**BAB II**

**TEORI DASAR**

Pada bab ini dibahas tentang Teknik Instrumentasi, Komponen Elektronika, *Mikrokontroler*, *Code Vision AVR*, Kendala Pengukuran Kedalaman dan Pengecekan Kebenaran Data Kedalaman.

**2.1 Teknik Instrumentasi**

Teknik instrumentasi adalah peralatan yang digunakan untuk pengukuran dan pengendalian suatu sistem. Sistem instrumentasi merupakan gabungan dari beberapa peralatan. Secara umum, sistem instrumentasi terdiri dari empat elemen dasar yaitu peralatan masukan *(input),* pengkondisi sinyal *(signal conditioning)*, sistem pengolah dan peralatan pencatat.

Peralatan masukan *(input)* merupakan peralatan pertama yang menerima besaran yang akan diukur. *Output* yang dihasilkan dari peralatan masukan berupa sinyal-sinyal listrik. Peralatan masukan (*input*) terdiri dari dua komponen yaitu sensor dan *transducer.*

Sensor merupakan bagian perangkat yang mendeteksi variabel fisik suatu media yang sedang diukur. *Transducer* merupakan bagian perangkat pengukur yang mengubah variabel yang terukur menjadi variabel dalam bentuk lain misalnya tegangan. Sensor dan *transducer* harus memiliki karakteristik sebagai berikut :

1. Ketelitian yang tinggi
2. Jangkauan pengukuran yang panjang
3. Mudah untuk dikalibrasi
4. Keandalan tinggi

Pengkondisi sinyal merupakan peralatan yang berfungsi untuk menerima sinyal yang dihasilkan dari peralatan masukan *(input)* dimana sinyal tersebut akan diperkuat. Sistem pengolah adalah peralatan yang berfungsi mengolah hasil penguatan sinyal oleh pengkondisi sinyal. Peralatan pencatat merupakan peralatan yang berfungsi mencatat dan menampilkan hasil pengolahan dari sistem pengolah agar dapat diamati oleh pengamat. Peralatan pencatat terdiri dari pencatat dengan gerak relatif dan pencatat dengan hasil digital. Skema sistem instrumentasi dapat dilihat pada gambar 2.1.



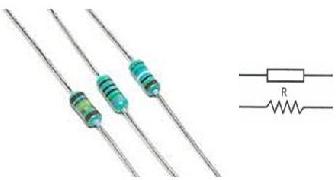
**Gambar 2.1 Skema Sistem Instrumentasi**

**2.2 Komponen Elektronika**

Komponen elektronika yang dipakai dalam tugas akhir ini adalah *Motor DC, Rotary* *Encoder,* Optocoupler, *(Light Emitting Diode)* LED*, Resistor*, *Multiturn*, *IC LM 324N, (Liquid Crystal Display)* LCD*,* dan *Mikrokontroller ATMega8535.*

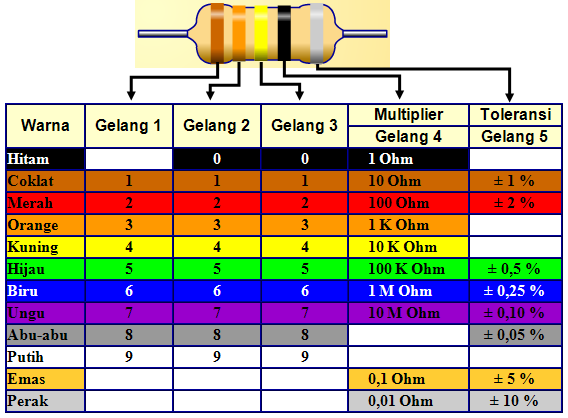
**2.2.1 Resistor**

*Resistor* adalah komponen elektronik yang digunakan untuk membatasi aliran arus listrik. Besarnya nilai tahanan dinyatakan dengan satuan *ohm* dilambangkan dengan Ω. Nilai tahanan *resistor* dilambangkan dengan rangkaian warna yang terdapat pada badan *resistor*. Bentuk dan simbol *resistor* dapat dilihat pada gambar 2.2 dan kode warna *resistor* dapat dilihat pada table 2. 1.



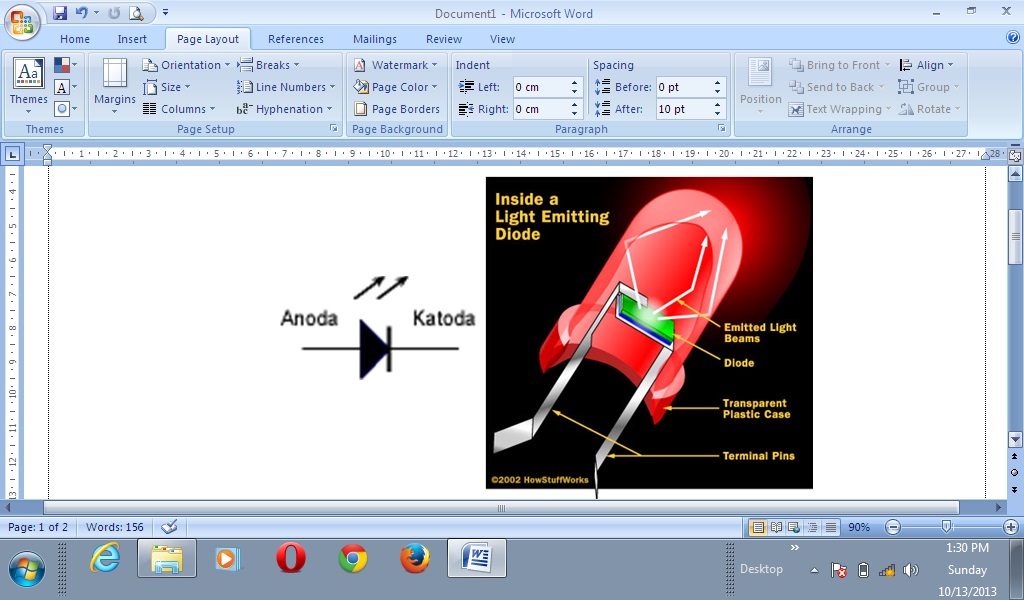
**Gambar 2.2 Bentuk Dan Simbol Resistor**

