PENGEMBANGAN PROGRAM UNTUK *PROTOTYPE* MESIN PEMBENGKOK BATANG SILINDER

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat Kelulusan Sarjana Strata-1

Program Studi Teknik Mesin

Universitas Pasundan Bandung

Oleh:

RIDWAN SOLIH

113030157

PRODI TEKNIK MESIN

FAKULTAS TEKNIK

UNIVERSITAS PASUNDAN

BANDUNG

2017

ABSTRAK

Mesin pembengkok *(bending)* adalah mesin yang digunakan untuk membengkokkan atau menekuk pipa dengan menggunakan *dies*. Bending merupakan pengerjaan dengan cara memberi tekanan pada bagian tertentu sehingga terjadi deformasi plastis pada bagian yang diberi tekanan. Pengerjaan bending biasanya dilakukan pada bahan plat atau pipa baja karbon rendah untuk menghasilkan suatu produk dari bahan plat atau pipa.

Di laboratorium otomasi robotika terdapat *Prototype* mesin pembengkok batang silinder yang dikendalikan dengan menggunakan *Software CodevisionAVR*. *Software CodevisionAVR* ini hanya dapat mengendalikan *Prototype* mesin pembengkok batang silinder untuk membuat produk dengan satu bentuk tertentu. Jika *Prototype* mesin pembengkok akan membuat bentuk yang lain maka programmer harus membuat program dari awal, karena dalam hal ini mikrokontroller digunakan untuk memerintah mekanisme penggerak pada *Prototype* mesin pembengkok. Berdasarkan hal tersebut timbul gagasan untuk mempermudah progammer dalam membuat program untuk mengendalikan prototype mesin pembengkok. Agar mikrokontroller yang awalnya digunakan untuk memerintah motor penggerak pada mekanisme prototype mesin pembengkok diubah fungsinya untuk menerjemahkan data yang dikirim dari komputer. Data yang dikirim komputer berupa data gerakan motor yang dapat diterjemahkan mikrokontroller menjadi sinyal digital.

KATA PENGANTAR

Assalamualaikum Wr. Wb

Alhamdulillah saya panjatkan puji syukur kehadirat Allah SWT yang telah memberikan rahmat serta hidayahNya sehingga saya bisa menyelesaikan tugas akhir saya dengan judul **"PENGEMBANGAN PROGRAM *PROTOTYPE* MESIN PEMBENGKOK BATANG SILINDER"**. Tugas akhir ini disusun dan diajukan sebagai syarat kelulusan program sarjana strata-1 di Prodi Teknik Mesin Fakultas Teknik Universitas Pasundan Bandung.

Tidak sedikit kesulitan yang saya hadapi dalam tugas akhir ini, karena dengan dorongan serta doa dan dukungan moril dari berbagai pihak akhirnya laporan ini dapat diselesaikan. Saya ucapkan rasa terima kasih saya kepada :

1. Kedua orangtua dan keluarga saya yang telah memberikan dukungan serta doa yang tak henti-hentinya
2. Bapak Ir. Rachmad Hartono, MT., selaku pembimbing 1 pada tugas akhir saya yang telah banyak memberikan bimbingan, arahan, dan dukungan moril selama proses perancangan dan penyusunan tugas akhir ini.
3. Bapak Ir. Syahbardia, MT., selaku pembimbing 1 pada tugas akhir saya yang telah banyak memberikan bimbingan, arahan, dan dukungan moril selama proses perancangan dan penyusunan tugas akhir ini dan selaku koordinator tugas akhir Prodi Teknik Mesin Universtitas Pasundan Bandung.
4. R. Ibrahim (Baim) dan Encu sukarya (Boncu) selaku peserta TA sebelumnya yang telah memberikan masukan dan dukungan moril sehingga saya bisa menyelesaikan tugas akhir ini
5. Casmadi, Resno, Rizki, Aris, Asep, Cecep, Tiko dan serta keluarga besar laboratorium PEOTRO yang menyempatkan waktu berdiskusi selama perancangan tugas akhir ini.
6. Keluarga besar BKR yang telah memotivasi saya.
7. Rekan-rekan di prodi teknik mesin mesin angakatan 2011 yang tidak bisa disebutkan satu persatu

Saya menyadari tugas akhir ini jauh dari kata sempurna dikarenakan keterbatasan waktu, pengetahuan, serta kemampuan yang dimiliki. Dengan demikian kritik dan saran membangun sangatlah diharapkan.

Akhir kata saya berharap tugas akhir ini dapat bermanfaat khususnya bagi saya dan umumnya bagi pembaca.

Wassalamualaikum Wr. Wb

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

Pada bab ini dibahas tentang latar belakang, tujuan, batasan masalah, metode pengumpulan data, dan sistematika penulisan.

1.1 Latar Belakang

Pada era globalisasi, peningkatan teknologi sangat dibutuhkan. Sistem otomatis banyak diterapkan di berbagai bidang termasuk di bidang pemesinan. Peralatan produksi yang serba otomatis menjadi faktor utama untuk meningkatkan efektivitas, efisiensi, ketelitian dan keamanan dalam melakukan proses produksi. Banyak perusahaan meng-*update* mesin perkakas manual menjadi mesin perkakas berbasis komputer. Perkembangan teknologi komputer saat ini telah mengalami kemajuan yang amat pesat. Komputer telah diaplikasikan ke dalam mesin perkakas diantaranya mesin bending, rolling, forging, dan shearing.

Mesin pembengkok (bending) adalah mesin yang digunakan untuk membengkokkan atau menekuk pipa dengan menggunakan dies. Bending merupakan pengerjaan dengan cara memberi tekanan pada bagian tertentu sehingga terjadi deformasi plastis pada bagian yang diberi tekanan. Pengerjaan bending biasanya dilakukan pada bahan plat atau pipa baja karbon rendah untuk menghasilkan suatu produk dari bahan plat atau pipa.

Di laboratorium otomasi robotika terdapat *prototype* mesin pembengkok batang silinder yang menggunakan *Software CodevisionAVR*. *Software CodevisionAVR* ini hanya dapat mengerjakan satu bentuk dalam satu program. Jika *prototype* mesin pembengkok akan membuat bentuk yang lain maka programmer harus membuat program dari awal. Berdasarkan hal tersebut timbul gagasan untuk mempermudah progammer dalam membuat program dengan menggunakan *Software Visual basic*.

1.2 Tujuan

Tujuan tugas akhir ini adalah mengembangkan program untuk mengendalikan *prototype* mesin pembengkok batang silinder. Pengembangan program bertujuan untuk mempermudah dan mempersingkat dalam proses pembuatan program untuk mengendalikan *prototype* mesin pembengkok untuk membuat produk dengan bentuk tertentu.

Pengendali *prototype* mesin pembengkok sebelumnya hanya menggunakan program CodeVision AVR. Program yang dibuat di CodeVision AVR hanya dapat digunakan untuk membuat satu jenis produk tertentu. Untuk membuat produk yang lain program CodeVision AVR perlu diubah. Proses mengubah program codevision AVR untuk produk yang lain membutuhkan waktu yang cukup lama. Untuk mempersingkat waktu dalam pembuatan program pada *prototype* mesin pembengkok memerlukan penambahan program. Program yang ditambahkan pada *prototype* mesin pembengkok yaitu program visual basic. Program visual basic berfungsi untuk mengirim data ke mikrokontroler. Mikrokontroler menerjemahkan data yang dikirim komputer menjadi sinyal digital yang berfungsi untuk menggerakkan motor.

1.3 Batasan Masalah

Masalah yang dibahas dalam tugas akhir ini dikhususkan pada beberapa bagian. Beberapa bagian tersebut adalah sebagai berikut :

1. Pada *prototype* mesin pembengkok batang silinder dilakukan pengembangan program. Pengembangan program ini dilakukan dengan cara menambahkan program *visual basic* pada program yang sudah ada.
2. Pada *visual basic* dibuat langkah-langkah proses pembengkokkan dengan cara memasukkan beberapa parameter, yaitu : jumlah putaran, delay, dan posisi motor.
3. Benda kerja yang dapat dibuat pada *prototype* mesin pembengkok berdiamater maksimal 5 mm dengan panjang maksimal 35 mm. Material yang digunakan sebagai benda kerja yang akan dibentuk pada *prototype* mesin pembengkok batang silinder adalah kawat elektroda E6013.

1.4 Metode Pengumpulan Data

Pada kegiatan ini ada beberapa metode pengumpulan data untuk menyelesaikan tugas akhir berjudul pengembangan sistem kontrol penggerak otomatis mesin pembengkok batang silinder. Beberapa metode pengumpulan data tersebut adalah studi pustaka, survei industri, diskusi.

1. Studi pustaka

   Studi pustaka adalah kegiatan pengumpulan data yang dilakukan dengan cara mempelajari literatur yang berhubungan dengan proses pengembangan sistem kontrol penggerak otomatis mesin pembengkok batang silinder.
2. Survei pasar

   Survei pasar adalah kegiatan mengumpulkan data dan survei di pasar tentang proses pengembangan mesin pembengkok batang silindris dan pembuatan serta ketersediaaan bahan-bahan yang dibutuhkan

beserta harganya. Hasil survei industri ini selanjutnya digunakan untuk proses pengembangan mesin pembengkok batang silinder.

