

BAB II TINJAUAN PUSTAKA

2.1 Sistem Perangkat Lunak

2.1.1 Citra

Secara harfiah, citra (*image*) adalah gambar pada bidang dwimatra (dua dimensi). Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek, dan objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh alat-alat optik, misalnya mata pada manusia, kamera, pemindai (*scanner*), dsb, sehingga bayangan objek yang disebut citra tersebut terekam.

Citra yang dimaksud disini adalah “citra diam” (*still image*). Citra diam adalah citra tunggal yang tidak bergerak. Untuk selanjutnya, citra diam disebut citra saja. Sedangkan “citra bergerak” (*moving image*) adalah rangkaian citra diam yang ditampilkan secara beruntun (*sequential*) sehingga memberi kesan pada mata kita sebagai gambar yang bergerak. Setiap citra didalam rangkaian itu disebut *frame*. Gambar-gambar yang tampak pada film layar lebar atau televisi pada dasarnya terdiri atas ratusan sampai ribuan *frame*.

Citra terdiri dari dua jenis, yaitu citra kontinu dan citra diskrit. Citra kontinu dihasilkan dari sistem optik yang menerima sinyal analog, misalnya mata manusia dan kamera analog. Citra diskrit dihasilkan melalui proses digitalisasi terhadap citra kontinu. Beberapa sistem optik dilengkapi dengan fungsi digitalisasi sehingga mampu menghasilkan citra diskrit, misalnya kamera digital dan *scanner*. Citra diskrit disebut juga citra digital.

Secara matematis, fungsi intensitas cahaya pada bidang dwimatra disimbolkan dengan $f(x,y)$ yang dalam hal ini :

(x,y) : koordinat pada bidang dwimatra (dua dimensi)

$f(x,y)$: intensitas cahaya (brightness) pada titik (x,y)

Karena cahaya merupakan bentuk energi, maka intensitas cahaya bernilai antara 0 sampai tidak berhingga,

$$0 \leq f(x,y) < \infty$$

Nilai $f(x,y)$ sebenarnya adalah hasil kali dari :

- $i(x,y)$, yaitu jumlah cahaya yang berasal dari sumbernya (*illumination*) nilainya antara 0 sampai tak berhingga, dan
- $r(x,y)$, derajat kemampuan objek memantulkan cahaya (*reflection*), nilainya antara 0 dan 1.

Proses pembentukan intensitas cahaya adalah sebagai berikut. Sumber cahaya menyinari permukaan objek. Jumlah pancaran iluminasi (*illumination*) cahaya yang diterima objek pada koordinat (x,y) adalah $i(x,y)$. Objek memantulkan cahaya yang diterimanya dengan derajat pantulan $r(x,y)$. Hasil kali antara $i(x,y)$ dengan $r(x,y)$ menyatakan intensitas cahaya pada koordinat (x,y) yang ditangkap oleh sensor visual pada sistem optik.

Jadi,

$$f(x,y) = i(x,y) \cdot r(x,y)$$

yang dalam hal ini,

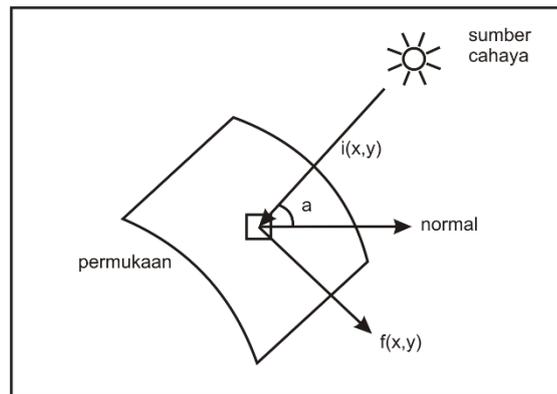
$$0 \leq i(x,y) < \infty$$

$$0 \leq r(x,y) \leq 1$$

sehingga,

$$0 \leq f(x,y) < \infty \quad \dots(1)$$

Untuk mempermudah pemahaman dapat dilihat pada gambar 2.1.



