

BAB II

TINJAUAN PUSTAKA

2.1 Sistem Biometrika

Biometrika atau *biometrics* berasal dari kata *bio* dan *metrics*. *Bio* berarti suatu yang hidup, dan *metrics* berarti mengukur. Biometrika berarti mengukur karakteristik pembeda (*distinguishing traits*) pada badan atau perilaku seseorang yang digunakan untuk melakukan pengenalan secara otomatis terhadap identitas orang tersebut, dengan membandingkannya dengan karakteristik yang sebelumnya sudah disimpan dalam database. Pengertian pengenalan secara otomatis pada definisi biometrika di atas adalah dengan menggunakan teknologi (komputer). Pengenalan terhadap identitas seseorang dapat dilakukan secara waktu nyata (*realtime*), tidak membutuhkan waktu berjam-jam ataupun berhari-hari untuk proses pengenalan itu. (Darma Putra, 2007).

Teknologi biometrika menawarkan autentikasi secara biologis memungkinkan sistem dapat mengenali penggunanya secara tepat. Terdapat beberapa metode diantaranya : *fingerprint scanning*, *retina scanning*, dan *DNA scanning*. *Fingerprint scanning* saat ini telah digunakan secara luas.

2.2 Sidik Jari

Sidik jari adalah gurat-gurat yang terdapat di kulit ujung jari. Sidik jari berfungsi untuk member gaya gesek lebih besar agar jari dapat memegang benda-benda lebih erat. Sistem pengamanan dengan menggunakan sidik jari sudah mulai dipergunakan di Amerika oleh seorang bernama E. Henry pada tahun 1902. Henry menggunakan metode sidik jari untuk melakukan identifikasi pekerja dalam rangka mengatasi pemberian upah ganda. Sistem Henry menggunakan pola *ridge* (*Ridge* = punggung alur pada kulit, baik pada tangan atau kaki), yang terpusat pola jari tangan, jari kaki, khususnya telunjuk. Untuk memperoleh gambar pola *ridge*, dilakukan dengan cara menggulung jari yang diberi tinta pada suatu kartu cetakan hingga dihasilkan suatu pola *ridge* yang unik bagi masing-masing individu. Para

pakar membuktikan bahwa tidak ada dua individu yang mempunyai pola *ridge* yang serupa. Pola *ridge* tidaklah diwariskan. Pola *ridge* dibentuk waktu embrio, dan tidak pernah berubah seumur hidup. Perubahan *ridge* hanya dapat terjadi akibat trauma, misal akibat luka-luka, terbakar, penyakit, atau penyebab lainnya. Sistem biometrika sidik jari merupakan sistem yang paling banyak digunakan saat ini karena memiliki tingkat akurasi yang tinggi dan mudah untuk diterapkan. Dari hasil penelitian, ditemukan 9 macam pola utama *papillary ridge*, antara lain:

1. *Loop* : Terdiri dari satu atau lebih kurva bebas dari *ridge* dan sebuah delta.
2. *Arch* : Membentuk pola dengan *ridge* berada diatas *ridge* yang lain dalam bentuk lengkungan umum.
3. *Whorl* : Pola ini terdiri dari satu atau lebih kurva bebas *ridge* dan dua buah delta.
4. *Tented Arch* : Pola ini terdiri dari paling tidak sebuah *ridge* yang melengkung keatas yang kemudian bercabang menjadi dua *ridge*.
5. *Double Loop*: Pola ini membentuk dua formasi lengkungan yang lalu berpisah, dengan dua titik delta.
6. *Central Pocket Loop* : Terdiri dari satu atau lebih kurva *ridge* dan dua titik delta.
7. *Accidental* : Pola ini mempunyai dua titik delta. Satu delta akan berhubungan dengan lengkungan keatas, dan delta yang lain terhubung dengan lengkungan yang lain.
8. *Composite* : Terdiri dari gabungan dua atau lebih pola yang berbeda.
9. *Lateral Pocket Loop* : Pola ini terdiri dari dua lengkungan yang terpisah. Ada dua titik dua delta.







Gambar 2.1 Contoh pola papillary ridge

Sekitar 60% orang memiliki pola sidik jari Loop. Sekitar 30% orang memiliki pola whorl, sekitar 5% berbentuk arch, dan 5% sisanya adalah bentuk-bentuk lainnya. Semua pola tersebut dapat dibedakan oleh mata biasa. Komputer dapat menganalisa garis-garis perubahan arah bentuk *ridge*, dengan kemampuan seperti mata manusia yang terlatih. Gambaran ukuran-ukuran karakteristik anatomi pola tersebut dapat digambarkan sebagai berikut:

Tabel 2.1 Variasi Pola *Ridge*

 <p>Ridge</p>	<i>Ridge</i>	Mempunyai ketegasan jarak ganda dari permulaan ke-akhir, sebagai lebar <i>ridges</i> satu dengan lainnya
 <p>Evading Ends</p>	<i>Evading Ends</i>	dua <i>ridge</i> dengan arah berbeda berjalan sejajar satu sama lain kurang dari 3mm.
 <p>Bifurcation</p>	<i>Bifurcation</i>	dua <i>ridge</i> dengan arah berbeda berjalan sejajar satu sama lain kurang dari 3mm.
 <p>Hook</p>	<i>Hook</i>	<i>Ridges</i> merobek; satu <i>ridges</i> tidaklah lebih panjang disbanding 3mm
 <p>Fork</p>	<i>Fork</i>	Dua <i>ridges</i> dihubungkan oleh seper tiga <i>ridges</i> tidak lebih panjang disbanding 3mm
 <p>Dot</p>	<i>Dot</i>	Bagian <i>ridges</i> adalah tidak lagi disbanding <i>ridges</i> yang berdekatan
 <p>Eye/Island</p>	<i>Eye</i>	<i>Ridges</i> merobek dan menggabungkan lagi di dalam 3mm

