

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Sistem Pengendalian Proses

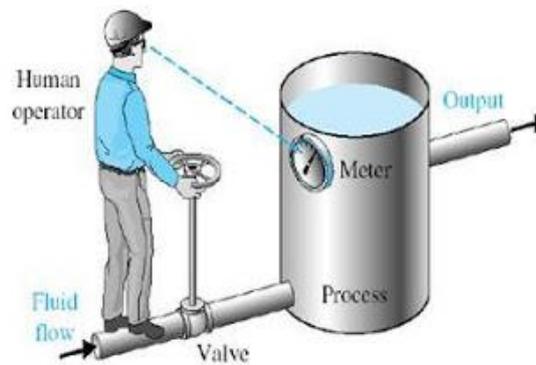
Sistem pengendalian atau sistem kontrol adalah susunan beberapa instrumentasi yang terangkai membentuk suatu sistem untuk mengendalikan sistem lain atau sistem itu sendiri. Sistem pengendalian yang diterapkan dalam teknologi proses disebut sistem pengendalian proses. Sistem pengendalian dalam suatu proses digunakan untuk mengendalikan suatu besaran/parameter sehingga parameter tersebut berada pada suatu harga atau range yang diinginkan. Hampir semua proses dalam dunia industri membutuhkan peralatan-peralatan otomatis untuk mengendalikan parameter - parameter prosesnya. (Arfy Sasmita, 2010)

Ada banyak parameter yang harus dikendalikan di dalam suatu proses. Di antaranya, yang paling umum, adalah tekanan (*pressure*) di dalam sebuah *vessel* atau pipa, aliran (*flow*) didalam pipa, suhu (*temperature*) di unit proses seperti *heat exchanger*, atau permukaan zat cair (*level*) disebuah tangki. Dan ada beberapa parameter lain yang cukup penting dan juga perlu dikendalikan karena kebutuhan spesifik proses, diantaranya: pH di industry petrokimia, warna produk di suatu fasilitas pencairan gas (NGL) dan sebagainya.

Gabungan serta kerja alat – alat pengendalian otomatis itulah yang dinamakan sistem pengendalian proses (*process control system*). Sedangkan semua peralatannya yang membentuk sistem pengendalian disebut instrumentasi pengendalian proses (*process control instrumentation*). (Frans Gunterus, 1994)

## 2.2 Prinsip Pengendalian Proses

Pada gambar 1 menunjukkan prinsip dari pengendalian proses, ketika seorang operator harus mengamati ketinggian *level*, kemudian mengevaluasi apakah *level* yang ada sudah seperti yang dikehendakinya. Kalau *level* tidak sama dengan yang dikehendakinya, operator harus memperkirakan seberapa banyak *valve* perlu lebih ditutup atau lebih dibuka. Selanjutnya, operator harus benar-benar mengubah bukaan *valve* sesuai dengan yang diperkirakan tadi.



Gambar 1. Pengendalian *Level* di Dalam Tangki Oleh Manusia

(Frans Gunterus, 1994)

Prinsip pengendalian proses, jika dikaji lebih jauh lagi, dalam mengendalikan proses operator mengerjakan empat langkah berikut:

### **Mengukur – Membandingkan – Menghitung – Mengoreksi**

Pada pengendalian level pada suatu tangki, operator mengamati ketinggian level, artinya operator sedang melakukan langkah **mengukur** process variable. Dalam hal ini yang berperan sebagai process variable adalah tinggi level pada tangki. Selanjutnya, operator akan melakukan langkah **membandingkan**, apakah hasil pengukuran tadi sesuai dengan apa yang dikehendakinya. Besar process

variable yang dikehendaki disebut set point (SP). Apabila terjadi selisih antara process variable dan set point, maka selisih tersebut disebut error.

$$\text{Error} = \text{set point} - \text{process variable}$$

*Process variable* bisa lebih besar atau bisa juga lebih kecil daripada *set point*. Oleh karena itu, *error* bisa negatif bisa juga positif. Berdasarkan besarnya *error* yang dihasilkan, operator menentukan ke arah mana dan seberapa besar koreksi bukaan *valve* perlu dilakukan. Bila *error* bernilai negatif (berarti *process variable* lebih besar dari *set point*) operator harus mengurangi *flow* dengan lebih menutup *valve*. Sebaliknya, bila *error* positif (berarti *process variable* lebih kecil dari *set point*) operator harus menambah *flow* dengan lebih membuka *valve*.

Seorang operator tidak akan sembarang membuka atau menutup *valve*. Ia juga akan memperkirakan seberapa banyak *valve* perlu lebih dibuka atau lebih ditutup. Pada tahapan itu, operator sebenarnya sedang melakukan langkah **menghitung**. Langkah berikutnya yang perlu dikerjakan oleh operator adalah mengubah bukaan *valve* sesuai dengan hasil perbandingan dan perhitungan tadi. Langkah terakhir inilah yang disebut dengan langkah **mengoreksi**.

Keempat langkah tersebut yang dilakukan oleh operator yaitu; mengukur, membandingkan, menghitung dan mengoreksi, seluruhnya dapat dikerjakan oleh instrumentasi. Manusia, kemudian, sama sekali tidak menentukan keempat langkah tadi. Operator hanya perlu menentukan besarnya *set point*, dan semuanya kan dikerjakan secara otomatis oleh instrument. Sistem pengendalian semacam itulah yang disebut sistem pengendalian otomatis (*automatic control system*). (Frans Gunterus, 1994)

### 2.3 Laju Alir Fluida

Fluida adalah suatu zat yang bisa mengalami perubahan-perubahan bentuknya secara continue/terus-menerus bila terkena tekanan/gaya geser walaupun relatif kecil atau bisa juga dikatakan suatu zat yang mengalir, kata fluida mencakup zat cair, gas, air, dan udara karena zat-zat ini dapat mengalir. Sebaliknya batu dan benda-benda keras (seluruh zat-zat padat tidak dapat dikategorikan sebagai fluida karena zat-zat tersebut tidak bisa mengalir secara continue). (Irma Suryani, 2013)

Laju alir adalah banyaknya zat yang mengalir dengan kecepatan tertentu persatuan waktu. Laju alir suatu fluida dapat dibedakan ke dalam 3 macam pola aliran, yaitu aliran laminar, aliran turbulen dan aliran dari campuran keduanya (transisi).

#### 1. Aliran Laminar

Aliran laminar terjadi ketika kecepatan fluida dalam pipa rendah dan partikel fluida bergerak lancar. Kecepatan dari partikel diseluruh fluida mengambil bentuk parabola. Reynold menetapkan bahwa untuk aliran laminar bilangan Reynoldnya di bawah 2000 ( $Re < 2000$ ).

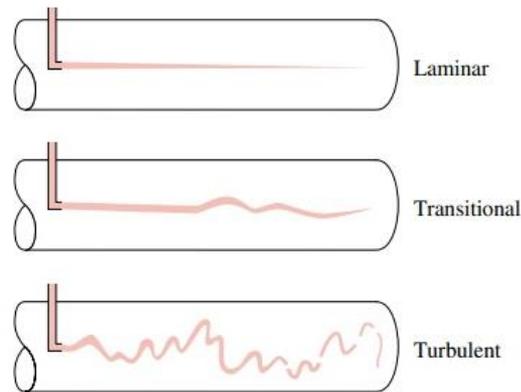
#### 2. Aliran Turbulen

Aliran turbulen terjadi ketika kecepatan aliran fluida tinggi dan pergerakan dari partikel-partikel fluida sangat tidak menentu karena mengalami percampuran serta putaran partikel antar lapisan. Aliran akan menjadi turbulen apabila bilangan Reynold lebih besar dari 4000 ( $Re > 4000$ ).

