

BAB 2

TINJAUAN PUSTAKA

2.1 Definisi Burung Kenari

Burung kenari merupakan salah satu jenis burung berkicau yang banyak digemari masyarakat Indonesia. Burung kenari yang menjadi favorit peternak diantaranya adalah burung kenari warna bulu kuning dan putih. Memelihara binatang telah menjadi bagian dari kehidupan manusia sejak dulu. Disamping menguntungkan, juga karena membawa kesenangan tersendiri yang terkadang tidak ternilai dengan uang. Dari sekian binatang piaraan, terdapat sebagian orang yang gemar memelihara burung yang suaranya merdu atau warna bulunya yang indah. Salah satu burung ocean yang digemari masyarakat karena keindahan warna bulu dan morfometrinya adalah burung kenari (*Serinus canarius*)[8].

Ada sekitar 99 jenis burung kenari (*serinus canarius*) yang telah dikenal. Ada sekitar 99 jenis kenari (*Serinus canarius*) yang telah dikenal. Munculnya berbagai kenari ini dikarenakan kenari mudah ditenakkan dan para peternak selalu membuat inovasi baru dalam perkawinan silang. Untuk mendapatkan warna baru yang lebih menarik, para peternak kenari cenderung menyilangkan burung kenari polos yang beda warna untuk memperoleh warna yang bervariasi. Sebagai contoh warna bulu kuning polos dengan putih polos akan mendapatkan warna campuran kuning dan putih atau kuning bond putih. Burung yang dengan warna bulu campuran, merupakan koleksi yang menarik. Tetapi banyak juga peternak yang mempertahankan warna dasar seperti kenari kuning dan putih. Budidaya inilah yang cenderung terjadi pada masyarakat, mereka memelihara kenari bukan hanya untuk mendengarkan alunan suara tapi lebih dari itu untuk mengagumi keindahan bulunya [9].

Pada pemilik burung kenari ini, pemilik tidak bisa terus mengontrol pemberian pakan pada burung miliknya sehingga pemberian pakan pada burung

menjadi terlaambat, atau pun memberi pakan terlalu cepat dari waktu yang telah ditentukan, sehingga membuat pertumbuhan burung menjadi sedikit lebih terhambat dan suara merdunya pun tidak terdengar. Maka dari itu perlu dibangun alat untuk dapat memberi pakan burung secara otomatis berdasarkan pola tingkah laku burung.

Berdasarkan permasalahannya, burung kenari dapat tumbuh normal dan suara merdunya-pun keluar jika pemberian pakan dilakukan secara teratur. Burung kenari pun memiliki beberapa ciri ketika merasa lapar yaitu, akan terus aktif bergerak atau gelisah.

2.2 Data dan Informasi

2.2.1 Data

Secara konseptual, data adalah deskripsi tentang benda, kejadian, aktivitas dan transaksi, yang tidak mempunyai makna atau tidak berpengaruh secara langsung kepada pemakai. Data sering kali disebut sebagai bahan mentah informasi.

Berikut adalah kutipan pengertian data dari sudut pandang yang berbeda :

1. Menurut kamus bahasa, data diartikan atau diterjemahkan sebagai fakta atau bahan-bahan keterangan.
2. Dari sudut pandang bisnis, terdapat pengertian data bisnis sebagai berikut:
“Business data is an organization’s description of things (resources) and events (transactions) that it faces.” Jadi data dalam bisnis merupakan gambaran organisasi tentang sesuatu (sumber) dan peristiwa (transaksi) yang terjadi.
3. Gordon B. Davis dalam buku *Management Information System: Conceptual Foundations, Structures and Development* yang dikutip oleh Teguh Wahyono dalam buku Sistem Informasi, menyebut data sebagai bahan mentah (*raw materias*) yang dirumuskan sebagai sekelompok lambang-lambang tidak acak yang menunjukkan jumlah atau tindakan, atau hal-hal lainnya [10].

Dari ketiga pengertian diatas, dapat diambil kesimpulan bahwa data adalah sebuah bahan informasi. Data juga terbentuk dari karakter, alphabet, angka, maupun simbol khusus untuk disusun dan diolah dalam bentuk struktur data, dan lain-lain.

2.2.2 Informasi

Informasi adalah data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat.

