningkatan model B+ dari model B:

- Monitor arus pada port USB yang berarti model B+ sekarang telah mendukung hot-plugging.
- Pembatas arus pada sumber daya 5V untuk HDMI yang berarti semua VGA konverter yang menggunakan daya dari kabel HDMI bisa digunakan.
- 14 pin GPIO tambahan.
- Dukungan EEPROM readout untuk papan ekspansi baru HAT.
- Kapasitas drive yang lebih tinggi untuk audio out analog, dari regulator terpisah, yang berarti kualitas audio DAC yang lebih baik.
- Tidak ada lagi masalah dengan backpowering (daya lain masuk dari USB port bukan dari port power), karena pembatas arus USB yang juga mencegah aliran balik, bersama dengan "dioda power ideal".
- Composite video out dipindahkan ke jack 3.5mm.
- Konektor sekarang dipindahkan ke dua sisi papan ketimbang menggunakan empat sisi papan.
- 4 lobang pasang yang ditaruh dengan posisi segi panjang sehingga memudahkan untuk pemasangan pada casing dll.

## 5. Model Compute Modul



Gambar 2.13 Model Compute Modul

*Compute Modul* diperuntukan bagi penggunaan industri, merupakan versi potongan yang hanya menyertakan *chip BCM2835*, 512MB *SDRAM* dan 4GB *eMMC* flash memori, dalam form factor berukuran kecil. Modul ini dihubungkan dengan papan dasar menggunakan konektor 200 pin *DDR2 SODIMM* yang telah dimodifikasi dan bukan merupakan perangkat yang kompatibel dengan *SODIMM*, namun hanya menggunakan konektor yang sama dengan *SODIMM*. Semua fitur dari *BCM2835* dipaparkan melalui konektor *SODIMM*, termasuk dua buah kamera dan *LCD port*, sementara model A dan B hanya memiliki satu.

*Compute modul* diharapkan dapat digunakan oleh perusahaan yang berharap untuk dapat mempercepat proses pengembangan dari produk baru, berarti hanya papan dasarnya saja yang perlu dibuat, dengan periperal yang

sesuai, dan dengan *Compute modul* yang menyediakan *CPU*, memori dan penyimpanan dengan perangkat lunak yang teruji dan terpercaya.

Tabel 2.1 Tabel Perbandingan Jenis-jenis *Raspberry PI*

<b>Feature</b>	<b>Model A</b>	<b>Model B</b>	<b>Model A+</b>	<b>Model B+</b>	<b>Compute Module</b>
BRCM2835 SoC	Yes	Yes	Yes	Yes	Yes
Standard SoC Speed	700Mhz	700Mhz	700Mhz	700Mhz	700Mhz
RAM	256MB	512MB*	256MB	512MB	512MB
Storage	Full SD	Full SD	Micro SD	Micro SD	4GB eMMC
Ethernet 10/100	No	Yes	No	Yes	No
HDMI output port	Yes	Yes	Yes	Yes	Yes
Composite video output	Yes	Yes	On 3.5mm jack	On 3.5mm jack	Yes
Number of USB2.0 ports	1	2	1	4	1
Expansion header	26	26	40	40	N/A
Number of available GPIO	17	17	26	26	48
3.5mm audio jack	Yes	Yes	Audio / Video	Audio / Video	N/A
Number of camera interface ports	1	1	1	1	2

Feature	Model A	Model B	Model A+	Model B+	Compute Module
(CSI-2)					
Number of LCD display interface ports (DSI)	1	1	1	1	2
Power (bare, approx, 5v)	300mA, 1.5W	700mA, 3.5W	-	650mA, 3W	-
Size	85 x 56 x 15mm	85 x 56 x 17mm	65 x 56 x 12mm	85 x 56 x 17mm	62 x 30 x 3mm

### 2.2.3 Webcam

Kamera Video Konferensi yang juga dikenal dengan Kamera Web (*Webcam*) adalah kamera yang dirancang penggunaannya pada *World Wide Web*. Kamera Web sekarang harganya turun dengan cepat yang menjadikannya sebagai suatu produk pasar massa. Pada kamera *web Xcam2* yang bentuknya kecil, tetapi mempunyai kemampuan kamera video yang dapat diletakkan di mana saja dan dapat mentransmisikan video berwarna ke TV, VCR atau pada *PC (Personal Computer)*.