Gambar 2.1. Pembentukan citra.

Nilai $i(x,y)$ ditentukan oleh sumber cahaya, sedangkan $r(x,y)$ ditentukan oleh karakteristik objek di dalam gambar. Nilai $r(x,y) = 0$ mengindikasikan penyerapan total, sedangkan $r(x,y) = 1$ menyatakan pemantulan total. Jika permukaan mempunyai derajat pemantulan 0, maka fungsi intensitas cahaya $f(x,y)$ juga 0. Sebaliknya, jika permukaan mempunyai derajat pemantulan 1, maka fungsi intensitas cahaya sama dengan iluminasi yang diterima oleh permukaan tersebut.

Intensitas f dari gambar hitam-putih pada titik (x,y) disebut derajat keabuan (*gray level*), yang dalam hal ini derajat keabuannya bergerak dari hitam ke putih, sedangkan citranya disebut citra hitam putih (*grayscale image*) atau citra monokrom (*monochrome image*).

Citra hitam putih disebut juga citra satu kanal (*channel*), karena warnanya hanya ditentukan oleh satu fungsi intensitas saja. Citra berwarna (*color image*) dikenal dengan nama citra spektral, karena warna pada citra disusun oleh tiga komponen warna yang disebut komponen RGB, yaitu merah (*red*), hijau (*green*), dan biru (*blue*). Intensitas suatu titik pada citra berwarna merupakan kombinasi dari tiga intensitas derajat keabuan merah ($f_{\text{merah}}(x,y)$), hijau ($f_{\text{hijau}}(x,y)$), dan biru ($f_{\text{biru}}(x,y)$).

2.1.2 DIGITALISASI CITRA

Agar dapat diolah dengan komputer, maka suatu citra harus direpresentasikan secara numerik dengan nilai-nilai diskrit. Representasi citra dari fungsi kontinu menjadi nilai-nilai diskrit disebut digitalisasi. Citra yang dihasilkan inilah yang disebut citra digital (*digital image*). Pada umumnya citra digital berbentuk empat persegi panjang, dan dimensi

si ukurannya dinyatakan sebagai tinggi kali lebar.

Citra digital yang tingginya N , lebarnya M , dan memiliki L derajat keabuan dapat dianggap sebagai fungsi :

$$f(x, y) \begin{cases} 0 \leq x \leq M \\ 0 \leq y \leq N \\ 0 \leq f \leq L \end{cases}$$

Citra digital yang berukuran $N \times M$ lazim dinyatakan dengan matriks yang berukuran N baris dan M kolom sebagai berikut :

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M) \\ f(1,0) & f(1,1) & \dots & f(1,M) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \quad \dots(2)$$

Indeks baris (i) dan indeks kolom (j) menyatakan suatu koordinat titik pada citra, sedangkan $f(i, j)$ merupakan intensitas derajat keabuan pada titik (i, j).

Masing-masing elemen pada citra digital (berarti elemen matriks) disebut *image element*, *picture element* atau *pixel*. Jadi citra yang berukuran $N \times M$ mempunyai NM buah *pixel*. Sebagai contoh, sebuah citra yang berukuran 256×256 *pixel* yang memiliki 65536 buah *pixel* direpresentasikan secara numerik dengan matriks yang terdiri dari 256 baris (diindeks dari 0 sampai 255) dan 256 kolom (diindeks dari 0 sampai 255) seperti contoh berikut :

$$\begin{bmatrix} 0 & 134 & 145 & \dots & \dots & 231 \\ 0 & 167 & 201 & \dots & \dots & 197 \\ 220 & 187 & 189 & \dots & \dots & 120 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 221 & 219 & 213 & \dots & \dots & 156 \end{bmatrix}$$

Pixel pertama pada koordinat (0, 0) mempunyai nilai intensitas 0 yang berarti warna *pixel* tersebut hitam, *pixel* kedua pada koordinat (0, 1) mempunyai intensitas 134 yang berarti warnanya antara hitam dan putih, dst.