Secara umum pengertian informasi adalah sekumpulan data atau fakta yang telah diproses dan dikelola sedemikian rupa sehingga menjadi sesuatu yang mudah dimengerti dan bermanfaat bagi penerimanya. Berikut pengertian informasi menurut para ahli :

1. Ladjamudin berpendapat bahwa informasi adalah data yang telah diolah menjadi bentuk yang lebih berarti dan berguna bagi penerimanya untuk mengambil keputusan masa kini maupun masa yang akan datang [11].
2. Kadir informasi merupakan data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan orang yang menggunakan data tersebut.
3. Gordon B. Davis Informasi adalah data yang telah diolah menjadi suatu bentuk yang penting bagi si penerima dan mempunyai nilai nyata yang dapat dirasakan dalam keputusan-keputusan yang sekarang atau keputusan-keputusan yang akan datang [12].
4. Jogiyanto yang dikutip oleh Machmud adalah: “Informasi diartikan sebagai data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya” [13].

Dari keempat penjelasan diatas, dapat disimpulkan bahwa informasi adalah data yang diproses kedalam bentuk yang lebih berarti bagi penerima dan berguna pada saat pengambilan keputusan.

2.3 Konsep Perancangan Sistem

Konsep perancangan sistem pada implementasi pembangunan alat pakan burung otomatis ini berbasis *internet of things*.

Cara kerja Internet of Things yaitu dengan memanfaatkan sebuah argumentasi pemrograman yang dimana tiap-tiap perintah argumennya ini menghasikan sebuah interaksi antara sesama mesin yang terhubung secara otomatis tanpa campur tangan manusia dan dalam jarak berapapun. Internetlah yang menjadi penghubung diantara kedua interaksi mesin tersebut.

2.4 Unified Modeling Language

Unified Modeling Language (UML) adalah suatu cara dalam menyelesaikan sebuah masalah dengan mendeskripsikan menjadi standar dalam dunia industri untuk visualisasi, rancangan dan dokumentasi sebuah sistem atau perangkat lunak. Dengan menggunakan UML kita dapat membangun model dalam segala bentuk dan jenis aplikasi perangkat lunak, aplikasi yang dibangun dapat berjalan pada perangkat lunak dengan jenis sistem operasi dan jaringan apapun [14].

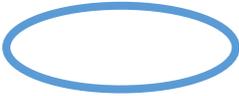
Selain itu, dengan menggunakan UML, programmer dapat mengerti, memahami dengan jelas maksud, tujuan dan arah rancangan sistem, sehingga sistem perangkat lunak yang dirancang dapat dibangun dengan bahasa program apapun. Namun karena UML merupakan suatu model pengembangan sistem perangkat lunak yang berbasis *object oriented* sehingga menggunakan bentuk class dan operation dalam konsep dasarnya, maka lebih cocok dalam pemrogramannya dengan menggunakan bahasa-bahasa pemrograman yang berbasis OOP (*Object Oriented Programming*) seperti C, C++, *Java*, VB dan lain sebagainya.

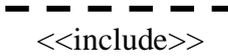
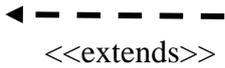
Dalam membangun suatu model sistem perangkat lunak dengan UML, digunakan bentuk-bentuk diagram atau symbol untuk merepresentasikan elemen-elemen dalam sistem. Bentuk diagram yang digunakan seperti *Use Case Diagram*, *Class Diagram*, *Sequence Diagram* dan *Activity Diagram*

2.4.1 Use Case Diagram

Use Case Diagram merupakan pemodelan untuk kelakuan (*behaviour*) sistem informasi yang akan dibuat. *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu :

Tabel 2.1 Use Case Diagram

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja.</p>
	<p>Aktor adalah <i>substraction</i> dari orang atau sistem lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>Use Case</i>, tetapi tidak memiliki kontrol terhadap <i>use case</i>.</p>

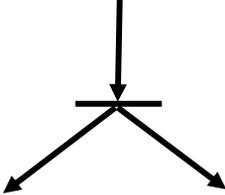
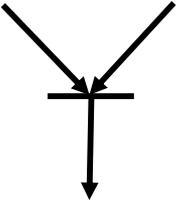
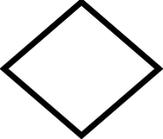
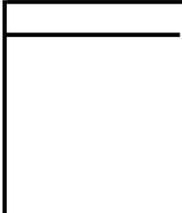
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengidentifikasi siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidentifikasi data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

2.4.2 Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity Diagram* yaitu:

