

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Pengertian Perancangan**

Desain atau perancangan dalam pengembangan perangkat lunak merupakan upaya untuk mengkonstruksi sebuah sistem yang memberikan kepuasan (mungkin informal) akan spesifikasi kebutuhan fungsional memenuhi target, memenuhi kebutuhan secara implisit dan eksplisit dari segi performansi maupun penggunaan sumber daya, kepuasan batasan pada proses desain dari segi biaya, waktu, dan perangkat. (Rosa A.S, M. Shalahuddin ; 2011: 21).

Desain atau perancangan adalah mentransformasikan kebutuhan detail menjadi kebutuhan yang sudah lengkap, dokumen desain sistem fokus pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan. (Rosa A.S, M. Shalahuddin ; 2011: 25).

Desain atau perancangan perangkat lunak adalah proses langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan perangkat lunak ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan. (Rosa A.S, M. Shalahuddin ; 2011: 27).

Perancangan adalah proses menuangkan ide dan gagasan berdasarkan teori-teori dasar yang mendukung. Proses perancangan dapat dilakukan dengan

cara pemilihan komponen yang akan digunakan, mempelajari karakteristik dan data fisiknya, membuat rangkaian skematik dengan melihat fungsi-fungsi komponen yang dipelajari, sehingga dapat dibuat alat yang sesuai dengan spesifikasi yang diharapkan. (<http://elib.unikom.ac.id/download.php?id=143047>: dikunjungi pada tanggal 13 Februari 2017).

## **II.2. Pengertian Dasar Sistem Kendali**

Sistem kendali merupakan suatu sistem yang keluarannya dikendalikan pada suatu nilai tertentu atau untuk mengubah beberapa ketentuan yang telah ditetapkan dari masukan ke sistem. Untuk merancang suatu sistem yang dapat merespon suatu perubahan tegangan dan mengeksekusi perintah berdasarkan situasi yang terjadi, maka diperlukan pemahaman tentang sistem kendali (*control system*). Sistem kendali merupakan suatu kondisi dimana sebuah perangkat (*device*) dapat dikendalikan sesuai dengan perubahan situasi.

Kendali (pengendalian) adalah sebuah proses atau upaya untuk mencapai tujuan. Dengan demikian, sistem kendali adalah kombinasi dari beberapa komponen yang bekerja secara sinergi dan terpadu untuk memperoleh hasil yang diinginkan (tujuan). Sebagai contoh sederhana dan akrab dengan aktivitas sehari-hari dari konsep kendali adalah saat mengendarai kendaraan. Tujuan yang diinginkan dari proses tersebut adalah berjalannya kendaraan pada lintasan (*track*) yang diinginkan. Ada beberapa komponen yang terlibat di dalamnya, misalnya pedal gas, *speedometer*, mesin (penggerak), rem, dan pengendara. Sistem kendali berkendara berarti kombinasi dari komponen-komponen tersebut yang

menghasilkan berjalannya kendaraan pada lintasan yang diinginkan. Ketika jalan lengang dan aturan memperbolehkan, pengendara mempercepat laju kendaraan dengan membuka pedal gas. Demikian pula, jika ada kendaraan lain di depan atau lampu penyeberangan berwarna merah maka pengendara menginjak rem dan menurunkan kecepatannya. Semua upaya itu dilakukan untuk mempertahankan kendaraan pada lintasan yang diinginkan.

Contoh lain dapat disebutkan berupa proses memindahkan barang oleh tangan kita. Pada proses tersebut, tujuannya adalah posisi atau letak barang yang diinginkan. Komponennya berupa tangan (dalam hal ini tentunya dengan otot tangan), mata, dan otak sebagai pengendali. Pada saat tangan bergerak untuk memindahkan barang, mata akan menangkap informasi tentang posisi pada saat itu. Informasi tersebut diproses otak untuk disimpulkan apakah posisinya sudah benar atau tidak. Selanjutnya, apabila posisinya masih belum tercapai maka otak akan memerintahkan otot tangan untuk bergerak memindahkan barang ke posisi yang diinginkan. Secara umum dapat dikatakan semua proses yang terjadi di alam pada hakikatnya adalah sebuah sistem kendali. (Asep Najmurokhman, 2010 : 1)