**Tabel 2.1 Kode Warna Resistor**



**2.2.2 Light Emitting Diode (LED)**

*Light Emitting Diode* (LED) merupakan komponen yang dapat mengeluarkan emisi cahaya. LED merupakan produk temuan lain setelah dioda, strukturnya juga sama dengan dioda, tetapi belakangan ditemukan bahwa elektron yang menerjang sambungan *P-N* juga melepaskan energi berupa energi panas dan energi cahaya. LED dibuat agar lebih efisien jika mengeluarkan cahaya untuk mendapatkan emisi cahaya pada semi konduktor. Bentuk dan simbol *Light Emitting Diode*(LED)dapat dilihat pada gambar 2.3.



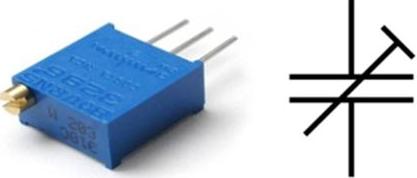
**Gambar 2.3 Bentuk Dan Simbol Light Emitting Diode (LED)**

LED terbuat dari berbagai material setengah penghantar campuran seperti *gallium arsenida fosfida (GaAsP), gallium fosfida (GaP),* dan *gallium aluminium arsenida (GaAsP).* Karakteristiknya yaitu jika diberi panjaran maju, pertemuannya mengeluarkan cahaya dan warna cahaya bergantung pada jenis dan kadar material pertemuan. Ketandasan cahaya berbanding lurus dengan arus maju yang mengalirinya. Dalam kondisi menghantar, tegangan maju pada LED merah adalah 1,6 sampai 2,2 volt, LED kuning 2,4 volt, LED hijau 2,7 volt. Sedangkan tegangan terbaik maksimum yang dibolehkan pada LED merah adalah 3 volt, LED kuning 5 volt, dan LED hijau 5 volt.

**2.2.3 Multiturn**

*Multiturn* adalah *resistor* yang nilai resistansinya dapat dibuah secara langsung baik dengan tuas yang telah tersedia atau menggunakan obeng. Ada dua jenis *resistor* variabel, yaitu *trimpot* (*trimer potensio*) dan *potensiometer*.

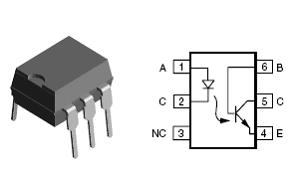
*Trimpot* (*trimer potensio*) merupakan *resistor* yang nilai *resistansinya* dapat diubah dengan memutar porosnya menggunakan obeng. Nilai *resistansi* dari *trimpot* tertulis pada badan *trimpot* tersebut menggunakan kode angka. Nilai yang tertulis pada badan *trimpot* merupakan nilai maksimum dari resistansi *trimpot* tersebut. Bentuk dan simbol *multiturn* dapat dilihat pada gambar 2.4.

****

**Gambar 2.4 Bentuk Dan Simbol Multiturn**

**2.2.4 Optocoupler 4N25**

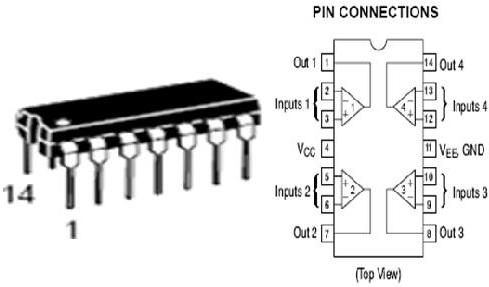
Optocoupler adalah suatu piranti yang terdiri dari 2 bagian yaitu transmitter dan receiver merupakan antara bagian cahaya dengan bagian deteksi sumber cahaya terpisah yang bekerja berdasarkan picu cahaya optic. Prinsip kerja dari *optocoupler* yaitu jika antara *phototransistor* dan *LED* terhalang maka *phototransistor* tersebut akan off sehingga output dari kolektor akan berlogika high. Sebaliknya jika antara *phototransistor* dan *LED* tidak terhalang maka phototransistor dan *LED* tidak terhalang maka phototransistor tersebut akan on sehingga output-nya akan berlogika low. Bentuk dari optocoupler dapat dilihat pada gambar 2.5.