Tabel 2.2 Daigram Activity

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
	<i>End point</i> , akhir aktivitas

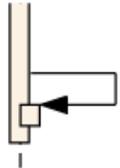
	<i>Activities</i> , menggambarkan suatu proses / kegiatan bisnis.
	<i>Fork</i> /percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

2.4.3 Diagram Urutan (*Sequences Diagram*)

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *Sequence Diagram* yaitu :

Tabel 2.3 Sequence Diagram

Gambar	Keterangan
--------	------------

	<p><i>Entity Class</i>, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.</p>
	<p><i>Boundary Class</i>, berisi kesimpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan form entry dan form cetak.</p>
	<p><i>Control Class</i>, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.</p>
	<p><i>Message</i>, simbol mengirim pesan antar <i>class</i>.</p>
	<p><i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.</p>
	<p><i>Activation</i>, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

2.4.4 Diagram Kelas (*Class Diagram*)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

Class Diagram secara khas meliputi : Kelas (*Class*), Relasi *Associations*, *Generalisation* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality*.

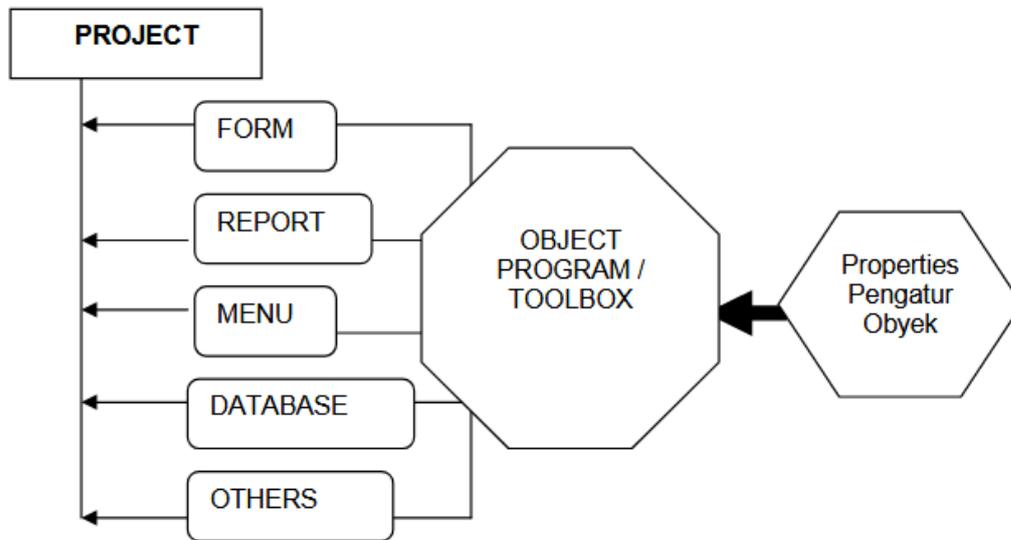
Tabel 2.4 Class Diagram

<i>Multiplicity</i>	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 dan maksimal 4.

2.5 Object Oriented Programming

Object Oriented Programming adalah sebuah istilah yang diberikan kepada bahasa pemrograman yang menggunakan teknik berorientasi pada sebuah objek dalam pembangunan program aplikasi, maksudnya bahwa orientasi pembuatan program tidak lagi menggunakan orientasi linear melainkan berorientasi pada objek-objek yang terpisah-pisah. Suatu perintah dalam bahasa ini diwakili oleh sebuah objek didalamnya berisi beberapa perintah-perintah standar sederhana.

Objek ini dikumpulkan dalam Modul form atau Report atau modul lain dan disusun didalam sebuah project. Gambaran tentang pemrograman ini seperti dibawah ini :



Gambar 2.1 Model Pemrograman OOP

2.6 Mikrokontroler

Mikrokontroler (pengendali mikro) merupakan sistem mikroprosesor lengkap yang terkandung dalam sebuah chip. Mikrokontroler berbeda dari mikroprosesor serba guna yang digunakan dalam sebuah PC, karena didalam sebuah mikrokontroler umumnya telah terdapat komponen pendukung sistem minimal mikroprosesor, yakni memori dan antarmuka I/O, bahkan ada beberapa jenis mikrokontroler yang memiliki fasilitas ADC, PLL, EEPROM dalam satu kemasan, sedangkan di dalam mikroprosesor umumnya hanya berisi CPU saja [15].