 <p>Eye/Island</p>	<i>Island</i>	<i>Ridges</i> merobek dan tidak bergabung lagi, kurang dari 3mm dan tidak lebih dari 6mm. Area yang terlampir adalah <i>Ridge</i> .
 <p>Enclosed Ridge</p>	<i>Enclosed Ridge</i>	<i>Ridges</i> tidak lebih panjang dibanding 6mm antara dua <i>ridges</i>
 <p>Enclosed Loop</p>	<i>Enclosed Loop</i>	yang tidak mempola menentukan pengulangan antar dua atau lebih <i>ridges</i> paralel
 <p>Specialty</p>	<i>Specialties</i>	<i>Rare ridge</i> membentuk seperti tanda tanya dan sangkutan pemotong

Area *papillary ridge* kadang-kadang dikenal sebagai *pattern area*. Masing-masing pola *papillary ridge* menghasilkan suatu bentuk pola area yang berbeda. Pusat gambar jari mencerminkan pola area, dikenal sebagai inti *core point*. Bagian *ridges* yang berwujud dua paralel yang berbeda mengelilingi pola area inti disebut *type lines*. Titik awal percabangan dua *ridge* disebut *delta*. Proses perpecahan sebuah garis menjadi dua garis *ridge* disebut *bifurcation*. Banyaknya persimpangan *ridge* di dalam pola area disebut suatu *ridge count*. Komputer Tomography dapat digunakan untuk mendeteksi titik-titik tersebut berdasarkan sumbu koordinat x-y. (EkoNugroho, 2009).

2.3 Sensor Sidik Jari

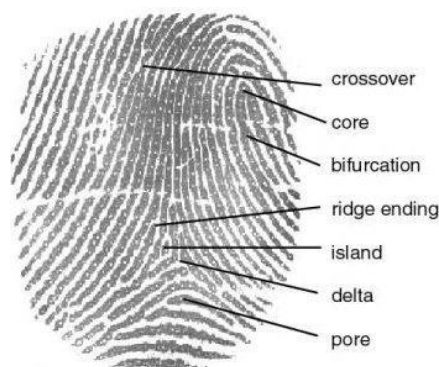
Sistem biometrika sidik jari merupakan sistem yang paling banyak digunakan saat ini, karena memiliki kecenderungan tingkat akurasi yang tinggi dan mudah diterapkan. Sifat yang dimiliki sidik jari antara lain:

1. *Perennial nature*, yaitu guratan-guratan pada sidik jari yang melekat pada kulit manusia seumur hidup.
2. *Immutability*, yaitu sidik jari seseorang tidak pernah berubah, kecuali mendapatkan kecelakaan yang serius.
3. *Individuality*, pola sidik jari adalah unik dan berbeda untuk setiap orang.



Gambar 2.2 Sensor Sidik Jari

Ciri khas sidik jari yang digunakan adalah sidik jari yang diidentifikasi dengan cara menganalisis detail dari guratan-guratan sidik jari yang dinamakan “*minutiae*” (Naslim Lathif, 2001). *Minutiae* berasal dari bahasa Inggris yang artinya barang tidak berarti atau rincian tidak penting dan terkadang diartikan sebagai detil. *Minutiae* sebenarnya merupakan rincian sidik jari yang tidak penting bagi kita, tetapi bagi sebuah mesin sidik jari itu adalah detil yang diperhatikan.



Gambar 2.3 Definisi sidik jari

Pemindai sidik jari saat ini sudah banyak digunakan, mulai dari absensi, sebagai *access control*, hingga sebagai identitas pribadi seperti pada SIM atau *passport*. Seperti halnya bagian tubuh yang lain, sidik jari terbentuk karena faktor genetik dan lingkungan. Kode genetik pada DNA memberi perintah untuk

terbentuknya janin yang secara spesifik membentuk hasil secara acak. Demikian juga halnya dengan sidik jari. Sidik jari memiliki bentuk unik bagi setiap orang. Artinya setiap orang memiliki bentuk sidik jari yang berbeda-beda meskipun terlahir kembar. Jadi, walaupun sidik jari terlihat seperti sama bila dilihat sekilas, buat penyidik terlatih atau dengan menggunakan *software* khusus akan terlihat perbedaannya.

Secara umum, sidik jari dapat dibedakan menjadi beberapa tipe menurut *Henry Classification System*, yaitu *loop pattern*, *whorl pattern*, dan *arch pattern*. Hampir $2/3$ manusia memiliki sidik jari dengan *loop pattern*, $1/3$ lainnya memiliki sidik jari dengan *whorl pattern*, dan hanya 5-10% yang memiliki sidik jari dengan *arch pattern*. Pola-pola sidik jari seperti ini yang digunakan untuk membedakan sidik jari secara umum. Namun, untuk mesin pembaca sidik jari, perbedaan seperti ini tidak cukup. Karena itulah, mesin sidik jari dilengkapi dengan pengenalan lain yang disebut *minutiae*.



