

## **BAB II**

### **KAJIAN PUSTAKA**

Bab ini memuat dua hal yaitu kajian pustaka dari beberapa buku tentang masalah yang diteliti beserta dimensi-dimensinya. Serta hasil penelitian terdahulu yang telah dilakukan oleh para peneliti sebelumnya tentang masalah yang sama. Selain itu bab ini juga berisi landasan teori mengenai komponen-komponen yang diperlukan sebagai dasar dalam penelitian yang dilakukan.

#### **2.1 Tinjauan Pustaka**

Untuk mendukung penulisan tugas akhir ini digunakan beberapa tinjauan pustaka yang diambil dari penelitian terdahulu dalam beberapa jurnal penelitian yang berkaitan dengan penelitian yang dilakukan oleh penulis.

##### **2.1.1 Rancang Bangun Alat Pemilalah Kualitas Kematangan Buah Naga Menggunakan Teknik Image Processing Dengan Metode Image Segmentation HSV**

Penelitian ini dilakukan oleh Sustiono, Wahyu Setyo Pambudi dari jurusan teknik elektro Universitas Internasional Batam. Dalam penelitian ini bertujuan untuk pemeriksaan kualitas kematangan produk buah naga yang telah dipanen. Dengan demikian pemeriksaan dan pengendalian kematangan buah naga menjadi masalah yang sangat penting dalam pengelolaan produk buah naga. Mengingat kualitas kematangan buah dirasakan langsung oleh pelanggan sebagai indikator kualitas yang paling utama.

Penentuan kualitas kematangan buah naga akan didasarkan pada warna kulit buah naga yang terlihat. Pemeriksaan kualitas kematangan buah naga menggunakan teknologi computer vision dengan metode segmentasi warna HSV. Computer vision adalah teknik pemeriksaan secara lebih cepat, konsisten dan objectif, dan secara luas dipakai untuk berbagai bidang industri. Penggunaan computer vision juga dipakai untuk kualitas pengawasan (quality inspection) dan menentukan rangking (grading) pada buah-buahan dan sayuran.

Dalam rancang bangun system conveyor otomatis pemilahan kualitas kematangan buah naga akan diterapkan segmentasi warna dengan metode deteksi warna HSV. Ruang lingkup warna HSV terdiri dari 3 elemen yaitu hue mewakili warna, saturation mewakili tingkat dominasi warna dan value mewakili tingkat kecerahan. Dengan demikian metode ini cenderung mendeteksi warna dan tingkat dominasi serta kecerahannya.

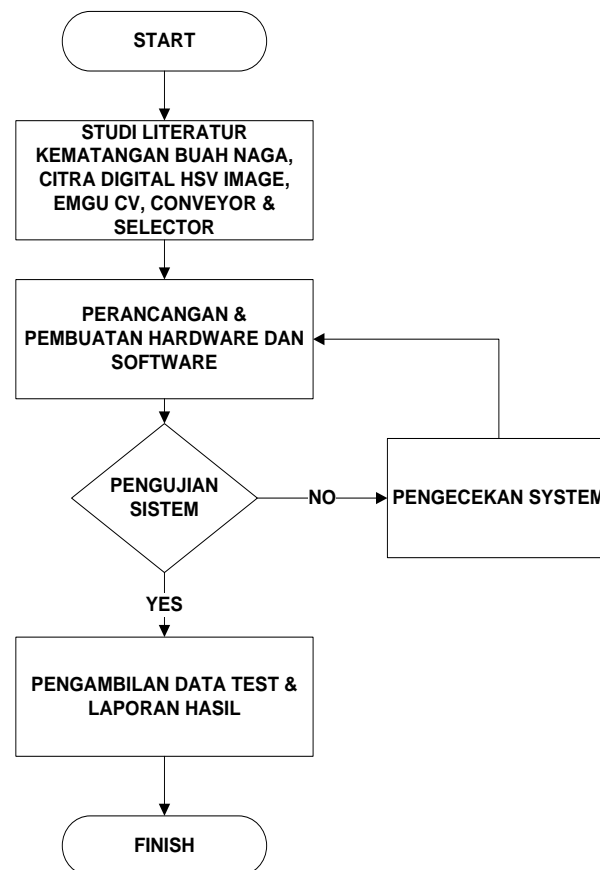
Berdasarkan beberapa referensi metode Image Segmentation HSV sangat sesuai digunakan untuk pemilahan benda yang didasarkan dengan warna benda tersebut. Sistem ini mengaplikasikan teknologi *Computer Vision* untuk mendeteksi dan memilah kematangan buah naga. Pertama-tama didefinisikan terlebih dahulu kualitas kematangan buah naga yang diinginkan sebagai nilai referensi pada proses pemilahan. Selanjutnya buah naga yang akan dipilah, diletakkan di atas conveyor, conveyor akan membawa buah naga ke posisi tertentu dimana *Computer vision* akan mengenali buah naga yang sudah matang dengan cara mengenali perubahan warna buah naga menggunakan kamera. Kamera akan mendeteksi kematangan buah naga berdasarkan perubahan warna kulit. Setelah proses deteksi oleh kamera, sebuah selector yang diletakkan di akhir conveyor digunakan untuk memilah buah naga menjadi 2 katagori, yaitu matang dan belum matang. Hasil pengujian dari preoses pemilahan buah naga dengan metode Image Segmentation HSV, setelah dilakukan 100 kali percobaan didapat presentase keberhasilan sebesar 86% dan waktu yang diperlukan untuk memilah satu buah naga antara 15 sampai 22 detik.