2.6.1 Sejarah Mikrokontroler

Perkembangan Mikrokontroler dimulai dari awal tahun 1970an. Motorola mengeluarkan seri mikrokontroler 6800 yang terus dikembangkan hingga sekarang menjadi 68HC05, 68HC11, 68HC12, dan 68HC16. Zilog juga mengeluarkan seri

mikroprosesor Z80-ny yang terkenal dan terus dikembangkan hingga kini menjadi Z180 dan kemudian diadopsi juga oleh mikroprosesor Rabbit. Intel mengeluarkan mikrokontrolernya yang populer di dunia yaitu 8051, yang karena begitu populernya maka arsitektur 8051 tersebut kemudian diadopsi oleh vendor lain seperti Philips, Siemens, Atmel, dan vendor-vendor lain dalam produk mikrokontroler mereka. Masih ada mikrokontroler populer lainnya seperti Basic Stamps, PIC dari Microchip, MSP 430 dari Texas Instrument dan masih banyak lagi [16].

2.7 Sejarah Arduino

Arduino adalah sebuah platform komputasi fisik *open source* berbasiskan Rangkaian *input / output* sederhana mengimplementasikan bahasa *processing*. Arduino dapat digunakan untuk mengembangkan objek interaktif mandiri atau dapat dihubungkan ke perangkat lunak pada komputer anda (seperti Flash, Pengolahan, VVVV, atau Max / MSP). Rangkaiannya dapat dirakit dengan tangan atau dibeli. IDE (*Integrated Development Environment*) Arduino bersifat *open source* [16].

2.7.1 Board Arduino

Arduino adalah sebuah *board* mikrokontroler yang bersifat *open source*, tidak hanya sekedar sebuah alat pengembangan, tetapi ia adalah sebuah kombinasi dari hardware; bahasa pemrograman dan *Integrated Development Environment*(IDE).

Arduino IDE sangat berperan untuk menulis program, ,emg-*compile* menjadi kode biner dan meng-*upload* ke dalam *memory* mikrokontroler. *Bootloader* Chip/IC pada *Board Arduino* telah diisi program *Arduino Bootloader*, yang memungkinkan kita meng-*upload* *code* program tanpa menggunakan *hardware* tambahan.

Bootloader akan aktif selama beberapa detik ketika *board* mengalami *reset*, hasil kompilasi dari *Arduino IDE* dapat dipergunakan dan dijalankan tidak hanya pada *Arduino board* tetapi juga dapat dijalankan di sistem mikrokontroler AVR yang sesuai bahkan tanpa *bootloader*. *Bootloader* adalah program inisiasi yang ukurannya kecil, dijalankan oleh CPU saat daya dihidupkan. Setelah *bootloader* selesai dijalankan, berikutnya program didalam RAM akan dieksekusi.

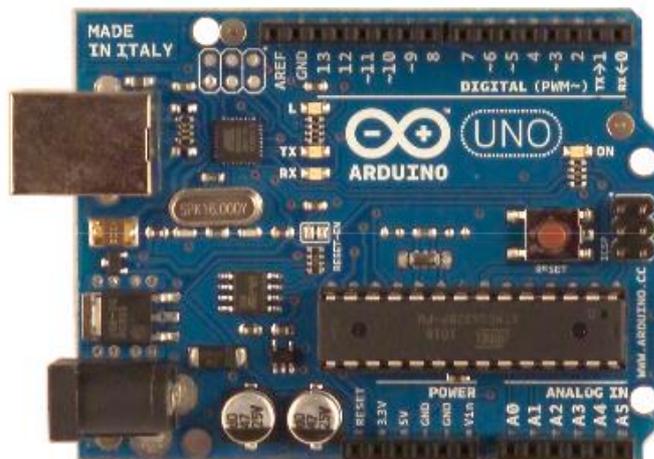
Berikut adalah beberapa contoh jenis-jenis papan arduino :

1. Arduino USB

Menggunakan USB sebagai antar muka pemrograman atau komunikasi komputer.