**

**Gambar 2.5 Bentuk Dan Simbol Optocoupler**

**2.2.5 LM324N**

*LM324N* merupakan komponen elektronik yang dapat digunakan sebagai komparator. *LM324N* umumnya dikenal dengan *Op Amp (Operational Amplifier). Op Amp (Operational Amplifier)* yaitu suatu rangkaian terintegrasi yang berisi berupa tingkat dan konfigurasi dan penguat diferensia. Bentuk fisik dan simbol *LM324N* dapat dilihat pada gambar 2.6.

****

**Gambar 2.6 Bentuk Dan Simbol LM324N**

Komparator berfungsi untuk membandingkan tegangan yang masuk pada kedua kaki *input Op Amp*. Untuk membandingkan kedua kaki *input* pada *Op Amp* salah satu kaki *input* diberi tegangan *referensi* dan kaki lainnya diberi tegangan pembanding. Jika tegangan pada kaki *non inverting input (+)* lebih besar atau sama dengan tegangan pada kaki *inverting input (-)* maka *output* akan berharga *high* (*1*). Jika tegangan pada kaki *non inverting input (+)* lebih kecil dari pada tegangan pada kaki *inverting input (-)* maka kaki *output* akan berharga *low* (*0*).

Untuk membandingkan kedua kaki *input* pada Op Amp, salah satu kaki *input* diberi tegangan referensi dan kaki lainnya diberi tegangan pembanding. Salah satu keunggulan *LM324N* adalah dapat beroperasi pada *voltase* 3.0 V sampai 32.0 V. Skematik *Op Amp* sebagai *komparator* dapat dilihat pada gambar 2.7.



**Gambar 2.7 Op Amp Sebagai Komparator**

Salah satu fungsi *Op Amp* lainnya adalah sebagai *differential amplifier.* Diferensial *amplifier* berfungsi sebagai penguat perbedaan tegangan yang masuk pada kedua kaki *input Op Amp*. Skematik *Op Amp* sebagai *differential amplifier* dapat dilihat pada gambar 2.8.



**Gambar 2.8 Op Amp Sebagai Differential Amplifier**

Nilai tegangan *output* pada *Op Amp* sebagai *differential amplifier* dapat diketahui dengan menggunakan persamaan :

Keterangan :

*Vout* = *Tegangan output (Volt).*

*Vref = Tegangan referensi (Volt).*

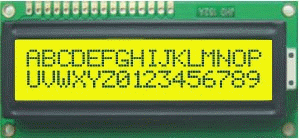
*Vin = Tegangan input (Volt).*

*R1 = R2 = Tahanan (Ohm).*

**2.2.6 Liquid Crystal Display (LCD)**

Liquid Crystal Display (LCD) adalah suatu jenis media tampil yang menggunakan kristal cair sebagai penampil utama. LCD sudah digunakan di berbagai bidang misalnya alat–alat elektronik seperti televisi, kalkulator ataupun layar komputer.

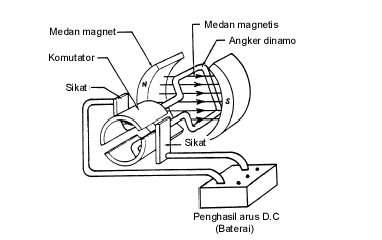
Tipe LCD dot matrik dengan jumlah karakter 16\*2. LCD sangat berfungsi sebagai penampil yang nantinya akan digunakan untuk menampilkan status kerja alat. Bentuk LCD 16\*2 dapat dilihat pada gambar 2.9.

****

**Gambar 2.9 Bentuk LCD 16\*2**

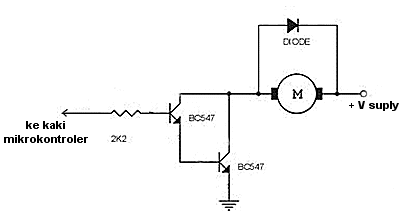
**2.2.7 Motor DC**

Motor listrik merupakan perangkat *electromagnetic* yang mengubah energi listrik menjadi energi mekanik. Energi mekanik ini digunakan untuk memutar *impeller* pompa, *fan*, *blower* menggerakan kompresor dan mengangkat bahan. Motor *dc* memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Kumparan medan pada motor *dc* disebut *stator* (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Jika terjadi putaran pada kumparan jangkar dalam pada medan magnet maka akan timbul tegangan (GGL) yang berubah-ubah arah pada setiap setengah putaran sehingga merupakan tegangan bolak-balik. Prinsip kerja dari arus searah adalah membalik *phase* tegangan dari gelombang yang mempunyai nilai positif dengan menggunakan komutator*,* dengan demikian arus yang berbalik arah dengan kumparan jangkar yang berputar dalam medan magnet. Bentuk motor paling sederhana memiliki kumparan satu lilitan yang bisa berputar bebas di antara kutub-kutub magnet permanen. Skematik motor *dc* sederhana dapat dilihat pada gambar 2.10.



**Gambar 2.10 Skematik Motor DC Sederhana**

Satu tegangan *DC* dari baterai menuju lilitan melalui sikat yang menyentuh komutator*,* dua segmen yang terhubung dengan dua ujung lilitan. Kumparan satu lilitan pada gambar di atas disebut angker dinamo. Angker dinamo adalah sebutan untuk komponen yang berputar di antara medan magnet. Skematik rangkaian *driver* dapat dilihat pada gambar 2.11.