Gambar 2.4 *Arch pattern*



Gambar 2.5 *Whorl Pattern*



Gambar 2.6 *Loop Pattern*

Untuk lebih jelasnya, *minutiae* pada sidik jari adalah titik-titik yang mengacu kepada *crossover* (persilangan dua garis), *core* (putar-balikan sebuah garis), *bifurcation* (percabangan sebuah garis), *ridge ending* (berhentinya sebuah garis), *island* (sebuah garis yang sangat pendek), *delta* (pertemuan dari tiga buah garis yang membentuk sudut) dan *pore* (percabangan sebuah garis yang langsung diikuti dengan menyatunya kembali percabangan tersebut sehingga membentuk 10 sebuah lingkaran kecil). Mesin pemindai sidik jari akan mencari titik-titik ini dan membuat pola dengan menghubungkan-hubungkan titik-titik ini. Pola yang didapat dengan menghubungkan titik-titik inilah yang nantinya akan digunakan untuk melakukan pencocokan bila ada jari yang menempel pada mesin sidik jari. Jadi, sebenarnya mesin sidik jari tidak mencocokkan pola yang didapat dari *minutiae-minutiae* ini. Mesin pemindai sidik jari bekerja dengan mengambil gambar dari sidik jari tersebut.

2.4 *Arduino*

Proyek *arduino* berawal di Ivre, italia pada tahun 2005. sekarang telah lebih dari 120.000 unit terjual sampai dengan 2010. Pendirinya adalah Massimo Banzi dan David Cuartiellez (*Sumber:www.academia.edu*. *Arduino* adalah pengendali mikro *single-board* yang bersifat *open source*, diturunkan dari *wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Perangkat kerasnya memiliki prosesor *atmel AVR* dan perangkat lunaknya memiliki bahasa

pemrograman sendiri. Saat ini *arduino* sangat populer di seluruh dunia. Banyak pemula yang belajar mengenal robotika dan elektronika lewat *arduino* karena mudah dipelajari. Tapi tidak hanya pemula, para *hobbyist* atau *professional* pun ikut senang mengembangkan aplikasi elektronik menggunakan *arduino*. Bahasa yang dipakai dalam *arduino* bukan *assembler* yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (*libraries*) yang terdapat pada *arduino*.

2.4.1 Kelebihan *Arduino*

Arduino menyederhanakan proses bekerja dengan mikrokontroler, sekaligus menawarkan berbagai macam kelebihan antara lain:

1. Murah

Papan (perangkat keras) *arduino* biasanya dijual relatif murah, dibandingkan dengan *platform* mikrokontroler pro lainnya. Jika ingin lebih murah lagi, tentu bisa dibuat sendiri dan itu sangat mungkin sekali karena semua sumber daya untuk membuat sendiri *arduino* tersedia lengkap di *website arduino* bahkan di *website-website* komunitas *arduino* lainnya. Tidak hanya cocok untuk *windows*, namun juga cocok bekerja di *linux*.

2. Sederhana dan mudah pemrogramannya

Perlu diketahui bahwa lingkungan pemrograman di *arduino* mudah digunakan untuk pemula, dan cukup fleksibel bagi mereka yang sudah tingkat lanjut. Untuk guru atau dosen, *arduino* berbasis pada lingkungan pemrograman *processing*, sehingga jika mahasiswa atau murid-murid terbiasa menggunakan *processing* tentu saja akan mudah menggunakan *arduino*.

3. Perangkat lunaknya *open source*

Perangkat lunak *arduino IDE (Integrated Development Environment)* dipublikasikan sebagai *open source*, tersedia bagi para pemrogram berpengalaman untuk pengembangan lebih lanjut. Bahasanya bisa

dikembangkan lebih lanjut melalui pustaka-pustaka C++ yang berbasis pada bahasa C untuk AVR (*Advance Virtual RISC*).

4. Perangkat kerasnya *open source*

Perangkat keras arduino berbasis mikrokontroler *ATMEGA8*, *ATMEGA168*, *ATMEGA328* dan *ATMEGA1280* (yang terbaru *ATMEGA2560*). Dengan demikian siapa saja bisa membuatnya (dan kemudian bisa menjualnya) perangkat keras *arduino* ini, termasuk *bootloader* tersedia langsung dari perangkat lunak *arduino IDE*-nya. Bisa juga menggunakan *breadboard* untuk membuat perangkat *arduino* beserta periferal-periferal lain yang dibutuhkan.

2.4.2 Soket *USB (Universal Serial Bus)*

Soket *USB* adalah soket kabel *USB* yang disambungkan ke komputer atau laptop, yang berfungsi untuk mengirimkan program ke *arduino* dan juga sebagai *port* komunikasi serial.

2.4.3 *Input* atau *Output Digital* dan *Input Analog*

Input atau *output* digital (*digital pin*) adalah pin-pin untuk menghubungkan *arduino* dengan komponen atau rangkaian digital, contohnya: jika ingin membuat *LED (Light Emitting Diode)* berkedip, *LED* tersebut bisa dipasang pada salah satu pin *input* atau *output* digital dan *ground*. komponen lain yang menghasilkan *output* digital atau menerima *input* digital bisa disambungkan ke pin-pin ini. *Input* analog (*analog pin*) adalah pin-pin yang berfungsi untuk menerima sinyal dari komponen atau rangkaian analog, contohnya: potensiometer, sensor suhu, sensor cahaya, dan lain-lain.

2.4.4 *Catu Daya*

Pin *catu daya* adalah pin yang memberikan tegangan untuk komponen atau rangkaian yang dihubungkan dengan *arduino*. Pada bagian *catu daya* ini pin V_{in} dan *Reset*. V_{in} digunakan untuk memberikan tegangan langsung kepada *arduino* tanpa

melalui tegangan pada *USB* atau adaptor, sedangkan *reset* adalah pin untuk memberikan sinyal reset melalui tombol atau rangkaian eksternal.