### **II.3. Pengertian Wasit Pertandingan**

Wasit adalah seorang yang memiliki wewenang untuk mengatur jalannya suatu pertandingan. Ada bermacam-macam istilah wasit. Dalam bahasa Inggris dikenal *referee*, *umpire*, *judge* atau *linesman*. Istilah wasit dalam bahasa Inggris *Referee* berasal dari sepak bola. Awalnya kapten dari setiap tim saling berkonsultasi untuk menyelesaikan perselisihan yang terjadi di lapangan.

Kemudian peran ini didelegasikan kepada seorang *umpire*. Setiap tim membawa *umpire*-nya masing-masing sehingga masing-masing kapten tim dapat berkonsentrasi kepada permainan. Akhirnya, seorang yang dianggap netral dinamai *referee* (dari *would be "referred to"*) bertindak sebagai orang yang akan menyelesaikan permasalahan jika *umpire* tidak bisa menyelesaikannya.

(Sumber : <https://id.wikipedia.org/wiki/Wasit>)

#### **II.4. Pengertian Sensor**

Dalam rangkaian elektronika untuk keperluan pengukuran atau deteksi, diperlukan suatu bagian yang disebut sensor. Sensor berfungsi untuk menubah besaran yang bersifat fisis atau suhu, tekanan, berat, atau intensitas cahaya menjadi besaran listrik (tegangan atau arus listrik).

Sensor adalah alat untuk mendeteksi atau mengukur sesuatu yang digunakan untuk mengubah variasi mekanis, *magnetic*, panas, dan kimia menjadi tegangan dan arus listrik. Dalam lingkungan sistem pengendali dan robotika, sensor memberikan kesamaan yang menyerupai mata, pendengaran, hidung, lidah yang kemudian akan diolah oleh kontroler sebagai otaknya. (Petruzella : 2010)

Sensor memiliki suatu ukuran yang disebut sensitivitas. Sensitivitas menunjukkan seberapa besar pengaruh perubahan nilai besaran fisis yang diukur oleh sensor terhadap keluaran dari sensor tersebut. Misalnya, sebuah sensor suhu yang tegangan keluarannya berubah 0,1 V jika terjadi perubahan suhu sebesar 1<sup>0</sup>C. Maka sensor suhu tersebut dapat dikatakan memiliki sensitivitas sebesar 0,1V/1<sup>0</sup>C. (Endra Pitowarno ; 2006 : 56)

Sensor yang baik memiliki ciri-ciri sebagai berikut:

1. Peka terhadap besaran yang akan diukur.
2. Tidak peka terhadap besaran lain yang tidak akan diukur.
3. Keberadaan sensor tidak mempengaruhi besaran yang akan diukur.

## II.5. Mikrokontroler Arduino

Arduino dikatakan sebagai sebuah platform dari physical computing yang bersifat *open source*. Pertama-tama perlu dipahami bahwa kata “*platform*” adalah sebuah pilihan kata yang tepat. Arduino tidak hanya sekedar sebuah alat pengembangan, tetapi ia adalah kombinasi dari *hardware*, bahasa pemrograman dan *Integrated Development Environment* (IDE) yang canggih. IDE adalah sebuah *software* yang sangat berperan untuk menulis program, meng-*compile* menjadi kode biner dan meng-*upload* ke dalam memori *microcontroller*. Ada banyak projek dan alat – alat yang dikembangkan oleh akademisi dan profesional dengan menggunakan Arduino, selain itu juga ada banyak modul-modul pendukung (sensor, tampilan, penggerak dan sebagainya) yang dibuat oleh pihak lain untuk bisa disambungkan dengan Arduino. Arduino berevolusi menjadi sebuah platform karena ia menjadi pilihan dan acuan bagi banyak praktisi.