****

**Gambar 2.11 Skematik Rangkaian Driver Sederhana**

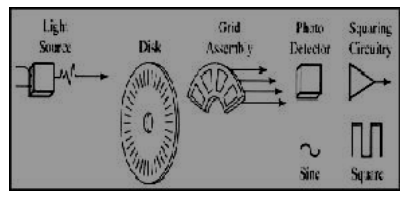
**2.2.8 Rotary Encoder**

*Rotary encoder* adalah perangkat elektromekanik yang dapat memonitor gerakan dan posisi. *Rotary encoder* umumnya menggunakan sensor optik untuk menghasilkan serial pulsa yang dapat diartikan menjadi gerakan, posisi dan arah. Sehingga posisi sudut suatu poros benda berputar dapat diolah menjadi informasi berupa kode digital oleh *rotary encoder* untuk diteruskan oleh rangkaian kendali. *Rotary encoder* umumnya digunakan pada pengendalian *robot, motor drive* dan sebagainya. Bentuk *rotary encoder type E40H-A-360-24* dapat dilihat pada gambar 2.12.

**

**Gambar 2.12 Rotary Encoder Type E40H-A-360-24**

*Rotary encoder* tersusun dari suatu piringan tipis yang memiliki lubang-lubang pada bagian lingkaran piringan. *LED* ditempatkan pada salah satu sisi piringan sehingga cahaya akan menuju piringan. Di sisi yang lain suatu *photo-transistor* diletakkan sehingga *photo-transistor* ini dapat mendeteksi cahaya dari *LED* yang berseberangan. Piringan tipis tadi dipasangkan dengan poros motor, atau perangkat berputar lainnya yang ingin kita ketahui posisinya sehingga ketika motor berputar piringan juga akan ikut berputar. Apabila posisi piringan mengakibatkan cahaya dari LED dapat mencapai *photo-transistor* melalui lubang-lubang yang ada, maka *photo-transistor* akan mengalami *saturasi* dan akan menghasilkan suatu pulsa gelombang persegi. Semakin banyak deretan pulsa yang dihasilkan pada satu putaran menentukan akurasi *rotary encoder* tersebut, akibatnya semakin banyak jumlah lubang yang dapat dibuat pada piringan menentukan akurasi *rotary encoder* tersebut. Sistem blok penyusunan *rotary encoder*  dilihat pada gambar 2.13.

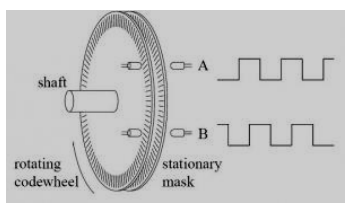
****

**Gambar 2.13 Sistem Blok Penyusunan Rotary Encoder**

Pada penelitian ini *rotary encoder* yang digunakan termasuk dalam kategori *incremental encoder*. *Incremental encoder* terdiri dari dua *track* atau *single track* dan dua *sensor* yang disebut *channel* A dan B, ketika poros berputar, deretan pulsa akan muncul pada masing-masing *channel* pada *frekuensi* yang proporsional dengan kecepatan putar sedangkan hubungan *phase* antara *channel* A dan B menghasilkan arah putaran. Putaran dapat dihitung dengan cara jumlah pulsa yang terjadi *resolusi* piringan. Untuk mengetahui arah putaran, dengan mengetahui *channel* mana yang *leading* terhadap *channel* satunya dapat kita tentukan arah putaran yang terjadi karena kedua *channel* tersebut akan selalu berbeda *phase* seperempat putaran (*quadrature signal*).

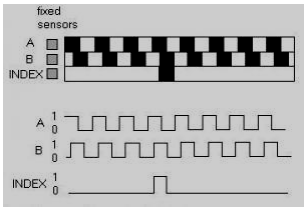
*Quadrature* sinyal atau sinyal berpangkat sebenarnya terdiri dari 2 sinyal yang biasanya dimisalkan dengan *I (untuk phase biasa)* dan *Q (untuk phase kuadrat)* atau dengan *RE (untuk sinyal Real)* dan *IM (untuk sinyal imaginer).* Letak perbedaan dari sinyal sebenarnya dengan sinyal *kuadrat* adalah terletak di *spektrum* dari sinyal *kuadrat* tidak *simetri* dengan *frequensi 0.*

Seringkali terdapat *output channel ketiga,* disebut *INDEX,* yang menghasilkan satu *pulsa* per putaran berguna untuk menghitung jumlah putaran yang terjadi. Susunan piringan untuk *incremantal encoder* dapat dilihat pada gambar 2.14.