2.4.5 Baterai atau Adaptor

Socket baterai atau adaptor digunakan untuk menyuplai arduino dengan tegangan dari baterai atau adaptor 9V pada saat arduino sedang tidak disambungkan kekomputer. Jika *arduino* sedang disambungkan kekomputer dengan *USB*, *Arduino* mendapatkan suplai tegangan dari *USB*, Jika tidak perlu memasang baterai atau adaptor pada saat memprogram arduino.

2.5 *Arduino Mega 2560*



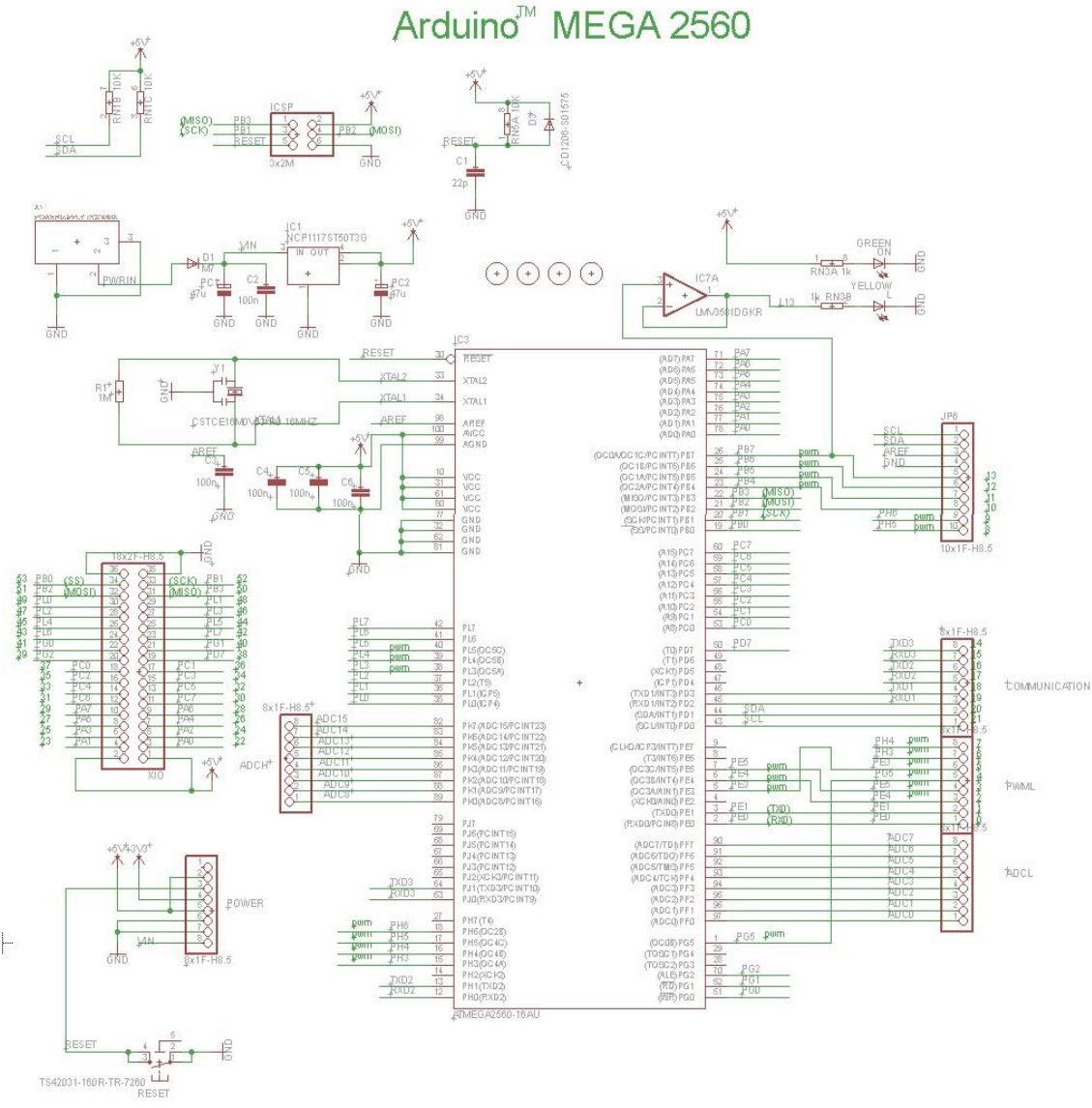
Gambar 2.7 Arduino Mega 2560

(Sumber : www.arduino.cc)

Arduino mega 2560 adalah papan mikrokontroler *Atmega 2560* berdasarkan (*datasheet*) memiliki 54 digital pin *input* atau *output* (dimana 15 pin dapat digunakan sebagai *output PWM* atau *Pulse Width Modulation*), 16 *analog input*, 4 *UART (Universal Asynchronous Receiver/Transmitter)*, *osilator* kristal 16 MHz, koneksi *USB*, *jack* listrik, *header ICSP (In-Circuit Serial Programing)*, dan tombol *reset*. Semuanya diperlukan untuk mendukung mikrokontroler, hanya menghubungkannya ke komputer dengan kabel *USB* atau *power* dengan adaptor *AC (Alternating Current) – DC (Direct Current)* atau baterai.