#### 3. Aliran Transisi

Aliran transisi merupakan aliran peralihan dari aliran laminar ke aliran turbulen. Keadaan peralihan ini tergantung pada viskositas fluida,

kecepatan dan lain-lain yang menyangkut geometri aliran. Bilangan Reynold aliran transisi yaitu berada pada ( $2000 < Re < 4000$ ).



Gambar 2. Pola Aliran Fluida  
(Kurniawan Teguh, 2012)

#### 2.4 Hukum Bernoulli

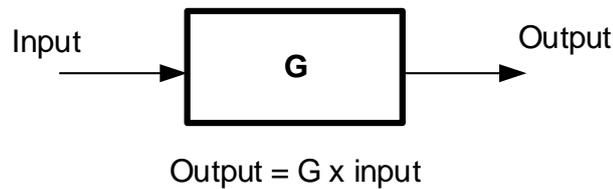
Hukum Bernoulli ditemukan pertama kali oleh Daniel Bernoulli, seorang matematikawan dari Swiss pada tahun 1700-an. Persamaan Bernoulli memiliki hubungan antara tekanan, kecepatan fluida dan elevasi dalam sistem aliran. Hukum Bernoulli menyatakan bahwa tekanan fluida di tempat yang kecepatannya tinggi lebih kecil daripada di tempat yang kecepatannya lebih rendah. Jadi semakin besar kecepatan fluida dalam suatu pipa maka tekanan yang dihasilkan semakin kecil dan sebaliknya semakin kecil kecepatan fluida dalam suatu pipa maka semakin besar tekanannya. Hukum ini diterapkan pada zat cair yang mengalir dengan kecepatan berbeda dalam suatu pipa. Prinsip Bernoulli adalah sebuah istilah di dalam mekanika fluida yang menyatakan bahwa peningkatan kecepatan aliran fluida akan menimbulkan penurunan tekanan pada aliran tersebut.

## 2.5 Dinamika Proses

Dinamika Proses adalah sifat-sifat proses yang terjadi di dalam suatu system pengendalian, ada *process variable* yang cepat berubah dengan berubahnya *manipulated variable* (bukan *control valve*), ada pula yang lambat berubah. Ada proses yang sifatnya lamban, ada yang reaktif, ada yang mudah stabil, dan ada pula yang mudah menjadi tidak stabil. Sehingga, pengendalian proses akan berbeda-beda.

Dalam dinamika proses sering dikaitkan dengan unsur kapasitas (*capacity*) dan kelambatan (*lag*). Dalam bahasa ilmu sistem pengendalian, dikatakan kapasitas proses tergantung pada sumber energi yang bekerja pada proses. Kalau sumber energi kecil dan kapasitas prosesnya besar, proses akan menjadi lambat. Kalau sumber energinya besar dan kapasitasnya prosesnya kecil, proses akan menjadi cepat.

Kata kapasitas dan kelambatan itulah yang kemudian dipakai sebagai standar (ukuran) untuk menyatakan dinamika proses secara kualitatif. Dalam bentuk kualitatif, proses dibedakan menjadi proses cepat dan proses lambat, atau kapasitas besar dan kapasitas kecil. Selain bentuk kualitatif, dinamika proses juga dinyatakan secara kuantitatif dalam bentuk *transfer function*. Secara umum, *transfer function* suatu elemen proses ditandai dengan huruf G, dan gambar dalam bentuk diagram kotak seperti pada gambar berikut:



Dimana G = Transfer Function Proses

Gambar 3. Diagram Kotak Sebuah Proses

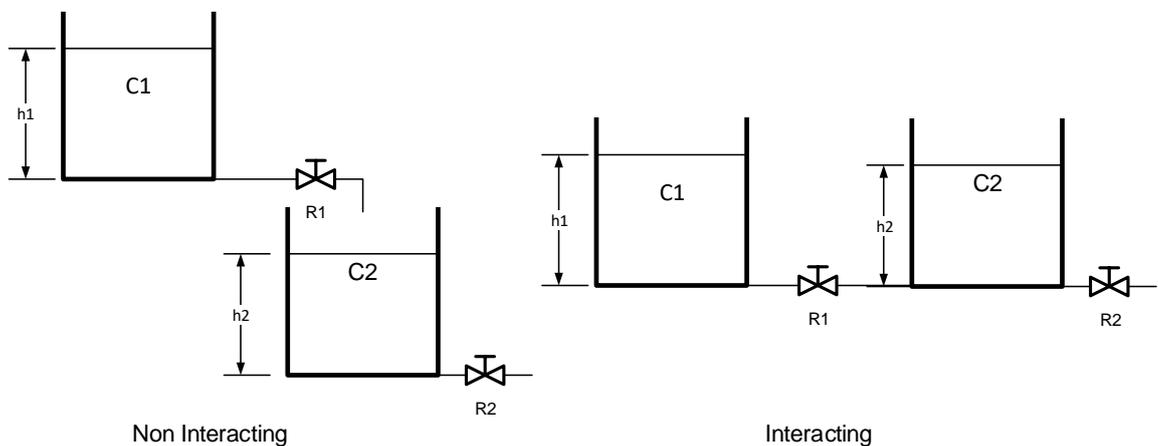
(Frans Gunterus, 1994)

Transfer function (G) mempunyai dua unsur gain, yaitu *steady state gain* yang sifatnya statik, dan *dynamic gain* yang sifatnya dinamik. Unsur *dynamic gain* muncul karena elemen proses mengandung unsur kelambatan. Oleh karena itu, bentuk *transfer function* elemen proses hampir pasti berbentuk persamaan matematika fungsi waktu yang ada dalam wujud persamaan differensial.

Persamaan differensial adalah persamaan yang menyatakan adanya kelambatan antara *input-output* suatu elemen proses. Semakin banyak pangkat persamaan differensial, semakin lambat dinamika proses. Sebuah elemen proses kemudian dinamai orde satu (*first order process*) karena persamaan differensialnya berpangkat satu. Dinamai proses orde dua (*second order process*) karena differensialnya berpangkat dua. Dinamai proses orde banyak (*higher order process*) karena differensialnya berorde banyak. Pangkat persamaan dalam differensial juga mencerminkan jumlah kapasitas yang ada di elemen proses. Suatu orde satu juga disebut *one capacity process* atau *single capacity process*. Proses orde dua juga disebut *two capacity process*. Proses orde banyak juga disebut *multicapacity process*. (Frans Gunterus, 1994)

## 2.6 Proses Orde Dua atau Orde Banyak

Salah satu cara untuk menyatakan hubungan *input-output* suatu proses adalah dengan menjabarkannya dalam bentuk matematik, yang disebut *transfer function*. Bentuk *transfer function* elemen proses, hampir selalu ada dalam bentuk persamaan diferensial. Bila persamaan diferensial itu berpangkat satu, prosesnya disebut proses orde satu. Bila persamaan diferensial itu berpangkat banyak, prosesnya disebut proses orde banyak.