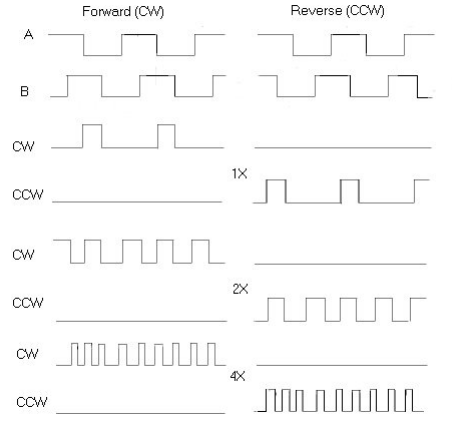


**Gambar 2.14 Susunan Piringan Untuk Incremental Encoder**

*Resolusi* keluaran dari sinyal *quadrature* A dan B dapat dibuat beberapa macam yaitu *1X, 2X* dan *4X.* *Resolusi* *1X*  hanya memberikan *pulsa* tunggal untuk setiap *siklus* salah satu *sinyal* A dan B, sedangkan *resolusi 4X* memberikan pulsa setiap *transisi* pada kedua *sinyal A* dan *B* menjadi empat kali *resolusi 1X* arah putaran dapat ditentukan melalui *level* salah satu *sinyal* selama *transisi* terhadap *sinyal* yang kedua. Pada contoh *resolusi 1X,* A = arah bawah dengan B = 1 menunjukkan arah putaran searah jarum jam, sebaliknya B = arah bawah dengan A = 1 menunjukkan arah berlawanan jarum jam. Skema pola keluaran *incremental encoder* dapat dilihat pada gambar 2.15. dan skema *output* dan arah putaran pada *resolusi* yang berbeda-beda dapat dilihat pada gambar 2.16.

****

**Gambar 2.15 Contoh Pola Keluaran Incremental Encoder**

****

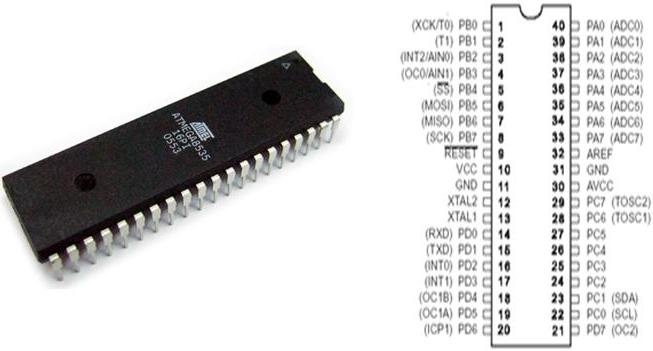
**Gambar 2.16 Output Dan Arah Putaran Pada Resolusi Yang Berbeda-beda**

**2.3 Mikrokontroller**

*Mikrokontroller* merupakan perangkat elektronika yang didalamnya terdapat rangkaian kontrol, *mikroprosesor, memori,* dan *input/output*. *Mikrokontroller* dapat diprogram menggunakan berbagai macam bahasa program. Bahasa program yang biasa digunakan untuk program *mikrokontroller* diantaranya bahasa *assembler*, bahasa *C*, bahasa *basic* dan lain-lain.

*Mikrokontroller*  biasanya digunakan untuk mengendalikan suatu proses secara otomatis seperti sistem kontrol mesin, *remote* kontrol*,* kontrol alat berat dan lain-lain. Dengan menggunakan *mikrokontroller* sistem kontrol akan menjadi lebih ringkas, lebih mudah dan lebih ekonomis.

Salah satu jenis *mikrokontroller* yang banyak digunakan untuk aplikasi kontrol adalah *ATMega8535. ATMega8535* merupakan salah satu *mikrokontroller* keluaran *Atmel*. *Atmel* adalah salah satu *vendor* yang bergerak di bidang mikro elektronika. Bentuk dan simbol *mikrokontroller ATMega8535* dapat dilihat pada gambar 2.17 .

****

**Gambar 2.17 Bentuk IC ATMega8535**

**2.4 CodeVisionAVR**

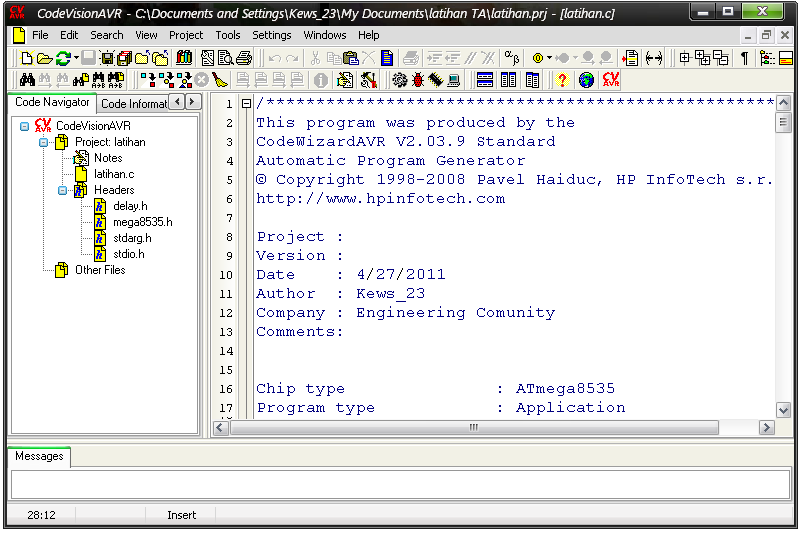
*AVR STUDIO* merupakan *software* khusus untuk bahasa *assembly*  yang mempunyai fungsi sangat lengkap. *Software* *AVR STUDIO* digunakan untuk menulis program, kompilasi, simulasi dan *download* program ke *IC mikrokontroller AVR.*

