

II. TINJAUAN PUSTAKA

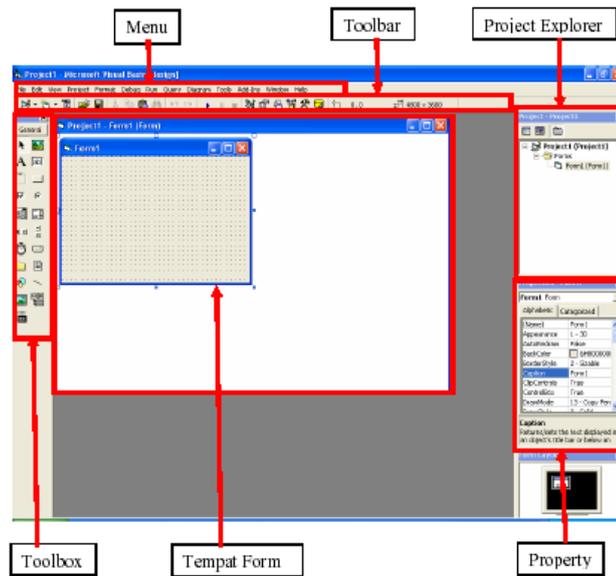
A. Microsoft Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi. Baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya dalam bentuk aplikasi *console*, aplikasi *Windows*, ataupun aplikasi *Web*. Visual Studio mencakup kompiler, SDK, *Integrated Development Environment* (IDE), dan dokumentasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di *Windows*) ataupun *managed code* (dalam bentuk *Microsoft Intermediate Language* di atas *.NET Framework*). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi *Silverlight*, aplikasi *Windows Mobile* (yang berjalan di atas *.NET Compact Framework*).

1. Antar Muka Visual Studio

Interface atau antar muka Visual Studio, berisi menu, *toolbar*, *toolbox*, *form*, *project explorer* dan *property* seperti terlihat pada Gambar 2.1 berikut:



Gambar. 2.1 *Interface* Visual Studio

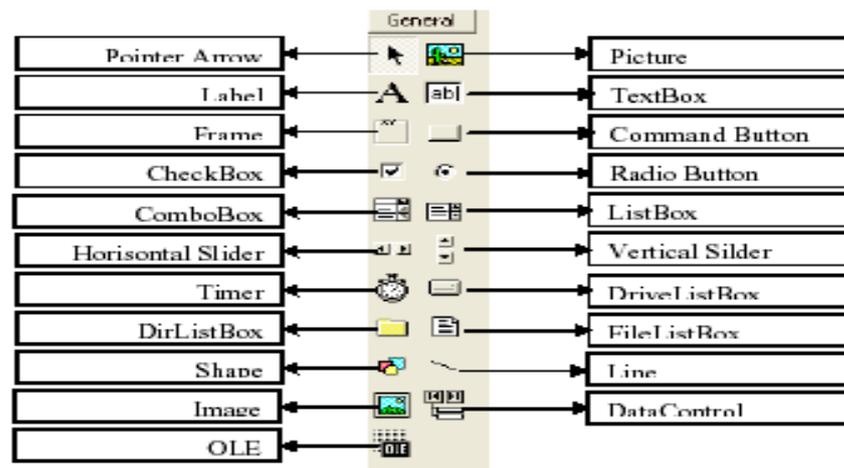
Pembuatan program aplikasi menggunakan Visual Studio dilakukan dengan membuat tampilan aplikasi pada *form*, kemudian diberi *script* program di dalam komponen-komponen yang diperlukan. *Form* disusun oleh komponen-komponen yang berada di [*Toolbox*], dan setiap komponen yang dipakai harus diatur propertinya lewat jendela [*Property*].

Menu pada dasarnya adalah operasional standar di dalam sistem operasi *windows*, seperti membuat form baru, membuat *project* baru, membuka *project* dan menyimpan *project*. Di samping itu terdapat fasilitas-fasilitas

pemakaian Visual Studio pada menu. Untuk lebih jelasnya Visual Studio menyediakan bantuan yang sangat lengkap dan detail dalam MSDN (*Microsoft Developer Network*).

a. *Toolbox*

Toolbox berisi komponen-komponen yang bisa digunakan oleh suatu *project* aktif, artinya isi komponen dalam *toolbox* sangat tergantung pada jenis *project* yang dibangun. Komponen standar dalam *toolbox* dapat dilihat pada Gambar. 2.2 berikut ini.



Gambar. 2.2 Komponen standar dalam *Toolbox*

Toolbox Visual Studio dengan semua kontrol intrinsik. Jendela *Toolbox* merupakan jendela yang sangat penting. Dari jendela ini dapat mengambil komponen-komponen (*object*) yang akan ditanamkan pada form untuk membentuk *user interface*.

b. Variabel

Variabel adalah tempat dalam memori komputer yang diberi nama (sebagai pengenal) dan dialokasikan untuk menampung data. Sesuai

data yang ditampung maka variabel harus mempunyai tipe data yang sesuai dengan isinya.

c. Operator

Operator digunakan untuk menghubungkan variabel dengan variabel lain untuk melakukan berbagai manipulasi dan pengolahan data

2. Konsep Dasar Pemrograman Dalam Visual Studio

Konsep dasar pemrograman Visual Studio adalah pembuatan form dengan mengikuti aturan pemrograman *Property*, Metode dan *Event*. Hal ini berarti:

- a. *Property*: Setiap komponen di dalam pemrograman Visual Studio dapat diatur propertinya sesuai dengan kebutuhan aplikasi.
- b. Metode: Bahwa jalannya program dapat diatur sesuai aplikasi dengan menggunakan metode pemrograman yang diatur sebagai aksi dari setiap komponen. Metode merupakan tempat untuk mengekspresikan logika pemrograman dari pembuatan suatu program aplikasi.