Langkah-langkah metode penelitian yang dilakukan yaitu mencari literatur ciri-ciri kematang buah naga didasarkan pada warna kulit buah naga yang nantinya akan digunakan sebagai penentuan thershold pada pengolahan citra digital. Mencari literatur teknik pengolahan citra digital untuk segmentasi HSV yang banyak dipakai pada pendeteksi warna pada suatu obyek terutama pada obyek buah-buahan. Serta referensi pengolahan citra digital menggunakan EmguCV. Dilakukan studi literature tentang penggerak conveyor dan selector menggunakan board arduino uno sebagai antar muka pada system conveyor berbasis computer dengan komunikasi melalui kabel USB. Penjelasan perancangan system dapat ditunjukkan dalam gambar berikut.



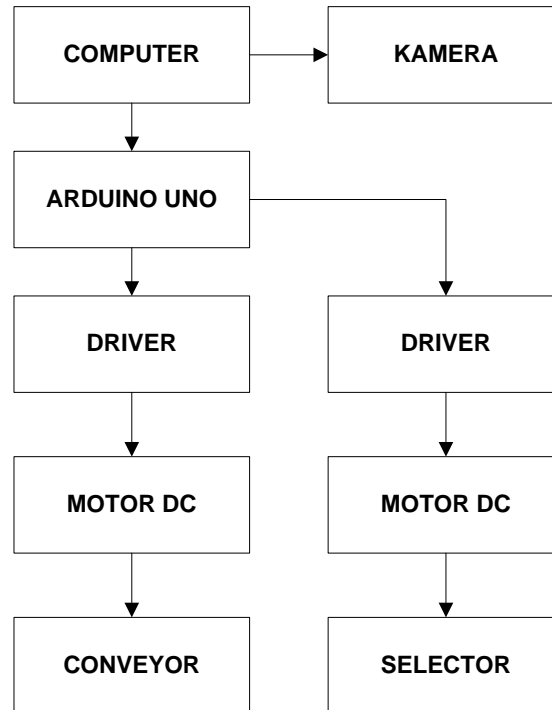
**Gambar 2.1** Perancangan System

Perancangan hardware system conveyor dibuat menggunakan controller board arduino uno sebagai antar muka untuk pengontrol motor conveyor, selector dan limit switch yang dipasang pada selector. Perancangan software dengan menggunakan bahasa pemrograman visual studio C# dengan emguCV dan image segmentation image HSV untuk mendeteksi kematangan buah naga serta penggunaan teknik pengolahan segmentasi gray image digunakan untuk mendeteksi buah pada conveyor, berikut flowchart penelitian:



**Gambar 2.2** Flowchart Penelitian

Dalam perancangan ini akan dibuat blok diagram dari system hardware rancang bangun alat pemilah kematangan buah Naga. Perangkat yang dipakai antara lain computer, webcam, board arduino uno, dua buah driver motor, dua motor DC, conveyor dan selector, berikut blok diagram system hardware.



**Gambar 2.3** Blok Diagram Sistem Hardware

Penjelasan dari blok diagram hardware sebagai berikut. Komputer digunakan sebagai fungsi kontrol dari semua peralatan yang dipakai dan program pada komputer dirancang menggunakan bahasa program Visual studio C#. Kamera webcam digunakan sebagai sensor pendeteksi buah naga pada conveyor dan juga sebagai sensor pendeteksi kematangan buah naga. Kamera dihubungkan dengan komputer menggunakan USB Kabel. Board arduino uno dipakai sebagai antar muka control motor conveyor dan motor selector melalui driver motor pada masing- masing motor. Karena arus dan tegangan keluaran dari arduino uno kecil atau terbatas maka diperlukan driver motor untuk menggerakan motor yang memerlukan arus dan tegangan lebih besar. Driver Motor conveyor terhubung dengan arduino uno melalui output digital PIN 11 dan PIN 12. Sedangkan driver motor selector terhubung dengan board arduino uno melalui output digital PIN 8, PIN9 dan PIN 10. Board Arduino uno juga berfungsi sebagai antar muka dari dua limit switch yang dipasang pada selector disisi kiri dan disisi kanan untuk mendeteksi saat terakhir keberadaan posisi selector. Limit switch terhubung dengan board Arduino uno melalui input digital, PIN 2 dan PIN 3. Computer berkomunikasi dengan board arduino uno menggunakan kabel USB.

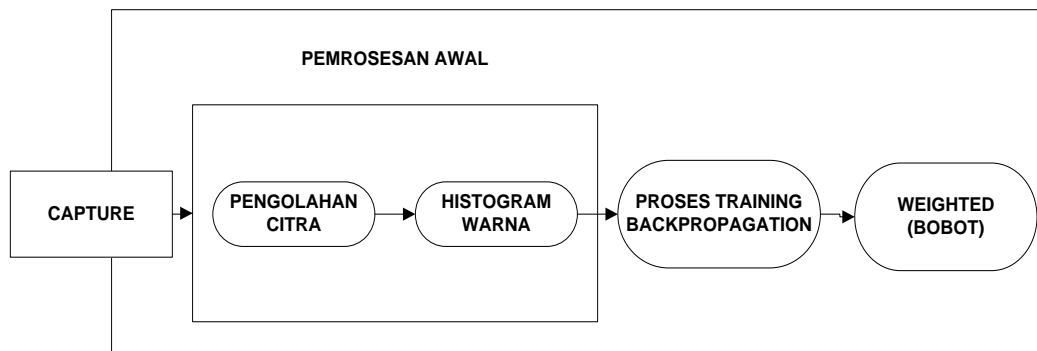
### **2.1.2 Identifikasi Kematangan Buah Tomat Menggunakan Metode *Backpropagation***