Gambar 4. Proses Orde Dua

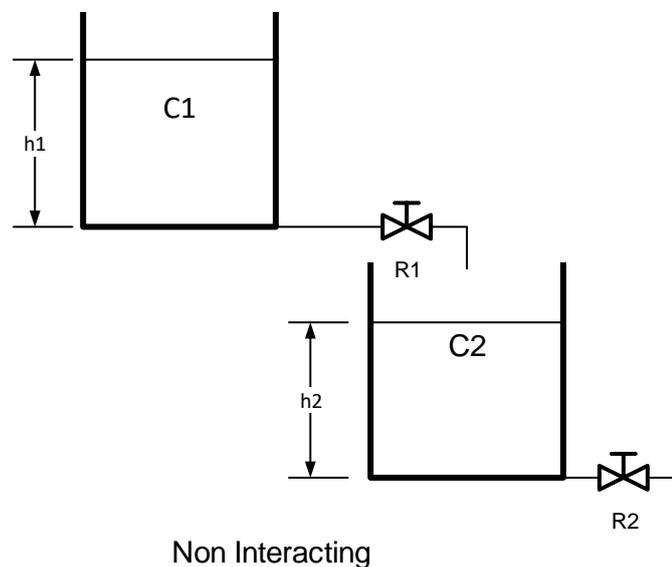
(Frans Gunterus, 1994)

Pangkat persamaan diferensial suatu elemen proses ternyata juga sama dengan jumlah unsur kapasitas yang ada didalam proses. Proses orde satu mempunyai satu unsur kapasitas, proses orde dua mempunyai dua unsur kapasitas, dan proses orde banyak juga mempunyai banyak unsur kapasitas. Proses orde dua merupakan gabungan dua proses orde satu. Di dalam konfigurasi *input-output*, keempat tangki itu dirangkaikan dalam dua konfigurasi yang berbeda

yaitu konfigurasi *interacting-capacities* dan konfigurasi *non-interacting capacities*. Pada proses orde dua *interacting-capacities* ketinggian level dikedua tangki akan mempengaruhi besarnya flow yang keluar dari tangki pertama sedangkan pada proses orde dua *non-interacting capacities* ketinggian level dikedua tangki jelas tidak saling mempengaruhi. (Frans Gunterus, 1994)

## 2.7 Proses Orde Dua Non-Interacting Capacities

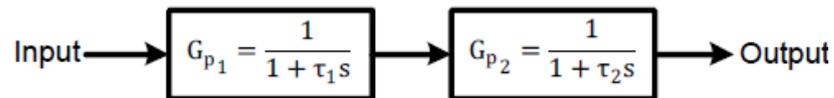
Proses orde dua merupakan gabungan dua proses orde satu. Pada proses orde dua *non-interacting*, ketinggian *level* di kedua tangki tidak saling mempengaruhi. *Level* di tangki kedua tidak akan mempengaruhi besar kecilnya laju alir yang keluar dari tangki pertama. Hal ini dapat dilihat pada gambar 5.



Gambar 5. Proses Orde Dua *Non-interacting*

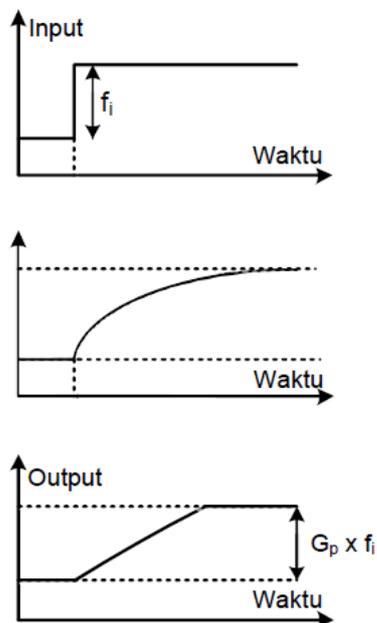
(Frans Gunterus, 1994)

Seperti pada proses orde satu, *transfer function* proses orde dua *non-interacting* juga merupakan persamaan diferensial fungsi waktu. Bahkan, persamaan diferensialnya sekarang berpangkat dua karena prosesnya memang mempunyai dua *lag time*.



Gambar 6. Transfer Function Proses Orde Dua Non-Interacting  
(Frans Gunterus, 1994)

Hubungan antara *input-output* proses orde dua non-interacting dapat dilihat pada Gambar 7.



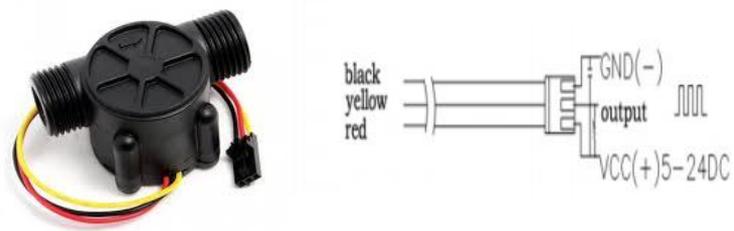
Gambar 7. Kurva Waktu Proses Orde Dua *Non-Interacting*  
(Frans Gunterus, 1994)

Kurva waktu tersebut menunjukkan tahap demi tahap perubahan yang terjadi pada *level* di tangki pertama ( $h_1$ ) atas perubahan  $F_i$  dan perubahan *level* di tangki kedua ( $h_2$ ) atas perubahan *level* di tangki pertama ( $h_1$ ). Karena sifat prosesnya tetap *self-regulation*, setelah ada gangguan keseimbangan dengan bertambahnya  $F_i$  sebanyak  $f_i$ , *level* di tangki pertama ( $h_1$ ) akan naik seperti layaknya proses orde satu *self-regulation*. Tangki kedua akan menerima penambahan *flow* dari tangki pertama yang naiknya sebanding dengan kenaikan *level* di tangki pertama ( $h_1$ ). Akibatnya, *level* di tangki kedua ( $h_2$ ) akan naik juga, tetapi secara jauh lebih lambat lagi. Bila pada keadaan akhir (*steady state*) ternyata *level* ( $h_2$ ) naik 20% sebagai akibat dari kenaikan  $F_i$  sebanyak 10%, *steady state gain* proses orde dua ini dikatakan sama dengan dua ( $G_p = 2$ ). (Frans Gunterus, 1994)