Event: Setiap komponen dapat beraksi melalui *event*, seperti *event click* pada *command button* yang tertulis dalam layar *script Command1_Click*.^[2]

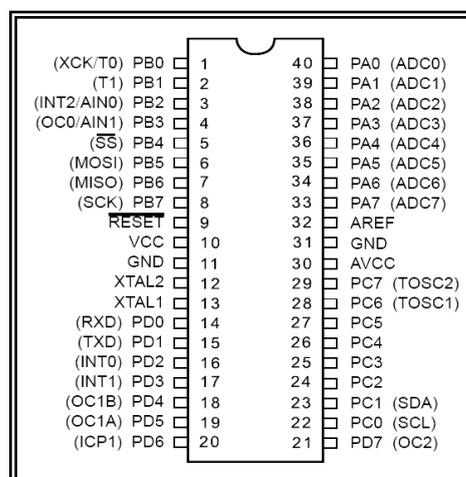
B. Mikrokontroler ATmega8535

Mikrokontroler ATmega8535 adalah salah satu jenis mikrokontroler keluarga AVR yang diproduksi oleh *Atmel Corporation*. ATmega8535

merupakan mikrokontroler 8 bit dengan arsitektur RISC (*Reduce Instruction Set Computer*). Fitur-fitur yang dimiliki oleh mikrokontroler ATmega8535 antara lain:

- a. Lebar data 8 bit.
- b. Memiliki 130 buah instruksi.
- c. Dapat mencapai kecepatan 16 MIPS (*Mega Instruction per Second*) pada frekuensi *clock* 16 MHz.
- d. Memiliki 32 x 8 register aplikasi umum.
- e. 8 k byte *flash memory* untuk memori program .
- f. 512 byte EEPROM untuk memori data *nonvolatile*.
- g. 512 byte SRAM.
- h. Dua 8 bit *timer/counter*.
- i. Satu 16 bit *timer/counter*.
- j. Empat saluran untuk penghasil sinyal PWM/*clock*.
- k. 8 saluran, 10 bit ADC.

Susunan pin mikrokontroler ATmega8535 tipe DIP (*dual in line package*) diperlihatkan pada Gambar. 2.3.



Gambar. 2.3. Konfigurasi Pin Mikrokontroler ATmega8535

Pada Gambar 3 dapat dilihat ATmega8535 memiliki empat buah *port* (terminal) masukan/keluaran yaitu *Port A*, *Port B*, *Port C*, dan *Port D*. *Port A* terdapat pada pin nomor 33 hingga 40 selain berfungsi sebagai *port* I/O digital, pin-pin ini juga dapat difungsikan sebagai saluran masukan sinyal analog yang akan diubah menjadi sinyal digital oleh ADC internal.

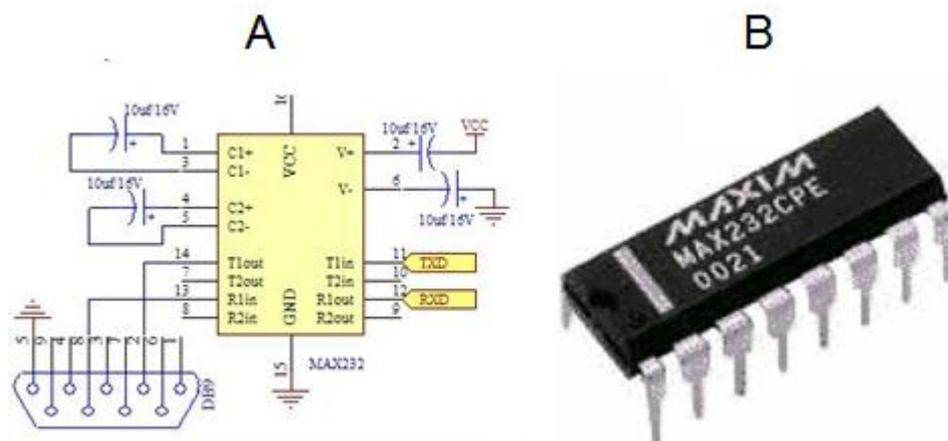
Port B terdapat pada pin nomor 1 hingga 8. Selain sebagai pin I/O digital biasa, pin-pin yang ada *Port B* juga memiliki fungsi khusus. Pin PB0 dan PB1 memiliki fungsi lain yaitu sebagai masukan sinyal *clock* eksternal untuk *timer/counter* 0 dan 1. Pin PB5 (MOSI), PB6 (MISO), dan PB7(SCK) memiliki fungsi lain sebagai saluran untuk sinyal ISP (*in sistem programming*).

Port D terdapat pada pin nomor 14 hingga 21. Selain berfungsi sebagai pin I/O digital biasa, pin-pin yang terdapat pada *port D* juga memiliki fungsi khusus. Pin PD2 (INT0) dan PD3 (INT1) berfungsi sebagai masukan untuk sinyal *interrupt* eksternal yang dapat menginterupsi alur program yang dieksekusi CPU. Pin PD7 (OC2) juga berfungsi sebagai pin untuk keluaran sinyal *clock/PWM* yang dihasilkan oleh timer 2 yang ada di dalam mikrokontroler. Pin *RESET* merupakan pin aktif rendah untuk mereset mikrokontroler. Dalam keadaan *reset*, alur program akan kembali ke alamat 0x0. Pin VCC dan GND adalah pin yang digunakan untuk penyedia tegangan mikrokontroler. Beda tegangan yang dapat diberikan berkisar antara 4,5–5,5V.^[3]

C. IC MAX 232

Untuk dapat berkomunikasi antara mikrokontroller dengan komputer, maka diperlukan suatu penyetaraan level tegangan. Besarnya level tegangan komunikasi serial (Level Tegangan RS232) adalah -25 s.d -3 V untuk logika high (1) dan +3 s.d +25 V untuk logika low (0). Hal ini sangat berbeda dengan level tegangan pada mikrokontroller (Level Tegangan TTL/CMOS) di mana untuk logika high (1) level tegangannya adalah 5 V dan untuk logika low (0) level tegangannya adalah 0 V.

Oleh karena itu diperlukan sebuah pengantarmuka yang dapat menyamakan level tegangan dari komunikasi serial pada komputer dengan mikrokontroller, yaitu IC RS232 produksi MAXIM yang disebut MAX232.