Penelitian ini dilakukan oleh Dila Deswari, Hendrick, M.T, Derisma, M.T dari jurusan Sistem Komputer, Fakultas Teknologi Informasi, Universitas Andalas. Dalam penelitian ini bertujuan mengidentifikasi kematangan buah nanas dengan menggunakan webcam sebagai pengolah citra dari gambar yang direkam (*image processing*) karena selama penelitian berlangsung proses yang dilakukan masih secara manual. Kelemahan pengklasifikasian manual sangat dipengaruhi subjektivitas operator sortir sehingga pada kondisi tertentu tidak konsisten proses pengklasifikasiannya. Perkembangan teknologi informasi memungkinkan identifikasi buah berdasarkan ciri warna dengan bantuan computer.

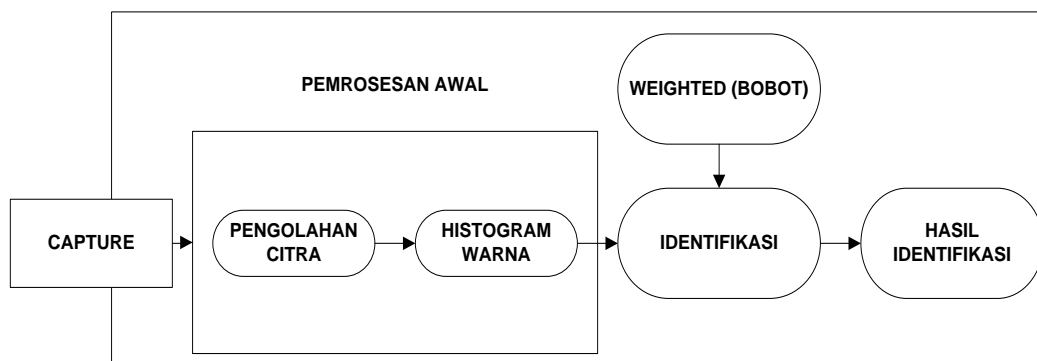
Cara komputasi ini dilakukan dengan pengamatan visual tidak langsung dengan menggunakan kamera sebagai pengolah citra dari gambar yang direkam (*image processing*) untuk kemudian diolah dengan menggunakan perangkat lunak computer. Pada tugas akhir ini penulis melakukan penelitian pada identifikasi tingkat kematangan buah tomat berdasarkan warna. Identifikasi ini menggunakan aplikasi jaringan syaraf tiruan (JST) dengan metode pembelajaran *backpropagation*. Objek yang diamati yaitu buah tomat yang berbeda tingkat kematangannya berdasarkan warna. Informasi citra dari buah tomat yang diamati menggunakan bantuan video kamera atau webcam. Webcam dan software diintegrasikan ke komputer menggunakan jaringan syaraf tiruan agar komputer dapat mengenali informasi citra buah tomat tersebut.

Metodologi yang dikerjakan pada tugas akhir ini ialah studi literature, pra proses meliputi capture buah tomat oleh webcam, citra hasil capture akan mengalami *image processing* yaitu proses histogram warna. Proses identifikasi kematangan buah tomat meliputi arsitektur *backpropagation* menggunakan 3 unit input, 4 node hidden layer dan 3 unit output. Kemudian rancangan pembelajaran (*training*) *backpropagation* Citra – histogram warna – pola – *training* – bobot. Dan rancangan proses identifikasi *backpropagation* Citra – histogram warna –

pola – indentifikasi (dari bobot). Proses perancangan software dapat ditunjukkan dalam gambar berikut ini.



**Gambar 2.4** Proses Pelatihan Data Training



**Gambar 2.5** Proses Identifikasi Data

Secara garis besar proses dapat dikelompokkan menjadi dua bagian yaitu proses pelatihan data (pembelajaran backpropagation) dan identifikasi kematangan buah tomat. Proses pelatihan berguna untuk melatih sistem dengan memasukkan data-data inputan ke dalam sistem Neural Network kemudian data tersebut diolah dengan menggunakan metode backpropagation.

Berdasarkan hasil dari penelitian yang dilakukan dapat diambil beberapa kesimpulan bahwa penelitian ini berhasil membangun perangkat lunak yang mengimplementasikan algoritma backpropagation dalam mengidentifikasi tingkat kematangan buah tomat berdasarkan warna menggunakan webcam dengan tingkat identifikasi 71,76%. Dari hasil identifikasi yang telah dilakukan menghasilkan tiga output identifikasi yaitu tomat masak 85%, muda 45%, dan setengah masak 85%. Identifikasi kematangan buah tomat masak dan setengah masak pada sistem ini lebih akurat dibanding identifikasi tomat muda karena pengaruh pencahayaan.