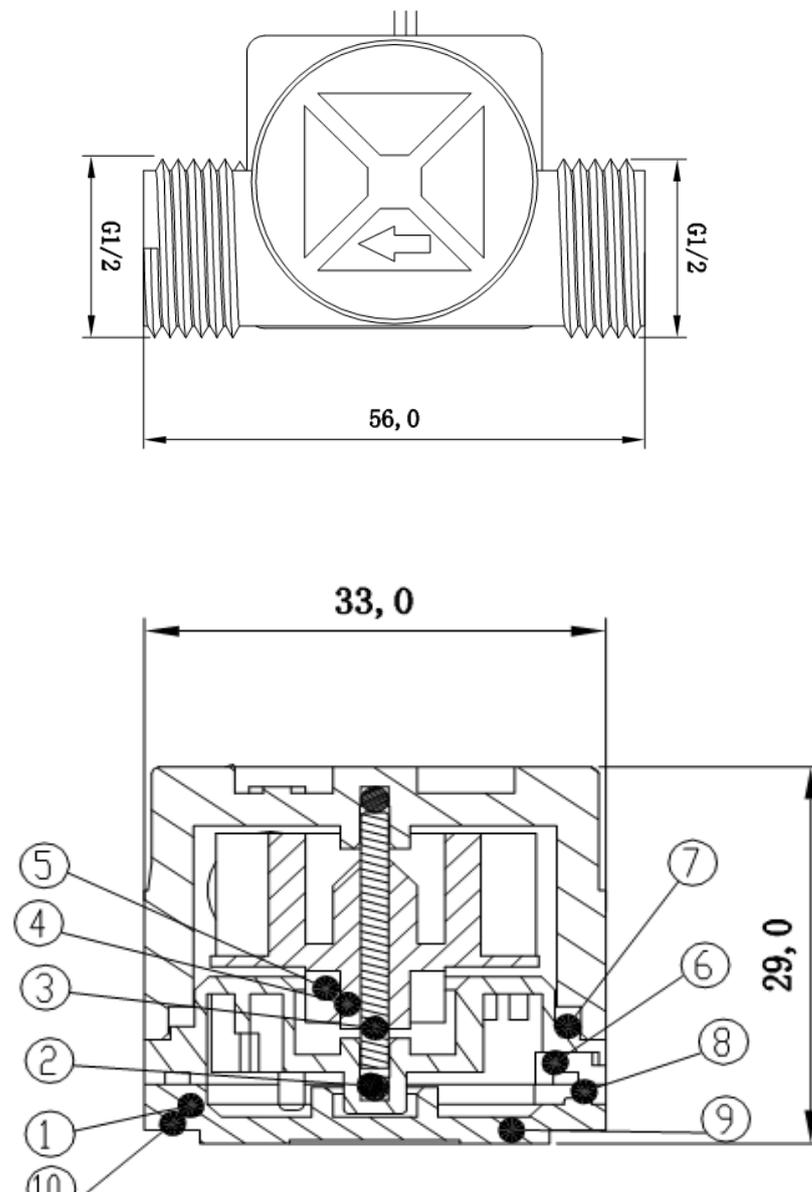
## **2.8 Water Flow Sensor**

### **2.8.1 Pengertian Water Flow Sensor**

Water Flow sensor terdiri dari tubuh katup plastik, rotor air, dan sensor hall efek. Ketika air mengalir melalui pipa dalam sensor ini, maka akan mengenai rotor dan membuatnya berputar. Kecepatan putar rotor akan berubah ketika kecepatan aliran air berubah pula. Output dari sensor hall-effect akan sebanding dengan pulsa yang digenerate rotor. Kelebihan sensor ini adalah hanya membutuhkan 1 sinyal (SIG) selain jalur 5V dc dan Ground. Pada *Water Flow Sensor* mempunyai 3 kabel, kabel warna merah untuk tegangan sensor, kabel warna hitam untuk ground dan kabel warna kuning untuk sinyal sensor. Contoh *water flow sensor* dapat dilihat pada gambar 8.



Gambar 8. Fisik dan Skematik Instalasi *Water Flow Sensor*



Gambar 9. Mechanic Dimensi *Water Flow Sensor*

Tabel 1. Komponen Water Flow Sensor

No	Nama Komponen	Jumlah	Material
1	Valve Body	1	<i>Glass fiber</i>
2	Stainless steel bead	1	<i>Stainless steel</i>
3	Axis	1	<i>Stainless steel</i>
4	Impeller	1	<i>POM</i>
5	Ring Magnet	1	<i>Ferrite</i>
6	Middle Ring	1	<i>Glass fiber</i>
7	O-seal Ring	1	<i>Rubber</i>
8	Electronic Seal Ring	1	<i>Rubber</i>
9	Cover	1	<i>Glass fiber</i>
10	Screw	4	<i>Stainless steel</i>
11	Cable	1	1007 AWG

### 2.8.2 Spesifikasi Water Flow Sensor

Tabel 2. Spesifikasi Water Flow Sensor

No	Spesifikasi	Jumlah
1	Tegangan Kerja	5V-24V
2	Arus Maksimum	15 mA (DC5V)
3	Berat Sensor	43 g
4	Diameter Eksternal	20 mm
5	Tingkat Aliran	0,5~ 60L / menit
6	Suhu Pengoperasian	0°C~ 80°
7	Operasi Kelembaban	35%~ 90% RH
8	Operasi Tekanan	1.75Mpa

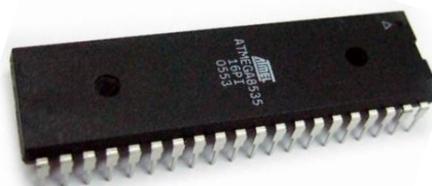
### 2.8.3 Prinsip Kerja Sensor

Prinsip kerja sensor ini adalah dengan memanfaatkan fenomena efek Hall. Efek Hall ini didasarkan pada efek medan magnetik terhadap partikel bermuatan yang bergerak. Ketika ada arus listrik yang mengalir pada device efek Hall yang

ditempatkan dalam medan magnet yang arahnya tegak lurus arus listrik, pergerakan pembawa muatan akan berbelok ke salah satu sisi dan menghasilkan medan listrik. Medan listrik terus membesar hingga gaya Lorentz yang bekerja pada partikel menjadi nol. Perbedaan potensial antara kedua sisi divais tersebut disebut potensial Hall. Potensial Hall ini sebanding dengan medan magnet dan arus listrik yang melalui *divice*. (Muhhamad Syarif, 2014).

## 2.9 Microcontroller ATMEGA 8

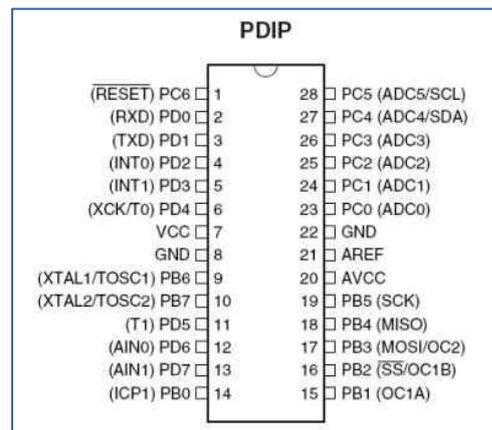
Mikrokontroler ATmega8 merupakan mikrokontroler keluarga AVR 8bit. ATMEGA 8 adalah mikrokontroler CMOS 8-bit daya rendah berbasis arsitektur RISC yang ditingkatkan. Kebanyakan instruksi dikerjakan pada satu siklus *clock*, ATMEGA 8 mempunyai throughput mendekati 1 MIPS per MHz membuat disain dari sistem untuk mengoptimasi konsumsi daya versus kecepatan proses. Beberapa tipe mikrokontroler jenis AVR yang sama dengan ATmega8 ini antara lain ATmega8535, ATmega16, ATmega32, ATmega328, dll. Yang membedakan antara beberapa mikrokontroler diatas adalah ukuran memori, banyaknya GPIO (pin input/output), peripheral (USART, timer, counter, dll). Dari segi ukuran fisik, ATmega8 memiliki ukuran fisik lebih kecil dibandingkan dengan beberapa mikrokontroler yang lain. Namun untuk segi memori dan periperial lainnya ATmega8 tidak kalah dengan yang lainnya karena ukuran memori dan periperialnya relatif sama dengan ATmega8535, ATmega32, dll, hanya saja jumlah GPIO lebih sedikit dibandingkan mikrokontroler yang lain.



Gambar 10. Mikrokontroller ATMEGA8

### 2.9.1 Fungsi dan Susunan Pin

Susunan pin – pin dari IC mikrokontroler ATMEGA 8 diperlihatkan pada gambar dibawah ini. IC ini tersusun dari 28 pin yang memiliki beberapa fungsi tertentu.



Gambar 11. Pin Out IC Mikrokontroler ATMEGA8

ATMega8 memiliki 3 buah PORT utama yaitu PORT B, PORT C, dan PORT D dengan total pin input/output sebanyak 28 pin yang masing – masing pin – nya memiliki fungsi yang sebagai input/output digital atau difungsikan sebagai periperial lainnya. Berikut akan dijelaskan tentang kegunaan dari masing – masing pin pada ATMega8.

1. VCC

Merupakan supply tegangan untuk digital

2. GND

Merupakan ground untuk semua komponen yang membutuhkan grounding

3. Port B

Adalah 8 buah pin mulai dari pin B.0 sampai dengan pin B.7. Tiap pin dapat digunakan sebagai input dan juga output. Port B merupakan sebuah 8-bit

bit-directional I/O port dengan internal pull-up resistor. Sebagai input, pin – pin yang terdapat pada port B yang secara eksternal diturunkan, maka akan mengeluarkan arus jika pull-up resistor diaktifkan. Jika ingin menggunakan tambahan kristal, maka cukup untuk menghubungkan kaki dari kristal ke kaki pada pin port B. Namun jika tidak digunakan, maka cukup untuk dibiarkan saja. Pengguna kegunaan dari masing – masing kaki ditentukan dari clock fuse setting-nya.