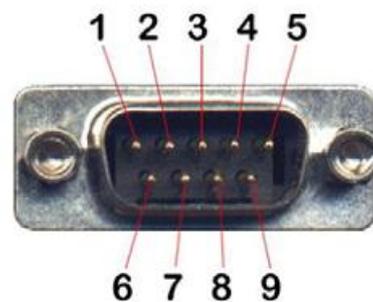


Gambar . 2.4 A. Skematik diagram koneksi IC MAX232 ke port serial
B. Tampak fisik IC Max 232m^[4]

Gambar 2.4 A memperlihatkan konfigurasi koneksi kaki IC MAX232 dengan port serial DB9, dan gambar 2.4 B memperlihatkan tampak fisik dari IC MAX232. MAX232 adalah saluran *driver/receiver* ganda yang termasuk pembangkit tegangan kapasitif yang menyediakan level tegangan RS232 dari sebuah sumber tegangan 5V. Setiap *receiver* pada IC MAX232 ini mengkonversikan level tegangan RS232 ke level tegangan TTL/CMOS sebesar 5 V. Setiap *receiver* ini mempunyai ambang batas sebesar 1.3 V, dan histeresis sebesar 0.5 V, serta dapat menerima masukan level tegangan ± 30 V. Sedangkan untuk setiap *driver* pada IC MAX232 ini mengkonversikan level tegangan masukan TTL/CMOS menjadi level tegangan RS232.

Konfigurasi Port Serial

Pada komputer IBM PC biasanya kita dapat menemukan dua konektor *port serial* DB-9 yang biasanya dinamai COM1 dan COM2.



Gambar. 2.5 Konektor *Serial* DB-9

Tabel. 2.1 Konfigurasi pin dan nama sinyal konektor *serial* DB-9

Pin	Nama	Direksi	Diskripsi
1	CD	←	<i>Carrier Detect</i>
2	RXD	←	<i>Receive Data</i>
3	TXD	→	<i>Transmit Data</i>

4	DTR	→	<i>Data Terminal Ready</i>
5	GND		<i>System Ground</i>
6	DSR	←	<i>Data Set Ready</i>
7	RTS	→	<i>Request to Send</i>
8	CTS	←	<i>Clear To Send</i>
9	RI	←	<i>Ring Indicator</i>

Gambar. 2.5 memperlihatkan susunan pin pada konektor serial dan tabel 1 menjelaskan nomor pin dan fungsinya. Untuk dapat menggunakan *port serial* kita perlu mengetahui alamatnya. *Base address* COM1 biasanya adalah 1016 (3F8h) dan COM2 biasanya 760 (2F8h). Alamat tersebut adalah alamat yang biasa digunakan, tergantung dari komputer yang digunakan. Tepatnya kita bisa melihat pada peta memori tempat menyimpan alamat tersebut, yaitu memori 0000.0400h untuk *base address* COM1 dan memori 0000.0402h untuk *base address* COM2. Setelah kita mengetahui *base address*-nya, kita dapat menentukan alamat register-register yang digunakan untuk komunikasi *port serial* ini.

Untuk konektivitas antara rangkaian pengendali dengan komputer digunakan komunikasi *serial* karena koneksi *serial* ini mempunyai keuntungan dibandingkan menggunakan komunikasi data secara paralel, yaitu :

- a. Pada komunikasi dengan kabel yang panjang, masalah rugi-rugi kabel (*cable loss*) tidak akan menjadi masalah yang besar daripada menggunakan kabel paralel. *Port serial* mentransmisikan '1' pada level tegangan -3 Volt sampai tegangan -25 Volt dan '0' pada level tegangan +3 Volt sampai +25 Volt, sedangkan *port* paralel

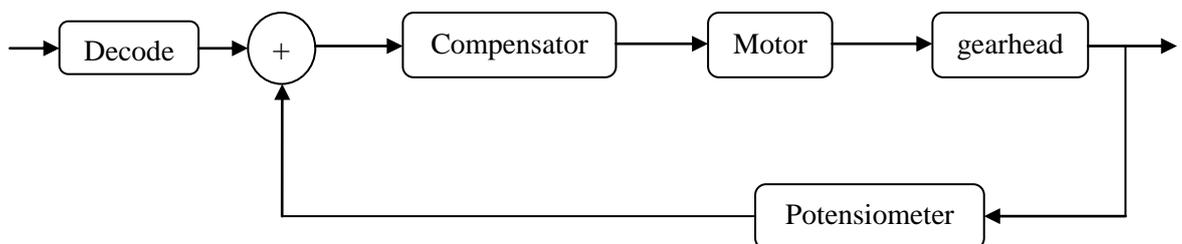
mentransmisikan '0' pada level tegangan 0 Volt dan '1' pada level tegangan 5 Volt.

- b. Dibutuhkan jumlah kabel yang lebih sedikit, bisa hanya menggunakan 3 kabel, yaitu saluran *transmit Data*, *Recieve Data* dan *Ground* (konfigurasi *Null Modem*)

Saat ini penggunaan mikrokontroler sudah dilengkapi dengan SCI (*Serial Communication Interface*) yang dapat digunakan untuk komunikasi dengan *port serial* komputer. ^[4]

D. Motor Servo

Motor servo adalah sebuah motor dengan sistem umpan balik tertutup di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor DC, serangkaian *gear*, potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor.



Gambar. 2.6 Cara Kerja Motor Servo

Karena motor DC servo merupakan alat untuk mengubah energi listrik menjadi energi mekanik, maka magnet permanent motor DC servo yang mengubah energi listrik ke dalam energi mekanik melalui interaksi dari dua medan magnet. Salah satu medan dihasilkan oleh magnet permanent dan yang satunya dihasilkan oleh arus yang mengalir dalam kumparan motor. Resultan dari dua medan magnet tersebut menghasilkan torsi yang membangkitkan putaran motor tersebut. Saat motor berputar, arus pada kumparan motor menghasilkan torsi yang nilainya konstan.

Secara umum terdapat 2 jenis motor servo. Yaitu :

1. motor servo standard dan motor servo *Continuous*. Servo motor tipe standar hanya mampu berputar 180 derajat. Motor servo standar sering dipakai pada sistem robotika misalnya untuk membuat “*Robot Arm*” (Robot Lengan).
2. Servo motor *continuous* dapat berputar sebesar 360 derajat. motor servo *Continuous* sering dipakai untuk *Mobile Robot*. Pada badan servo tertulis tipe servo yang bersangkutan.