3. Diskusi

   Diskusi adalah kegiatan melakukan diskusi dengan dosen pembimbing tentang pengembangan sistem kontrol penggerak otomatis mesin pembengkok batang silinder.

1.5 Sistematika Penulisan

Laporan ini disusun bab demi bab dan ter*dir*i dari 5 bab. Isi masing-masing bab adalah sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini dibahas tentang latar belakang masalah, batasan masalah, tujuan, pengumpulan data dan sistematika penulisan.

BAB II DASAR TEORI

Pada bab ini dibahas tentang teori – teori dasar mesin pembengkok batang silinder, pengenalan komponen elektronika, pengenalan microcontroller, pengenalan program AVR, dan pengenalan *visual basic*.

BAB III MODIFIKASI MEKANISME PENJEPIT BENDA KERJA, PENGUJIAN RANGKAIAN ELEKTRONIKA, DAN PEMBUATAN PROGRAM SISTEM PENGENDALI MEKANISME SIMULATOR MESIN PEMBENGKOK BATANG SILINDER

Pada bab ini dibahas tentang modifkasi mekanisme penjepit benda kerja, pengujian rangkian elektronika yang akan digunakan, dan program yang akan dimuat di microcontroller untuk mengendalikan sistem mekanisme simulator mesin pembengkok.

BAB IV PENGUJIAN DAN ANALISA *PROTOTYPE* MESIN PEMBENGKOK BATANG SILINDER

Pada bab ini dibahas tentang pengujian program pengendalian mekanisme simulator mesin pembengkok dengan menggunakan material uji dan analisa hasil pengujian pengendalian proses pembengkokkan.

BAB V KESIMPULAN DAN SARAN

Pada bab ini dibahas tentang kesimpulan dan saran.

DAFTAR PUSTAKA

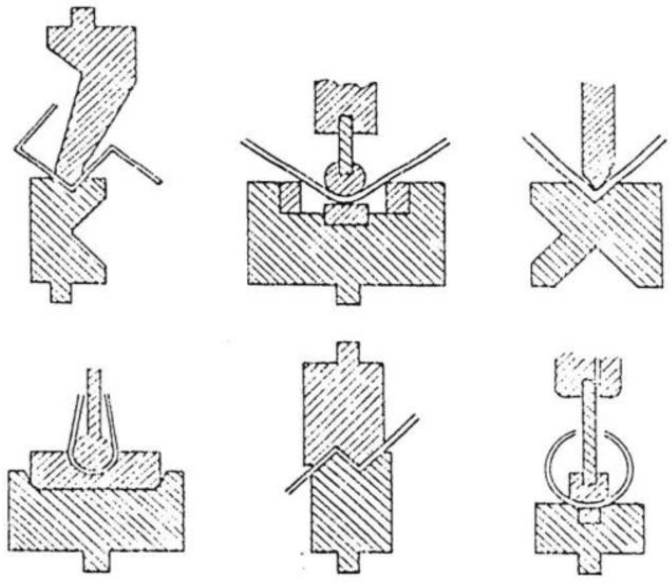
BAB II

TEORI DASAR

Pada bab ini dibahas tentang teori-teori dasar proses pembengkokan, pengenalan komponen elektronika, pengenalan *microcontroller* ATMega8535, pengenalan motor *stepper*, pengenalan *driver* motor *stepper*, pengenalan motor *servo* AC, pengenalan encoder, pengenalan program AVR, dan pengenalan *visual basic*.

2.1 Proses Pembengkokan

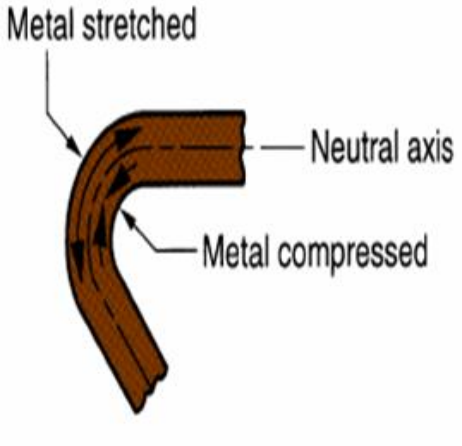
Proses pembengkokkan *(bending process)* adalah proses *forming* secara *cold working* yang menyebabkan deformasi plastis logam. Logam yang mengalami deformasi plastis mengalami perubahan penampang dengan bantuan tekanan piston pembentuk dan cetakan *(dies)*. Logam dapat berubah bentuk menjadi bengkok akibat tekanan mesin sederhana dengan menggunakan proses *bending*. Proses *bending* biasanya memakai dies berbentuk U, V, W atau yang lainnya. Proses *Bending* menyebabkan logam di sisi luar sumbu netral mengalami tarikan, sedangkan di sisi lainnya mengalami tekanan. Skema jenis-jenis pembengkokan dapat dilihat pada gambar 2.1.



Gambar 2.1 Skema jenis-jenis proses pembengkokan (*bending*)

Proses *bending* mengakibatkan logam bagian luar mengalami tegangan tarik sementara permukaan logam mengalami tegangan tekan. Tempat kedudukan bagian logam yang tidak mengalami tekanan dikenal sebagai sumbu netral tekukan. Sumbu netral tekukan terjadi karena kekuatan luluh lebih besar daripada kekuatan tarik sehingga logam bagian luar terdeformasi lebih dahulu daripada bagian dalamnya. Selama proses bending selalu terjadi penegangan

(*stretching*) dan pengerutan (*shrinking*). Jika radius pembentukan relatif kecil terhadap part maka benda kerja cenderung terjadi stretch saat proses pembentukan. Proses penekukkan logam dapat dilihat pada gambar 2.2.



Gambar 2.2 Proses penekukkan logam

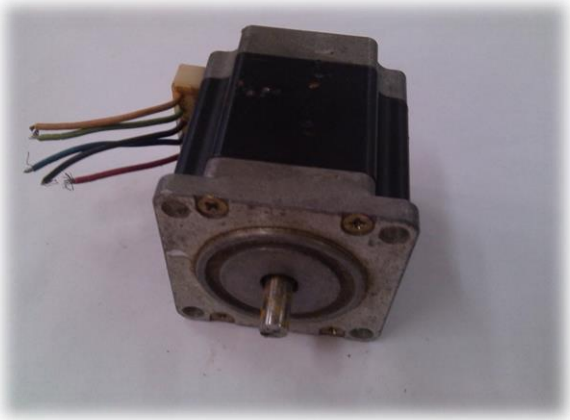
Dari gambar 2.2 dapat dilihat bahwa sumbu netral biasanya terletak diantara 1/3 dan 1/2 permukaan bagian dalam logam. Karena ketidaksamaan deformasi, logam akan mengalami sedikit penipisan di daerah tekukan. Penipisan akan lebih terlihat di tengah-tengah material yang secara bebas ditarik sepanjang sumbu tekukan. Dilihat dari sisi bagian dalam tekukan, sangat mungkin tekanan di bagian bawah logam dapat mengakibatkan *upsetting*. *Upsetting* mengakibatkan logam bertambah panjang dalam arah sejajar dengan sumbu tekukan. Efek upsetting akan semakin terlihat pada proses penekukan material yang tebal tetapi sempit.

2.2 Motor *Stepper*

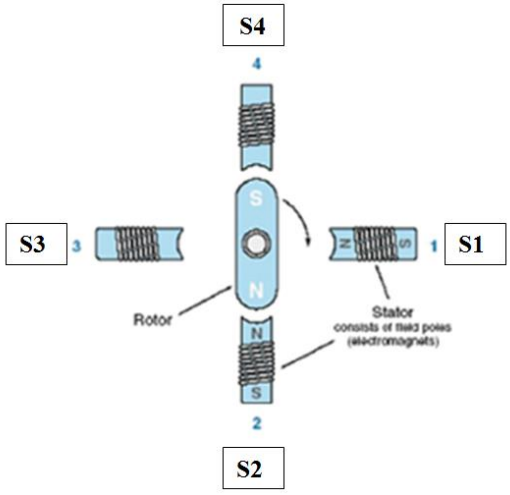
Motor *stepper* adalah motor DC yang khusus berputar dalam suatu derajat yang tetap yang disebut langkah (*step*). Satu *step* motor *stepper* bernilai antara 0,9° - 1,8°. Pada motor *stepper* terdapat *rotor* dan *stator*. *Rotor* adalah permanen magnet sedangkan *stator* adalah elektromagnet. *Rotor* akan bergerak ketika *stator* diberi aliran listrik. Aliran listrik yang dihasilkan stator dapat membangkitkan medan magnet dan membuat *rotor* menyesuaikan dengan kutub magnet yang dimilikinya. Tampilan motor *stepper* dan skematis posisi rotor dan stator dapat dilihat pada gambar 2.3 dan gambar 2.4.

Motor *stepper* dapat digerakkan jika lilitan *stator* dialiri listrik secara bergantian dan berurutan. Prinsip pengendalian motor *stepper* dapat dilihat pada gambar 2.5 dan tabel 2.1. Jika seluruh saklar dalam keadaan terbuka (OFF alias berkondisi 0), maka motor berada dalam

keadaan diam. Jika saklar ditutup dan dibuka secara bergantian dengan urutan sebagai berikut T_A, T_B, T_C, T_D, maka motor akan bergerak sejauh 4 langkah (4 x 1,8^o).