*CodeVisionAVR* merupakan *software C-cros compiler. Software Code Vision AVR* dapat ditulis dalam bahasa *C. CodeVisionAVR* memiliki *(Integrated Development Environtment)* IDE yang lengkap. *CodeVisionAVR* digunakan untuk penulisan program, *compile*, *link*, dan pembuatan kode bahasa mesin *(assembler).*

Proses *download*  program ke *IC mikrokontroller AVR* dapat dilakukan dengan menggunakan *system download* secara *In-System Programing* (ISP). ISPmengijinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial *SPI*. Langkah-langkah untuk menjalankan program *CodeVisionAVR* adalah sebagai berikut :

1. Membuka CodeVisionAVR

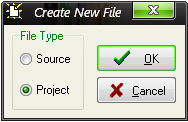
Langkah untuk membuka *CodeVisionAVR* dilakukan dengan memilih menu *Start Menu → All Program CodeVisionAVR → CodeVisionAVR Compiler.* Setelah langkah tersebut dilakukan pada monitor akan tampil tampilan seperti gambar 2.18.



**Gambar 2.18 Tampilan Awal CodeVisionAVR**

1. Membuat Project Baru

Langkah untuk membuat *project* baru dilakukan dengan memilih menu *File → New,* kemudian memilih *Project → OK.*Setelah langkah tersebut dilakukan pada monitor akan tampil dialog *Create New Fil*e seperti pada gambar 2.19.



**Gambar 2.19 Konfirmasi Membuat Project Atau File**

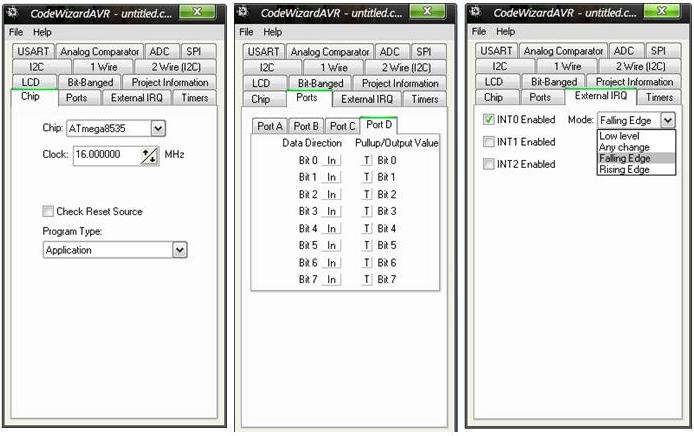
Jika tombol *OK* dipilih maka akan muncul kotak dialog *Confirm.* Kotak dialog ini menanyakan apakah *CodeWizardAVR* akan digunakan atau tidak. Jika *Codewizard* akan digunakan maka tekan tombol *YES.* Kotak dialog *confirm* seperti pada gambar 2.20.

****

**Gambar 2.20 Konfirmasi Penggunaan CodeWizardAVR**

1. Melakukan Konfigurasi Project

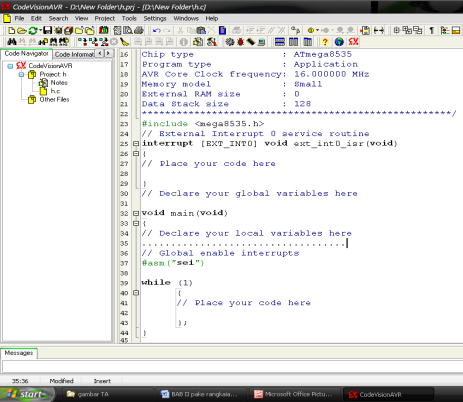
Langkah untuk *menkonfigurasi project* dilakukan dengan memilih *tab-tab* yang ada pada kotak dialog *CodeWizardAVR. Tab-tab* yang ada pada kotak dialog *CodeWizard* terdiri dari *tab* *USART*, *tab* *Analog Comperator*, *tab ADC, tab SPI, tab l2C, tab 1 Wire, tab 2Wire(l2C), tab LCD, tab Bit-Banged, tab Project Information, tab Chip, tab Port* dan *tab External IRQ.* Sebagian *tab-tab* dapat pada kotak dialog *CodeWizardAVR* seperti pada gambar 2.21.



**Gambar 2.21 Konfigurasi Program Melalui CodeWizardAVR**

1. Membuat Kode Program

Langkah untuk membuat kode program dilakukan dengan membuat *file* dan menamainya terlebih dahulu kemudian menyimpannya pada folder yang telah ditentukan. Langkah-langkah pembuatan *file* dilakukan dengan memilih *menu* *File, Generate, save and Exit* kemudian *file source* (\*.c), *file project* (\*.prj) dan *file* project *code wizard (\*.cwp)* diberi nama dan disimpan pada *folder* yang telah ditentukan. Setelah langkah tersebut, pada monitor akan tampil program yang siap diisi oleh program yang dibuat. Form program seperti pada gambar 2.22.