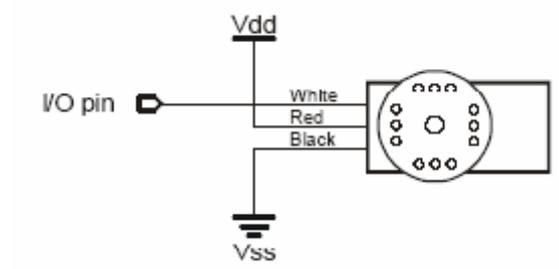
Motor servo merupakan sebuah motor dc kecil yang diberi sistem gear dan potensiometer sehingga dia dapat menempatkan “*horn*” servo pada posisi yang dikehendaki. Karena motor ini menggunakan sistem *close loop* sehingga posisi “*horn*” yang dikehendaki bisa dipertahankan. “*Horn*” pada servo ada dua jenis. Yaitu *Horn* “X” dan *Horn* berbentuk bulat, seperti pada Gambar 2.7



a.

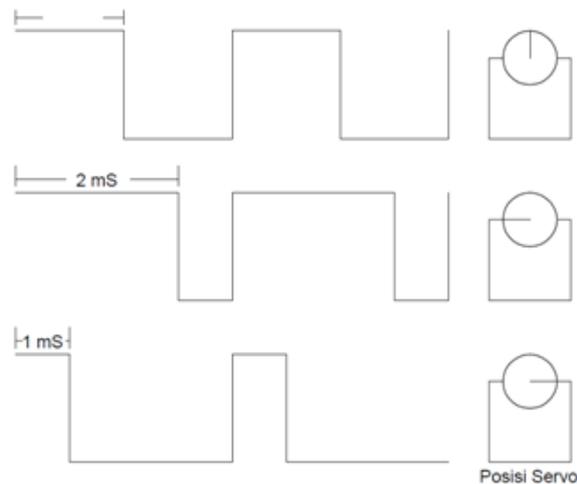


b.

Gambar. 2.7 a. Servo dengan *Horn* Bulatb. Servo dengan *horn* x^[5]Gambar. 2.8 Kaki Motor Servo^[5]

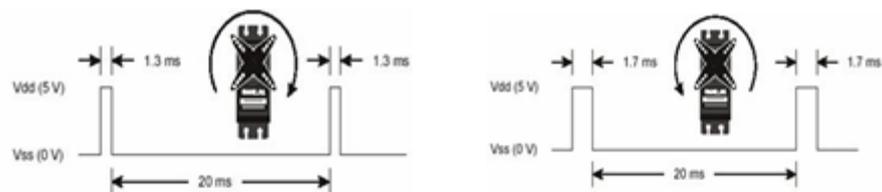
Pengendalian gerakan batang motor servo dapat dilakukan dengan menggunakan metode PWM. (*Pulse Width Modulation*). Teknik ini menggunakan sistem lebar pulsa untuk mengemudikan putaran motor. Sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor. Tampak pada gambar 9 dengan pulsa 1.5 mS pada periode selebar 2 mS maka sudut dari sumbu motor akan berada pada posisi tengah. Semakin lebar pulsa OFF maka akan semakin

besar gerakan sumbu ke arah jarum jam dan semakin kecil pulsa OFF maka akan semakin besar gerakan sumbu ke arah yang berlawanan dengan jarum jam.



Gambar. 2.9 Sinyal Pulsa Servo^[5]

Untuk menggerakkan motor servo ke kanan atau ke kiri, tergantung dari nilai delay yang kita berikan. Untuk membuat servo pada posisi *center*, berikan pulsa 1.5ms. Untuk memutar servo ke kanan, berikan pulsa $\leq 1.3\text{ms}$, dan pulsa $\geq 1.7\text{ms}$ untuk berputar ke kiri dengan *delay* 20ms, seperti ilustrasi berikut:



- Motor Servo akan bekerja secara baik jika pada bagian pin kontrolnya diberikan sinyal PWM dengan frekuensi 50Hz.

- Dimana pada saat sinyal dengan frekuensi 50Hz tersebut dicapai pada kondisi *Ton duty cycle* 1.5ms, maka rotor dari motor akan berhenti tepat di tengah-tengah (sudut 0° / netral).
- Pada saat *Ton duty cycle* dari sinyal yang diberikan kurang dari 1.5ms, maka rotor akan berputar ke arah kiri 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 1.5ms, maka rotor akan berputar ke arah kanan dengan membentuk sudut yang linier pula terhadap besarnya *Ton duty cycle*, dan bertahan diposisi tersebut. ^[5]

E. Pengertian Barcode

Barcode atau kode batang adalah sekumpulan data yang digambarkan dengan garis dan jarak spasi (ruang). *Barcode* menggunakan urutan garis batang vertikal dan jarak antar garis untuk mewakili angka atau simbol lainnya. Dengan demikian, setiap ketebalan garis batang dan jarak antara garis satu dengan yang lain selalu berbeda sesuai dengan isi data yang dikandung oleh kode batang atau *barcode* tersebut.



Gambar. 2.10 *Barcode* ^[11]

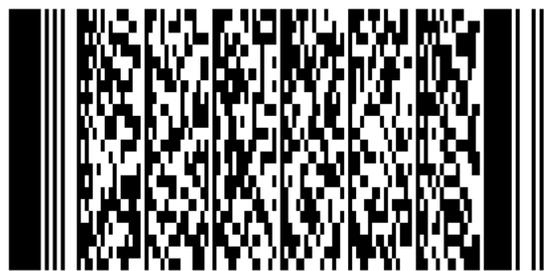
Jenis *barcode* yang dikenal saat ini adalah *barcode linear* 1D (1 dimensi) yang berupa rangkaian garis dengan ketebalan yang bervariasi dan berbentuk persegi panjang.



One-Dimensional Barcode

Gambar. 2.11 *Barcode* 1 dimensi^[12]

Serta juga terdapat jenis *barcode matriks* 2D (2 dimensi) yang datanya diwakili oleh simbol-simbol yang berbentuk persegi, titik, heksagon dan bentuk geometri lainnya pada gambar yang berada dalam sebuah bujur sangkar. Untuk jenis *barcode matriks* ini dapat memasukkan data sampai ratusan karakter dalam sebuah *barcode*, berbeda dengan *barcode linear* yang kemampuan menyimpan datanya terbatas.