Gambar 2.3 Motor *stepper*



Gambar 2.4 Skematis Posisi *rotor* dan *stator* pada motor *stepper*

Tabel 2.1 Arah Putaran Motor *Stepper*

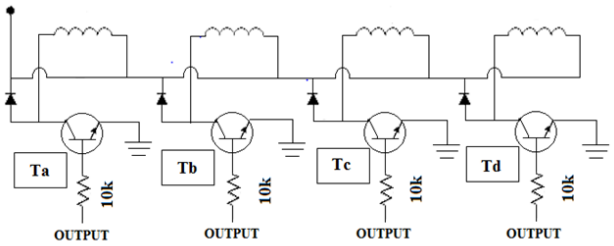
Ta	Tb	Tc	Td	Gerakan
0	0	0	0	**X**
1	**0**	**0**	**0**	**CW**
0	**1**	**0**	**0**	**CW**
0	**0**	**1**	**0**	**CW**
0	**0**	**0**	**1**	**CW**

Tabel 2.2 Arah Putaran Motor *Stepper* (lanjutan)

Ta	Tb	Tc	Td	Gerakan
0	0	0	1	CCW
0	0	1	0	CCW
0	1	0	0	CCW
1	0	0	0	CCW

CW : *Clock* Wise (Searah jarum Jam)

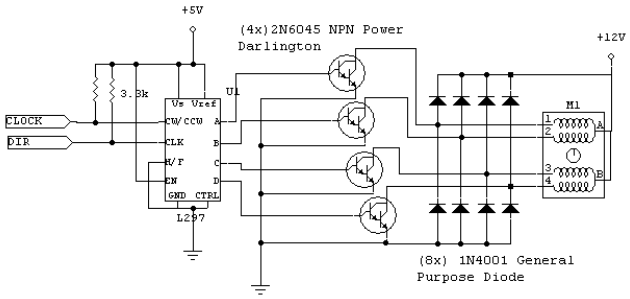
CCW : Counter *Clock* Wise (Berlawanan dengan arah jarum jam)



Gambar 2.5 Rangkaian Penggerak Motor *Stepper* Menggunakan Transistor

2.3 *Driver* Motor *Stepper*

Driver motor *stepper* adalah *driver* yang berfungsi sebagai penggerak motor *stepper*. *Driver* motor *stepper* memiliki dua kaki input dan empat kaki *output*. Kedua kaki input *driver* motor *stepper* yaitu kaki *dir (direction)* untuk mengatur arah putaran pada motor *stepper* dan kaki *clock* untuk memberikan sinyal. Keempat kaki *output* pada *driver* motor *stepper* yaitu A, B, C, dan D. Skematis *driver* motor *stepper* dapat dilihat pada gambar 2.6.



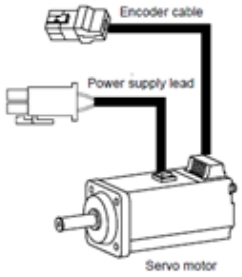
Gambar 2.6 Skematis *Driver* Motor *Stepper*

Prinsip kerja *driver* motor *stepper* yaitu jika ada perubahan sinyal di kaki *clock* maka sinyal high di kaki *output* bergeser. Jika kaki *dir* bernilai high maka sinyal high di kaki *output*

bergeser dari A, B, C dan D (CW). Jika kaki *dir* bernilai low maka sinyal high di kaki *output* bergeser dari kaki D, C, B dan A (CCW).

2.4 Motor *servo* AC

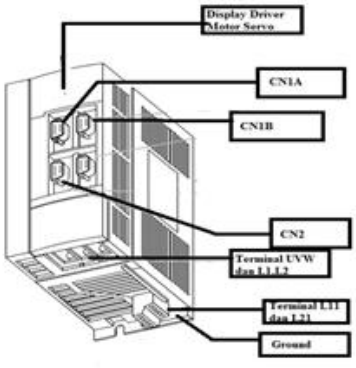
Motor *servo* AC adalah motor AC 3 phasa yang dilengkapi dengan *driver* motor *servo* dan *encoder*. Motor *servo* AC merupakan alat yang digunakan sebagai penggerak dalam sistem *servo*. Sistem *servo* mengandalkan umpan balik berupa posisi dan kecepatan untuk mengontrol setiap aksi. Motor *servo* mengubah energi listrik menjadi energi mekanik. Energi listrik yang diubah menjadi energi mekanik berupa gerak rotasi poros motor *servo*. Skematis motor *servo* AC AC MR-J2S-40A dapat dilihat pada gambar 2.7.



Gambar 2.7 Skematis Motor *Servo* AC AC MR-J2S-40A

2.5 *Driver* Motor *Servo* AC

Driver motor *servo* adalah perangkat yang digunakan untuk penguat sinyal dan pengolah sinyal error. Sinyal yang diolah digunakan untuk mengoreksi perbedaan antara sinyal input dengan sinyal umpan balik. Skematis *driver* motor *servo* AC MR-C dan *driver* motor *servo* AC MR-J2S-40A dapat dilihat pada gambar 2.8.



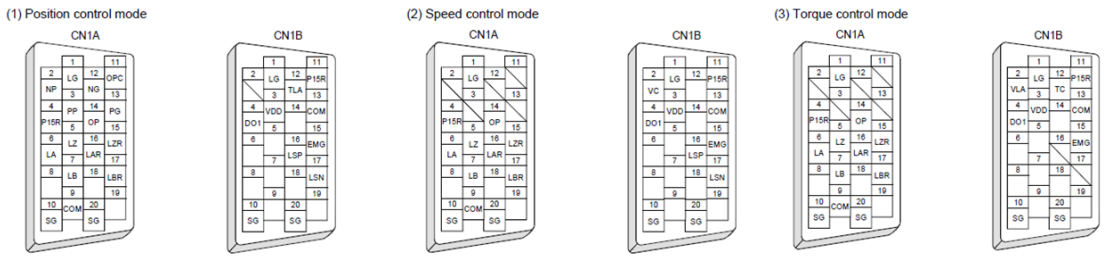
Gambar 2.8 Skematis *driver* Motor *Servo* AC MR-J2S-40A

Driver motor *servo* AC MR-J2S-40A memiliki tiga *Control mode* yang dapat digunakan. Ketiga *Control mode* yaitu: *Position control mode*, *Speed control mode*, dan *Torque control mode*. *Control Mode* pada motor *servo* AC MR-J2S-40A dapat diketahui dengan cara men*setting* parameter 0. Keterkaitan Parameter 0 dan *Control Mode* dapat dilihat pada tabel 2.2. *Control Mode* yang dipilih menentukan beberapa kaki di pin *conector* CN1A, CN1B dan beberapa parameter pelengkap *mode* yang dipilih *driver* motor *servo* AC MR-J2S-40A. Skematis beberapa kaki di pin *conector* CN1A dan CN1B sesuai *mode* motor *servo* AC dapat dilihat pada table 2.2.

Tabel 2.2 Parameter 0 *Driver* Motor *Servo* AC MR-J2S-40A

Class	No.	Symbol	Name and function	Initial value	Unit	Setting range	Control mode
Basic parameters	0	*STY	Control mode, regenerative option selection Used to select the control mode and regenerative option. 0 0 Select the control mode. 0:Position 1:Position and speed 2:Speed 3:Speed and torque 4:Torque 5:Torque and position Selection of regenerative option 00: Regenerative option or regenerative option is not used with 7kW or less servo amplifier (The built-in regenerative resistor is used.) Supplied regenerative resistors or regenerative option is used with 11kW or more servo amplifier 01: FR-RC, FR-BU2, FR-CV 02: MR-RB032 03: MR-RB12 04: MR-RB32 05: MR-RB30 06: MR-RB50 (Cooling fan is required) 08: MR-RB31 09: MR-RB51 (Cooling fan is required) 0E: When regenerative resistors supplied to 11k to 22kW are cooled by cooling fans to increase capability The MR-RB65, 66 and 67 are regenerative options that have encased the GRZG400-2Ω, GRZG400-1Ω and GRZG400-0.8Ω, respectively. When using any of these regenerative options, make the same parameter setting as when using the GRZG400-2Ω, GRZG400-1Ω or GRZG400-0.8Ω (supplied regenerative resistors or regenerative option is used with 11kW or more servo amplifier). POINT • Wrong setting may cause the regenerative option to burn. • If the regenerative option selected is not for use with the servo amplifier, parameter error (AL.37) occurs.	0000		Refer to Name and function column.	P•S•T

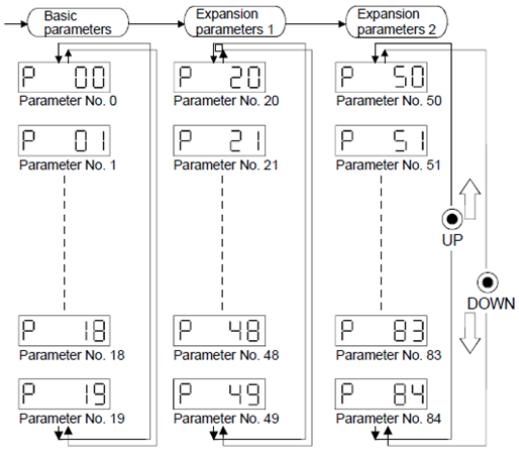
Dari tabel 2.2 dapat dilihat bahwa pada parameter 0 terdapat 4 karakter yang harus diisi pada *driver* motor *servo* AC MR-J2S-40A. Karakter di kolom 1 dan di kolom 2 menentukan *driver* motor *servo* AC yang digunakan untuk mengendalikan motor *servo* AC. Karakter di kolom 3 bernilai 0. Karakter di kolom 4 digunakan untuk mensetting *control mode* yang digunakan *driver* motor *servo* AC MR-J2S-40A.