*Webcam (Web Camera)* secara sederhana terdiri atas *digital* kamera yang tersambung dengan komputer. Webcam terintegrasi dengan komputer dengan melalui *port USB* (sebelumnya kamera terhubung ke komputer melalui port parallel). Cara kerja *webcam* tidak jauh berbeda dengan cara kerja kamera tradisional yang berbasis film, yaitu memilih obyek yang akan direkam dengan menggunakan jendela pengintai. Kemudian bayangan objek

tersebut difokuskan oleh lensa ke sebuah peralatan peka cahaya yang dapat berupa sensor *CCD (Charge Coupled Device)* atau *CMOS (Complementary Metal-Oxide Semiconductor)*. Setiap elemen sensor mengkonversi cahaya ke tegangan listrik yang sesuai dengan kecerahan (brightness) yang diteruskan ke sebuah *ADC (Analog to Digital Converter)* yang menerjemahkan fluktuasi tegangan dari *CCD* ke dalam kode biner. Keluaran dari *ADC* dikirimkan ke sebuah *DSP (Digital Signal Processor)* yang menyesuaikan kontras dan detail dari gambar, serta mengkompres gambar sebelum dikirim ke media penyimpanan. Semakin terang cahaya yang dihasilkan sensor *CCD*, semakin tinggi tegangan dan semakin besar resolusi piksel yang dihasilkan komputer menyebabkan semakin besar detail yang ditangkap oleh kamera.

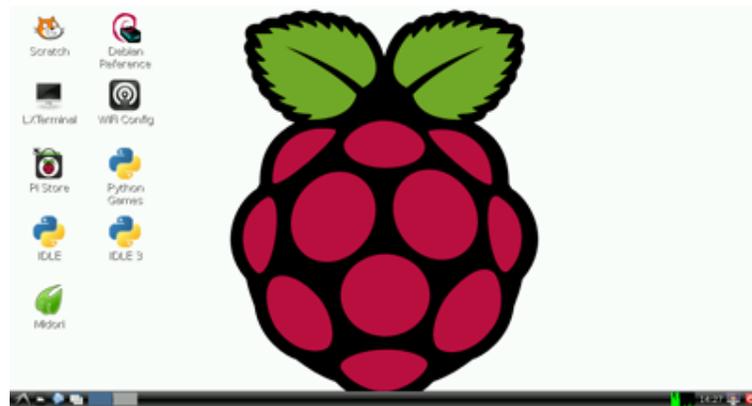
## **2.3 Perangkat Lunak (Software)**

### **2.3.1 Definisi Perangkat Lunak**

Pengembangan perangkat lunak diarahkan pada realisasi sistem aplikasi yang mampu menunjang proses transaksi ekonomi yang cepat dan aman, serta pengambilan keputusan yang benar dan cepat. Harga yang terjangkau dan daya saing pada tingkat internasional merupakan salah satu kriteria yang dipersyaratkan, khususnya mendukung kebijakan substitusi impor. Perangkat lunak sistem operasi dengan kehandalan tinggi dan kebutuhan sumber daya memori maupun prosesor yang minimal serta fleksibel terhadap perangkat keras maupun program aplikasi yang baru, merupakan prioritas yang harus dikembangkan. Program aplikasi juga perlu dikembangkan, terutama yang terkait dengan sektor perekonomian, industri,

pendidikan, maupun pemerintahan. Dalam mempercepat pengembangan dan pendayagunaan perangkat lunak, perlu pula ditinjau implementasi konsep open source. Penerapan konsep open source ini diharapkan mampu menggalakkan industri perangkat lunak dengan partisipasi seluruh lapisan masyarakat tanpa melakukan pelanggaran hak cipta.

### 2.3.2 Sistem Operasi pada *Raspberry Pi Raspbian Wheezy*



Gambar 2.14 Antarmuka Raspbian

*Raspbian* adalah sistem operasi gratis yang didasarkan pada *Debian* dan dioptimisasi untuk perangkat keras *Raspberry Pi*. Dan menyediakan lebih dari 35.000 paket program bundel perangkat lunak yang telah di pra-compile dalam format yang bagus agar mudah dipasang pada *Raspi* Anda.

Saat ini *Raspbian* masih dalam pengembangan aktif dengan perhatian pada meningkatkan stabilitas dan performa dari sebanyak-banyaknya paket *Debian*. *Raspbian* tidaklah berafiliasi dengan *Raspberry Pi Foundation*. *Raspbian* diciptakan oleh tim kecil yang berdedikasi yang merupakan

penggemar dari perangkat keras *Raspberry Pi*, tujuan pendidikan dari *Raspberry Pi Foundation* dan, tentunya juga dari *Debian Project*. *Raspbian* merupakan sistem operasi umum yang paling banyak orang gunakan pada *Raspberry Pi*.