2.5.1 Schematic Arduino Mega 2560

Adapun gambar *schematic* dari rangkaian arduino mega 2560, dapat dilihat pada Gambar 2.2 dibawah ini :



Gambar 2.8 Schematic Arduino Mega 2560

(Sumber : www.arduino.cc)

2.5.2 Spesifikasi

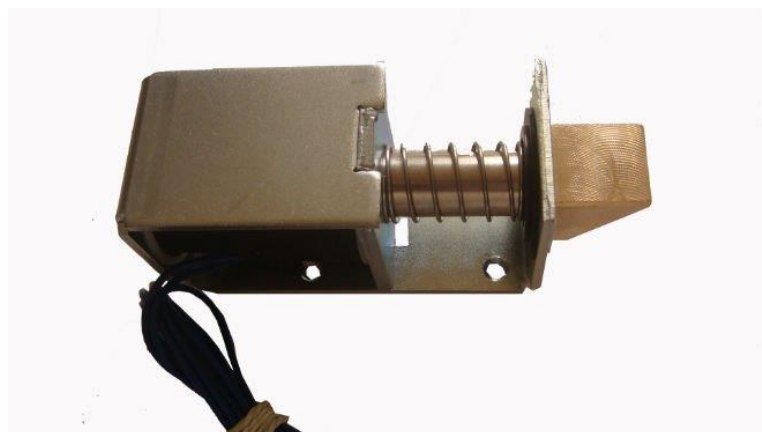
Adapun spesifikasi data-data mengenai arduino mega 2560, dapat dilihat pada Tabel 2.2 berikut :

Tabel 2.2 Spesifikasi *Arduino Mega 2560*

<i>Microcontroller</i>	Atmega 2560
<i>Operating Voltage</i>	5V
<i>Input Voltage (recommended)</i>	7-12V
<i>Input Voltage (limits)</i>	6-20V
<i>Digital I/O Pins</i>	54 (of which 15 provide PWM output)
<i>Analog Input Pins</i>	16
<i>DC Current per I/O Pin</i>	40 mA
<i>DC Current for 3.3V Pin</i>	50 mA
<i>Flash Memory</i>	256 KB of which 8 KB used by bootloader
<i>SRAM</i>	8 KB
<i>EEPROM</i>	4 KB
<i>Clock Speed</i>	16 MHz
Power	12 V

2.6 Solenoid Door Lock

Solenoid Door Lock ini berfungsi sebagai aktuator. Didalam solenoid terdapat kawat yang melingkar pada inti besi. Sistem kerja solenoid ini adalah NC (Normalu Close). Katup Solenoid akan tertarik jika ada tegangan dan sebaliknya katup solenoid akan memanjang jika tidak ada tegangan.



Gambar 2.9 Solenoid door lock
(Sumber: <http://www.waferstar.com>)

Tabel 2.3 Spesifikasi *Solenoid Doorlock*

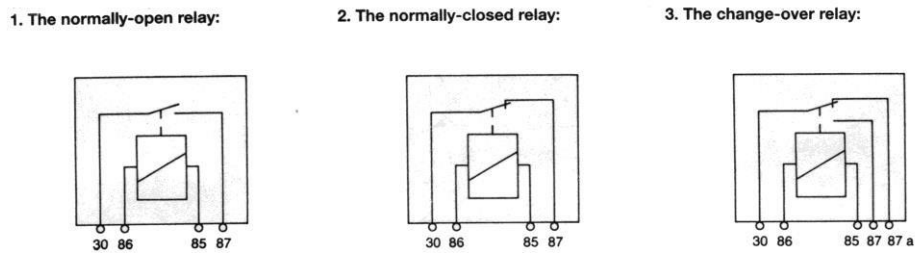
<i>Product</i>	<i>Door Solenoid 12v</i>
<i>Material</i>	<i>Metal</i>
<i>Input Voltage (recommended)</i>	DC 12V 2A
<i>Color</i>	<i>Silver</i>
<i>Cable Length</i>	24 cm

2.7 Relay

Relay adalah komponen elektronika berupa saklar elektronik yang digerakkan oleh arus listrik. Secara prinsip, relay merupakan saklar dengan lilitan kawat pada batang besi (solenoid) di dekatnya. Ketika solenoid dialiri arus listrik, tuas akan tertarik karena adanya gaya magnet yang terjadi pada solenoid sehingga kontak saklar akan menutup. Pada saat arus dihentikan, gaya magnet akan hilang, tuas akan kembali ke posisi semula dan kontak saklar kembali terbuka. Relay biasanya digunakan untuk menggerakkan arus/tegangan yang besar (misalnya peralatan listrik 4 ampere AC 220 V) dengan memakai arus/tegangan yang kecil (misalnya 0.1 ampere 12 Volt DC). Relay yang paling sederhana ialah relay elektromekanis yang memberikan pergerakan mekanis saat mendapatkan energi listrik.

Dalam pemakaiannya biasanya relay yang digerakkan dengan arus DC dilengkapi dengan sebuah dioda yang di-paralel dengan lilitannya dan dipasang terbaik yaitu anoda pada tegangan (-) dan katoda pada tegangan (+). Ini bertujuan untuk mengantisipasi sentakan listrik yang terjadi pada saat relay berganti posisi dari on ke off agar tidak merusak komponen di sekitarnya.