Contoh :

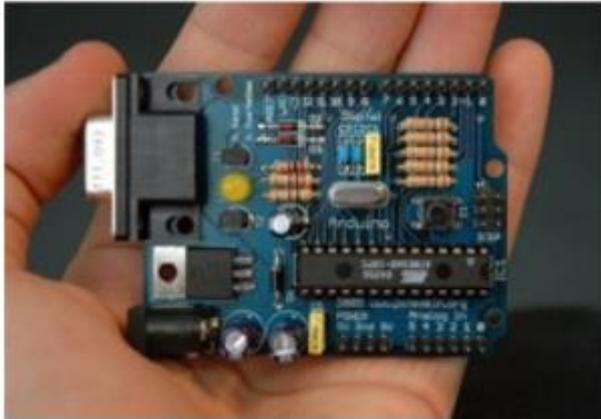
- a) Arduino uno
- b) arduinoDuemilanove
- c) arduino NG(Nouva Generazione)
- d) arduino Extreme dan Arduino Extreme v2
- e) arduino USB dan Arduino USB v2.0



Gambar 2.2 Board Arduino NG

2. Arduino Serial

Menggunakan RS232 sebagai antar muka pemrograman atau komunikasi komputer

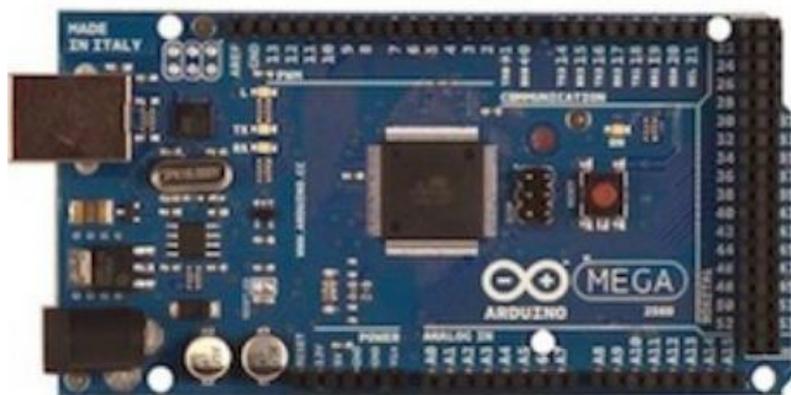


Gambar 2.3 Board Arduino Serial

3. Arduino Mega 2560

Papan arduino dengan spesifikasi yang lebih tinggi, dilengkapi tambahan pin digital, pin analog, port serial dan sebagainya. Contoh :

- a) Arduino Mega
- b) Arduino Mega 2560

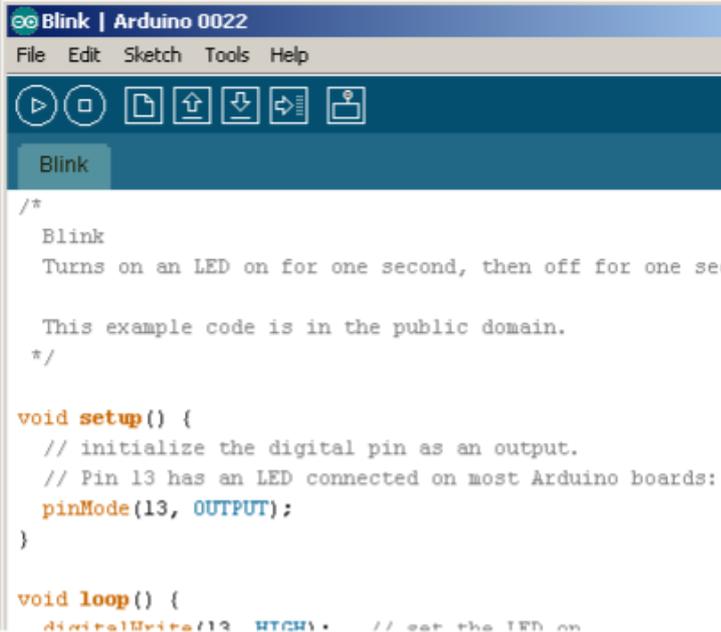


Gambar 2.4 Board Arduino Mega 2560

2.7.2 Software IDE Arduino

IDE *Arduino* adalah software yang sangat canggih ditulis dengan menggunakan Java. IDE *Arduino* terdiri dari :

- a) *Editor* program, sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *Processing*.
- b) *Compiler*, sebuah modul yang mengubah kode program (bahasa *Processing*) menjadi kode biner. Bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa *Processing*. Yang bisa dipahami oleh mikrokontroler adalah kode biner. Itulah sebabnya *compiler* diperlukan dalam hal ini.
- c) *Uploader*, sebuah modul yang memuat kode biner dari komputer kedalam *memory* di dalam papan *Arduino*.