#### 4. Port C

Port C merupakan sebuah 7-bit bi-directional I/O yang di dalam masing – masing pin terdapat pull-up resistor. Jumlah pin-nya hanya 7 buah mulai dari C.0 sampai dengan pin C.6. Sebagai keluaran / output, port C memiliki karakteristik yang sama dalam hal kemampuan menyerap arus ( sink ) ataupun mengeluarkan arus ( source)

#### 5. Reset/PC6

Jika RSTDISBL Fuse diprogram, maka PC6 akan berfungsi sebagai pin I/O. Untuk diperhatikan juga bahwa pin ini memiliki karakteristik yang berbeda dengan pin – pin yang terdapat pada port C. Namun jika RSTDISBL Fuse tidak diprogram, maka pin ini akan berfungsi sebagai input reset. Dan jika level tegangan yang masuk ke pin ini rendah dan pulsa yang ada lebih pendek dari pulsa minimum, maka akan menghasilkan suatu kondisi reset meskipun clock-nya tidak berkerja.

#### 6. Port D

Port D merupakan 8-bit bi-directional I/O dengan internal pull-up resistor. Fungsi dari port ini sama dengan port – port yang lain. Hanya saja pada port ini tidak terdapat kegunaan-kegunaan yang lain. Pada port ini hanya

berfungsi sebagai masukan dan keluaran saja atau biasa disebut dengan I/O.

#### 7. AVCC

Pada pin ini memiliki fungsi sebagai power supply tegangan untuk ADC. Untuk pin ini harus dihubungkan secara terpisah dengan VCC karena pin ini digunakan untuk analog saja. Bahkan jika ADC pada AVR tidak digunakan, tetap saja disarankan untuk menghubungkan secara terpisah dengan VCC. Cara menghubungkan AVCC adalah melewati low-pass.

#### 8. AREF

Merupakan pin referensi analog jika menggunakan ADC. Pada AVR status Register mengandung beberapa informasi mengenai hasil dari kebanyakan hasil eksekusi intruksi aritmatik. Informasi ini dapat digunakan untuk altering arus program sebagai kegunaan untuk meningkatkan performa pengoperasian. Perlu diketahui bahwa register ini di-update setelah semua operasi ALU ( Arithmetic Logic Unit ). Hal tersebut seperti yang telah tertulis dalam datasheet khususnya pada bagian Instruction Set Reference

### 2.9.2 Jenis dan Ukuran Memori pada ATMEGA8

Flash memori merupakan lokasi penyimpanan program yang di buat. File hex hasil kompilasi program nantinya akan dimasukkan ke mikrokontroler melalui alat yang disebut downloader/programmer. File hex tersebut nantinya akan disimpan pada sebuah memori yang disebut flash memori. Pada saat melakukan proses pemograman (*coding*) biasanya diperlukan variabel atau tempat menampung data. Pada saat program dijalankan oleh mikrokontroler, kemudian

terdapat proses yang melibatkan variabel tersebut ( misalnya operasi aritmatika ) maka data dari variabel tersebut akan disimpan pada memori yang bernama SRAM. Kemudian jika ingin menyimpan sebuah data seperti halnya pada flashdisk ( data tidak hilang ketika tidak ada aliran listrik ) maka data tersebut dapat di simpan pada sebuah memori yang bernama EEPROM. EEPROM sama halnya seperti hardisk, flashdisk yang ada pada komputer yaitu sebagai tempat penyimpanan data yang tidak terpengaruh terhadap aliran listrik.

### 2.9.3 Fitur pada ATmega8

1. Saluran I/O sebanyak 28 buah terbagi menjadi 3 port.
2. ADC sebanyak 6 saluran dengan 4 saluran 10 bit dan 2 saluran 8 bit.
3. Tiga buah timer counter, dua diantaranya memiliki fasilitas pembanding.
4. CPU dengan 32 buah register
5. Watchdog timer dan oscillator internal.
6. SRAM sebesar 1K byte.
7. Memori flash sebesar 8K Bytes system Self-programable Flash
8. Unit interupsi internal dan eksternal.
9. Port antarmuka SPI.
10. EEPROM sebesar 512 byte.
11. Port USART ( *Universal Synchronous and Asynchronous Serial Receiver and Transmitter* ) untuk komunikasi serial.

### 2.10 Arduino

Arduino adalah pengendali mikro single-board yang bersifat *open-source*, yang dirancang untuk memudahkan penggunaan elektronik dalam berbagai

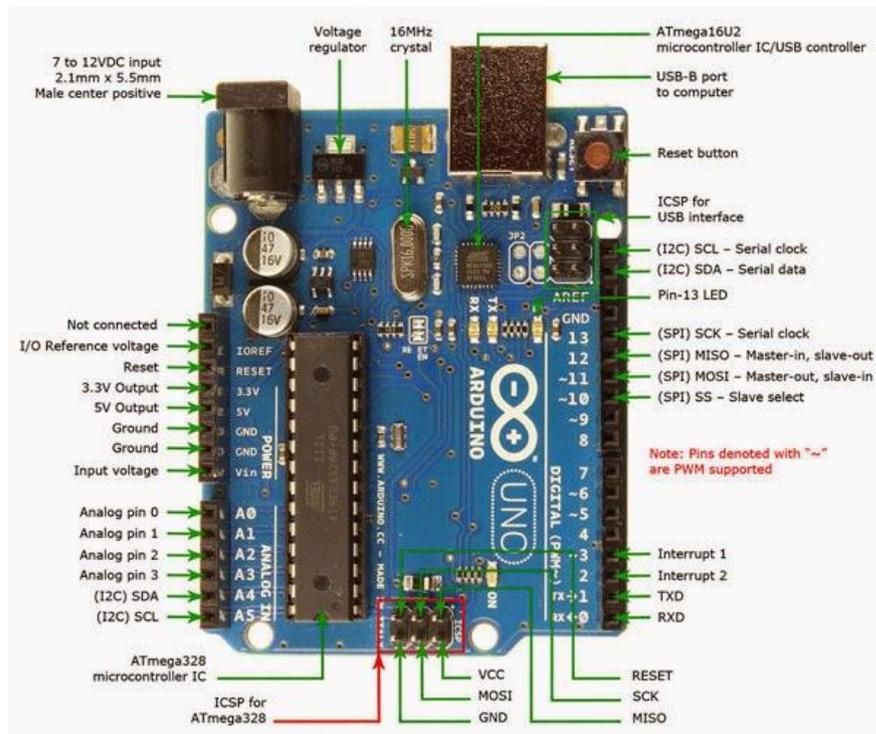
bidang. Hardware (perangkat keras)-nya memiliki prosesor Atmel AVR dan software (perangkat lunak)-nya memiliki bahasa pemrograman sendiri. Open source IDE yang digunakan untuk membuat aplikasi mikrokontroler yang berbasis platform arduino. Mikrokontroler single-board yang bersifat open source hardware dikembangkan untuk arsitektur mikrokontroler AVR 8 bit dan ARM 32 bit. Dari pengertian di atas, dapat disimpulkan bahwa Arduino adalah kit atau papan rangkaian elektronik open source yang di dalamnya terdapat komponen utama yaitu sebuah chip mikrokontroler dengan jenis AVR. Mikrokontroler itu sendiri adalah chip atau IC (integrated circuit) yang bisa diprogram menggunakan komputer. Tujuan menanamkan program pada mikrokontroler adalah agar rangkaian elektronik dapat membaca input, memproses input tersebut dan kemudian menghasilkan output seperti yang diinginkan. Jadi mikrokontroler bertugas sebagai otak yang mengendalikan input, proses, dan output sebuah rangkaian elektronik.