Salah satu yang membuat arduino memikat hati banyak orang adalah karena sifatnya *open source*, baik untuk *hardware* maupun *software*-nya. Komponen utama didalam papan Arduino adalah sebuah *microcontroller* 8 bit dengan merk Atmega yang dibuat oleh perusahaan Atmel Corporation.

Berbagai papan Arduino menggunakan tipe Atmega yang berbeda-beda tergantung dari spesifikasinya, sebagai contoh Arduino Uno menggunakan Atmega328 sedangkan Arduino Mega 2560 yang lebih canggih menggunakan Atmega2560 (Feri Djuandi,2011: 2).

### II.5.1. Arduino Mega 2560

Arduino Mega 2560 adalah sebuah papan mikrokontroler berbasis Atmega 2560 (*datasheet*). Mempunyai 54 pin digital *input/output* (dimana 14 pin dapat digunakan sebagai keluaran PWM), 16 pin input analog, 2 UARTs (*Hardware serial ports*), sebuah *crystal oscillator* 16 MHz, sebuah penghubung USB, sebuah colokan listrik, ICSP *header*, dan tombol kembali. Setiap isi dari Arduino Mega 2560 membutuhkan dukungan mikrokontroler; koneksi mudah antara Arduino mega 2560 ke komputer dengan sebuah kabel USB atau daya dengan AC to DC adaptor atau baterai untuk memulai. Arduino mega cocok sebagai rancangan pelindung untuk Arduino *Deumilanove* atau *Diecimila*



**Gambar II.1** Arduino Mega 2560

Sumber: <http://www.arduino.cc/>

### II.5.2. Arsitektur Arduino Mega 2560

Arduino Mega 2560 terbentuk dari prosesor yang dikenal dengan Mikrokontroler ATmega 2560. Mikrokontroler ATmega 2560 memiliki beberapa fitur / spesifikasi yang menjadikannya sebagai solusi pengendali yang efektif untuk berbagai keperluan. Fitur-fitur tersebut antara lain :

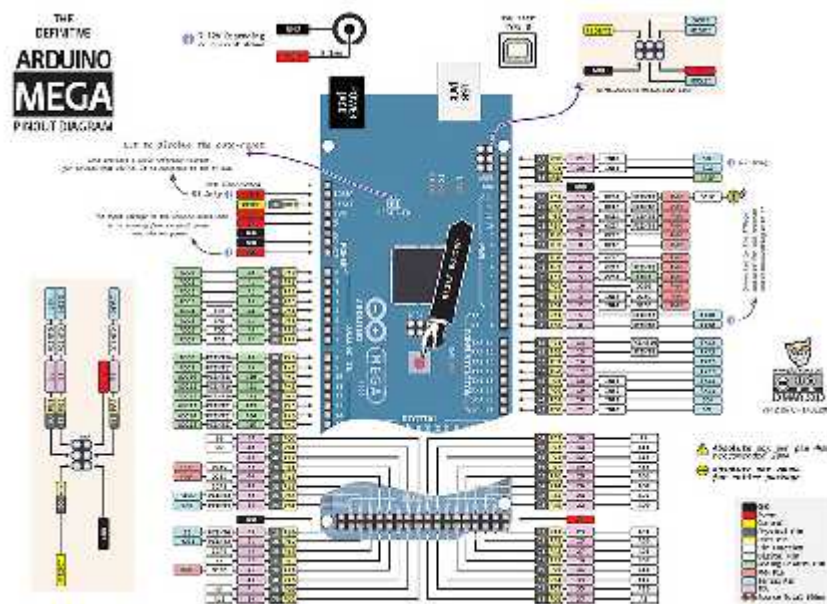
1. Tegangan Operasi sebesar 5 V
2. Tegangan input sebesar 6 – 20 V tetapi yang direkomendasikan untuk ATmega 2560 sebesar 7 – 12 V.
3. Pin digital I/O sebanyak 54 pin dimana 14 pin merupakan keluaran dari PWM.
4. Pin input analog sebanyak 16 pin
5. Arus DC pin I/O sebesar 40 mA sedangkan Arus DC untuk pin 3.3V sebesar 50 mA
6. Flash memory 156 Kb yang mana 8 Kb digunakan oleh *bootloader*.
7. SRAM 8 Kbyte
8. EEPROM 4 Kbyte
9. Serta mempunyai 2 Port UARTs untuk komunikasi serial.