Kebanyakan relay yang ditemui hanya memiliki tiga kondisi, yakni normally open (NO), normally close (NC), dan change-over (CO). Kondisi NO akan terjadi ketika relay diberi tegangan maka saklar akan terbuka. Kondisi NC merupakan kebalikan dari NO dimana saklar akan tertutup ketika relay diberi tegangan. Sedangkan kondisi CO merupakan kondisi dimana relay akan mengubah posisi saklar ketika diberi tegangan.



Gambar 2.10 Kondisi relay ketika *normally open* (NO), *normally close* (NC), dan *change-over* (CO)

(Sumber : <http://www3.telus.net/cbradley/relay3.JPG>)

2.8 Push Button

Push Button adalah saklar tekan yang berfungsi sebagai pemutus atau penyambung arus listrik dari sumber arus ke beban listrik. Suatu sistem saklar tekan push button terdiri dari saklar tekan start, stop reset dan saklar tekan untuk emergency. Push button memiliki kontak NC (normally close) dan NO (normally open).

Prinsip kerja Push Button adalah apabila dalam keadaan normal tidak ditekan maka kontak tidak berubah, apabila ditekan maka kontak NC akan berfungsi sebagai stop (memberhentikan) dan kontak NO akan berfungsi sebagai start (menjalankan)

2.9 Mengenal *Visual Basic.Net*

Pada zaman dahulu ada sebuah bahasa pemrograman yang diberi nama Basic (Beginner's All-purpose Symbolic Instruction Code). Sesuai dengan namanya, Basic ditujukan sebagai bahasa yang paling sederhana bagi mereka yang tidak terlalu familiar dengan dunia pemrograman. Pada tahun 1991 Microsoft mengeluarkan Visual Basic, pengembangan dari Basic yang berubah dari sisi pembuatan antarmukanya. Visual Basic sampai sekarang masih menjadi salah satu bahasa pemrograman terpopuler di dunia.

Pada akhir tahun 1999, teknologi.Net diumumkan Microsoft memosisikan teknologi tersebut sebagai platform untuk membangun XML Web *services*. XML Web *services* memungkinkan aplikasi tipe apa pun dapat berjalan pada sistem computer dengan tipe manapun dan dapat mengambil data yang tersimpan pada server dengan tipe apa pun melalui internet.

Microsoft Visual Basic.Net adalah visual basic yang direkayasa kembali untuk digunakan pada platform.Net sehingga aplikasi yang dibuat menggunakan Visual basic.Net dapat berjalan pada sistem computer apa pun, dan dapat mengambil data dari server dengan tipe apa pun asalkan terinstal.Net framework.

Berikut ini perkembangan Visual Basic. Net :

- a. Visual Basic. Net 2002 (VB 7.0)
- b. Visual Basic. Net 2003 (VB 7.1)
- c. Visual Basic. 2005 (VB 8.0)
- d. Visual Basic. 2008 (VB 9.0)
- e. Visual Basic. 2010 (VB 10.0)
- f. Visual Basic. 2012 (VB 11.0)
- g. Visual Basic. 2013

Pada umumnya Visual basic. Net terpaket dalam Visual Studio .Net. Pada distribusinya, terdapat berbagai versi Visual Studio .Net yaitu versi Professional, Premium dan yang paling lengkap adalah versi berbayar dengan harga yang sangat mahal. Visual Studio .Net Professional dibanderol dengan harga \$499, versi premium dihargai \$5.468 sedangkan versi Ultimate harganya adalah \$11.899.

Kelebihan Visual Basic .Net antara lain:

1. Sederhana dan mudah dipahami.
2. Mendukung GUI.
3. Menyederhanakan *deployment*.
4. Menyederhanakan pengembangan perangkat lunak.
5. Mendukung penuh OOP.
6. Mempermudah pengembangan aplikasi berbasis web.
7. Migrasi ke VB .Net dapat dilakukan dengan mudah.
8. Banyak digunakan oleh *programmer-programmer* di seluruh dunia.

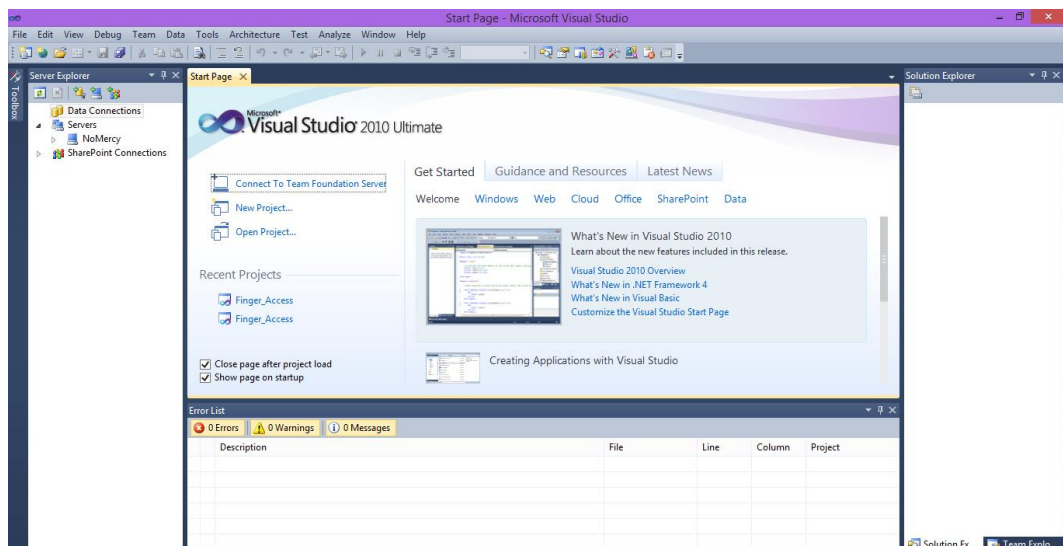
2.9.1 Lingkungan kerja *Visual Basic.Net*

Pada saat pertama kali dijalankan *Visual Basic 2010 Ultimate*, akan menampilkan sebuah jendela *Splash Visual Studio 2010 Ultimate* pada gambar 2.11.