Gambar 2.9

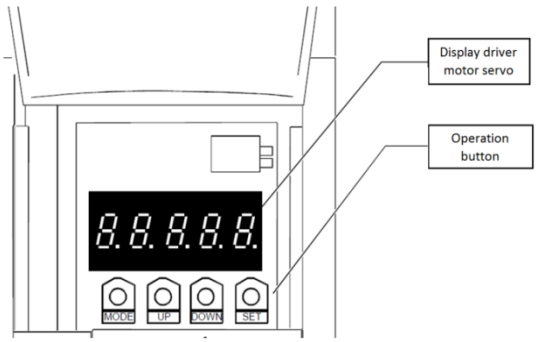
Kaki di Pin *Conector* CN1A dan CN1B *Driver* Motor *Servo* AC MR-J2S-40A

Driver motor *servo* AC MR-J2S-40A juga memiliki parameter motor *servo* AC yang dikelompokkan menjadi 3 jenis yaitu: *Basic* parameter (parameter 0-19), *Expansion* parameter1 (parameter 20-49), dan *Expansion* parameter2 (parameter 50-84). Skematis kelompok parameter *driver* motor *servo* AC MR-J2S-40A dapat dilihat pada gambar 2.10.



Gambar 2.10 Skematis Kelompok Parameter *Driver* Motor *Servo* AC MR-J2S-40A

Driver motor *servo* AC MR-J2S-40A memiliki *display driver* dan *operation button*. *Display driver* motor *servo* AC MR-J2S-40A ter*dir*i dari lima digit karakter. Karakter yang terdapat pada *display driver* digunakan untuk menampilkan status *display indicator driver* motor *servo*, status parameter *driver* motor *servo* dan status alarm *driver* motor *servo*. *Display driver* motor *servo* dan *operation button* MR-J2S-40A dapat dilihat pada gambar 2.11. Pada beberapa paragraf berikutnya dibahas cara men*setting* parameter *driver* motor *servo* AC MR-J2S-40A.



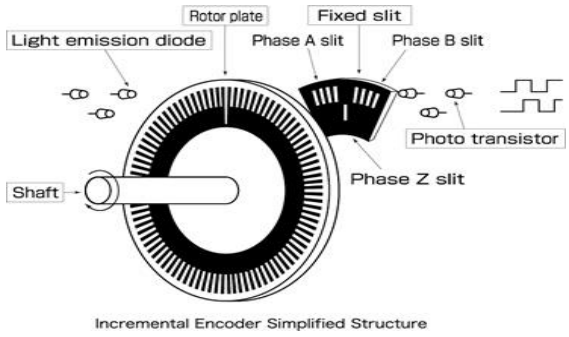
Gambar 2.11 *Display Driver* Motor *Servo* dan *Operation Button* Tipe MR–J2S-40A

Men*setting* parameter *driver* motor *servo* MR-J2S-40A dilakukan dengan melakukan beberapa langkah. Beberapa langkah tersebut yaitu:

1. Status *display* motor *servo* diubah dari status *display indicator* C menjadi status parameter yang dikelompokan, yaitu: *Basic* parameter (diawali P00), *Expansion* parameter1 (diawali P20), dan *Expansion* parameter2 (diawali P50). Status diubah dengan cara tombol *mode* ditekan beberapa kali sampai *display driver* motor *servo* menunjukan status parameter.
2. Tombol set ditekan sehingga karakter yang terdapat pada display *driver* motor *servo* berkedip. Tombol *up* atau *down* ditekan untuk mengubah karakter berkedip yang dimiliki oleh status parameter *driver* motor *servo*. Karakter yang diubah akan muncul di *display driver* motor *servo*.
3. Tombol *set* kembali ditekan untuk mensetting status parameter yang diinginkan.

2.6 *Encoder*

Encoder adalah perangkat yang digunakan untuk mengukur jumlah putaran poros motor dan menentukan arah putaran poros motor. Pada poros *encoder* diletakkan piringan. Pada bagian tepi piringan dibuat beberapa celah dengan jarak yang seragam. Pada bagian kiri piringan diletakkan LED (*Light Emission Diode*), pada bagian kanan diletakkan *photo transistor*. Jika cahaya dari LED tidak terhalang oleh piringan (berkas sinar melewati celah) maka *photo transistor* bernilai *high*. Jika cahaya dari LED terhalang oleh piringan (berkas sinar melewati celah) maka *photo transistor* bernilai *low*. Jumlah perubahan kondisi dari *high* ke *low* atau sebaliknya mencerminkan jumlah putaran poros motor. Konstruksi *encoder* dapat dilihat pada gambar 2.12.



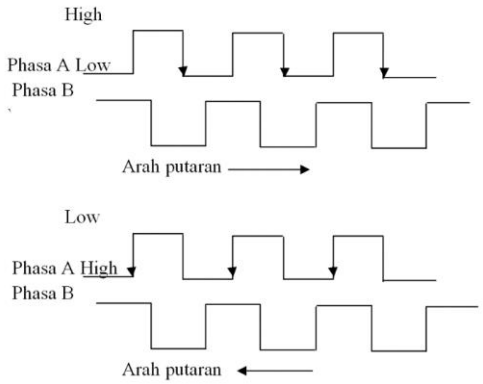
Gambar 2.12 Konstruksi Encoder

Dari gambar 2.12 dapat dilihat bahwa pada konstruksi encoder terdapat fixed slit (celah tetap) yang terbagi menjadi 3 fasa, yaitu : fasa A, fasa B, dan fasa Z. Fasa A & B digunakan untuk menunjukkan kecermatan encoder sementara fasa Z digunakan untuk menyatakan satu pulse untuk satu putaran. Jumlah putaran dihitung berdasarkan sinyal di *output* A/B. Bentuk sinyal pada kaki *output* A dan B dapat dilihat pada gambar 2.13.

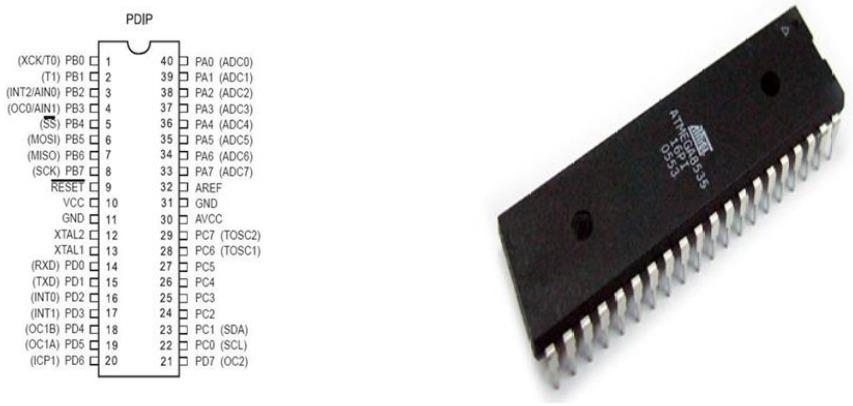
Dari gambar 2.13 dapat dilihat gelombang yang ada pada encoder. Selisih antara gelombang A dan B adalah (1/4) dari panjang gelombang. Kaki A digunakan untuk mengukur jumlah putaran sementara kaki B digunakan untuk menentukan arah putaran poros. Sebagai contoh: bila arah putaran ke kanan, ketika terjadi perubahan kondisi pada kaki A dari high ke low kondisi kaki B bernilai low. Bila arah putaran ke kiri, maka ketika terjadi perubahan pada kaki A dari high ke low kondisi kaki B bernilai high.

2.7 Mikrokontroller ATMega8535

Mikrokontroller adalah perangkat elektronika. Di dalam mikrokontroller terdapat rangkaian kontrol, mikroprosesor, memori, serta *input/output*. Mikrokontroller dapat diprogram menggunakan berbagai macam bahasa pemrograman. Bahasa pemrograman yang dapat digunakan untuk mikrokontroller diantaranya adalah bahasa *assembler*, bahasa C, dan bahasa *basic*. Mikrokontroller dapat digunakan untuk mengendalikan suatu proses secara otomatis seperti sistem kontrol mesin, remote kontrol, sistem kontrol robot dan lain-lain. Salah satu jenis mikrokontroller yang banyak digunakan adalah ATMega8535 Skema dan bentuk Mikrokontroller ATmega8535 dapat dilihat pada gambar 2.14.



Gambar 2.13 Bentuk Sinyal Pada Kaki *Output* A dan B



Gambar 2.14 Skema dan Bentuk Mikrokontroler ATMega8535

2.8 CodeVision AVR

CodeVision AVR adalah salah satu alat bantu pemrograman (*Programming Tool*). CodeVision AVR digunakan untuk membuat program (urutan instruksi). Program yang dibuat dimasukkan dan disimpan di dalam mikrokontroler. Tampilan awal *Code Vision AVR* dapat dilihat pada gambar 2.15.