**Gambar II.2 ATmega 2560 pada Arduino Mega 2560**

(Atmel Corporation.2014)

### II.5.3. Konfigurasi Pin Arduino Mega 2560



**Gambar II.3 Konfigurasi Pin Arduino Mega 2560**

Sumber: <https://arduino-info.wikispaces.com/>

Pada gambar II.3 terlihat konfigurasi pin pada *board* Arduino Mega 2560.

Pin-pin tersebut terdiri dari:

1. VCC adalah tegangan catu digital
2. GND adalah Ground
3. Port A (PA7..PA0) Port A adalah sebuah port I/O 8 bit dua arah dengan *internal pull-up resistor* (dipilih untuk masing-masing bit). Penyangga output Port A memiliki karakter penggerak karakteristik dengan kedua sink tinggi dan kemampuan sumber. Sebagai input, pin Port A eksternal *pulled low* sumber arus jika resistor pull-up aktif. Pin port A dinyatakan tri ketika sebuah kondisi *reset* menjadi aktif, bahkan jika waktu tidak

berjalan. Port A juga menyajikan fungsi dari berbagai fitur spesial dari Atmega640/1280/1281/2560/2561.

4. Port B (PB7..PB0) Port B adalah sebuah port I/O 8 bit dua arah dengan *internal pull-up resistor* ( dipilih untuk masing-masing bit). Penyangga output Port B memiliki karakter penggerak karakteristik dengan kedua sink tinggi dan kemampuan sumber. Sebagai input, pin Port A eksternal *pulled low* sumber arus jika resistor pull-up aktif. Pin port A dinyatakan tri ketika sebuah kondisi *reset* menjadi aktif, bahkan jika waktu tidak berjalan. Port B mempunyai kemampuan bergerak lebih baik daripada port lainnya.
5. Port C (PC7..PC0) Port C adalah sebuah port I/O 8 bit dua arah dengan *internal pull-up resistor* ( dipilih untuk masing-masing bit). Penyangga output Port C memiliki karakter penggerak karakteristik dengan kedua sink tinggi dan kemampuan sumber. Sebagai input, pin Port C eksternal *pulled low* sumber arus jika *resistor pull-up* aktif. Pin port C dinyatakan tri ketika sebuah kondisi reset menjadi aktif, bahkan jika waktu tidak berjalan.
6. Port D (PD7..PD0) Port D adalah sebuah port I/O 8 bit dua arah dengan *internal pull-up resistor* ( dipilih untuk masing-masing bit). Penyangga output Port D memiliki karakter penggerak karakteristik dengan kedua sink tinggi dan kemampuan sumber. Sebagai input, pin Port D eksternal *pulled low* sumber arus jika *resistor pull-up* aktif. Pin port D dinyatakan

tri ketika sebuah kondisi reset menjadi aktif, bahkan jika waktu tidak berjalan.

7. Port E (PE7..PE0) Port E adalah sebuah port I/O 8 bit dua arah dengan *internal pull-up resistor* ( dipilih untuk masing-masing bit). Penyangga output Port E memiliki karakter penggerak karakteristik dengan kedua sink tinggi dan kemampuan sumber. Sebagai input, pin Port E eksternal *pulled low* sumber arus jika *resistor pull-up* aktif. Pin port E dinyatakan tri ketika sebuah kondisi reset menjadi aktif, bahkan jika waktu tidak berjalan.
8. Port F (PF7..PF0) Port F disajikan sebagai masukan analog ke A/D converter. Port F juga menyajikan sebuah port I/O 8 bit dua arah, jika A/D Converter tidak digunakan. Pin port dapat menyediakan *internal pull-up resistor* ( dipilih untuk masing-masing bit). Penyangga output Port F memiliki karakter penggerak karakteristik dengan kedua sink tinggi dan kemampuan sumber. Sebagai input, pin Port F eksternal *pulled low* sumber arus jika *resistor pull-up* aktif. Pin port F dinyatakan tri ketika sebuah kondisi *reset* menjadi aktif, bahkan jika waktu tidak berjalan. Jika antarmuka JTAG mengizinkan, *pull-up resistor* pada pin PF7(TDI), PF5(TMS), dan PF4(TCK) akan iaktifkan bahkan jika terjadi *reset*. Port F juga menyajikan fungsi dari antarmuka JTAG.
9. Port G (PG7..PG0) Port G adalah sebuah port I/O 6 bit dua arah dengan *internal pull-up resistor* (dipilih untuk masing-masing bit). Penyangga output Port G memiliki karakter penggerak karakteristik dengan kedua