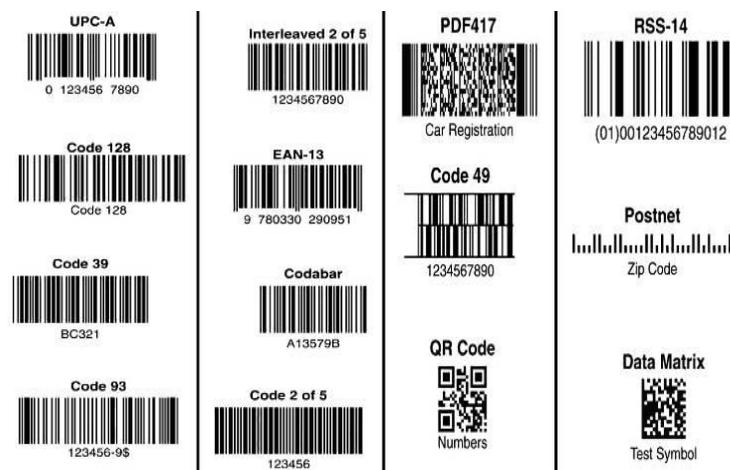


Gambar. 2.12 *Barcode* 2 Dimensi^[12]

Terdapat beberapa standar kode dalam *barcode* sesuai dengan kegunaan dan tujuan pemakaian *barcode*, seperti pada daftar berikut :

- *Uniform Product Code (UPC)* : Digunakan untuk *checkout* penjualan, persediaan, dan sebagainya pada toko retail.
- *Code 39 (Code 3 of 9)* : Digunakan untuk identifikasi, inventarisasi, dan pengiriman pelacakan.
- *POSTNET* : Digunakan untuk kode pos *encoding* di US mail.
- *European Article Number (EAN)* : Digunakan untuk sebuah superset dari UPC yang memungkinkan digit ekstra untuk identifikasi negara.
- *Japanese Article Number (JAN)* : Dhampir serupa dengan EAN, namun penggunaan jenis ini lebih banyak digunakan di Jepang.
- *Bookland* : Bekerja berdasarkan nomor ISBN dan digunakan pada sampul buku.
- *ISSN bar code* : Bekerja berdasarkan nomor ISSN, digunakan pada majalah di luar AS.
- *Code 128* : Digunakan dalam preferensi untuk Code 39 karena lebih kompak.
- *Interleaved 2 of 5* : Digunakan dalam industri pelayaran dan gudang.
- *Codabar* : Digunakan oleh Federal Express, di perpustakaan dan bank darah.
- *Magnetic Ink Character Recognition (MICR)* : Sebuah font khusus yang digunakan untuk nomor di bagian bawah cek bank.
- *OCR-A* : Format pengenalan karakter optik yang digunakan pada sampul buku, untuk nomor ISBN agar bisa dibaca oleh manusia.

- OCR-B : Digunakan untuk mempermudah pembacaan *barcode* versi UPC, EAN, JAN, *Bookland*, dan ISSN dan Code 39.
- *Maxicode* : Digunakan oleh *United Parcel Service*.
- PDF417 : Suatu jenis *barcode* 2-D baru yang dapat *encode* sampai 1108 *byte* informasi; dapat terkompresi seperti pada sebuah portabel file data (PDF).



Gambar. 2.13 Jenis Jenis *Barcode*^[6]

Alat yang digunakan untuk membaca *barcode* adalah *barcode scanner*. Penggunaan *barcode scanner* sangat mudah sehingga pengguna (operator) hanya memerlukan sedikit latihan. *Barcode scanner* dapat membaca informasi atau data dengan kecepatan yang jauh lebih tinggi dari pada mengetikkan data dan *barcode scanner* memiliki tingkat ketelitian yang lebih tinggi.

Metoda Pengkodean Ada Dua Sistem yaitu:

A. *Binary coding* (Pengkodean Biner)

Dua ukuran *bar* dan *space* digunakan untuk meng-*encode*-kan data. *Bar* dan spasi dapat diubah ke dalam kode biner dengan mudah, yang kemudian diubah (menggunakan sebuah tabel) ke dalam karakter ASCII.

B. *Proportional coding*

Ada beberapa ukuran yang berbeda pada *bar* dan *space*. Ukuran pada *bar* / *space* dan urutan dari *bar* dan *space* mendefinisikan karakter yang dipresentasikan. Kode tersebut lebih sulit dibaca (kemungkinan tidak mudah mentranslasikannya ke biner) dan diperlukan ketelitian yang lebih dalam mencetak dan men-scanning kode bar.

Pada umumnya ada 4 ukuran yang berbeda pada bar dan spasi yang digunakan untuk meng-*encode*-kan data. Contoh jenis kode bar yang menggunakan teknik *encoding* ini adalah USS Code 128.

Pengkodean data dalam sebuah kode bar dilakukan sebagai berikut :

- Sebuah *fixed number* pada bar digunakan per karakter. Hal ini berarti bahwa jika sebuah bar tidak terbaca, maka kode bar tersebut tidak akan dapat dibaca.
- Jumlah karakter yang mungkin yang dapat di-*encoded* dalam beberapa jenis kode bar lebih tinggi dibandingkan dengan jumlah karakter yang valid. Hal ini berarti jika ukuran dari salah satu bar/spasi salah

terbaca, maka karakter ilegal akan terbaca. Oleh karena itu, kecepatan substitusi karakter menjadi sangat rendah.

Keuntungan penggunaan *barcode*, antara lain adalah :

- Proses *Input Data* lebih cepat, karena : *Barcode Scanner* dapat membaca atau merekam data lebih cepat dibandingkan dengan melakukan proses *input* data secara manual.
- Proses *Input Data* lebih tepat, karena : Teknologi *barcode* mempunyai ketepatan yang tinggi dalam pencarian data.
- Penelusuran informasi data lebih akurat karena teknologi *barcode* mempunyai akurasi dan ketelitian yang sangat tinggi.
- Mengurangi biaya, karena dapat menghindari kerugian dari kesalahan pencatatan data dan mengurangi pekerjaan yang dilakukan secara manual secara berulang-ulang.
- Peningkatan Kinerja Manajemen, karena dengan data yang lebih cepat, tepat dan akurat maka pengambilan keputusan oleh manajemen akan jauh lebih baik dan lebih tepat, yang nantinya akan sangat berpengaruh dalam menentukan kebijakan perusahaan.
- Memiliki nilai tawar lebih tinggi atau prestise serta kemampuan bersaing dengan saingan atau kompetitor akan lebih terjaga.