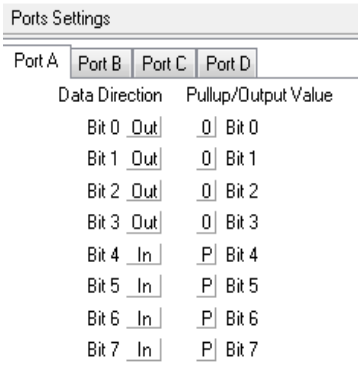
Program *CodeVision* AVR mempunyai beberapa fasilitas. Beberapa fasilitas tersebut adalah *Input/Output* dan *USART*. Beberapa fasilitas Program *CodeVision* AVR akan dibahas di beberapa paragraf berikutnya.

Gambar 2.15 Tampilan Awal Code Vision AVR

2.8.1 Input/*output*

Input/output merupakan fasilitas yang dimiliki mikrokontroler agar mikrokontroler dapat menerima sinyal masukan (*input*) dan memberikan sinyal keluaran (*output*). Sinyal *input* maupun sinyal *output* berupa data digital 1 (*high*) yang mewakili tegangan 5 volt dan data digital 0 (*low*) yang mewakili tegangan 0 volt. Mikrokontroler ATMega 8535 memiliki empat buah port delapan

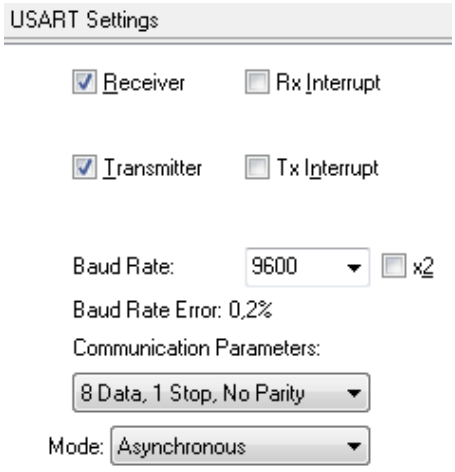
bit yang dapat difungsikan sebagai port *input* maupun sebagai port *output*. Port-port tersebut adalah port A, port B, port C, dan port D. Setiap port mikrokontroller mempunyai tiga buah *register bit*, yaitu DDRx.y, PORTx.y, dan PINx.y. Huruf x mewakili nama port, sedangkan y mewakili nama *bit*. Pada *bit* DDx.y terdapat I/O *address* PORTx sedangkan pada *bit* PINx.y terdapat pada I/O *address* PINx. *Register* PORTx.y digunakan untuk memberi nilai keluaran *high/low* (pada saat difungsikan sebagai *output*). Tampilan pengaturan port mikrokontroller dapat dilihat pada gambar 2.16.



Gambar 2.16 Tampilan Pengaturan PORT Mikrokontroller

2.8.2 *USART*

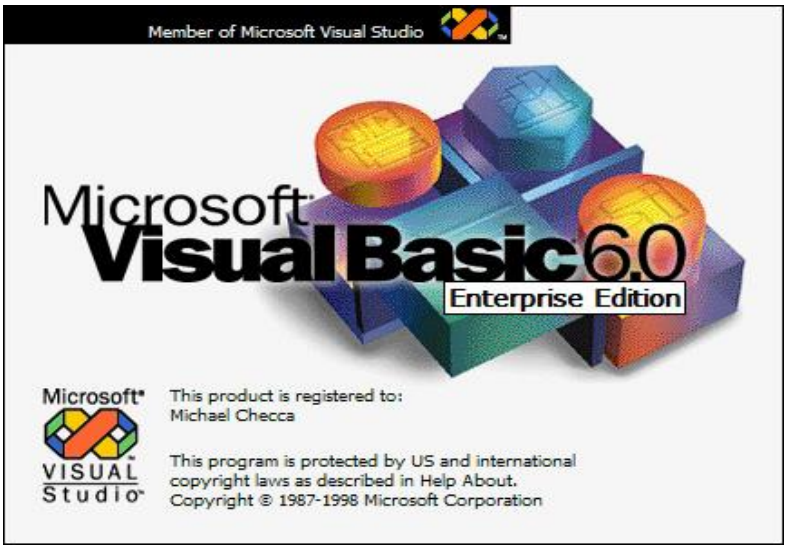
USART adalah fasilitas yang dimiliki mikrokontroller agar mikrokontroller dapat melakukan komunikasi serial antara mikrokontroller dengan komputer. Komunikasi serial dilakukan untuk mengirim data dari mikrokontroller menuju komputer atau mengirim data dari komputer menuju mikrokontroller. Tampilan pengaturan *USART* pada *CodeVision* AVR dapat dilihat pada gambar 2.17.



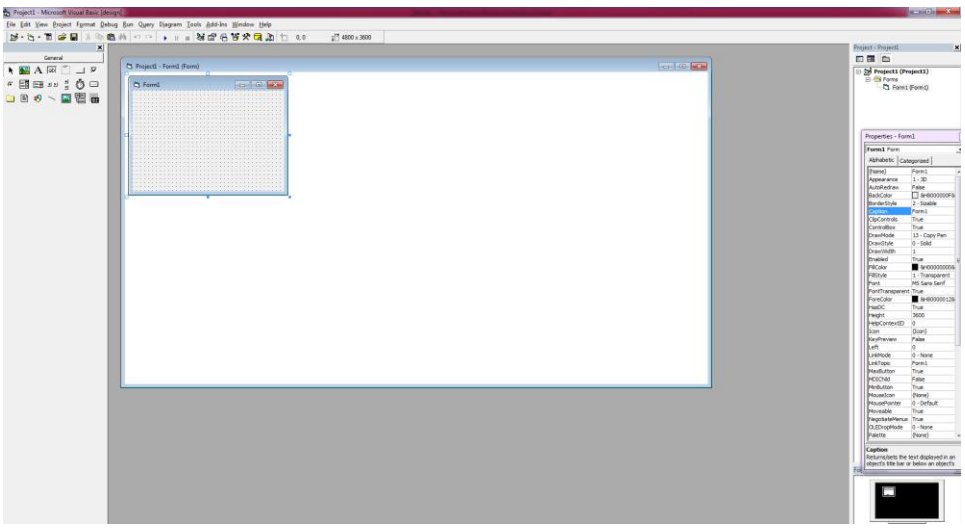
Gambar 2.17 Tampilan Pengaturan *USART* pada *CodeVision AVR*

2.9 *Visual Basic*

Visual Basic adalah salah satu bahasa pemograman komputer. Bahasa pemograman adalah bahasa perintah yang dapat dimengerti oleh komputer untuk melakukan perintah-perintah tertentu. Bahasa pemograman *Visual Basic* yang dikembangkan oleh *Microsoft*, merupakan pengembangan dari pendahulunya yaitu bahasa pemograman BASIC (*Beginner's Allpurpose Symbolic Instruction Code*). Tampilan awal *Visual Basic* dapat dilihat pada gambar 2.18 dan gambar 2.19.



Gambar 2.18 Tampilan Awal *Visual Basic*

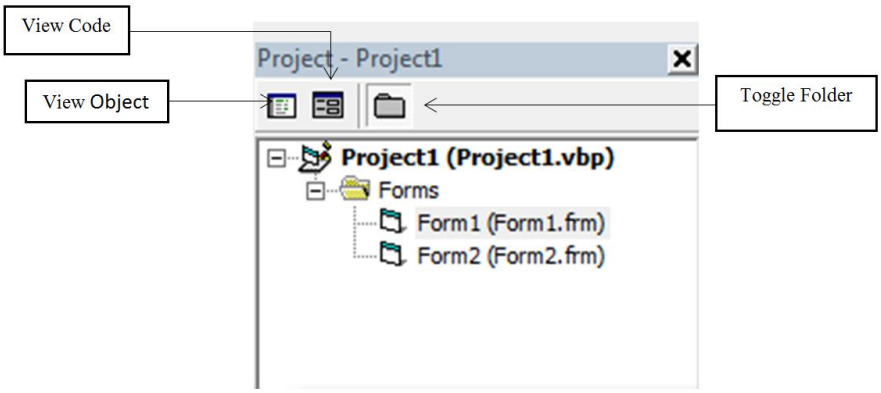


Gambar 2.19 Tampilan Awal Visual Basic

Microsoft Visual Basic 6.0 dikembangkan pada era 1950-an. *Microsoft Visual Basic 6.0* merupakan salah satu *development tool* yaitu alat bantu untuk membuat berbagai macam program komputer. Pembuatan program dikhususkan untuk komputer berbasis operasi *windows*. *Visual basic* 6.0 mempunyai beberapa fasilitas. Beberapa fasilitas *Visual Basic* 6.0 adalah *project*, *from* dan *toolbox*. Beberapa fasilitas *Visual Basic* 6.0 akan dibahas di beberapa paragraf berikutnya.

2.9.1 *Project*

Project adalah sekumpulan modul yang didalamnya terdapat *form* beserta kodenya. *Project* dapat disimpan dalam *file* berformat .vbf. *File* akan menyimpan keseluruhan komponen program termasuk pilihan *project*, pilihan *environment*, dan pilihan *file* .exe. Pada jendela *project* terdapat tiga *icon*, yaitu *icon view code*, *icon view object* dan, *icon toggle folder*. Tampilan *project* pada program *Visual Basic* 6.0 dapat dilihat pada gambar 2.20.