Proses digitalisasi citra ada 2 tahap, yaitu :

1. Digitalisasi spasial (x, y), sering disebut *sampling*
2. Digitalisasi intensitas f (x, y), sering disebut kuantisasi.

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

2.1.3 CITRA WARNA

a. Citra Warna 8 bit

Setiap *pixel* dari citra warna (8 bit) hanya diwakili oleh 8 bit dengan jumlah warna maksimum yang dapat digunakan adalah 256 warna. Ada dua jenis citra warna 8 bit. Pertama, citra warna 8 bit dengan menggunakan palet warna 256 dengan setiap paletnya memiliki pemetaan nilai (*colormap*) RGB tertentu. Model ini lebih sering digunakan. Kedua, setiap *pixel* memiliki format 8 bit seperti yang ditunjukkan pada gambar 2.2 Gambar 2.3 merupakan contoh dari citra warna 8 bit.

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
R	R	R	G	G	G	B	B

Gambar 2.2. 8 bit *true color*.



Gambar 2.3. Citra warna 8 bit.

b. Citra warna 16 bit

Citra warna 16 bit (biasanya disebut sebagai citra *highcolor*) dengan setiap pixelnya diwakili dengan 2 *byte memory* (16 bit).

Warna 16 bit memiliki 65.536 warna. Dalam formasi bitnya, nilai merah dan biru mengambil tempat di 5 bit di kanan dan dikiri. Komponen hijau memiliki 5 bit ditambah 1 bit ekstra. Pemilihan komponen hijau dengan deret 6 bit dikarenakan penglihatan manusia lebih sensitif terhadap warna hijau. Deret dari citra *high color* ditunjukkan oleh gambar 2.4 dan gambar 2.5 adalah contoh citra 16 bit.

Bit-15	Bit-14	Bit-13	Bit-12	Bit-11	Bit-10	Bit-9	Bit-8	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
R	R	R	R	R	G	G	G	G	G	G	B	B	B	B	B

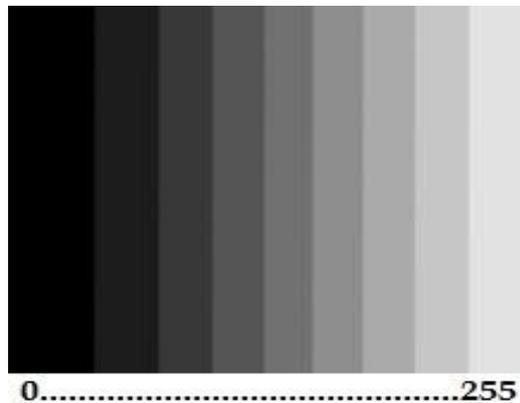
Gambar 2.4. 16 bit *high color*.



Gambar 2.5. Citra warna 16 bit.

2.1.4 Derajat Keabuan (Grayscale)

Citra keabuan adalah citra yang setiap pixelnya hanya berisi nilai intensitas dengan kata lain citra ini hanya memiliki 1 dasar warna yaitu abu-abu, hanya saja intensitasnya yang berbeda. Pada citra digital 8 bit, setiap warna dalam satu pixel yaitu merah, hijau, dan biru memiliki nilai yang sama. Sedangkan sebagai pembeda antara pixel 1 dan lainnya adalah intensitasnya dimana untuk warna lebih terang (putih) itu nilainya mendekati 255 dan warna lebih gelap itu nilainya mendekati 0.



Gambar 2.6. Derajat Keabuan dalam skala 8 bit.