Kelebihan Arduino antara lain :

- Tidak perlu perangkat *chip programmer* karena di dalamnya sudah ada *bootloader* yang akan menangani *upload* program dari komputer.
- Sudah memiliki sarana komunikasi USB, sehingga pengguna laptop yang tidak memiliki port serial/RS323 bisa menggunakannya.
- Memiliki modul siap pakai (*shield*) yang bisa ditancapkan pada *board* arduino. Contohnya *shield* GPS, *Ethernet*, dan lain-lain.

### **2.10.1 Bagian-Bagian Papan Arduino**

Dengan mengambil contoh sebuah papan Arduino tipe USB, bagian-bagiannya dapat dijelaskan sebagai berikut:



Gambar 12. Bagian-bagian papan Arduino

#### a) 14 pin input/output digital (0-13)

Berfungsi sebagai input atau output, dapat diatur oleh program. Khusus untuk 6 buah pin 3, 5, 6, 9, 10 dan 11, dapat juga berfungsi sebagai pin analog output dimana tegangan output-nya dapat diatur. Nilai sebuah pin output analog dapat diprogram antara 0 – 255, dimana hal itu mewakili nilai tegangan 0 – 5V.

#### b) USB

Berfungsi untuk:

- Memuat program dari komputer ke dalam papan
- Komunikasi serial antara papan dan computer
- Memberi daya listrik kepada papan

**c) Sambungan SV1**

Sambungan atau *jumper* untuk memilih sumber daya papan, apakah dari sumber eksternal atau menggunakan USB. Sambungan ini tidak diperlukan lagi pada papan Arduino versi terakhir karena pemilihan sumber daya eksternal atau USB dilakukan secara otomatis.

**d) Q1 = Kristal (quartz crystal oscillator)**

Jika mikrokontroler dianggap sebagai sebuah otak, maka kristal adalah jantungnya karena komponen ini menghasilkan detak-detak yang dikirim kepada mikrokontroler agar melakukan sebuah operasi untuk setiap detak-nya. Kristal ini dipilih yang berdetak 16 juta kali per detik (16MHz).

**e) Tombol Reset S1**

Untuk me-reset papan sehingga program akan mulai lagi dari awal. Perhatikan bahwa tombol reset ini bukan untuk menghapus program atau mengosongkan mikrokontroler.

**f) In = Circuit Serial Programming (ICSP)**

Port ICSP memungkinkan pengguna untuk memprogram mikrokontroler secara langsung, tanpa melalui *bootloader*. Umumnya pengguna Arduino tidak melakukan ini sehingga ICSP tidak terlalu dipakai walaupun disediakan.

**g) IC 1 = Mikrokontroler Atmega**

Komponen utama dari papan Arduino, di dalamnya terdapat CPU, ROM dan RAM.

**h) X1 = Sumber Daya External**

Jika hendak disuplai dengan sumber daya eksternal, papan Arduino dapat diberikan tegangan dc antara 9-12V.

### **i) 6 Pin Input analog (0-5)**

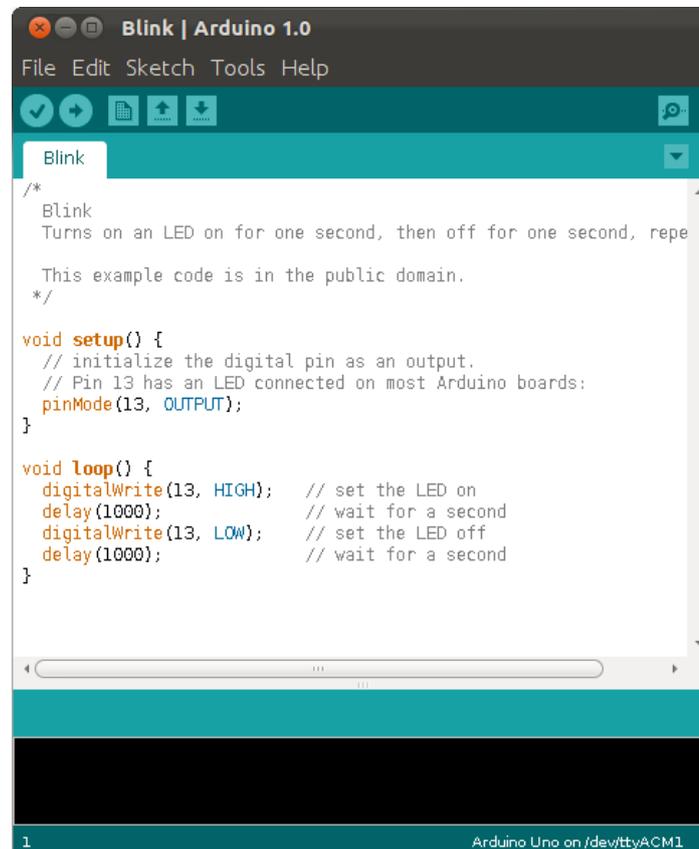
Pin ini sangat berguna untuk membaca tegangan yang dihasilkan oleh sensor analog, seperti sensor flow. Program dapat membaca nilai sebuah pin input antara 0 – 1023, dimana hal itu mewakili nilai tegangan 0 – 5V.

#### **2.10.2 IDE (*Integrated Development Environment*) Arduino**

IDE merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui software inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler.

Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan library C/C++ yang biasa disebut *Wiring* yang membuat operasi input dan output menjadi lebih mudah. Arduino IDE ini dikembangkan dari software *Processing* yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino. Program yang ditulis dengan menggunakan *Arduino Software* (IDE) disebut sebagai sketch. Sketch ditulis dalam suatu editor teks dan disimpan dalam file dengan ekstensi *.ino*. Teks editor pada *Arduino Software* memiliki fitur seperti cutting/paste dan seraching/replacing sehingga memudahkan kamu dalam menulis kode program. Pada Software Arduino IDE, terdapat semacam message box berwarna hitam yang berfungsi menampilkan status, seperti pesan error,

compile, dan upload program. Di bagian bawah paling kanan Software Arduino IDE, menunjukkan board yang terkonfigurasi beserta COM Ports yang digunakan.



Gambar 13. Tampilan Arduino IDE

Bagian – bagian IDE Arduino:

- *New*  
Befungsi untuk membuat *Sketch* baru
- *Open*  
Befungsi untuk membuka *sketch* yang pernah dibuat dan membuka kembali untuk dilakukan editing atau re-upload pada Arduino.
- *Verify*  
Befungsi untuk melakukan checking kode yang dibuat apakah sudah sesuai dengan kaidah pemrograman yang ada atau belum.

- *Save*  
Berfungsi untuk menyimpan *Sketch* yang telah dibuat
- *Upload*  
Berfungsi untuk melakukan kompilasi program atau kode yang dibuat menjadi bahasa yang dapat dipahami oleh Arduino.
- *Serial Monitor*  
Berfungsi untuk membuka serial monitor. Serial monitor merupakan jendela yang menampilkan data apa saja yang dikirimkan atau dipertukarkan antara arduino dengan sketch pada port serialnya. Serial Monitor sangat berguna ketika ingin membuat program atau melakukan *debugging* tanpa menggunakan LCD pada Arduino. Serial monitor ini dapat digunakan untuk menampilkan nilai proses, nilai pembacaan, bahkan pesan error.

### **2.11 Borland Delphi**

Borland Delphi adalah suatu bahasa pemrograman (*development language*) yang digunakan untuk merancang suatu aplikasi program. Delphi termasuk dalam pemrograman bahasa tinggi (*high level language*) artinya adalah perintah-perintah programnya menggunakan bahasa yang mudah dipahami oleh manusia. Bahasa pemrograman Delphi disebut bahasa prosedural artinya mengikuti urutan tertentu. Dalam membuat aplikasi perintah-perintah, Delphi menggunakan lingkungan pemrograman visual.