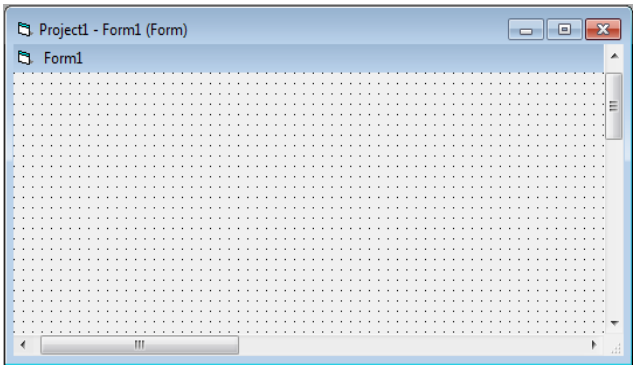


Gambar 2.20 Bentuk *Project*

Icon view code digunakan untuk menampilkan jendela editor kode program. *Icon view object* digunakan untuk menampilkan bentuk *form*. *Icon toggle folder* digunakan untuk menampilkan *folder* atau tempat penyimpanan *folder*.

2.9.2 *Form*

Form adalah *object* yang digunakan sebagai tempat perancangan program. Pada *form* terdapat garis titik-titik yang disebut *grid*. *Grid* sangat berguna membantu pengaturan tata letak *object* yang dimasukkan kedalam *form*. Kursor *mouse* akan menposisikan *object* pada titik-titik *grid*. Bentuk *form* pada *Visual Basic* 6.0 dapat dilihat pada gambar 2.21.



Gambar 2.21 Bentuk *Form*

2.9.3 *ToolBox*

ToolBox adalah kotak alat yang berisi *object-object* tertentu yang akan dimasukkan ke dalam jendela *form*. *Object* yang terdapat di *Toolbox* dapat ditambah. Penambahan *object* di tollbox dilakukan dengan cara kursor diarahkan ke *toolbox*, tombol kanan *mouse* diklik, kemudian tombol *mouse* kiri diklik memilih *component*. Beberapa *object* yang biasa digunakan adalah *Textbox, CommandButton, Listbox, MSCommonDialog,* dan *MSComm*. Beberapa *object* yang terdapat di *toolbox* dapat dilihat pada gambar 2.22.

Gambar 2.22 Beberapa *Object* yang Terdapat di *Toolbox*

2.9.4 *CommandButton*

CommanButton merupakan *object visual basic* yang digunakan untuk mengeksekusi perintah tertentu. *CommanButton* hampir selalu muncul pada semua aplikasi. Tampilan *CommanButton* dapat dilihat pada gambar 2.23.

Gambar 2.23 *CommanButton*

2.9.5 *TextBox*

TextBox merupakan *object Visual Basic* yang digunakan untuk memasukkan nilai yang diperlukan oleh suatu aplikasi program. *TextBox* dapat juga menampilkan suatu hasil perhitungan maupun nilai suatu hasil pengukuran. Tampilan *TextBox* dapat dilihat pada gambar 2.24.

Gambar 2.24 *TextBox*

2.9.6 *ListBox*

ListBox merupakan *object Visual Basic* yang digunakan untuk menampilkan hasil nilai setelah melakukan eksekusi program atau pada saat program berlangsung. Tampilan *ListBox* dapat dilihat pada gambar 2.25.

Gambar 2.25 *ListBox*

Agar dapat mengkoneksikan database dari *Microsoft Access* ke *Visual Basic* perlu dilakukan beberapa langkah pengaturan. Beberapa langkah tersebut yaitu:

1. Kursor diarahkan ke *object Adodc.* Tombol mouse kanan diklik, kemudian Adodc Properties dipilih. *Adodc Properties* dipilih dengan cara tombol mouse kiri diklik.
2. Setelah *Adodc* dipilih akan muncul jendela *Property Pages*. Jendela *Property Pages* menampilkan beberapa tombol. Dari beberapa tombol pilihlah *general*.
3. Setelah tombol *general* dipilih kursor diarahkan ke *Use Connection String* dan *Build* dipilih. *Build* dipilih dengan cara tombol mouse kiri diklik.
4. Setelah *Build* dipilih akan muncul jendela *Data Link Property*. Jendela *Data Link Property* menampilkan beberapa tombol. Dari beberapa tombol pilihlah *Provider*.
5. Setelah *Provider* dipilih kursor diarahkan ke *Microsoft Jet 4.0 OLE DB Provider*. *Microsoft Jet 4.0 OLE DB Provider* dipilih dengan cara tombol mouse kiri diklik.
6. Setelah *Microsoft Jet 4.0 OLE DB Provider* dipilih kursor diarahkan ke tombol *Next*. Tombol *Next* dipilih dengan cara tombol mouse kiri diklik.
7. Setelah Tombol *Next* dipilih akan muncul jendela tombol *Connecction. Database* diisi pada *Textboxt* nomer 1. *Textbox* nomer 1 diisi dengan cara kursor diarahkan ke sebelah kanan *Textbox* nomer 1 yang terdapat tombol pencarian database.

2.9.7 *MSCommonDialog*

MSCommonDialog merupakan *object Visual Basic* yang dapat digunakan untuk menampilkan *dialog box* yang berkaitan dengan penyimpanan maupun pemuatan suatu *file*. Tampilan *MSCommonDialog* dapat dilihat pada gambar 2.26.

Gambar 2.26 *MSCommonDialog*

2.9.8 *MSComm*

MSComm merupakan *object* yang digunakan untuk melakukan komunikasi serial. Objek *MSComm* disimbolkan dengan gambar telepon. Tampilan *MSComm* dapat dilihat pada gambar 2.27.

Gambar 2.27 *MSComm*

Agar dapat melakukan komunikasi secara serial, beberapa properti *object MSComm* perlu diubah. Beberapa properti tersebut adalah:

1. *CommPort*, properti ini diisi dengan jalur komunikasi serial yang akan digunakan,
2. *RTreshold*, properti ini diisi harga satu (1),
3. *SRhreshold,*properti ini diisi harga satu (1), dan
4. *Settings*, properti ini diisi sesuai dengan kecepatan pengiriman data dan jenis komunikasi serial yang dipilih.

Penerimaan data secara serial dilakukan dengan cara memindahkan karakter yang ada pada properti *input* ke memori. Sintaks penulisan perintah penerimaan data secara serial adalah buffer = MSComm1.Input.

Pengiriman data secara serial dilakukan dengan cara mengisi properti *output* dengan karakter yang dikirim. Sintaks penulisan perintah pengiriman data secara serial adalah MSComm1.*Output* = chr (angka).

Ketika komputer selesai mengirim atau menerima data secara serial, program akan secara otomatis mengeksekusi MSComm1.OnComm(). Bila komputer selesai menerima data, maka harga properti ComEvent = ComEvReceive. Bila komputer selesai mengirim data, maka harga properti ComEvent = ComEvSend.

BAB III

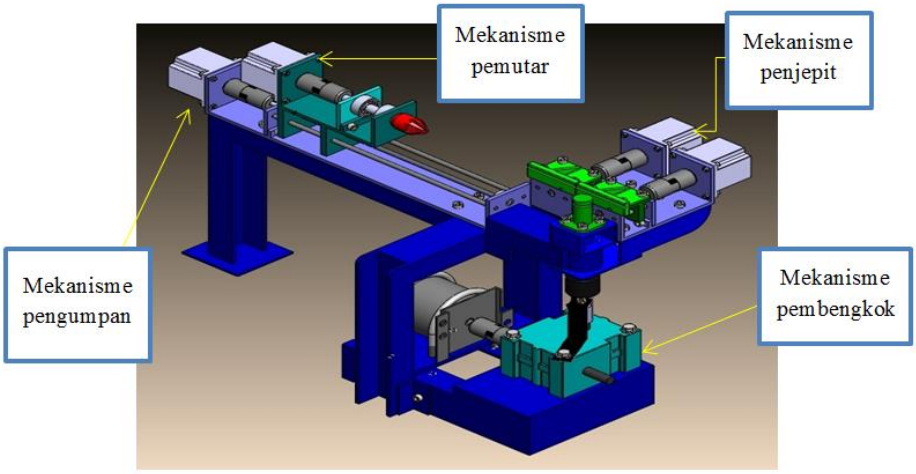
PENGENDALIAN MEKANISME *PROTOTYPE* MESIN PEMBENGKOK BATANG SILINDER DAN MODIFIKASI MEKANISME PENJEPIT BENDA KERJA

Pada bab ini dibahas tentang perangkat mekanik *Prototype* mesin pembengkok batang silinder, konstruksi motor *servo* AC MR-J2S-40A, konstruksi motor *stepper*, *driver* motor *stepper IC L297*, komponen utama motor *servo*, rangkaian mikrokontroller ATMega 8535, rangakaian pengendali *Prototype* mesin pembengkok batang silinder, *CodeVisionAVR*, dan modifikasi mekanisme penjepit.

3.1 *Prototype* Mesin Pembengkok Batang Silinder

Prototype mesin pembengkok batang silinder merupakan mesin yang digunakan untuk mensimulasikan proses pembengkokan batang silinder. Pembengkokan batang silinder umumnya menggunakan dua buah cetakan. Kedua buah cetakan digunakan untuk menjepit material. Salah satu cetakan diberi gaya sehingga material mengalami perubahan bentuk.