The image shows a screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 0022". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for running, stopping, saving, uploading, and downloading. The main editor area displays the following code:

```
/*
  Blink
  Turns on an LED on for one second, then off for one second...

  This example code is in the public domain.
  */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
```

Gambar 2.5 Antarmuka *Software IDE Arduino*

Komponen-komponen dan fungsi dari *software IDE Arduino* adalah sebagai berikut:

a. *File*

Tabel 2.5 Pilihan Menu File

<i>New</i>	Membuat <i>sketch</i> baru
<i>Open</i>	Membuka <i>file sketch</i> yang sudah disimpan
<i>Sketchbook</i>	Membuka <i>file sketch</i> yang pernah dibuat
<i>Examples</i>	Membuka contoh-contoh <i>file sketch</i> yang berisi berbagai macam aplikasi yang disediakan oleh <i>Arduino</i>
<i>Close</i>	Menutup <i>sketch</i>
<i>Save</i>	Menyimpan <i>sketch</i>
<i>Save As</i>	Menyimpan <i>sketch</i> dengan nama lain
<i>Upload to I/O Board</i>	Mengunggah program ke <i>board</i>
<i>Page Setup</i>	Mengatur ukuran halaman pada pencetak
<i>Print</i>	Mencetak <i>sketch</i>
<i>Preferences</i>	Mengatur setting IDE <i>Arduino</i>
<i>Quit</i>	Keluar dari IDE <i>Arduino</i>

b. *Menu Edit*

Cut, Copy, Copy for Forum, Copy as HTML, Paste, Select All, Comment, Increase Indent, Decrease Indent, Find, Find Next

c. *Menu Sketch*

Tabel 2.6 Pilihan Menu Sketch

<i>Verify/Compile</i>	Mengompilasi program
<i>Stop</i>	Menghentikan kompilasi
<i>Show sketch folder</i>	Menampilkan <i>folder</i> dari <i>sketch</i> yang sedang dibuka
<i>Import Library</i>	Mengambil <i>header</i> librari dari fungsi-fungsi tambahan
<i>Add File</i>	Menambah buka file <i>sketch</i> pada jendela yang sama

d. *Menu Tools***Tabel 2.7 Pilihan Menu Tools**

<i>Auto Format</i>	Mengatur <i>format sketch</i> secara otomatis
<i>Archive Sketch</i>	Menyimpan <i>sketch</i> dalam bentuk Zip file (kompresi)
<i>Fix Encoding & Reload</i>	Membatalkan perubahan <i>sketch</i> dan mengambil ulang <i>sketch</i> sebelumnya yang telah disimpan
<i>Serial Monitor</i>	Mengaktifkan jendela tampilan komunikasi serial pada komputer
<i>Board</i>	Menentukan jenis <i>board Arduino</i> yang digunakan
<i>Serial Port</i>	Menentukan <i>port serial</i> yang digunakan untuk mengunggah program dan tersambung pada <i>board Arduino</i>
<i>Burn Bootloader</i>	Memasukan <i>bootloader</i> pada mikrokontroler yang ada pada <i>board Arduino</i> melalui ICSP

2.8.3 Bahasa Pemograman *Arduino*

Syahwil menyatakan, bahwa banyak bahasa yang bisa digunakan untuk program mikrokontroler, misalnya bahasa *assembly*. Namun dalam pemrograman arduino bahasa yang dipakai adalah bahasa C [15].

Kusuma menyatakan, bahwa akar bahasa C adalah bahasa BCPL yang dikembangkan oleh Martin Richards pada tahun 1967. Bahasa C adalah bahasa *standard*, artinya suatu program yang ditulis dengan versi bahasa C tentu akan dapat dikompilasikan dengan versi bahasa C yang lain dengan sedikit modifikasi. Beberapa alasan mengapa bahasa C banyak digunakan, diantaranya sebagai berikut [17] :

- a) Bahasa C tersedia hampir disemua jenis komputer
- b) Kode bahasa C bersifat *portable*
- c) Bahasa C hanya menyediakan sedikit kata-kata kunci.
- d) Proses *executable program* bahasa C lebih cepat
- e) Dukungan pustaka yang banyak.
- f) C adalah bahasa yang terstruktur
- g) Selain bahasa tingkat tinggi, C juga dianggap sebagai bahasa tingkat menengah
- h) Bahasa C adalah *compiler*.