sink tinggi dan kemampuan sumber. Sebagai input, pin Port G eksternal *pulled low* sumber arus jika *resistor pull-up* aktif. Pin port G dinyatakan tri ketika sebuah kondisi reset menjadi aktif, bahkan jika waktu tidak berjalan.

10. Port H (PH7..PH0) Port H adalah sebuah port I/O 8 bit dua arah dengan internal *pull-up resistor* ( dipilih untuk masing-masing bit). Penyangga output Port H memiliki karakter penggerak karakteristik dengan kedua sink tinggi dan kemampuan sumber. Sebagai input, pin Port H eksternal *pulled low* sumber arus jika *resistor pull-up* aktif. Pin port H dinyatakan tri ketika sebuah kondisi *reset* menjadi aktif, bahkan jika waktu tidak berjalan.

11. Port J (PJ7..PJ0) Port J adalah sebuah port I/O 8 bit dua arah dengan *internal pull-up resistor* ( dipilih untuk masing-masing bit). Penyangga output Port J memiliki karakter penggerak karakteristik dengan kedua sink tinggi dan kemampuan sumber. Sebagai input, pin Port J *eksternal pulled low* sumber arus jika *resistor pull-up* aktif. Pin port J dinyatakan tri ketika sebuah kondisi *reset* menjadi aktif, bahkan jika waktu tidak berjalan.

12. Port K (PK7..PK0) Port K disajikan sebagai masukan analog ke A/D *converter*. Port K adalah sebuah port I/O 8 bit dua arah dengan *internal pull-up resistor* ( dipilih untuk masing-masing bit). Penyangga output Port K memiliki karakter penggerak karakteristik dengan kedua sink tinggi dan kemampuan sumber. Sebagai input, pin Port K eksternal *pulled low*

sumber arus jika resistor *pull-up* aktif. Pin port K dinyatakan tri ketika sebuah kondisi reset menjadi aktif, bahkan jika waktu tidak berjalan.

13. Port L (PL7..PL0) Port L adalah sebuah port I/O 8 bit dua arah dengan *internal pull-up resistor* ( dipilih untuk masing-masing bit). Penyangga *output* Port L memiliki karakter penggerak karakteristik dengan kedua sink tinggi dan kemampuan sumber. Sebagai input, pin Port L eksternal *pulled low* sumber arus jika *resistor pull-up* aktif. Pin port L dinyatakan tri ketika sebuah kondisi reset menjadi aktif, bahkan jika waktu tidak berjalan.

14. Reset Input reset. Sebuah level rendah pada pin ini untuk lebih panjang dari pada panjang minimum pulsa akan menghasilkan sebuah reset, bahkan jika waktu tidak berjalan. Panjang minimum pulsa dijelaskan pada “Sistem dan karakter reset” pada halaman 360. Pulsa terpendek tidak dijamin menghasilkan sebuah reset .

15. XTAL1

Input ke *inverting amplifier oscillator* dan input ke internal jalur operasi waktu.

16. XTAL2 Keluaran dari *inverting oscillator amplifier*

17. AVCC AVCC merupakan pin tegangan catu untuk port F dan A/D *Converter*. AVCC dapat terhubung secara eksternal ke VCC, bahkan jika ADC tidak digunakan jika ADC digunakan, ADC akan terhubung ke VCC melalui sebuah *low pass filter*.