Adapun cara mengkonversi citra RGB menjadi Grayscale adalah sebagai berikut.

$$f(x, y) = \frac{(R(x,y)+G(x,y)+B(x,y))}{3} \quad \dots(3)$$

2.1.5 *Thresholding*

Thresholding merupakan teknik dari segmentasi gambar yang paling sederhana. Teknik ini merupakan teknik yang membagi gambar menjadi 2 warna, yang biasanya merupakan warna hitam dan putih. Untuk menggunakan teknik ini diperlukan teknik *grayscale* terlebih dahulu, artinya ketika gambar asli telah berubah menjadi gambar *grayscale*, maka akan di tarik sebuah titik yang merupakan nilai dari suatu intensitas *grayscale*. Diumpamakan nilai intensitas yang diambil adalah 100, maka nilai intensitas *grayscale* yang di atas seratus akan kita beri nilai 255 dan nilai yang dibawah 100 akan kita beri nilai 0. Otomatis hal ini akan memberikan warna kontras pada gambar tersebut. Nilai 100 inilah yang merupakan titik tengah atau yang di sebut nilai *threshold*.

Nilai *threshold* dapat kita tentukan atau kita kalibrasi nilainya agar pembagian warna terhadap objek pada gambar dapat terbagi sesuai dengan kebutuhan seperti yang terlihat pada gambar 2.9.



Gambar 2.7. Citra biner.

Metode thresholding secara umum dibagi menjadi dua, yaitu :

1. Thresholding global

Thresholding dilakukan dengan mempartisi histogram dengan menggunakan sebuah *threshold* (batas ambang) global T , yang berlaku untuk seluruh bagian pada citra.

2. Thresholding adaptif

Thesholding dilakukan dengan membagi citra menggunakan beberapa sub citra. Lalu pada setiap sub citra, segmentasi dilakukan dengan menggunakan threshold yang berbeda.

Thresholding diimplementasikan setelah dilakukan proses perbaikan kontras citra menggunakan fungsi Contrast-limited adaptive histogram equalization (CLAHE). Thresholding dikatakan global jika nilai threshold T hanya bergantung pada $f(x,y)$, yang melambangkan tingkat keabuan pada titik (x,y) dalam suatu citra.

2.2 Sistem Perangkat Keras

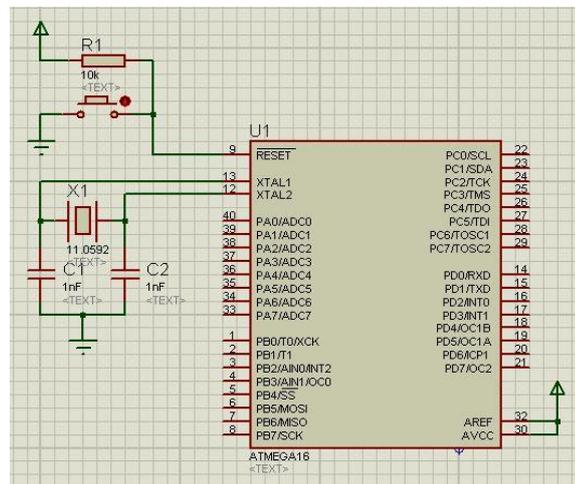
2.2.1 Mikrokontroler

Mikrokontroler adalah salah satu dari bagian dasar dari suatu sistem komputer. Meskipun mempunyai bentuk yang jauh lebih kecil dari suatu komputer pribadi dan komputer *mainframe*, mikrokontroler dibangun dari elemen-elemen dasar yang sama. Seperti umumnya komputer, mikrokontroler adalah alat yang mengerjakan instruksi-instruksi (program) yang diberikan kepadanya. Program ini menginstruksikan komputer untuk melakukan jalinan yang panjang dari aksi-aksi sederhana untuk melakukan tugas yang lebih kompleks.