### **2.1.3 Aplikasi Kematangan Tomat Berdasarkan Warna dengan Metode *Linear Discriminant Analysis (LDA)***

Penelitian ini dilakukan oleh Ghazali Eko K. Subha, Galuh M., M. Burhanuddin, M. Izzun Niam dari Universitas Brawijaya Malang. Dalam penelitian ini, warna menjadi salah satu karakteristik yang menonjol pada buah dan sayur di daerah tropis sebagai inti dalam mengidentifikasi kematangan buah tomat. Identifikasi kematangan tomat dilakukan dengan melakukan ekstraksi warna. Dan dengan menggunakan algoritma LDA (Linear Discriminant Analysis), tomat dikelompokkan dalam dua kelas yang telah ditentukan yaitu kelas matang dan kelas mentah. Hasil penelitian ini diharapkan dapat menggantikan cara mendeteksi kematangan tomat yang dilakukan secara manual.

Buah dan sayur merupakan salah satu makanan yang bisa dikatakan tidak terlepas dari kehidupan manusia. Buah dan sayur terkenal akan kandungan vitamin dan antioksidan yang mampu memperbaiki regenerasi sel, mencegah penyakit tertentu, dan banyak manfaat lainnya yang bisa kita ambil dari buah. Setiap pemilik kebun buah selalu berusaha meningkatkan kualitas perkebuna mereka, baik dari segi kualitas buah itu sendiri maupun kualitas pelayanan terhadap konsumen. Oleh karena itu, seiring berkembangnya teknologi, mereka berlomba memanfaatkan teknologi untuk peningkatan kualitas tersebut. Salah satunya pemanenan dilakukan secara otomatis (robot pemanen).

Robot semacam ini telah dikembangkan di Jepang untuk memanen buah stroberi. Pada prinsipnya, hal semacam ini menerapkan ekstraksi warna. Warna menjadi salah satu ciri yang mudah diketahui untuk menentukan apakah buah atau sayur siap dipanen atau belum. Dalam penelitian ini menerapkan pengklasifikasian objek. Klasifikasi bertujuan untuk menetapkan kelas yang telah ditetapkan untuk setiap contoh. Hal ini dapat membantu untuk memahami data yang ada dan dapat digunakan untuk memprediksi bagaimana kasus baru akan berperilaku. Salah satu metode pengklasifikasian yang digunakan adalah algoritma LDA (Linear Discriminant Analysis) untuk mengklasifikasikan buah dalam kelas matang atau kelas mentah. Oleh karena itu, kami

mengembangkan salah satu dari teknologi itu. Software deteksi kematangan tomat, untuk menentukan apakah tomat telah matang (siap panen) atau tidak.

Pada penelitian ini metode yang digunakan dalam mengklasifikasikan buah tomat digunakan metode algoritma *LDA (Linear Discriminant Analysis)*. *LDA* bekerja berdasarkan analisa matrik penyebaran (*scatter matrix analysis*) yang bertujuan menemukan suatu proyeksi optimal sehingga dapat memproyeksikan data input pada ruang dengan dimensi yang lebih kecil dimana semua pola (pattern) dapat dipisahkan semaksimal mungkin. Karenanya untuk tujuan pemisahan tersebut maka *LDA* akan mencoba untuk memaksimalkan penyebaran data-data input diantara kelas-kelas yang berbeda dan sekaligus juga meminimalkan penyebaran input pada kelas yang sama. Perbedaan antar kelas direpresentasikan oleh matriks  $S_b$  (*scatter between class*) dan perbedaan dalam kelas direpresentasikan oleh matriks  $S_w$  (*scatter within class*).

Dalam *Discriminant Analysis*, variabel yang dependent (Y) merupakan kelas dan variabel yang independent (X) merupakan fitur objek yang mendeskripsikan kelas tersebut. Variabel dependent selalu berupa kategori (skala nominal) sedangkan variabel independent berupa skala pengukuran seperti interval atau rasio. Dengan metode *LDA* tersebut diharapkan kita dapat menentukan apakah buah tomat masuk ke kelas matang atau kelas mentah.

Pengumpulan data yang dibutuhkan dalam penelitian ini, akan dibahas pada bagian ini. Proses pengumpulan data dilakukan dalam beberapa tahap seperti dibawah ini :

1. Mengambil gambar tomat, disarankan gambar tomat dengan ukuran 300x300 pixel.
2. Melakukan cropping pada bagian gambar tomat untuk menghindari warna yang tidak dibutuhkan dalam proses pengambilan nilai RGB. Ukuran gambar yang diproses setelah melakukan cropping adalah 210x210 pixel.
3. Menghitung rata-rata nilai RGB.
4. Kualifikasi gambar menjadi 2 kelas yaitu matang dan mentah.
5. Memasukkan data RGB ke dalam tabel untuk diproses pada tahap pelatihan data.



Pada tahap ini akan dijelaskan tentang langkah-langkah dalam penghitungan LDA untuk mendapatkan nilai maksimum sehingga kita dapat menentukan apakah buah tomat itu masuk ke kelas matang atau kelas mentah. Data latih yang digunakan dalam penelitian ini sejumlah 50 data latih. Perlu dilakukan ekstraksi fitur, sebelum dilakukan pelatihan data. Hasil ekstraksi fitur tomat yaitu fitur red, green, dan blue seperti pada table 1.