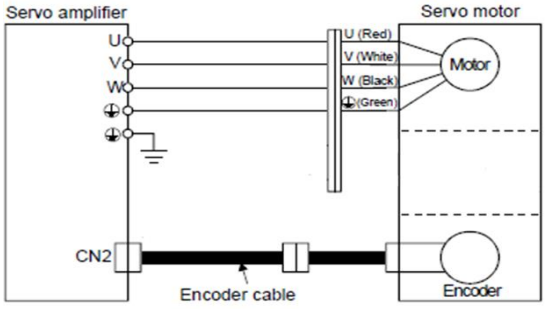
Prototype mesin pembengkok batang silinder memiliki beberapa sub-sistem yaitu, mekanisme sistem pengumpan material, mekanisme sistem pemutar material, mekanisme sistem penjepit material, dan mekanisme sistem pembengkok material. Mekanisme pengumpan, mekansime pemutar, dan mekanisme penjepit menggunakan motor *stepper* sebagai motor penggerak. Mekanisme pembengkok menggunakan motor *servo* AC sebagai motor penggerak. Perangkat mekanisme *Prototype* mesin pembengkok dapat dilihat pada gambar 3.1.



Gambar 3.1 *Prototype* Mesin Pembengkok Batang Silinder

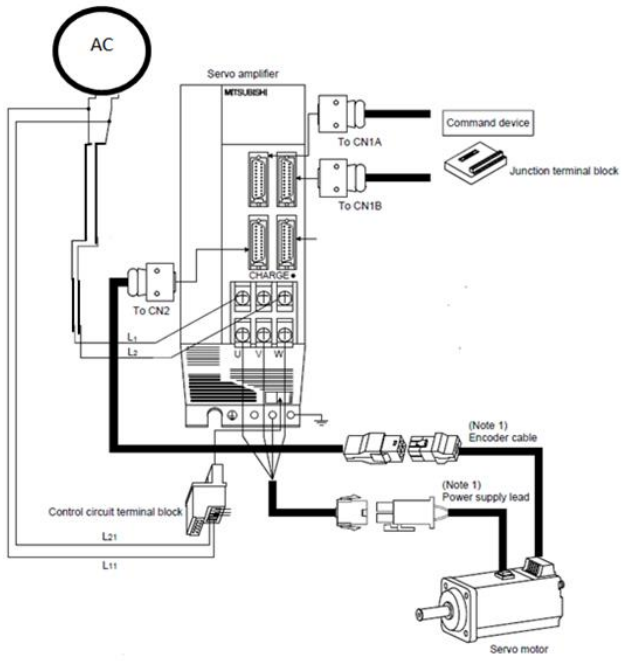
3.2 Motor *Servo* AC HC-KFS43

Motor *servo* AC HC-KFS43 adalah motor *servo* yang memiliki daya sebesar 400 watt, torsi maksimum 3,8 N.m dan kecepatan putar maksimum 4500 rpm. Skematis instalasi motor *servo* AC HC-KFS43 dapat dilihat pada gambar 3.2.



Gambar 3.2 Instalasi Motor *Servo* AC HC-KFS43

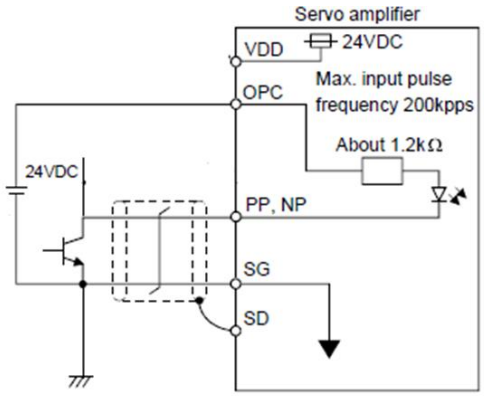
Driver motor *servo* AC MR-J2S-40A adalah *driver* yang dapat digunakan untuk mengendalikan motor *servo* AC HC-KFS43. Instalasi *driver* motor *servo* AC MR-J2S-40A dapat dilihat pada gambar 3.3.



Gambar 3.3 Instalasi *Driver* Motor *Servo* AC HC-KFS43

Dari skematis instalasi *driver* motor *servo* AC MR-J2S-40A dapat dijelaskan bahwa *wiring system* motor *servo* AC ter*dir*i dari:

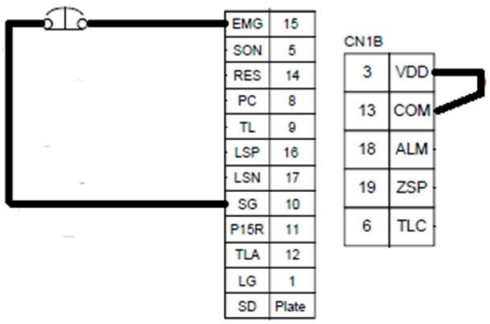
1. Kabel *power supply* AC dihubungkan ke terminal kabel L1 dan kabel L2 yang terdapat pada *driver* motor *servo* AC MR-J2S-40A.
2. Kaki L11 dihubungkan ke terminal L1 dan kaki L21 dihubungkan ke kaki L2. Hubungan ini perlu dilakukan agar *driver* motor *servo* AC aktif.
3. Kabel UVW dan kabel Ground dihubungkan ke motor *servo* AC sesuai warna yang sudah ditentukan.
4. Pin *conector* CN2 dihubungkan ke *encoder* motor *servo* AC MR-J2S-40A.
5. Pin *conector* CN1A dihubungkan ke beberapa kaki yang terdapat di terminal *conector* CN1A. Beberapa kaki pada *pin connector* CN1A yang dapat dipakai sesuai *position control mode* adalah PP (kaki 2), NP (kaki 3), SG (kaki 10) dan OPC (kaki 11). Skematis instalasi CN1A dapat dilihat pada gambar 3.4.
6. Pin *connector* CN1B dihubungkan ke beberapa kaki yang terdapat di terminal *connector* CN1B. Beberapa kaki pada pin conector CN1B yang dapat dipakai sesuai *position control mode* adalah VDD (kaki 3), COM (kaki 13), SG (kaki 10) dan EMG (kaki 15). Skematis instalasi CN1B dapat dilihat pada gambar 3.5.



Gambar 3.4 Skematis instalasi Pin *Conector* CN1A *Driver* HC-KFS43

Dari skematis instalasi pin *conector* CN1A dapat dilihat bahwa kaki OPC (kaki 2) dihubungkan ke VCC 24 Volt. Kaki SG (kaki 10) dihubungkan Ground 24Volt. Kaki PP (kaki 2) dan kaki NP (kaki 3) dihubungkan ke kaki SG menggunakan transistor. Kaki basis transistor dapat diberi *pulse* dengan dihubungkan ke mikrokontroller. Prinsip kerja *driver* motor *servo* AC MR-J2S-40A adalah, jika ada perubahan kondisi dari high ke low atau low ke high (koneksitas) di kaki PP atau NP yang dihubungkan dengan kaki SG maka poros motor *servo* AC bergeser satu *step*.

Dari skematis instalasi pin *conector* CN1B dapat dijelaskan bahwa terdapat beberapa kaki yang saling dihubungkan. Beberapa kaki di *pin conector* CN1B yang saling dihubungkan yaitu: kaki VDD (kaki 3) dengan kaki COM (kaki 13) dan kaki EMG (kaki 15) dengan kaki SG (kaki 10). Kaki VDD dan kaki COM dihubungkan agar CN1B dapat dialiri listrik secara internal. Kaki EMG dan kaki SG dihubungkan agar status EMG off berubah menjadi status EMG on.



Gambar 3.5 Skematis instalasi Pin Conector CN1B *Driver* Motor *Servo* MR-J2S-40A

Driver motor *servo* AC MR-J2S-40A memiliki daya keluaran 400 watt. *Input driver* motor *servo* AC MR-J2S-40A dapat menggunakan 1 phasa atau 3 phasa. *Control mode* yang digunakan pada sistem kendali *underground parking* adalah *position control mode*. Untuk menentukan control mode *driver* motor *servo* pada position control mode diperlukan pengaturan beberapa paramaeter. Beberapa parameter yang diatur sesuai *Position control mode* dapat dilihat pada tabel 3.1, tabel 3.2, tabel 3.3 dan tabel 3.4.

Parameter 0 digunakan untuk menentukan mode pengaturan (*control mode*) yang digunakan motor *servo* AC HC-KFS43. Parameter 0 mempunyai 4 digit karakter. Karakter di kolom 1 dan di kolom 2 menentukan *driver* motor *servo* AC yang digunakan untuk mengendalikan motor *servo* AC. Karakter di kolom 3 diisi karakter 0. Karakter di kolom 4 diisi karakter 0. Karakter 0 yang diisi di kolom 4 berfungsi untuk mengendalikan motor *servo* AC dengan *mode position control mode*.