Mikrokontroler AVR (*Alf and Vegard's Risc processor*) standar memiliki arsitektur 8 bit, dimana semua instruksi dikemas dalam kode 32-bit, dan sebagian besar instruksi dieksekusi dalam 1 (satu) siklus clock. AVR berteknologi RISC (*Reduced Instruction Set Computing*), sedangkan seri MCS51 berteknologi CISC (*Complex Instruction Set Computing*). AVR dapat dikelompokkan menjadi empat kelas, yaitu keluarga ATtiny, keluarga AT86RFxx, keluarga ATmega, dan keluarga AT86RFxx. Pada dasarnya, yang membedakan masing-masing kelas adalah memori, peripheral, dan fungsinya.

Salah satu tipe mikrokontroler AVR yang memiliki fitur memuaskan adalah ATmega 16. Mikrokontroler ini telah banyak digunakan pada penelitian robot bergerak otonom dengan alasan *downloader* yang mudah dibuat dan digunakan, serta ukuran *flash memory* yang lumayan besar, yakni 16 KB. Karena kelebihan

yang dimilikinya, maka mikrokontroler ATmega 16 dipilih sebagai unit kontroler untuk mengontrol motor servo yang akan digunakan pada penelitian ini dan untuk mengetahui susunan standar pin dari ATmega 16 beserta sistem minimumnya dapat dilihat pada gambar 2.13.



Gambar 2.8 Sistem minimum ATmega 16.

Untuk merancang sistem minimum sebuah mikrokontroller dibutuhkan beberapa komponen tambahan untuk membuat mikrokontroller tersebut menjadi terintegrasi. Komponen yang dibutuhkan dalam aplikasi rangkaian sistem minimum tersebut yaitu rangkaian pembangkit frekuensi kerja mikrokontroller yang diaplikasikan dengan kristal 11,0592 MHz. Sebuah sistem reset yang aktif *low* untuk memulai siklus kerja baru pada setiap perubahan interaksi kerja mikrokontroller.

2.2.2 Motor Servo

Motor servo adalah sebuah motor DC yang dilengkapi rangkaian kendali dengan sistem closed feedback yang terintegrasi dalam motor tersebut. Pada motor servo posisi putaran sumbu (*axis*) dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo.



Gambar 2.9. Motor Servo

2.2.2.1 Konstruksi Motor Servo

Motor servo disusun dari sebuah motor DC, gearbox, variabel resistor (VR) atau potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas maksimum putaran sumbu (axis) motor servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang pada pin kontrol motor servo. Motor servo adalah motor yang mampu bekerja dua arah (CW dan CCW) dimana arah dan sudut pergerakan rotornya dapat dikendalikan dengan memberikan variasi lebar pulsa (duty cycle) sinyal PWM pada bagian pin kontrolnya.



Gambar 2.10. Konstruksi Motor Servo

2.2.2.2 Jenis Motor Servo

1. Motor Servo Standar 180°

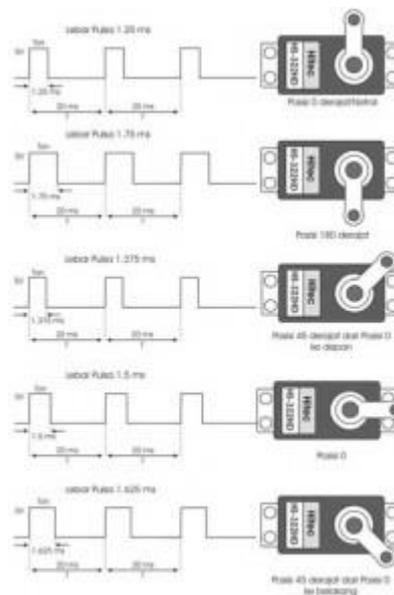
Motor servo jenis ini hanya mampu bergerak dua arah (CW dan CCW) dengan defleksi masing-masing sudut mencapai 90° sehingga total defleksi sudut dari kanan – tengah – kiri adalah 180° .