Delphi merupakan generasi penerus dari Turbo Pascal. Pemrograman Delphi dirancang untuk beroperasi dibawah sistem operasi Windows. Program ini mempunyai beberapa keunggulan, yaitu produktivitas, kualitas, pengembangan

perangkat lunak, kecepatan kompilasi, pola desain yang menarik serta diperkuat dengan bahasa perograman yang terstruktur dalam struktur bahasa perograman *Object Pascal*.

Sebagian besar pengembang Delphi menuliskan dan mengkompilasi kode program di dalam lingkungan pengembang aplikasi atau *Integrated Development Environment (IDE)*. Lingkungan kerja IDE ini menyediakan sarana yang diperlukan untuk merancang, membangun, mencoba, mencari atau melacak kesalahan, serta mendistribusikan aplikasi. Sarana-sarana inilah yang memungkinkan pembuatan prototipe aplikasi menjadi lebih mudah dan waktu yang diperlukan untuk mengembangkan aplikasi menjadi lebih singkat.

### **2.11.1 File-file Penyusun Project**

Sepintas sebuah program aplikasi yang dapat dibuat dengan menggunakan Delphi hanya terdiri dari file *project* dan sebuah unit. Namun kenyataannya terdapat beberapa file yang dibentuk pada saat membangun sebuah program aplikasi. Berikut ini merupakan file-file penyusun projek yang terdapat pada program Delphi, yaitu :

#### **1. File Project (.D pr ) dan file Unit (.P as )**

Sebuah program Delphi terbangun dari modul-modul source code yang disebut unit. Delphi menggunakan sebuah file projek (.D pr) untuk menyimpan program utama. File sumber untuk unit biasanya berisi sebagian besar kode di dalam aplikasi, file ini ditandai dengan ekstensi (.P as). Setiap aplikasi atau projek terdiri atas file projek tunggal atau lebih dalam file unit.

## 2. File *Form* (. Dfm )

File *form* adalah file biner yang dibuat oleh Delphi untuk menyimpan informasi yang berkaitan dengan *form*.

## 3. File Resource (. Res )

File resource merupakan file biner yang berisi sebuah ikon yang digunakan oleh project. File ini secara terus menerus di-update atau diubah oleh Delphi sehingga file ini tidak bisa diubah oleh pemakai. Dengan menambahkan file resource pada aplikasi dan menghubungkan dengan file project dapat menggunakan editor resource, misalnya editor untuk membuat file resource.

## 4. File Project Options (. Dof ) dan File Desktop Settings (. Dsk )

File *project options* merupakan file yang berisi *options-options* dari suatu *project* yang dinyatakan melalui perintah Options dari menu *Project*. Sedang file *desktop setting* berisi *option-option* yang dinyatakan melalui perintah *Environment Options* dari menu *Tools*. Perbedaan di antara kedua jenis file tersebut adalah bahwa file *project options* dimiliki oleh setiap project sedangkan file desktop setting dipakai untuk lingkungan Delphi. Apabila ada kerusakan pada kedua jenis file tersebut dapat mengganggu proses kompilasi. Prosedur yang dapat kita tempuh untuk menangani gangguan tersebut adalah dengan menghapus kedua jenis file tersebut yaitu . **Dof** dan . **Dsk** karena kedua file tersebut akan terbentuk secara otomatis pada saat menyimpan *project*.

## 5. File *Backup* (.~d p, . ~d f, . ~p a)

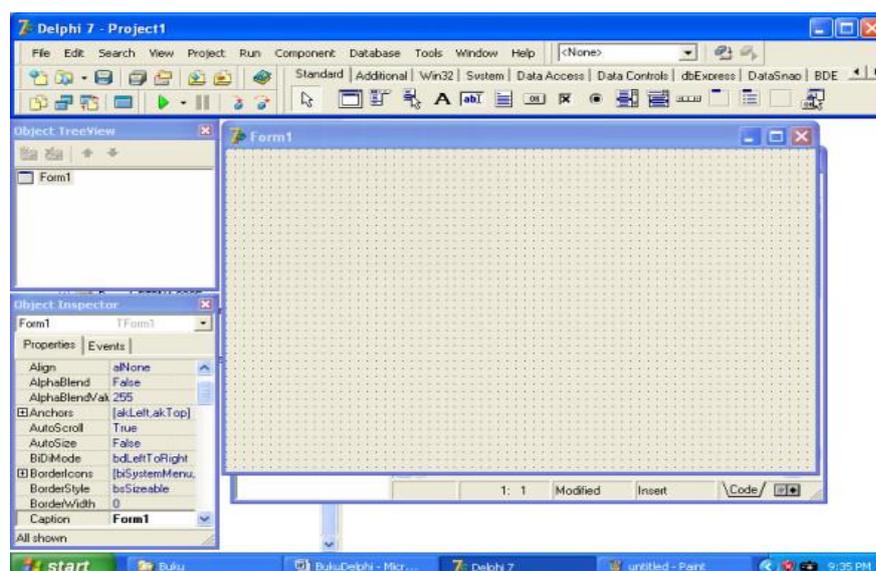
File-file dengan ekstensi di atas merupakan file *backup* dari suatu *project*, *form* dan unit. Ketiga jenis file tersebut akan terbentuk pada saat proses penyimpanan untuk yang kedua kalinya. Karena ketiga file tersebut berjenis *backup* (cadangan) maka ketiga jenis file tersebut berisi salinan terakhir dari file-file utama sebelum disimpan lebih lanjut.

## 6. File jenis lain

File-file dengan ekstensi lain yang dapat ditemukan dalam folder tempat penyimpanan program aplikasi selain yang memiliki ekstensi yang telah disebutkan pada umumnya adalah file-file yang dibentuk oleh compiler dan beberapa file Windows yang digunakan Delphi.

### 2.11.2 Integrated Development Environment (IDE) Delphi

IDE (Integrated Development Environment) merupakan lingkungan/wilayah dimana seluruh tools atau komponen-komponen yang dibutuhkan untuk merancang atau membangun aplikasi program. Berikut adalah gambar dari keseluruhan tampilan Delphi



Gambar 14. Tampilan Delphi

Secara umum IDE Delphi di kelompokkan kepada 8 bagian yaitu :

### 1. Main Menu

Merupakan penunjuk ke seluruh fasilitas yang disediakan aplikasi Delphi.



Gambar 15. Menu Pemrograman Delphi

### 2. Toolbar/Speedbar

Merupakan Icon (*Shortcut*) yang dirancang untuk lebih memudahkan menjangkau fasilitas yang ada pada Delphi.



Gambar 16. Toolbar Delphi

### 3. Component Palette

Merupakan komponen-komponen VCL (*Visual Component Library*) yang dikelompokkan kedalam Tab-tab, komponen komponen inilah yang akan digunakan untuk merancang interface atau antar muka aplikasi.