### **2.3.3 *OpenCV (Open Source Computer Vision Library)***

*OpenCV (Open Source Computer Vision Library)* adalah sebuah pustaka perangkat lunak yang ditujukan untuk pengolahan citra dinamis secara *real-time*, yang dibuat oleh *Intel*, dan sekarang didukung oleh Willow Garage dan Itseez. Program ini bebas dan berada dalam naungan sumber terbuka dari lisensi BSD. Pustaka ini merupakan pustaka lintas platform. Program ini didedikasikan sebagian besar untuk pengolahan citra secara *real-time*. Jika pustaka ini menemukan pustaka *Integrated Performance Primitives* dari intel dalam sistem komputer, maka program ini akan menggunakan rutin ini untuk mempercepat proses kerja program ini secara otomatis.

*OpenCV* pertama kali diluncurkan secara resmi pada tahun 1999 oleh *Inter Research* sebagai lanjutan dari bagian proyek bertajuk aplikasi intensif berbasis *CPU*, *real-time ray tracing* dan tembok penampil 3D. Para kontributor utama dalam proyek ini termasuk mereka yang berkecimpung dalam bidang optimasi di Intel Russia, dan juga Tim Pustaka Performansi Intel. Pada awalnya, tujuan utama dari proyek *OpenCV* ini dideskripsikan sebagai berikut :

- Penelitian penginderaan citra lanjutan tidak hanya melalui kode program terbuka, tetapi juga kode yang telah teroptimasi untuk infrastruktur penginderaan citra.
- Menyebarkan ilmu penginderaan citra dengan menyediakan infrastruktur bersama di mana para pengembang dapat menggunakannya secara bersama-sama, sehingga kode akan tampak lebih mudah dibaca dan ditransfer.
- Membuat aplikasi komersial berbasis penginderaan citra, di mana kode yang telah teroptimasi tersedia secara bebas dengan lisensi yang tersedia secara bebas yang tidak mensyaratkan program itu harus terbuka atau gratis.

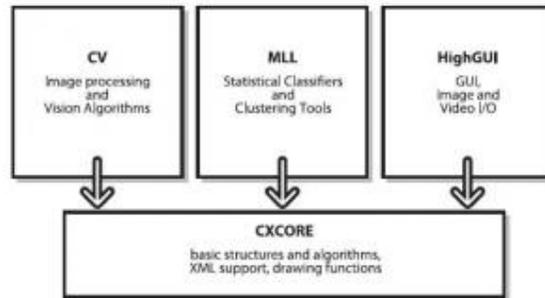
*OpenCV* juga dikenal sebagai sebuah *API (Application Programming Interface) Library* yang sudah sangat familiar pada Pengolahan Citra *Computer Vision*. *Computer Vision* itu sendiri adalah salah satu cabang dari Bidang Ilmu Pengolahan Citra (*Image Processing*) yang memungkinkan komputer dapat melihat seperti manusia. Dengan *vision* tersebut komputer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu objek. Beberapa pengimplementasian dari *Computer Vision* adalah *Face Recognition, Face Detection, Face/Object Tracking, Road Tracking*, dll. *OpenCV* adalah library *Open Source* untuk *Computer Vision* untuk *C/C++*, *OpenCV* didesain untuk aplikasi *real-time*, memiliki fungsi-fungsi akuisisi yang baik untuk *image/video*.

*OpenCV* sendiri terdiri dari 5 *library*, yaitu :

- *CV* : untuk algoritma *Image processing* dan *Vision*.

- *ML* : untuk *machine learning library*
- *Highgui* : untuk *GUI, Image dan Video I/O*.
- *CXCORE* : untuk dukungan struktur data dan aljabar linier  
*support XML* dan fungsi-fungsi grafis
- *CvAux* : fungsi tambahan (eksperimental)

Struktur dan konten *OpenCV*:

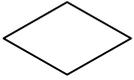


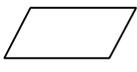
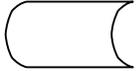
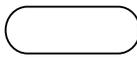
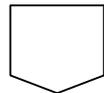
Gambar 2.15 Struktur dan konten *OpenCV*

#### 2.3.4 Flowchart

*Flowchart* adalah gambaran urutan kerja sistem dengan menggunakan simbol - simbol tertentu yang digambarkan dalam bentuk tabel yang terpecah atas jalur - jalur berdasarkan *entity*.

Tabel 2.2 Simbol - simbol *Flowchart*

Simbol	Nama	Keterangan
	Process	Digunakan untuk mendefinisikan mekanisme perekaman, proses, dan pelaporan.
	Decision	Digunakan untuk mendefinisikan adanya 2 pilihan, proses percabangan, penyeleksian data.