2. Motor Servo Continuous

Motor servo jenis ini mampu bergerak dua arah (CW dan CCW) tanpa batasan defleksi sudut putar (dapat berputar secara kontinyu).

2.2.2.3 Pulsa Kendali Motor Servo

Operasional motor servo dikendalikan oleh sebuah pulsa selebar ± 20 ms, dimana lebar pulsa antara 0.5 ms dan 2 ms menyatakan akhir dari range sudut maksimum. Apabila motor servo diberikan pulsa dengan besar 1.5 ms mencapai gerakan 90° , maka bila kita berikan pulsa kurang dari 1.5 ms maka posisi mendekati 0° dan bila kita berikan pulsa lebih dari 1.5 ms maka posisi mendekati 180° .



Gambar 2.11. Pulsa Kendali Motor Servo

Motor Servo akan bekerja secara baik jika pada bagian pin kontrolnya diberikan sinyal PWM dengan frekuensi 50 Hz. Dimana pada saat sinyal dengan frekuensi 50 Hz tersebut dicapai pada kondisi Ton duty cycle 1.5 ms, maka rotor

dari motor akan berhenti tepat di tengah-tengah (sudut 0°/ netral). Pada saat Ton duty cycle dari sinyal yang diberikan kurang dari 1.5 ms, maka rotor akan berputar ke berlawanan arah jarum jam (Counter Clock wise, CCW) dengan membentuk sudut yang besarnya linier terhadap besarnya Ton duty cycle, dan akan bertahan diposisi tersebut. Dan sebaliknya, jika Ton duty cycle dari sinyal yang diberikan lebih dari 1.5 ms, maka rotor akan berputar searah jarum jam (Clock Wise, CW) dengan membentuk sudut yang linier pula terhadap besarnya ton duty cyc, dan bertahan diposisi tersebut.

2.3 USB Atmel AVR ISP K-125R

K-125R merupakan USB Atmel AVR ISP Programmer / downloader untuk semua Tipe AVR. Berbeda dengan tipe Nue-125, K-125 yang dilengkapi dengan komunikasi USB to Serial TTL.

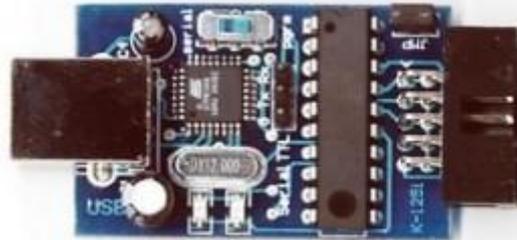
Alat ini membantu dalam memprogram Mikrokontroler AVR semudah memasang sebuah USB konektor pada komputer PC/ Laptop anda. Kelengkapan seperti buku manual, software AVRprog, AVR OSP II, CodeVision AVR, AVR Studio 4, Mikrobasic for AVR dan beberapa pendukung lainnya akan membantu dalam memprogram Mikrokontroler AVR/ AT89.

Jadi kapanpun dan dimanapun dapat melakukan pemrograman Mikrokontroler AVR anda dengan mudahnya baik menggunakan OS windows XP, windows Vista dan windows 7 32/ 64bit.

Spesifikasi:

- a. Format file yang didukung adalah * .hex
- b. Target ISP untuk semua AVR
- c. Terdapat Port komunikasi USB to Serial TTL
- d. Kompatibel dengan Windows XP, Vista dan win7 32/ 64bit
- e. Kompatibel Software: AVRdude, AVR studio 4, AVR OSP II, Bascom-AVR IDE, CodeVision AVR 1.25.9 dan AVRprog (rekomendasi gunakan AVRprog untuk speed download tercepat) .
- f. Tidak membutuhkan catu daya tambahan dari luar

- g. Terdapat selector jumper untuk power board mikrokontroler AVR jika membutuhkan power dari USB