18. AREF AREF adalah pin referensi analog untuk A/D *Converter*. (Atmel Corporation.2014).

## II.6. Bahasa Pemrograman

### II.6.1. Bahasa Pemrograman C

Struktur penulisan bahasa C secara umum terdiri atas empat blok, yaitu:

1. *Header*,
2. Deklarasi konstanta global dan atau variabel,
3. Fungsi dan atau prosedur (bisa di bawah program utama),
4. Program utama.

Secara umum, pemrograman C paling sederhana dilakukan dengan hanya menuliskan program utamanya saja, yaitu:

```

/* fungsi utama */
void main()
{
    Statemen-statemen;
}
/*fungsi-fungsi lain yang ditulis oleh pemrogram komputer*/
Fungsi_fingsi_lain()
{
    Statemen-statemen;
}

```

(M. Ary Heryanto, ST dan Ir. Wisnu Adi P. ; 2008 : 8)

### II.6.2. Pemrograman *IDE* Arduino

Arduino memiliki open-source yang memudahkan untuk menulis kode dan meng-upload board ke arduino. Arduino *IDE* (*Integrated Development Enviroment*) ini merupakan media yang digunakan untuk memberikan informasi

kepada arduino sehingga dapat memberikan output sesuai dengan apa yang diinginkan. Aplikasi arduino IDE ini dapat dijalankan di windows, Mac OS X, dan linux (Moh. Kamalul Wafi, 2014: 2). berikut merupakan gambaran tampilan arduino IDE :



**Gambar II.4. Tampilan Arduino IDE**

(Sumber: *arduino.stackexchange.com* dikunjungi pada tanggal 13 Februari 2017)

Dalam arduino terhubung dengan arduino *IDE* ini dengan hanya menekan tombol RESET. tombol ini dirancang untuk menjalankan program yang telah di upload ke arduino *board*, tombol ini juga terhubung dengan ATMEga 328 melalui kapasitor 100nf. *IDE (Integrated Development Enviroment)* arduino merupakan

pemrograman dengan menggunakan bahasa C. Setiap program *IDE* arduino (biasa disebut *sketch*) mempunyai dua buah fungsi yang harus ada, yaitu :

a. `void setup() { }`

Semua kode didalam kurung kurawal akan dijalankan hanya satu kali ketika program *IDE* Arduino dijalankan untuk pertama kalinya.

b. `void loop() { }`

Fungsi ini akan dijalankan setelah setup (fungsi *void* setup) selesai. Setelah dijalankan satu kali fungsi ini akan dijalankan lagi, dan lagi secara terus menerus sampai catu daya (*power*) dilepaskan.

*Compiler* merupakan modul yang berfungsi mengubah bahasa *processing* (kode pemrograman) kedalam kode biner karena kode biner adalah satu-satunya bahasa program yang dipahami oleh mikrokontroler. Sedangkan *upload* program adalah modul yang berfungsi memasukkan kode biner kedalam mikrokontroler. Pada *software* Arduino *IDE* memiliki fitur *compiler* sedangkan untuk *upload* program menggunakan *USBisp* yang dihubungkan ke port *ISP* pada papan rangkaian mikrokontroler. Pada proses ini akan merubah bahasa pemrograman dari digital ke bahasa analog yang dapat dipahami mikrokontroler (Anandya Bagus Venesa dan Wibowo Basuki Dwi, 2004 : 5).