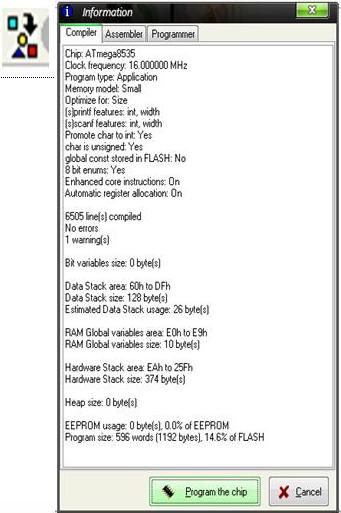
**Gambar 2.22 Form Program**

1. Mengconfigure Program

Langkah untuk meng-*configure* program dilakukan dengan memilih menu *Project → Configure → AfterBuild → program the Chip → Ok.*

1. Mengcompile Program

Langkah untuk meng-*compile* program dilakukan dengan memilih *menu Project / compile* atau dengan memilih *icon* → Program. Jika penulisan kode program benar, maka akan tampil kotak dialog *information*. Kotak dialog *informatioan* seperti pada gambar 2.23.

****

**Gambar 2.23 Kotak Dialog Information Configure**

**2.5 Interupsi**

*Interupsi* adalah fasilitas pada *mikrokontroller* untuk mengalihkan suatu program utama yang sedang berjalan dengan memprioritaskan program yang diutamakan (program *interupsi*). Pada saat program utama sedang dikerjakan oleh *CPU*, program dapat berhenti secara tiba-tiba untuk sementara waktu karena ada perintah lain *(interrupt)* yang harus dikerjakan terlebih dahulu oleh *CPU*. Setelah *interupsi* selesai dikerjakan, program utama akan dilanjutkan.

**2.6 Kendala Pengukuran Kedalaman**

Faktor-faktor yang menyebabkan kesalahan pengukuran kedalaman (*Depth Error*) umumnya disebabkan bukan dari sensor kedalaman (*Depth Encoder*) namun karena faktor diluar itu.

**2.6.1** **Tegangan Winch Cable**

Tegangan kabel pada waktu turun akan berbeda dengan tegangan kabel pada waktu bandul naik. Saat turun beban kabel hanya sebesar berat bandul dalam *fluida* (berat bandul di udara terbuka akan lebih besar dari pada berat bandul di dalam *fluida* karena pengaruh gaya ke atas sesuai *Hukum Archimedes*) atau W dikurangi dengan gesekan bandul dengan *fluida* (semakin besar *viskositas fluida* dan semakin cepat bergerak gesekan ini akan semakin besar atau *ff*), gesekan ini justru menghambat laju dari bandul ketika turun, secara langsung menyebabkan berkurangnya tegangan kabel (Tegangan Kabel = *w - (ff + Fa*) dimana *Fa*adalah (Gaya Keatas sesuai *Hukum Archimedes*). Disisi lain pada saat naik, tegangan kabel dipengaruhi oleh berat bandul dan ditambah dengan besarnya gaya gesek antara bandul dengan dinding sumur *(fd),* mengingat pada saat bandul naik menekan dinding sumur. Besarnya *Tegangan Kabel = (w + fd + ff) –FA.*

Faktor-faktor yang menyebabkan perbedaan tegangan kabel saat turun dan saat naik. Semakin besar perbedaan tersebut kemungkinan akan menyebabkan perbedaan pengukuran saat turun dan saat naik semakin besar pula. Hal-hal yang menyebabkan perbedaan tegangan kabel saat naik dan turun semakin besar adalah:

* Kecepatan saat bandul turun terlalu besar sehingga kabel menjadi kendor (akan mempengaruhi *ff* ).
* *Mud/fluida* didalam sumur mempunyai *viskositas* tinggi (akan mempengaruhi *ff* dan *FA*).
* Bandul mempunyai tekanan, sehingga gaya gesek bandul terhadap dinding sumur menjadi besar (akan mempengaruhi *fd*) Secara *frontal* dapat di ibaratkan mengukur karet yang kendor dan karet yang kencang maka panjang akan berbeda.

**2.6.2** **Slip**

*Slip* antara *Pulley* dengan *winch cable* dapat terjadi manakala kabel dalam konsidi kendor. Terutama ini terjadi pada saat bandul turun dengan cepat, sangat fatal akibatnya jika terjadi *slip* antara poros *encoder* dengan poros *pulley* (umumnya *error* yang terjadi sangat besar).

**2.6.3** **Pergeseran Posisi**

Posisi *winch* bergeser pada saat pengukuran, ini dapat terjadi jika *winch* tidak diikat saat melakukan pengukuran. Saat turun mungkin tidak akan mengubah posisi *winch* namun pada saat bandul naik, tegangan kabel naik, ini dapat menyebabkan *winch* bergeser maju ke arah lubang bor. Tentu hal ini akan menyebabkan pembacaan kedalaman menjadi tidak benar. Faktor lain adalah posisi dari *tripod*. Apabila kondisi tanah di sekitar lubang bor lembek, saat bandul naik, *tripod* akan megalami tekanan yang cukup besar. Kondisi tanah yang lembek dapat menyebabkan posisi *pulley* pada *tripod* turun, turunnya posisi *pulley* pada *tripod* akan menambah daftar penyebab *error*.