No.	Red	Green	Blue	Keterangan
1.	233	72	38	Matang
2.	223	78	38	Matang
3.	185	61	45	Matang
4.	192	57	42	Matang
5.	144	28	23	Matang
6.	96	150	61	Mentah
7.	158	183	52	Mentah
8.	207	195	20	Mentah
9.	122	129	53	Mentah
10.	190	201	133	Mentah

**Tabel 2.1** Dataset RGB Gambar Buah Tomat

Pengujian data dilakukan untuk menentukan kelas objek k baru. Tahap-tahap dalam pengujian data meliputi penginputan yang merupakan tahap memasukkan objek baru berupa gambar. Dari gambar, di ambil nilai rata-rata RGB sebagai  $X_k$  untuk dimasukkan ke dalam rumus  $f_i$ . Kemudian klasifikasi dari proses penginputan, dicari nilai  $f_i$  untuk dibandingkan nilai  $f_1$  dan  $f_2$ -nya. Jika nilai  $f_1 > f_2$  maka masuk ke kelas matang, namun jika nilai  $f_1 < f_2$  maka masuk ke kelas mentah.

## 2.2 Sistem

Sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*systema*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi, di mana suatu model matematika seringkali bisa dibuat.

Sistem juga merupakan kesatuan bagian-bagian yang saling berhubungan yang berada dalam suatu wilayah serta memiliki item-item penggerak. Pada prinsipnya, setiap sistem selalui terdiri atas empat elemen yaitu objek yang dapat

berupa bagian, elemen, ataupun variable yang dapat berupa benda fisik, abstrak, ataupun keduanya sekaligus tergantung kepada sifat sistem tersebut. Ada beberapa elemen-elemen yang membentuk sebuah sistem yaitu tujuan, masukan, proses, keluaran, batas, mekanisme pengendalian dan umpan balik serta lingkungan.

Setiap sistem memiliki tujuan yang menjadi pemotivasi dalam mengarahkan sistem. Tanpa tujuan sistem menjadi tak terarah antara satu sistem dengan sistem yang lainnya. Keluaran merupakan hasil dari proses sistem yang dapat berupa informasi, saran, cetakan laporan, dan sebagainya.

### 2.3 Buah Nanas

Nanas, nenas, atau ananas (*Ananas comosus* (L.) Merr.) adalah sejenis tumbuhan tropis yang berasal dari Brasil, Bolivia, dan Paraguay. Tumbuhan ini termasuk dalam familia nanas-nanasan (Famili *Bromeliaceae*). Perawakan (habitus) tumbuhannya rendah, herba (menahun) dengan 30 atau lebih daun yang panjang, berujung tajam, tersusun dalam bentuk roset mengelilingi batang yang tebal. Buahnya dalam bahasa Inggris disebut sebagai *pineapple* karena bentuknya yang seperti pohon pinus. Nama 'nanas' berasal dari sebutan orang Tupi untuk buah ini: *anana*, yang bermakna "buah yang sangat baik". Burung penghisap madu (*hummingbird*) merupakan penyerbuk alamiah dari buah ini, meskipun berbagai serangga juga memiliki peran yang sama.



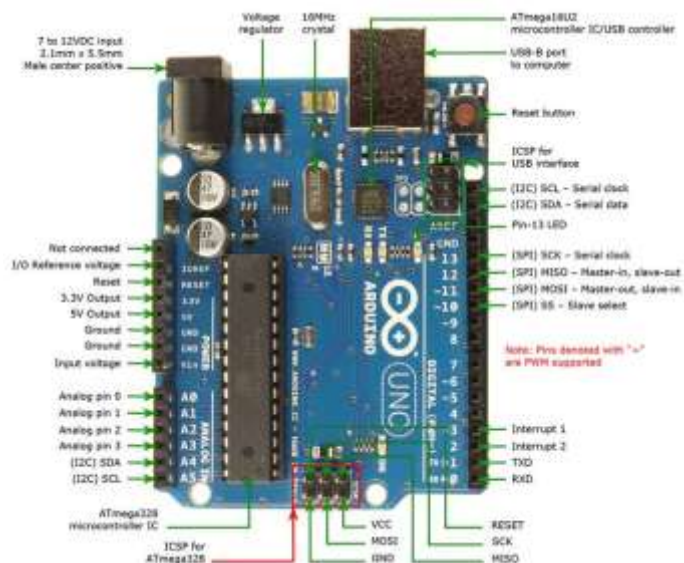
**Gambar 2.6** Buah Nanas

Buah nanas sebagaimana yang dijual orang bukanlah buah sejati, melainkan gabungan buah-buah sejati (bekasnya terlihat dari setiap 'sisik' pada kulit buahnya) yang dalam perkembangannya tergabung—bersama-sama dengan tongkol (spadix) bunga majemuk—menjadi satu 'buah' besar. Nanas yang dibudidayakan orang sudah kehilangan kemampuan memperbanyak

secara seksual, namun ia mengembangkan tanaman muda (bagian 'mahkota' buah) yang merupakan sarana perbanyakan secara vegetatif. nanas meningkatkan pencernaan dan mengurangi jerawat. Di Indonesia, provinsi Lampung merupakan daerah penanaman nanas utama, dengan beberapa pabrik pengolahan nanas juga terdapat di sana.

## 2.4 Arduino

Arduino UNO adalah sebuah board mikrokontroler yang didasarkan pada ATmega328. Arduino UNO mempunyai 14 pin digital input/output (6 di antaranya dapat digunakan sebagai output PWM), 6 input analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah power jack, sebuah ICSP header, dan sebuah tombol reset. Arduino UNO memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkannya ke sebuah computer dengan sebuah kabel USB atau mensuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai untuk memulainya.