Jadi, kesimpulannya *barcode* sangat aman. Kesalahan dalam pembacaan sangat tidak mungkin.^[6]

F. *Barcode scanner*

Barcode Scanner adalah alat yang digunakan untuk membaca kode-kode berbentuk garis-garis vertikal yang terdapat pada kebanyakan produk-produk *consumer good*. Penggunaan *barcode scanner* ini mempunyai dua keuntungan tambahan. Pertama akan memperkecil kesalahan *input* yang disebabkan kesalahan operator komputer atau kasir. Kedua, penggunaan *barcode scanner* mempercepat proses *entry* data sehingga mengurangi jumlah antrian yang panjang.

Berikut ini adalah tipe-tipe *barcode* dengan pengelompokan berdasarkan jenis pilihan:

1. *Barcode Scanner Genggam Desktop*

Barcode scanner genggam atau desktop adalah jenis *barcode scanner* yang paling umum dijumpai. *Scanner* jenis ini sering dijumpai pada toko swalayan yang digunakan di kasir-kasir.



Gambar. 2.14 Scanner Genggam^[7]

Barcode scanner ini cocok digunakan untuk jenis barang yang dalam pen-scan-an memerlukan fleksibilitas gerak tinggi. Misalnya pada toko fashion atau toko *baby shop*. *Barcode scanner* ini juga cocok digunakan di bagian *backoffice* dari sebuah supermarket, atau bahkan ditaruh di gudang.

2. *Barcode scanner desktop* dengan *stand*

Tipe ini adalah sebuah *barcode scanner desktop* yang dilengkapi dengan duduk-an (*stand*). Penggunaan *stand* akan membuat tatanan lebih rapi. Namun untuk penggunaan yang disebutkan di atas, seperti untuk toko baju, maka penggunaan *stand* ini justru akan mempersulit kasir - karena setelah melakukan *stand* harus mengembalikan pada posisinya.



Gambar. 2.15 *Scanner desktop* dengan *stand* ^[7]

Scanner barcode jenis ini memiliki sensor dengan fitur *auto-sensing* atau fitur *continuous scan*, hal ini juga sangat membantu untuk penggunaan di minimarket. Artinya *barcode scanner* tetap berada di posisi pada *stand*-nya, kemudian barang yang didekatkan ke arah *scanner*, dan tanpa menekan tombol, *barcode scanner* akan melakukan scanning otomatis.

3. *Barcode scanner omni directional*

Barcode scanner jenis *omni* merupakan pengembangan dari *barcode scanner* desktop dengan *stand*. Pada *barcode desktop* sinar yang keluar dari *barcode scanner* berjumlah 1 garis. Berarti *users* harus secara tepat meletakkan posisi sinar mendekati melintang (tegak lurus) dengan posisi *barcode* pada produk. Hal ini terkadang akan sangat menyulitkan dan membuat proses scanning berlangsung lama. *Barcode scanner omni* mengatasi masalah tersebut. Pada *barcode scanner omni*, sinar yang keluar tidak hanya satu sinar, namun banyak (biasanya sekitar 20 sinar) dengan posisi yang berbeda-beda. Saat *user* melewati barang tersebut pada posisi yang cukup dekat, *barcode scanner* sudah akan mengenali barang tersebut.



Gambar. 2.16 *Scanner Omni Directional* ^[7]

4. *Barcode scanner in-counter*

Barcode scanner ini sama dengan *barcode scanner omni*. Perbedaannya hanya pada peletakkannya saja. Apabila *barcode scanner omni* diletakkan di atas meja, maka posisi sensor dari *barcode scanner* diletakan didalam meja kasir dan menghadap ke atas. Hal ini dirasa membant untuk mempercepat *scanning* atas suatu produk.



Gambar. 2.17 *Scanner in-counter* ^[7]

5. *Barcode scanner wirelles (RF/Bluetooth)*

Saat ini permintaan *customer* akan kebutuhan *barcode scanner* semakin meningkat. Maka dibuatlah *barcode scanner wireless* baik yang menggunakan teknologi RF maupun teknologi *bluetooth* tanpa kabel membuat *barcode* ini lebih fleksibel dan mudah dibawa dalam radius tertentu. *Barcode* jenis ini dapat dipakai untuk keperluan apapun, baik di supermarket, toko fashion, *backoffice*, gudang, atau bahkan industri. Namun karena *barcode* tipe ini masih cukup mahal harganya.



Gambar. 2.18 *Scanner wirelles (RF/Bluetooth)* ^[7]

6. *Barcode scanner* untuk industri

Barcode scanner industri sebenarnya bisa berupa *barcode scanner desktop* (umumnya tanpa stand) atau *barcode scanner wireless*. Yang membedakan adalah bahwa *barcode* ini dibuat untuk ketahanan atau *reliability* yang lebih bagus dengan asumsi bahwa

penggunaan di industri mempunyai risiko lebih tinggi untuk jatuh, tertindih, terkena cairan, dan sebagainya. Biasanya *barcode scanner* untuk industri juga dilengkapi dengan pelindung berbahan karet tebal.



Gambar. 2.19 *Scanner* Industri^[7]

Selain klasifikasi barcode scanner berdasarkan penggunaan, tentunya masih ada klasifikasi barcode berdasarkan teknologinya, baik itu CCD, *Laser Diode*, *Red Diode*, dan lain-lain.^[7]

G. Pengertian Sistem Informasi

Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung

operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi serta menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

a. Komponen Sistem Informasi

Komponen-komponen sistem informasi adalah sebagai berikut :

1. Blok Masukan

Blok masukan terdiri atas metode-metode dan media untuk menangkap data yang dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok Model

Blok model terdiri atas kombinasi prosedur, logika, dan model matematika yang akan memanipulasi data input dan data yang tersimpan dalam basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumen yang berguna untuk tingkat manajemen dan semua pemakai sistem.

4. Blok Teknologi

Teknologi merupakan kotak alat dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, serta membantu pengendalian sistem secara keseluruhan.