**2.6.4** **Kecepatan**

Seperti dikemukakan pada tegangan *winch cable,* bahwa kecepatan selain menyebabkan besarnya gaya gesek *fluida* juga dapat menyebabkan *slip*. Dua faktor ini secara *akumulasi* dapat mempengaruhi besarnya *error* yang terjadi, idealnya kecepatan turun harus sama dengan kecepatan waktu naik. Apalagi dalam sumur yang mempunyai *viskositas* tinggi, kecepatan turun harus lebih kecil dari pada waktu naik. Untuk alasan-alasan kecepatan *operasi* pengukuran, pada lubang sumur bor dengan *fluida* air *(fresh water)* sebaiknya bandul turun tidak lebih dari 6 meter/menit.

**2.7 Pengecekan Kebenaran Data Kedalaman**

Secara berkala setidaknya 1 bulan sekali *operator* wajib mengecek kebenaran data kedalaman dengan membandingkan panjang kabel yang terukur di komputer dengan panjang kabel yang terukur dengan alat ukur panjang (meteran) Tarik kabel hingga setidaknya 25.00 meter, ukurlah dengan meteran apakah panjang kabel benar 25.00 meter. Setelah itu gulung kabel, pada posisi 0 (nol) penunjukan pada komputer harus sama. Pengujian ini dilakukan dengan cara menarik *cable head* secara mendatar agar tegangan kabel relatif sama. Apabila dalam pengukuran ini tidak *sinkron* antara pengukuran meteran dengan yang ditampilkan di komputer atau *LC, operator* harus melakukan kalibrasi.

Perlu diketahui bahwa :

* *Encoder* hanya ada kondisi baik atau rusak dan *software Visual Basic,* hanya mengenal dua kata tersebut. Jika Baik maka *encoder* akan mengirim sinyalnya ke *Mikrokontrol* dan *LCD* akan menampilkan responnya jika respon tidak ada maka *encoder* rusak.
* Dalam kondisi ideal (tidak ada *slip*, tegangan kabel saat naik dan turun sama), jika tidak terjadi *error* (saat bandul kembali di permukaan tanah), belum tentu data *kalibrasinya* benar. Karena dalam kondisi ini, koefisien *kalibrasi* benar atau tidak, jika bandul diturunkan dari permukaan (0), saat kembali lagi di permukaan pasti akan kembali 0 (nol).

**2.7.1 Pemecahan Masalah**

*Depth error* dalam pengukuran pasti akan terjadi karena faktor-faktor seperti dijelaskan di atas. Toleransi *error* adalah 0.2 % dari total kedalaman. Artinya *logging* untuk kedalaman 100 meter, masih ditolerir jika terjadi *error* hingga 20 cm. Lebih dari itu harus dilakukan pengukuran ulang, setelah terlebih dahulu dicari penyebabnya. Perlu diketahui bahwa jika di permukaan diperoleh *error* kedalaman 10 cm untuk 100 meter, berarti *error* di tiap titik kedalaman tidaklah sama. Di *bottom* *(TD)* dapat dikatakan mempunyai *error* 0 (nol) artinya data kedalamannya benar, sementara pada kedalaman 75 meter mempunyai *error* sekitar 2.5 cm,pada kedalaman 50 meter *error* 5 cm, pada kedalaman 25 cm *error* 7.5 cm dan di permukaan 10 cm. *Formula* berikut dapat dipergunakan untuk menentukan besarnya *error* di masing-masing kedalaman.

*Ed=(E \* d) / TD*

Dimana:

*Ed = Besarnya Error Di Kedalaman Di Tentukan (d)*

*E = Besarnya Error Dipermukaan*

*D = Kedalaman Yang Ditentukan*

*TD = Total Depth Logging (Total Kedalaman Yang Di logging)*

Formula tersebut di atas tidak dapat dipakai sebagai koreksi untuk kedalaman yang lebih dari 330 meter (1000 ft) mengingat permasalahannya sangat kompleks. Kedalaman yang melebihi 100 ft harus diperhitungkan berat kabel, streng koefisien dari *winch cable* dll.

**2.7.2 Referensi Kedalaman**

Fungsi utama dari pengukuran sumur bor (*logging*) selain mendapatkan *geophysical* data dari formasi juga sebagai koreksi kedalaman dari data pemboran (*lithologi* atau hasil *coring*). Ini harus dipergunakan sebagai referensi, karena data pemboran umumnya hanya mengandalkan dari panjang *rod,* sementara panjang *rod* dan *colar* (sambungan antar *rod*) ada yang dapat menutup/tersambung sempurna namun ada juga yang tidak. Panjang *rod* tidak dapat dipergunakan sebagai patokan kedalaman, mengingat untuk *rod-rod* yang sudah lama terpakai (apalagi sudah *ditap* ulang, bengkok, atau terpilin) panjangnya tidak lagi standar. Kita dapat dengan mudah menentukan total *depth* dari pemboran dengan menjumlahkan panjang *bit, shaft,* dan *rod.* Namun mengingat panjang *rod* belum tentu seragam maka kedalaman pemboran yang ditentukan dari jumlah rod yang berada di dalam lubang bor tidak dapat dipergunakan sebagai referensi data geologi.