	Input / Output data	Digunakan untuk mendefinisikan data yang diinputkan dan data yang dioutputkan ke dalam sistem.
	Manual Input	Digunakan untuk mendefinisikan fungsi pemasukan data atau <i>key in</i>
	Stored Data	Digunakan untuk mendefinisikan <i>file</i> referensi, <i>file</i> master ataupun <i>file</i> temporer yang digunakan dalam proses.
	Display	Digunakan untuk endefinisikan keluaran dalam bentuk layar ( <i>screen</i> ).
	Manual Operation	Suatu proses yang tidak dilakukan oleh komputer (manual).
	Terminator	Permulaan atau akhir sebuah alur program.
	Connector	Digunakan untuk menghubungkan alir data satu proses ke proses lainnya dalam halaman / lembar yang sama.
	Offline Connector	Digunakan untuk menghubungkan alir data yang satu dengan yang lain dalam halaman berbeda.

### 2.3.5 Bahasa Pemrograman *python*

*Python* adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. *Python* diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif.

*Python* mendukung multi paradigma pemrograman, utamanya; namun tidak dibatasi; pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Salah satu fitur yang tersedia pada *python* adalah sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis. Seperti halnya pada bahasa pemrograman dinamis lainnya, *python* umumnya digunakan sebagai bahasa skrip meski pada praktiknya penggunaan bahasa ini lebih luas mencakup konteks pemanfaatan yang umumnya tidak dilakukan dengan menggunakan bahasa skrip. *Python* dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi.

Saat ini kode *python* dapat dijalankan di berbagai platform sistem operasi, beberapa diantaranya adalah:

- *Linux/Unix*
- *Windows*
- *Mac OS X*
- *Java Virtual Machine*
- *OS/2*
- *Amiga*
- *Palm*
- *Symbian* (untuk produk-produk *Nokia*)

### a. Sejarah

*Python* didistribusikan dengan beberapa lisensi yang berbeda dari beberapa versi. Lihat sejarahnya di *Python Copyright*. Namun pada prinsipnya *Python* dapat diperoleh dan dipergunakan secara bebas, bahkan untuk kepentingan komersial. Lisensi *Python* tidak bertentangan baik menurut definisi *Open Source* maupun *General Public License (GPL)*

*Python* dikembangkan oleh Guido van Rossum pada tahun 1990 di *CWI*, Amsterdam sebagai kelanjutan dari bahasa pemrograman *ABC*. Versi terakhir yang dikeluarkan *CWI* adalah 1.2.

Tahun 1995, Guido pindah ke *CNRI* sambil terus melanjutkan pengembangan *Python*. Versi terakhir yang dikeluarkan adalah 1.6. Tahun 2000, Guido dan para pengembang inti *Python* pindah ke *BeOpen.com* yang merupakan sebuah perusahaan komersial dan membentuk *BeOpen PythonLabs*. *Python 2.0* dikeluarkan oleh *BeOpen*. Setelah mengeluarkan *Python 2.0*, Guido dan beberapa anggota tim *PythonLabs* pindah ke *DigitalCreations*.

Saat ini pengembangan *Python* terus dilakukan oleh sekumpulan pemrogram yang dikoordinir Guido dan *Python Software Foundation*. *Python Software Foundation* adalah sebuah organisasi non-profit yang dibentuk sebagai pemegang hak cipta intelektual *Python* sejak versi 2.1 dan dengan demikian mencegah *Python* dimiliki oleh perusahaan komersial. Saat ini distribusi *Python* sudah mencapai versi 2.6.1 dan versi 3.0

Nama *Python* dipilih oleh Guido sebagai nama bahasa ciptaannya karena kecintaan Guido pada acara televisi *Monty Python's Flying Circus*.

Oleh karena itu seringkali ungkapan-ungkapan khas dari acara tersebut seringkali muncul dalam korespondensi antar pengguna *Python*.

#### **b. Fitur**

Beberapa fitur yang dimiliki Python adalah:

- memiliki kepustakaan yang luas; dalam distribusi *Python* telah disediakan modul-modul 'siap pakai' untuk berbagai keperluan.
- memiliki tata bahasa yang jernih dan mudah dipelajari.
- memiliki aturan *layout* kode sumber yang memudahkan pengecekan, pembacaan kembali dan penulisan ulang kode sumber.
- berorientasi obyek.
- memiliki sistem pengelolaan memori otomatis (*garbage collection*, seperti *java*)
- modular, mudah dikembangkan dengan menciptakan modul-modul baru; modul-modul tersebut dapat dibangun dengan bahasa *Python* maupun *C/C++*.
- memiliki fasilitas pengumpulan sampah otomatis, seperti halnya pada bahasa pemrograman *Java*, *python* memiliki fasilitas pengaturan penggunaan ingatan komputer sehingga para pemrogram tidak perlu melakukan pengaturan ingatan komputer secara langsung.
- memiliki banyak fasilitas pendukung sehingga mudah dalam pengoperasiannya.

**c. Masukan / Pengeluaran**

Contoh masukan :

```
nama = raw_input("Masukkan nama Anda: ")
```

Contoh keluaran :

```
print "Halo", nama, ":"
```