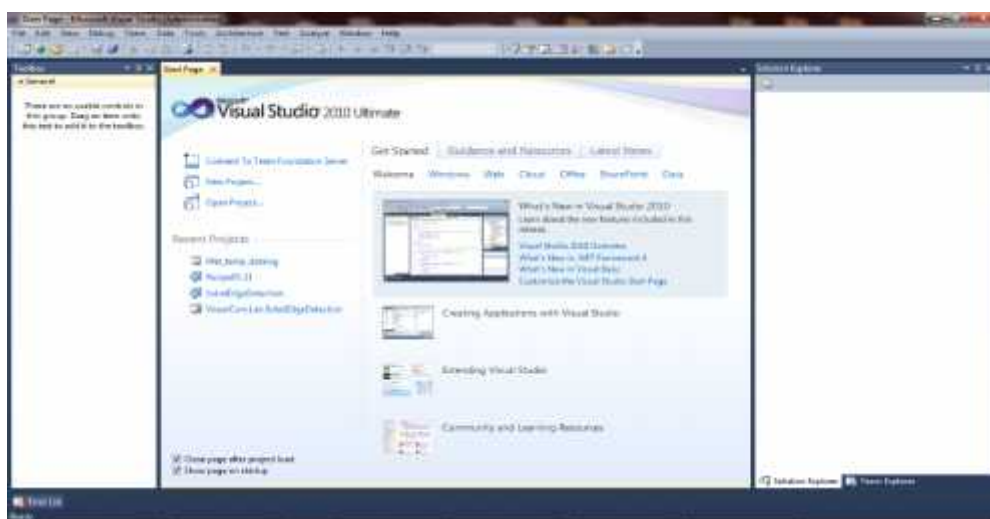
### II.6.3. Visual Studio 2010

Visual Studio 2010 merupakan edisi terbaru dari Visual Studio sebelumnya, yaitu Visual Studio 2008. Visual Studio merupakan lingkungan kerja (*IDE – Integrated Development Environment*) yang digunakan untuk

pemrograman .NET yang dapat digunakan untuk beberapa bahasa pemrograman, seperti Visual Basic (VB), C# (baca *C Sharp*), Visual C++, J# (baca *J Sharp*), F# (baca *F Sharp*) dan lain – lain.

Bahasa pemrograman Visual Basic merupakan salah satu bahasa yang sangat populer hingga kini dan merupakan salah satu solusi untuk menciptakan aplikasi pada sistem operasi Windows, baik Windows 7, Windows Server 2008 dan Windows Mobile 6.1. Hal ini dikarenakan kemudahan yang diberikan Visual Basic dan *IDE* Visual Studio yang digunakan untuk menciptakan sebuah aplikasi. (Wahana Komputer, 2010 : 2 )

Berikut adalah tampilan dari Visual Studio 2010 :



**Gambar II.5. IDE Visual Studio 2010.**



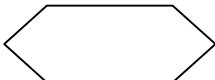
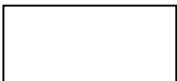
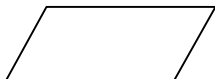
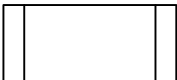
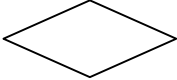


Sumber : [www.microsoft.com](http://www.microsoft.com)

## II.7. Flowchart

Prinsip kerja dari pemodelan di atas dapat digambarkan melalui *flowchart*. *Flowchart* adalah gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta pernyataannya. Gambaran ini dinyatakan dengan simbol dan

dengan demikian setiap simbol menggambarkan proses tertentu. Hubungan antar proses digambarkan dengan garis penghubung (Zarlis et al, 2007). *Flowchart* disusun dengan simbol-simbol. Simbol ini dipakai sebagai alat bantu menggambarkan proses di dalam program. Adapun tabel simbol *flowchart* pada Tabel II.1:

**Tabel II.1. Tabel Simbol Flowchart**

<b>SIMBOL</b>	<b>NAMA</b>	<b>FUNGSI</b>
	<b>TERMINATOR</b>	Permulaan/akhir program
	<b>GARIS ALIR (FLOW LINE)</b>	Arah aliran program
	<b>PREPARATION</b>	Proses inisialisasi/pemberian harga awal
	<b>PROSES</b>	Proses perhitungan/proses pengolahan data
	<b>INPUT/OUTPUT DATA</b>	Proses input/output data, parameter, informasi
	<b>PREDEFINED PROCESS (SUB PROGRAM)</b>	Permulaan sub program/proses menjalankan sub program
	<b>DECISION</b>	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	<b>ON PAGE CONNECTOR</b>	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	<b>OFF PAGE CONNECTOR</b>	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda

(Sumber : Dr. Suarga, M. sc., M. Math. , Ph D. , 2012.)