Gambar 2.12 Modul K-125R

2.4 Bahasa Pemrograman C

Bahasa pemrograman C merupakan salah satu bahasa pemrograman komputer. Dibuat pada tahun 1972 oleh Dennis Ritchie untuk Sistem Operasi Unix di Bell Telephone Laboratories. Meskipun C dibuat untuk memprogram sistem dan jaringan komputer namun bahasa ini juga sering digunakan dalam mengembangkan software aplikasi. C juga banyak dipakai oleh berbagai jenis platform sistem operasi dan arsitektur komputer, bahkan terdapat beberapa compiler yang sangat populer telah tersedia. C secara luar biasa memengaruhi bahasa populer lainnya, terutama C++, C# yang merupakan ekstensi dari C.

2.4.1 Mengkompilasi Program

Suatu source program C baru dapat dijalankan setelah melalui tahap kompilasi dan penggabungan. Tahap kompilasi dimaksudkan untuk memeriksa source-program sesuai dengan kaidah-kaidah yang berlaku di dalam bahasa pemrograman C. Tahap kompilasi akan menghasilkan *relocatable object file*. File-file objek tersebut kemudian digabung dengan perpustakaan-fungsi yang sesuai. untuk menghasilkan suatu *executable-program*. Shortcut yang digunakan untuk mengkompilasi :

- a. CTRL + F9 Æ dipakai untuk menjalankan program yang telah kita buat atau bisa juga dengan mengklik tombol debug pada tool bar.
- b. ALT + F9 Æ dipakai untuk melakukan pengecekan jika ada yang error

pada program yang telah kita buat.

2.4.2 Pengenalan bahasa C#

C# adalah bahasa pemrograman baru yang diciptakan oleh Microsoft yang dikembangkan dibawah kepemimpinan Anders Hejlsberg yang telah menciptakan berbagai macam bahasa pemrograman termasuk Borland Turbo C++ dan orland Delphi. Bahasa C# juga telah di standarisasi secara internasional oleh ECMA. Seperti halnya bahasa pemrograman yang lain, C# bisa digunakan untuk membangun berbagai macam jenis aplikasi, seperti aplikasi berbasis windows (desktop) dan aplikasi berbasis web serta aplikasi berbasis web services.

2.4.2.1 Feature dalam C#

Sebagai bahasa pemrograman baru C# banyak mengadopsi feature dari beberapa bahasa perogrmaan terkenal dan banyak komunitasnya tetapi yang paling dominan adalah Java adapun komposisinya adalah sebagai berikut 70% Java, 10% C++, 5% Visual Basic, 15% baru.

Feature yang sama dengan JAVA antara lain: Object-orientation (single inheritance), Interfaces, Exceptions, Threads, Namespaces (like Packages), Strong typing, Garbage Collection, Reflection, Dynamic loading Code.

Sedangkan Feature yang sama dengan C++ antara lain: (Operator) Overloading, Pointer arithmetic in unsafe code, Some syntactic details. Adapun Feature barunya jika dikomper dengan Java adalah sebagai berikut : Reference and output parameters, Objects on the stack (structs), Rectangular arrays, Enumerations, Unified type system, goto Versioning, Component-based programming, Properties, Events, Delegates, Indexers, Operator overloading foreach statements, Boxing/unboxing, Attributes.

2.4.2.2 Penulisan Kode C#

Langkah-langkah penulisan kode C#

- a. Kode program diawali dengan mendeklarasikan nama Class atau namespace
- b. Aplikasi dibuka dengan tanda “{“ dan pada akhir kode ditutup dengan tanda

“}”.

- c. Aplikasi C# dibangun oleh satu atau beberapa fungsi yang diletakkan di dalam sebuah Class dengan ketentuan sebagai berikut .