Gambar 2.11 Tampilan jendela *Splash Visual Studio 2010 Ultimate*

Setelah jendela *Splash Visual Studio 2010 Ultimate* muncul kemudian akan keluar sebuah start page microsoft visual studio seperti gambar 2.12

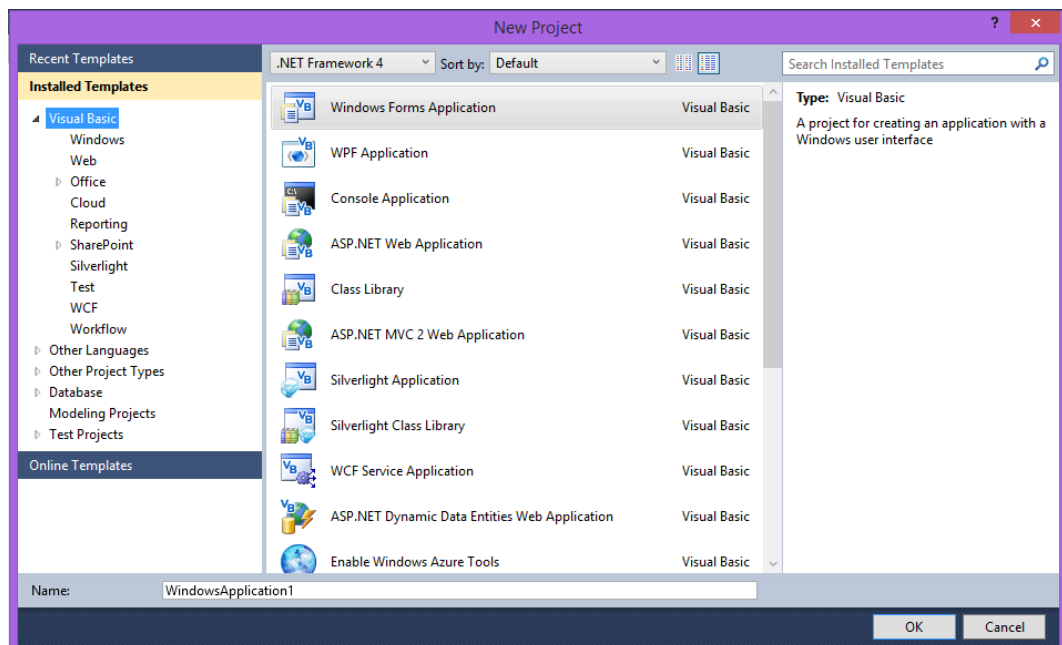


Gambar 2.12 Tampilan Start Page Microsoft Visual Studio

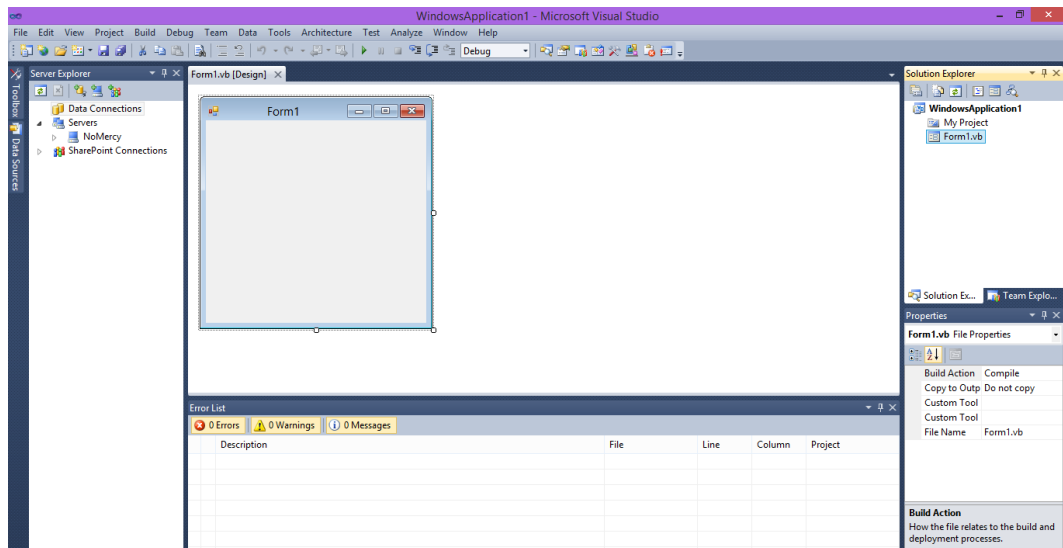
Untuk membuka proyek yang ada gunakan tombol *Open Project* atau langsung mengklik pada daftar proyek yang ditampilkan sedangkan untuk membuat sebuah proyek baru baru, klik tombol *New Project*.

Setelah itu akan muncul kotak dialog *New project*. Pada kotak pilih *Other Languages > Visual Basic > Windows > Windows Form Application*. Untuk memberi nama proyek dapat dilakukan pada bagian *name*, tentukan posisi penyimpanan *file-file* proyek dan tentukan nama *Salution*-nya dan tekan OK (gambar 2.3). Selanjutnya muncul Visual Basic 2010 IDE tempat untuk membangun aplikasi visual basic.