**Gambar 2.7** Arduino UNO R3

Arduino Uno berbeda dari semua board Arduino sebelumnya, Arduino UNO tidak menggunakan chip driver FTDI USB-to-serial. Sebaliknya, fitur-fitur Atmega16U2 (Atmega8U2 sampai ke versi R2) diprogram sebagai sebuah pengubah USB ke serial. Revisi 2 dari board Arduino Uno mempunyai sebuah resistor yang menarik garis 8U2 HWB ke ground, yang membuatnya lebih mudah

untuk diletakkan ke dalam DFU mode. Revisi 3 dari board Arduino UNO memiliki fitur-fitur baru sebagai berikut:

- Pinout 1.0: ditambah pin SDA dan SCL yang dekat dengan pin AREF dan dua pin baru lainnya yang diletakkan dekat dengan pin RESET, IOREF yang memungkinkan shield-shield untuk menyesuaikan tegangan yang disediakan dari board. Untuk ke depannya, shield akan dijadikan kompatibel/cocok dengan board yang menggunakan AVR yang beroperasi dengan tegangan 5V dan dengan Arduino Due yang beroperasi dengan tegangan 3.3V. Yang ke-dua ini merupakan sebuah pin yang tak terhubung, yang disediakan untuk tujuan kedepannya
- Sirkuit RESET yang lebih kuat
- Atmega 16U2 menggantikan 8U2

Arduino UNO dapat disuplai melalui koneksi USB atau dengan sebuah power suplai eksternal. Sumber daya dipilih secara otomatis. Suplai eksternal (non-USB) dapat diperoleh dari sebuah adaptor AC ke DC atau battery. Adaptor dapat dihubungkan dengan mencolokkan sebuah center-positive plug yang panjangnya 2,1 mm ke power jack dari board. Kabel lead dari sebuah battery dapat dimasukkan dalam header/kepala pin Ground (Gnd) dan pin Vin dari konektor POWER.

Board Arduino UNO dapat beroperasi pada sebuah suplai eksternal 6 sampai 20 Volt. Jika disuplai dengan yang lebih kecil dari 7 V, kiranya pin 5 Volt mungkin mensuplai kecil dari 5 Volt dan board Arduino UNO bisa menjadi tidak stabil. Jika menggunakan suplai yang lebih dari besar 12 Volt, voltage regulator bisa kelebihan panas dan membahayakan board Arduino UNO. Range yang direkomendasikan adalah 7 sampai 12 Volt. Pin-pin dayanya adalah sebagai berikut:

- VIN. Tegangan input ke Arduino board ketika board sedang menggunakan sumber suplai eksternal (seperti 5 Volt dari koneksi USB atau sumber tenaga lainnya yang diatur). Kita dapat menyuplai tegangan melalui pin ini, atau jika penyuplaian tegangan melalui power jack, aksesnya melalui pin ini.
- 5V. Pin output ini merupakan tegangan 5 Volt yang diatur dari regulator pada board. Board dapat disuplai dengan salah satu suplai dari DC power jack (7-

12V), USB connector (5V), atau pin VIN dari board (7-12). Penyuplaian tegangan melalui pin 5V atau 3,3V membypass regulator, dan dapat membahayakan board. Hal itu tidak dianjurkan.

- 3V3. Sebuah suplai 3,3 Volt dihasilkan oleh regulator pada board. Arus maksimum yang dapat dilalui adalah 50 mA.
- GND. Pin ground.

Arduino UNO mempunyai sejumlah fasilitas untuk komunikasi dengan sebuah komputer, Arduino lainnya atau mikrokontroler lainnya. Atmega 328 menyediakan serial komunikasi UART TTL (5V), yang tersedia pada pin digital 0 (RX) dan 1 (TX). Sebuah Atmega 16U2 pada channel board serial komunikasinya melalui USB dan muncul sebagai sebuah port virtual ke software pada komputer. Firmware 16U2 menggunakan driver USB COM standar, dan tidak ada driver eksternal yang dibutuhkan. Bagaimanapun, pada Windows, sebuah file inf pasti dibutuhkan. Software Arduino mencakup sebuah serial monitor yang memungkinkan data tekstual terkirim ke dan dari board Arduino. LED RX dan TX pada board akan menyala ketika data sedang ditransmit melalui chip USB-to-serial dan koneksi USB pada komputer (tapi tidak untuk komunikasi serial pada pin 0 dan 1).

Dari pada mengharuskan sebuah penekanan fisik dari tombol reset sebelum sebuah penguploadan, Arduino Uno didesain pada sebuah cara yang memungkinkannya untuk direset dengan software yang sedang berjalan pada komputer yang sedang terhubung. Salah satu garis kontrol aliran hardware (DTR) dari ATmega8U2/16U2 sihubungkan ke garis reset dari ATmega328 melalui sebuah kapasitor 100 nanofarad. Ketika saluran ini dipaksakan (diambil rendah), garis reset jatuh cukup panjang untuk mereset chip. Software Arduino menggunakan kemampuan ini untuk memungkinkan kita untuk mengupload kode dengan mudah menekan tombol upload di software Arduino. Ini berarti bahwa bootloader dapat mempunyai sebuah batas waktu yang lebih singkat, sebagai penurunan dari DTR yang dapat menjadi koordinasi yang baik dengan memulai penguploadan. Pengaturan ini mempunyai implikasi. Ketika Arduino Uno dihubungkan ke sebuah komputer lain yang sedang running menggunakan OS Mac X atau Linux, Arduino Uno mereset setiap kali sebuah koneksi dibuat dari

software (melalui USB). Untuk berikutnya, setengah-detik atau lebih, bootloader sedang berjalan pada Arduino UNO.