Nama suatu fungsi pada C# harus diawali dengan huruf, atau garis bawah “_” yang kemudian bisa diikuti oleh huruf, angka atau garis bawah. Pada bagian akhir nama fungsi digunakan tanda kurung buka dan kurung tutup “()”. Penamaan fungsi tidak boleh mengandung spasi. Awal dan akhir suatu fungsi di mulai dengan tanda “{” dan diakhiri dengan tanda “}”. Penulisan komentar (tulisan yang tidak di eksekusi) dapat dibuat sebagai berikut Komentar satu baris dengan menggunakan tanda “//” Komentor yang lebih dari satu baris dengan diawali tanda “/*” dan diakhiri oleh “*/”

Contoh program yang paling sederhana untuk file Hallo.cs class HelloWorld

```
{
// Bagian utama program C# à ini adalah contoh komentar 1 baris
public static void Main()
{
System.Console.WriteLine("Hello, World");
}
/* ini cantoh komentar
lebih dari satu baris */
```

2.5 Flowchart

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program (Adelia, 2011). *Flowchart* menolong *analyst* dan *programmer* untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. *Flowchart* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut. *Flowchart* adalah bentuk gambar/diagram yang mempunyai aliran satu atau dua arah secara sekuensial. *Flowchart* digunakan untuk merepresentasikan maupun mendesain program. Oleh karena itu *flowchart* harus bisa merepresentasikan komponen-komponen dalam bahasa pemrograman.

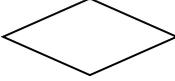
Ada beberapa macam *flowchart* diantaranya :

Sistem *flowchart* dapat didefinisikan sebagai bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan-urutan dari prosedur-prosedur yang ada di dalam sistem. Bagan alir sistem menunjukkan apa yang dikerjakan di sistem.

1. Bagan alir dokumen (*document flowchart*) atau disebut juga bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya.
2. Bagan alir skematik (*schematic flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem. Perbedaannya adalah, bagan alir skematik selain menggunakan simbol-simbol bagan alir sistem, juga menggunakan gambar-gambar komputer dan peralatan lainnya yang digunakan. Maksud penggunaan gambar-gambar ini adalah untuk memudahkan komunikasi kepada orang yang kurang paham dengan simbol-simbol bagan alir. Penggunaan gambar-gambar ini memudahkan untuk dipahami, tetapi sulit dan lama menggambarnya.
3. Bagan alir program (*program flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dibuat dari derivikasi bagan alir sistem.
4. Bagan alir program dapat terdiri dari dua macam, yaitu bagan alir logika program (*program logic flowchart*) dan bagan alir program komputer terinci (*detailed computer program flowchart*). Bagan alir logika program digunakan untuk menggambarkan tiap-tiap langkah di dalam program komputer secara logika. Bagan alat- logika program ini dipersiapkan oleh analis sistem. Gambar berikut menunjukkan bagan alir logika program. Bagan alir program komputer terinci (*detailed computer program flow-chart*) digunakan untuk menggambarkan instruksi-instruksi program komputer secara terinci. Bagan alir ini dipersiapkan oleh pemrogram.

5. Bagan alir proses (*process flowchart*) merupakan bagan alir yang banyak digunakan di teknik industri. Bagan alir ini juga berguna bagi analisis sistem untuk menggambarkan proses dalam suatu prosedur.

Tabel 2.1 Beberapa simbol pada *Flowchart*

Bagan	Nama	Fungsi
	<i>Terminator</i>	Awalatauakhir program
	<i>Flow</i>	Arahaliran program
	<i>Preparation</i>	inisialisasi/pemberiannilaiawal
	<i>Proccess</i>	Proses/pengolahan data
	<i>Input/Output Data</i>	input/output data
	Sub Program	sub program
	<i>Decision</i>	Seleksiataukondisi
	<i>On Page Connector</i>	Penghubungbagian-bagian <i>flowchart</i> padahalaman yang sama
	<i>Off Page Connector</i>	Penghubungbagian-bagian <i>flowchart</i> padahalaman yang berbeda