Pada IDE Visual Basic 2010 untuk *windows application default* telah terdapat sebuah *form*. Form tersebut bernama Form1. Pada *form* inilah tempat meletakkan kontrol-kontrol atau komponen untuk membuat sebuah aplikasi *windows form* dan dari kontrol-kontrol inilah yang biasanya disebut dengan GUI. Pada IDE Visual Studio 2010 terdapat *menu bar*, *toolbar*, *salution explorer*, dan *properties windows*.



Gambar 2.13 Kotak Dialog New Project

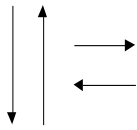
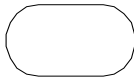
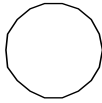
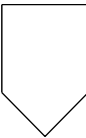




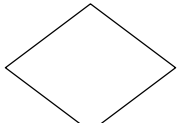

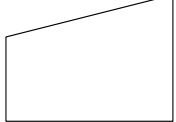
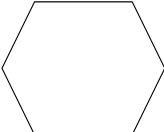

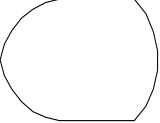

Gambar 2.14 IDE Visual Basic 2010

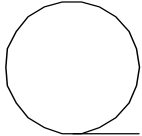


2.10 Simbol – Simbol Flochart

Flowchart disusun dengan symbol - simbol. Simbol ini dipakai sebagai alat bantu menggambarkan proses di dalam program.

Tabel 2.3 Simbol – simbol flowchart.

No	Simbol	Keterangan
1.		Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.
2.		Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan
3.		Yaitu simbol untuk keluar – masuk atau penyambungan proses dalam lembar / halaman yang sama.
4.		Yaitu simbol untuk keluar – masuk atau penyambungan proses pada lembar / halaman yang berbeda.

5.		Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer
6.		Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer
7.		Simbol pemilihan proses berdasarkan kondisi yang ada.
8.		Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya
9.		Simbol untuk pemasukan data secara manual on-line keyboard
10.		Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage.
11.		Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure
12.		Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.
13.		Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.

14.		Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
15.		Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
16.		Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

2.11 Bahasa C

Sejarah Perkembangan Bahasa C Berasal dari bahasa BCPL (Basic Combined Programming Language) oleh MARTIN RICHARD, Cambridge tahun 1967. KEN THOMPSON membuat bahasa B untuk dipakai pada komputer DEC PDP -7 di bawah sistem operasi UNIX pada Bell laboratory, Murray Hill, New Jersey tahun 1970. Bahasa B merupakan suatu bahasa pemrograman yang tidak memiliki jenis suatu data seperti halnya PL/M. Berdasarkan gambaran bahasa B, DENNIS RITCHIE menulis bahasa C. Nama C diambil berdasarkan urutan sesudah B dari bahasa BCPL.

Tujuan bahasa C pada mulanya untuk membentuk suatu sistem operasi yang akan digunakan pada mesin komputer DEC PDP-11 yang baru. Pada tahun 1975, sistem operasi UNIX versi 6 dan bahasa C mulai diberikan kepada Universitas maupun Akademi. Dan pada tahun 1979, system operasi UNIX versi 7 dikeluarkan dengan bahasa C.

Sistem operasi ini (versi 7) seluruhnya ditulis dalam bahasa C. Pada 1978 Dennis Ritchie dan Brian Kernighan kemudian mempublikasikan buku The C Programming Language yang semakin memperluas pemakaiannya dan dijadikan standar oleh ANSI (American National Standard Institute) pada tahun 1989. C kemudian dikembangkan lagi oleh Bjarne Stroustrup menjadi C++ (1986). C dan/atau C++ banyak digunakan (sehingga menjadi „standar“) sebagai bahasa pemrograman untuk membuat sistem operasi.

Untuk dapat memahami bagaimana suatu program ditulis, maka struktur dari program harus dimengerti terlebih dahulu. Tiap bahasa komputer mempunyai struktur program yang berbeda. Jika struktur dari program tidak diketahui, maka akan sulit bagi pemula untuk memulai menulis suatu program dengan bahasa yang bersangkutan. Struktur dari program C terdiri dari koleksi satu / lebih fungsi-fungsi. Fungsi pertama yang harus ada di program C sudah ditentukan namanya, yaitu bernama `main()` . Suatu fungsi di dalam program C dibuka dengan kurung kurawal buka “{” dan ditutup dengan kurung kurawal tutup “}”.

Di antara kurung kurawal 7 dapat dituliskan statemen-statemen program C dan pada setiap statemen diakhiri dengan tanda titik koma “;”. Bahasa C dikatakan sebagai bahasa pemrograman terstruktur, karena strukturnya menggunakan fungsi-fungsi sebagai program-program bagian (subroutine). Fungsi-fungsi selain fungsi utama merupakan program-program bagian. Fungsi-fungsi ini dapat ditulis setelah fungsi utama atau diletakkan pada file pustaka dan akan dipakai di suatu program, maka nama judulnya (header file) harus dilibatkan di dalam program yang menggunakan preprocessor directive `#include` .

Header file merupakan file yang berisi dengan prototype (judul, nama, sintak) dari sekumpulan fungsi-fungsi pustaka tertentu. Jadi file ini hanya berisi dengan prototype dari fungsi-fungsi pustaka, sedangkan fungsi-fungsi pustakanya sendiri disimpan dalam file pustaka (library file dengan nama extension file -nya adalah `.LIB`).