Ketika Arduino UNO diprogram untuk mengabaikan data yang cacat/salah untuk menahan beberapa bit pertama dari data yang dikirim ke board setelah sebuah koneksi dibuka. Jika sebuah sketch sedang berjalan pada board menerima satu kali konfigurasi atau data lain ketika sketch pertama mulai, memastikan bahwa software yang berkomunikasi menunggu satu detik setelah membuka koneksi dan sebelum mengirim data ini. Arduino Uno berisikan sebuah jejak yang dapat dihapus untuk mencegah reset otomatis. Pad pada salah satu sisi dari jejak dapat disolder bersama untuk mengaktifkan kembali. Pad itu diberi label "RESET-RN" Kita juga dapat menonaktifkan reset otomatis dengan menghubungkan sebuah resistor 110 ohm dari tegangan 5V ke garis reset.

## 2.5 Sensor Warna TCS

TCS3200 and TCS3210 merupakan konverter yang diprogram untuk mengubah warna menjadi frekuensi yang tersusun atas konfigurasi silicon photodiode dan konverter arus ke frekuensi dalam IC CMOS monolithic yang tunggal. Keluaran dari sensor ini adalah gelombang kotak (duty cycle 50%) frekuensi yang berbanding lurus dengan intensitas cahaya (irradiance). Keluaran frekuensi skala penuh dapat diskalakan oleh satu dari tiga nilai-nilai yang ditetapkan via dua kontrol pin input. Masukan digital dan keluaran digital memungkinkan antarmuka langsung ke mikrokontroler atau sirkuit logika lainnya. Tempat output enable (OE) output dalam keadaan impedansi tinggi untuk beberapa unit dapat berbagi jalur masukan mikrokontroler. didalam TCS3200, konverter cahaya ke frekuensi membaca sebuah array 8x8 dari photodiode, 16 photodiode mempunyai penyaring warna biru, 16 photodiode mempunyai penyaring warna merah, 16 photodiode mempunyai penyaring warna hijau, dan 16 photodiode untuk warna terang tanpa penyaring. Dalam TCS3210, converter cahaya ke frekuensi membaca sebuah array 4x6 dari photodiode, 6 photodiode mempunyai penyaring warna biru, 6 photodiode mempunyai penyaring warna hijau, 6 photodiode mempunyai penyaring warna merah, dan 6 photodiode untuk warna terang tanpa penyaring. 4 tipe warna dari photodiode telah diintegrasikan

untuk meminimalkan efek ketidak seragaman dari insiden irradiance. Semua photodiode dari warna yang sama telah terhubung secara parallel. Pin S2 dan S3 digunakan untuk memilih grup dari photodiode(merah, hijau, biru, jernih) yang telah aktif.



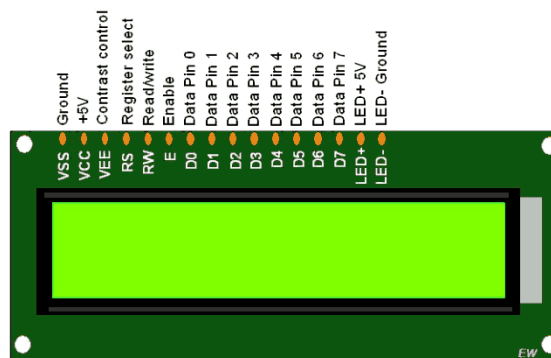
**Gambar 2.8** Sensor Warna TCS

Fitur-fitur yang dimiliki sensor warna TCS 3200 maupun TCS 3210 ialah sebagai berikut

- Konversi Tinggi Resolusi Intensitas Cahaya ke Frekuensi.
- Warna Diprogram dan Full Skala Frekuensi Keluaran
- Berkomunikasi Langsung Dengan Microcontroller
- Pasokan tunggal Operasi (2,7 V sampai 5,5 V)
- Mempunyai Power Down Fitur
- Kesalahan Nonlinier Biasanya 0,2% pada 50 kHz
- Stabil 200 ppm / ° C Koefisien Suhu
- Bebas Timbal (Pb) dan RoHS
- Kompatibel Paket “Surface Mount”

## **2.6 LCD (*Liquid Crystal Display*)**

Display elektronik adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik. LCD (*Liquid Cristal Display*) adalah salah satu jenis display elektronik yang dibuat dengan teknologi CMOS logic yang bekerja dengan tidak menghasilkan cahaya tetapi memantulkan cahaya yang ada di sekelilingnya terhadap front-lit atau mentransmisikan cahaya dari back-lit. LCD (*Liquid Cristal Display*) berfungsi sebagai penampil data baik dalam bentuk karakter, huruf, angka ataupun grafik.