5. Blok Basis Data

Basis data merupakan kumpulan data yang saling berhubungan antara yang satu dengan yang lain. Basis data tersimpan dalam perangkat keras komputer dan untuk menggunakannya diperlukan perangkat lunak untuk memanipulasi.

6. Blok Kendali

Blok kendali berguna untuk mengendalikan sistem informasi agar dapat berjalan dengan baik sesuai dengan keinginan.

b. Perangkat Pemodelan Sistem

Langkah- langkah yang dilakukan dalam perancangan sistem adalah membuat pemecahan masalah secara logika.

Alat bantu yang digunakan adalah :

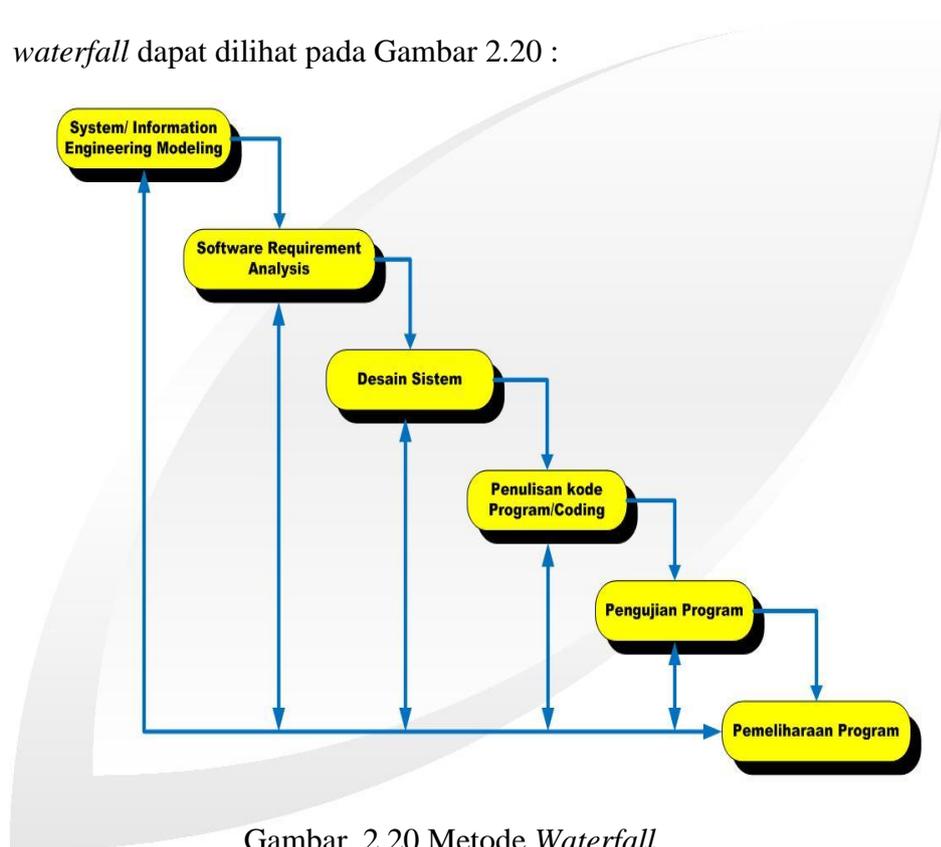
1. *Entity Relationship Diagram (ERD)*
2. *Data Context Diagram (DCD)*
3. *Data flow diagram (DFD)*

H. Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal spesifikasi sistem sampai pemeliharaan sistem setelah digunakan. Dalam tugas akhir ini akan menggunakan model pengembangan perangkat lunak *modified waterfall*, dikarenakan metode ini cukup sederhana dan lengkap.

Model pengembangan perangkat lunak *modified waterfall* melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap analisis, desain, *coding*, pengujian, dan pemeliharaan. Disebut dengan *modified waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.

Sebagai contoh tahap desain harus menunggu selesainya tahap sebelumnya yaitu tahap *requirement*. Secara umum tahapan pada model *modified waterfall* dapat dilihat pada Gambar 2.20 :



Gambar. 2.20 Metode *Waterfall*

Berikut adalah penjelasan dari tahap-tahap yang dilakukan di dalam model ini menurut Pressman:

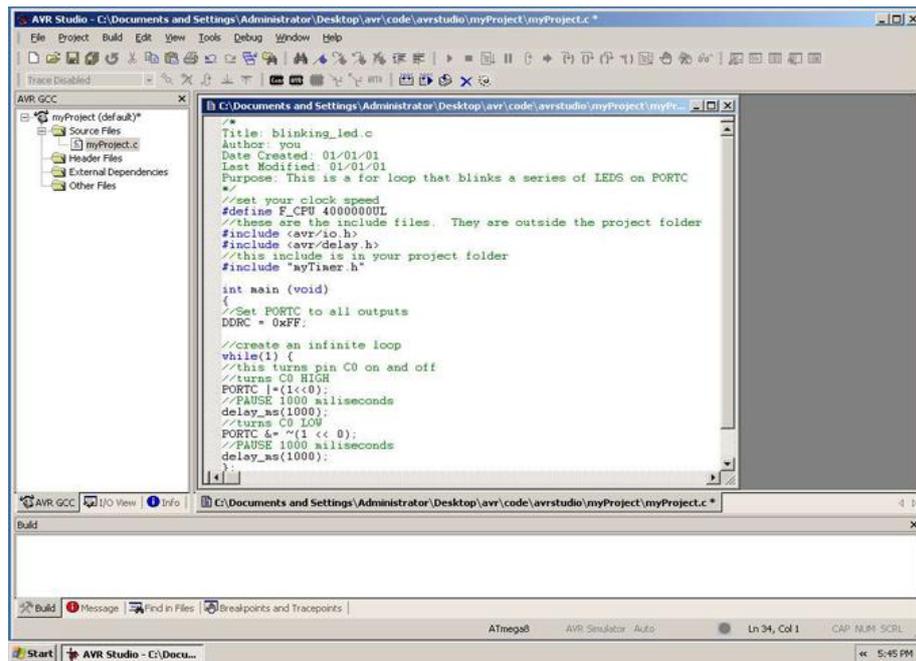
- a. **System / Information Engineering and Modeling.** Permodelan ini diawali dengan mencari kebutuhan dari keseluruhan sistem yang akan diaplikasikan ke dalam bentuk perangkat lunak. Hal ini sangat penting, mengingat perangkat lunak harus dapat berinteraksi dengan elemen-elemen yang lain seperti perangkat keras, basis data, dan sebagainya. Tahap ini sering disebut dengan *Project Definition*.
- b. **Software Requirements Analysis.** Proses pencarian kebutuhan diintensifkan dan difokuskan pada perangkat lunak. Untuk mengetahui sifat dari program yang akan dibuat, maka para *software engineer* harus mengerti tentang domain informasi dari perangkat lunak.
- c. **Desain Sistem.** Proses ini digunakan untuk mengubah kebutuhan-kebutuhan diatas menjadi representasi ke dalam bentuk “*blueprint*” perangkat lunak sebelum *coding* dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya.
- d. **Penulisan Kode Program/Coding.** Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses *coding*. Tahap ini merupakan implementasi dari tahap design yang secara teknis nantinya dikerjakan oleh *programmer*.
- e. **Pengujian/Testing.** Sesuatu yang dibuat haruslah diujicobakan. Demikian juga dengan perangkat lunak. Semua fungsi-fungsi perangkat lunak harus diujicobakan, agar perangkat lunak bebas dari *error*, dan hasilnya harus

benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

- f. Pemeliharaan Program.** Pemeliharaan suatu perangkat lunak diperlukan, termasuk di dalamnya adalah pengembangan, karena perangkat lunak yang dibuat tidak selamanya hanya seperti itu. Ketika dijalankan mungkin saja masih ada *errors* kecil yang tidak ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada perangkat lunak tersebut. Pengembangan diperlukan ketika adanya perubahan dari eksternal perusahaan seperti ketika ada pergantian sistem operasi, atau perangkat lainnya.^[8]

I. AVR Studio 4

AVR studio adalah sebuah *software Integrated Development Environment* (IDE) yang dibuat oleh ATMEL untuk membuat aplikasi pemrograman 8 bit pada mikrokontroler AVR. Pada dasarnya AVR studio 4 menggunakan bahasa pemrograman *Assembler*. Bahasa Pemrograman *Assembler* adalah bahasa tingkat rendah yang memiliki keunggulan tersendiri namun sulit untuk dipahami. Selain menggunakan bahasa pemrograman *Assembler*, AVR Studio 4 juga dapat menggunakan bahasa C sebagai bahasa pemrograman. Bahasa C adalah bahasa pemrograman tingkat menengah yang lebih mudah untuk dipelajari bila dibandingkan dengan bahasa *Assembler*.



Gambar. 2.21 Lembar Kerja AVR Studio 4

Untuk membuat program C dengan menggunakan AVR Studio 4, selain membutuhkan *software* AVR studio 4, programmer juga membutuhkan *include* file yang terdapat di dalam program WinAVR. WinAVR adalah sebuah *software opensource* yang dibuat untuk memprogram mikrokontroler AVR yang bekerja pada sistem operasi *windows*.

J. Bahasa C untuk Mikrokontroler

Pembuatan program mikrokontroler dalam bahasa tingkat-tinggi (*High-Level Language*, disingkat HLL), misalnya bahasa 'C', memungkinkan untuk mengurangi waktu pengembangan secara signifikan jika dibandingkan dengan Bahasa *Assembly*. Biasanya, sebuah program yang ditulis dalam bahasa tingkat-tinggi akan lebih terstruktur dibandingkan

program yang sama yang ditulis dalam *Assembly*. Dengan demikian, akan lebih mudah melakukan pelacakan (*debugging*).

Kelemahan utama penulisan program dalam bahasa tingkat-tinggi adalah programnya menjadi lebih besar dan lambat dibandingkan jika ditulis dalam Bahasa *Assembly*. Bagaimanapun juga, seiring dengan jumlah baris program yang bertambah, gap ukuran antara kode yang ditulis dalam bahasa tingkat-tinggi dan yang ditulis dalam *Assembly* menjadi mengecil.

Untuk para pengguna AVR, titik temu (antara bahasa tingkat-tinggi dan *Assembly*) kesamaan ukuran sekitar 4K. Program yang ditulis dalam bahasa tingkat-tinggi umumnya tidak akan lebih cepat dibandingkan dituliskan dalam *Assembly*. Jika kecepatan eksekusi pada beberapa bagian program menjadi bagian yang kritis, solusinya adalah menuliskan bagian kritis tersebut dalam *Assembly*, selebihnya kerangka program dan bagian-bagian yang tidak kritis dalam bahasa tingkat-tinggi

Pemrograman Bahasa C untuk mikrokontroler sudah umum digunakan. Berikut kerangka listing program bahasa C untuk memprogram mikrokontroller:

```
#include < [library1.h] > // Opsional*
#include < [library2.h] > // Opsional
#define [nama1] [nilai] ; // Opsional
#define [nama2] [nilai] ; // Opsional
[global variables] // Opsional
[functions` prototype] // Opsional
void main(void) // Program Utama harus ada
```

```
{ [Deklarasi local variable/constant] [Isi Program Utama] }
```

Perlu diperhatikan dalam pemrograman C, Pendeklarasian *library* harus ada sesuai dengan kode program yang digunakan.^[9]

K. Bahasa Pemrogram VB.NET (Visual Basic)

Microsoft Visual Basic .NET adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak diatas sistem .NET Framework dengan menggunakan bahasa BASIC. Dengan menggunakan ini, para *programmer* dapat membangun aplikasi Windows Forms, aplikasi web berbasis ASP.NET, dan juga aplikasi *command-line*. Program ini dapat diperoleh secara terpisah dari beberapa produk lainnya (seperti Microsoft Visual C++, Visual C#, atau Visual J#), atau juga dapat diperoleh secara terpadu dalam Microsoft Visual Studio .NET.

Bahasa Visual Basic .NET sendiri menganut paradigma bahasa pemrograman berorientasi objek yang dapat dilihat sebagai evolusi dari Microsoft Visual Basic versi sebelumnya yang diimplementasikan di atas .NET Framework. Banyak sekali perubahan yang dilakukan oleh Microsoft antara Visual Basic Clasic dengan Visual Basic .NET, dan Visual Basic .NET ini tidak kompatibel dengan versi terdahulu.^[10]