Gambar 17. Component Pallette Delphi

#### 4. Form Designer

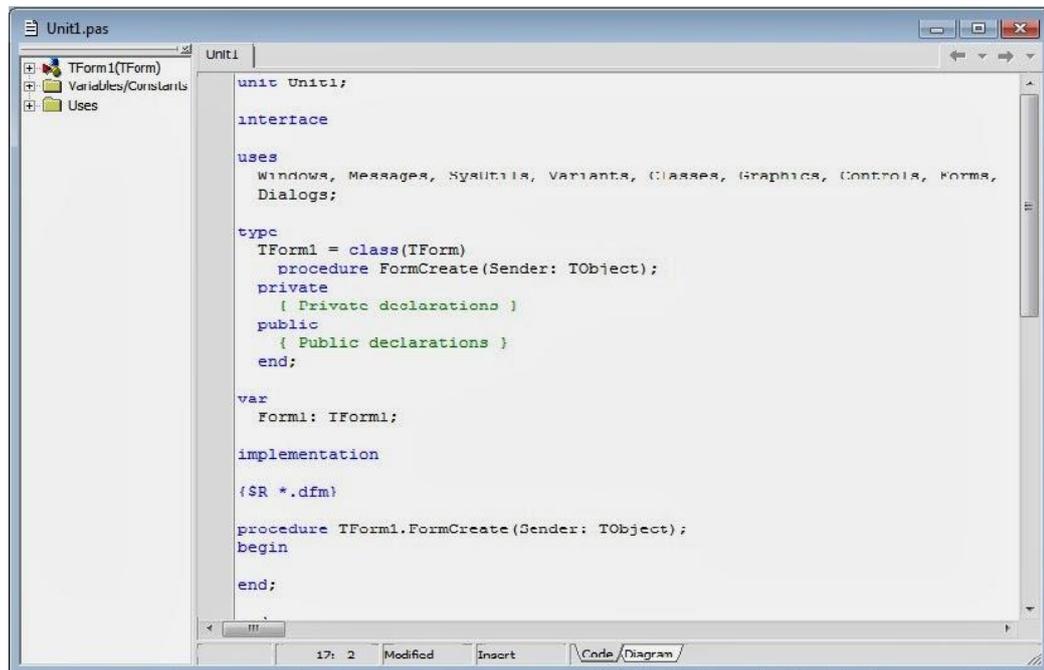
Merupakan interface (antar muka) aplikasi yang akan dibangun, Form akan menampung seluruh komponen yang akan digunakan dalam proses perancangan sebuah aplikasi dengan Delphi.



Gambar 18. Form Designer Delphi

#### 5. Code Editor

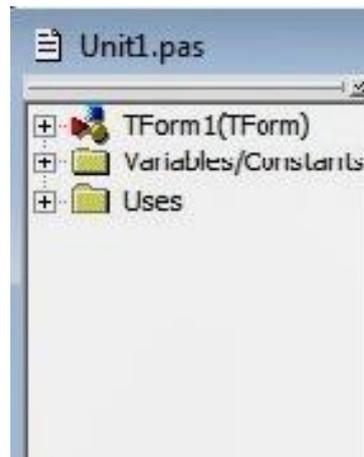
Code editor merupakan tempat untuk menuliskan kode program menggunakan bahasa object Pascal. Kode program tidak perlu di tulis secara keseluruhan karena Delphi sudah menyediakan blok atau kerangka untuk menulis kode program.



Gambar 19. Code Editor Delphi

## 6. Code Explorer

Digunakan untuk memudahkan berpindah antar file unit di dalam jendela code editor. Code explorer berisi daftar yang menampilkan semua tipe, class, properti, method, variabel global, rutin global yang telah didefinisikan di dalam unit. Saat memilih sebuah item dalam code explorer, kursor akan berpindah menuju implementasi dari item yang dipilih di dalam code editor.



Gambar 20. Code Explorer Delphi

## 7. Object Inspector

Object inspector digunakan untuk mengubah properti atau karakteristik dari suatu komponen. Terdiri dari 2 tab yaitu :

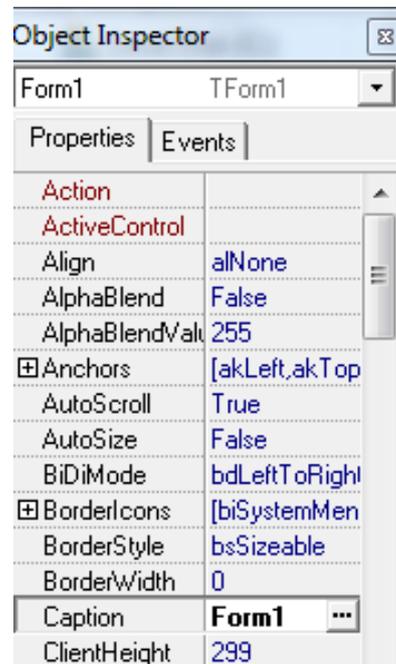
### a. Properties

Digunakan untuk menentukan setting suatu objek. Satu objek memiliki beberapa properti yang dapat diatur langsung dari object inspector maupun melalui kode program. Setting ini mempengaruhi cara kerja objek tersebut saat aplikasi dijalankan.

### b. Event

Merupakan bagian yang dapat diisi dengan kode program tertentu yang berfungsi untuk menangani event-event (berupa sebuah procedure) yang dapat direspon oleh sebuah komponen. Event adalah peristiwa atau kejadian yang diterima oleh suatu objek, misal: klik, drag, dan lain-lain. Event yang diterima objek akan memicu Delphi menjalankan kode program yang ada didalamnya. Misalnya ingin sesuatu dikerjakan pada saat form ditutup, maka

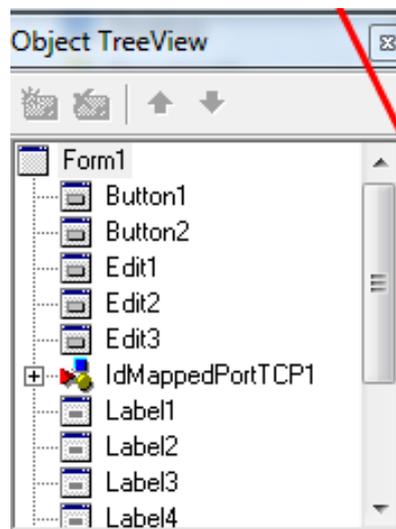
untuk menyatakan tindakan tersebut (berupa sebuah procedure)  
menggunakan OnClose



Gambar 21. Object Inspector Delphi

#### 8. Object Tree View

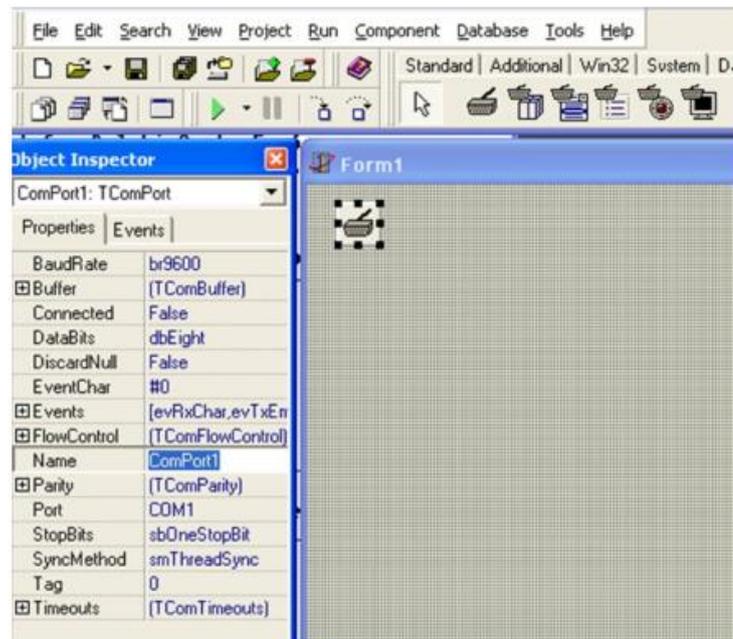
Object tree view berisi daftar komponen yang sudah diletakkan di form designer.



Gambar 22. Object Tree View Delphi

### 2.11.3 Kode Perintah Pada Delphi

Berikut adalah contoh bahasa pemrograman pada Delphi yang dapat menghubungkan ke mikrokontroler.



Gambar 23. Bahasa Pemrograman pada Delphi