**Gambar 2.9** LCD (*Liquid Crystal Display*)

Cara kerja LCD secara umum adalah port RW diberi logika rendah “0”. Bus data terdiri dari 4-bit atau 8-bit. Jika jalur data 4-bit maka yang digunakan ialah DB4 sampai dengan DB7. Sebagaimana terlihat pada table diskripsi, interface LCD merupakan sebuah parallel bus, dimana hal ini sangat memudahkan dan sangat cepat dalam pembacaan dan penulisan data dari atau ke LCD. Kode ASCII yang ditampilkan sepanjang 8-bit dikirim ke LCD secara 4-bit atau 8 bit pada satu waktu. Jika mode 4-bit yang digunakan, maka 2 nibble data dikirim untuk membuat sepenuhnya 8-bit (pertama dikirim 4-bit MSB lalu 4-bit LSB dengan pulsa clock EN setiap nibblenya). Jalur kontrol EN digunakan untuk memberitahu LCD bahwa mikrokontroller mengirimkan data ke LCD. Untuk mengirim data ke LCD program harus menset EN ke kondisi high “1” dan kemudian menset dua jalur kontrol lainnya (RS dan R/W) atau juga mengirimkan data ke jalur data bus. Saat jalur lainnya sudah siap, EN harus diset ke “0” dan tunggu beberapa saat (tergantung pada datasheet LCD), dan set EN kembali ke high “1”. Ketika jalur RS berada dalam kondisi low “0”, data yang dikirimkan ke LCD dianggap sebagai sebuah perintah atau instruksi khusus (seperti bersihkan layar, posisi kursor dll). Ketika RS dalam kondisi high atau “1”, data yang dikirimkan adalah data ASCII yang akan ditampilkan dilayar. Misal, untuk menampilkan huruf “A” pada layar maka RS harus diset ke “1”. Jalur kontrol R/W harus berada dalam kondisi low (0) saat informasi pada data bus akan dituliskan ke LCD. Apabila R/W berada dalam kondisi high “1”, maka program akan melakukan query (pembacaan) data dari LCD. Instruksi pembacaan hanya satu, yaitu Get LCD status (membaca status LCD), lainnya merupakan instruksi penulisan. Jadi hampir setiap aplikasi yang menggunakan LCD, R/W selalu diset ke “0”. Jalur data dapat terdiri 4 atau 8 jalur (tergantung mode yang dipilih



pengguna), DB0, DB1, DB2, DB3, DB4, DB5, DB6 dan DB7. Mengirim data secara paralel baik 4-bit atau 8-bit merupakan 2 mode operasi primer.

Mode 8-bit sangat baik digunakan ketika kecepatan menjadi keutamaan dalam sebuah aplikasi dan setidaknya minimal tersedia 11 pin I/O (3 pin untuk kontrol, 8 pin untuk data). Sedangkan mode 4 bit minimal hanya membutuhkan 7-bit (3 pin untuk kontrol, 4 pin untuk data). Bit RS digunakan untuk memilih apakah data atau instruksi yang akan ditransfer antara mikrokontroller dan LCD. Jika bit ini di set ( $RS = 1$ ), maka byte pada posisi kursor LCD saat itu dapat dibaca atau ditulis. Jika bit ini di reset ( $RS = 0$ ), merupakan instruksi yang dikirim ke LCD atau status eksekusi dari instruksi terakhir yang dibaca.

## 2.7 Arduino IDE

IDE itu merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui software inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. IC mikrokontroler Arduino telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler.



**Gambar 2.10** Logo Arduino

Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan library C/C++ yang biasa disebut *Wiring* yang membuat

operasi input dan output menjadi lebih mudah. Arduino IDE ini dikembangkan dari software Processing yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino.

## 2.8 Bahasa Pemrograman C

Bahasa Pemrograman C adalah sebuah bahasa pemrograman komputer yang bisa digunakan untuk membuat berbagai aplikasi (*general-purpose programming language*), mulai dari sistem operasi (seperti Windows atau Linux), antivirus, software pengolah gambar (*image processing*), hingga *compiler* untuk bahasa pemrograman, dimana C banyak digunakan untuk membuat bahasa pemrograman lain yang salah satunya adalah PHP.

Meskipun termasuk *general-purpose programming language*, yakni bahasa pemrograman yang bisa membuat berbagai aplikasi, bahasa pemrograman C paling cocok merancang aplikasi yang berhubungan langsung dengan Sistem Operasi dan hardware. Ini tidak terlepas dari tujuan awal bahasa C dikembangkan. Bahasa pemrograman C dibuat pertama kali oleh Dennis M. Ritchie pada tahun 1972. Saat itu Ritchie bekerja di Bell Labs, sebuah pusat penelitian yang berlokasi di Murray Hill, New Jersey, Amerika Serikat.

Ritchie membuat bahasa pemrograman C untuk mengembangkan sistem operasi UNIX. Sebelumnya, sistem operasi UNIX dibuat menggunakan bahasa assembly (*assembly language*). Akan tetapi bahasa assembly sendiri sangat rumit dan susah untuk dikembangkan. Dengan tujuan mengganti bahasa *assembly*, peneliti di Bell Labs membuat bahasa pemrograman B. Namun bahasa pemrograman B juga memiliki beberapa kekurangan, yang akhirnya di lengkapi oleh bahasa pemrograman C. Dengan bahasa C inilah sistem operasi UNIX ditulis ulang. Pada gilirannya, UNIX menjadi dasar dari banyak sistem operasi modern saat ini, termasuk Linux, Mac OS (iOS), hingga sistem operasi Android.