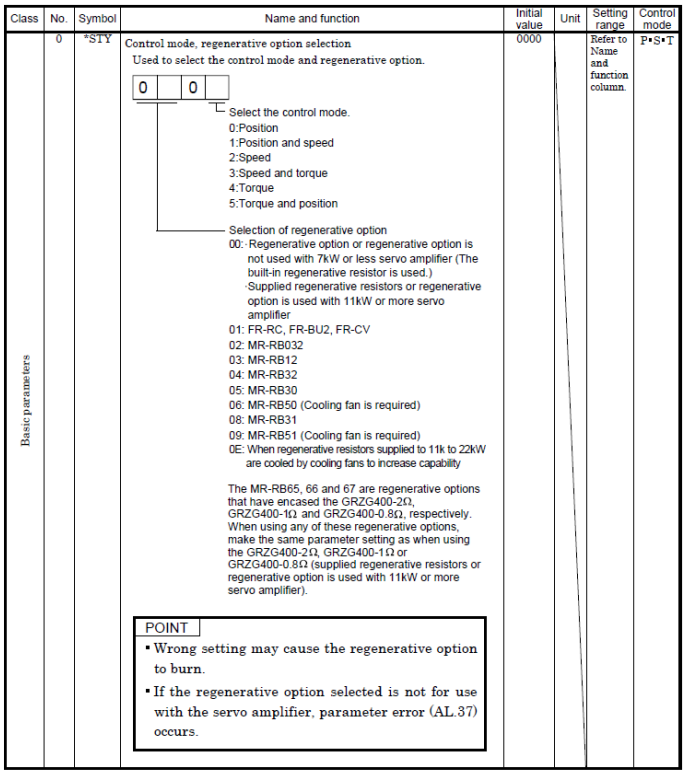
Dari tabel 3.2 dapat dilihat bahwa parameter 1 digunakan untuk untuk memilih filter input sinyal pin CN1B di kaki 19 dan sistem deteksi posisi *absolute.* Parameter 1 mempunyai 4 digit karakter. Karakter di parameter 1 diisi sesuai initial value yang sudah di tentukan oleh parameter 1.

Dari tabel 3.3 dapat dilihat bahwa parameter 2 digunakan untuk pengaturan *driver* terhubung secara otomatis dengan motor *servo* AC. Parameter 2 digunakan sesuai seleksi tingkat respon pengaturan otomatis, karakter di kolom 2 parameter 2 diisi karakter 1. Karakter di kolom 4 diisi.

karakter 5. Karakter 5 yang diisi di kolom 4 memiliki response level diantara low response dan middle response dengan machine resonanse frequency guidline sebesar 35Hz.

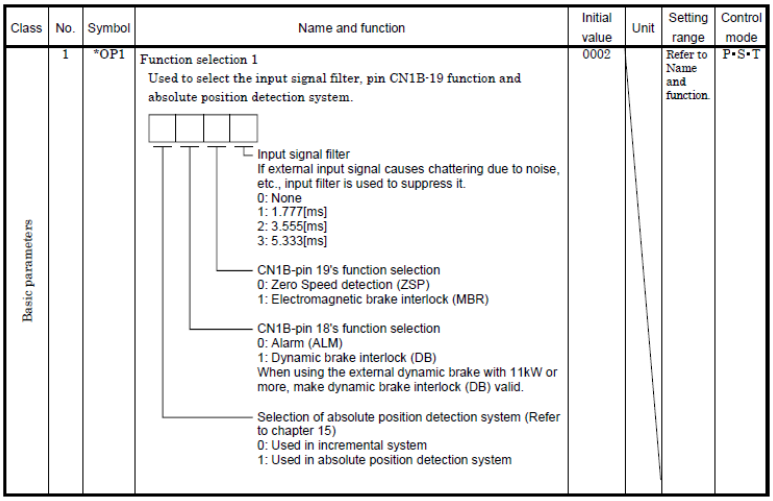
Dari tabel 3.4 dapat dilihat bahwa parameter 3 dan parameter 4 digunakan untuk mengatur resolusi *pulse*/putaran dari motor *servo* AC. Resulosi motor *servo* AC HC-KFS43 ditetapkan dengan nilai 131072. Resolusi jumlah *pulse*/putaran dari motor *servo* dapat dicari dengan menggunakan persamaan 3.1.

Tabel 3.1 Parameter 0 *Driver* Motor *Servo* MR-J2S-40A



Class	No.	Symbol	Name and function	Initial value	Unit	Setting range	Control mode
Basic parameters	0	*STY	Control mode, regenerative option selection Used to select the control mode and regenerative option. 0 0 Select the control mode. 0:Position 1:Position and speed 2:Speed 3:Speed and torque 4:Torque 5:Torque and position Selection of regenerative option 00: ·Regenerative option or regenerative option is not used with 7kW or less servo amplifier (The built-in regenerative resistor is used.) ·Supplied regenerative resistors or regenerative option is used with 11kW or more servo amplifier 01: FR-RC, FR-BU2, FR-CV 02: MR-RB032 03: MR-RB12 04: MR-RB32 05: MR-RB30 06: MR-RB50 (Cooling fan is required) 08: MR-RB31 09: MR-RB51 (Cooling fan is required) 0E: When regenerative resistors supplied to 11k to 22kW are cooled by cooling fans to increase capability The MR-RB65, 66 and 67 are regenerative options that have encased the GRZG400-2Ω, GRZG400-1Ω and GRZG400-0.8Ω, respectively. When using any of these regenerative options, make the same parameter setting as when using the GRZG400-2Ω, GRZG400-1Ω or GRZG400-0.8Ω (supplied regenerative resistors or regenerative option is used with 11kW or more servo amplifier). POINT • Wrong setting may cause the regenerative option to burn. • If the regenerative option selected is not for use with the servo amplifier, parameter error (AL.37) occurs.	0000		Refer to Name and function column.	P•S•T

Tabel 3.2 Parameter 1 *Driver* Motor *Servo* MR-J2S-40A



Class	No.	Symbol	Name and function	Initial value	Unit	Setting range	Control mode
Basic parameters	1	*OP1	Function selection 1 Used to select the input signal filter, pin CN1B-19 function and absolute position detection system. Input signal filter If external input signal causes chattering due to noise, etc., input filter is used to suppress it. 0: None 1: 1.777[ms] 2: 3.555[ms] 3: 5.333[ms] CN1B-pin 19's function selection 0: Zero Speed detection (ZSP) 1: Electromagnetic brake interlock (MBR) CN1B-pin 18's function selection 0: Alarm (ALM) 1: Dynamic brake interlock (DB) When using the external dynamic brake with 11kW or more, make dynamic brake interlock (DB) valid. Selection of absolute position detection system (Refer to chapter 15) 0: Used in incremental system 1: Used in absolute position detection system	0002		Refer to Name and function.	P•S•T

Beberapa parameter di*setting* untuk melengkapi sistem kendali *Position control mode.* Parameter pelengkap sistem kendali *Position control mode* dapat dilihat pada tabel 3.5, tabel 3.6 dan tabel 3.7.

Dari tabel 3.6 dapat dilihat bahwa parameter 21 digunakan untuk memilih input *pulse* yang diinginkan. Parameter 21 mempunyai 4 digit karakter. Karakter di kolom 1 dan di kolom 2

diisi dengan karakter 0. Karakter di kolom 3 diisi dengan karakter 0. Karakter 0 yang diisi di kolom 3 berfungsi untuk mengendalikan motor *servo* AC dengan *input* yang dimulai dari *positive logic*. Karakter di kolom 4 diisi dengan karakter 0. Karakter 0 yang diisi di kolom 4 berfungsi untuk menentukan putaran poros motor *servo* AC.

Tabel 3.3 Parameter 2 *Driver* Motor *Servo* MR-J2S-40A

Class	No.	Symbol	Name and function	Initial value	Unit	Setting range	Control mode
Basic parameters	2	ATU	Auto tuning Used to selection the response level, etc. for execution of auto tuning. Refer to chapter 7. 0 [ ] 0 [ ] Response level setting Set value / Response level / Machine resonance frequency guideline 1 Low response 15Hz 2 20Hz 3 25Hz 4 30Hz 5 35Hz 6 45Hz 7 Middle response 55Hz 8 70Hz 9 85Hz A 105Hz B 130Hz C 160Hz D 200Hz E High response 240Hz F 300Hz · If the machine hunts or generates large gear sound, decrease the set value. · To improve performance, e.g. shorten the settling time, increasethe set value. Gain adjustment mode selection (For more information, refer to section 7.1.1.) Set value / Gain adjustment mode / Description 0 Interpolation mode: Fixes position control gain 1 (parameter No. 6). 1 Auto tuning mode 1: Ordinary auto tuning. 2 Auto tuning mode 2: Fixes the load inertia moment ratio set in parameter No. 34. Response level setting can be changed. 3 Manual mode 1: Simple manual adjustment. 4 Manual mode 2: Manual adjustment of all gains.	7kW or less: 0105 11kW or more: 0102		Refer to Name and function column.	P•S

Tabel 3.4 Parameter 3 dan Parameter 4 *Driver* Motor *Servo* MR-J2S-40A

3	CMX	Electronic gear numerator Used to set the electronic gear numerator value. For the setting, refer to section 5.2.1. Setting "0" automatically sets the resolution of the servo motor connected. For the HC-MFS series, 131072 pulses are set for example.	1		0 1 to 65535	P
4	CDV	Electronic gear denominator Used to set the electronic gear denominator value. For the setting, refer to section 5.2.1.	1		1 to 65535	P

$$\frac{\text{Jumlah Pulse /Putaran}}{\text{Resolusi Motor}} = \frac{CDV}{CMX}$$ .....persamaan 3.1.

Keterangan:

CDV = *Elektronik gear denominator*

CMX = *Elektronik gear numerator*

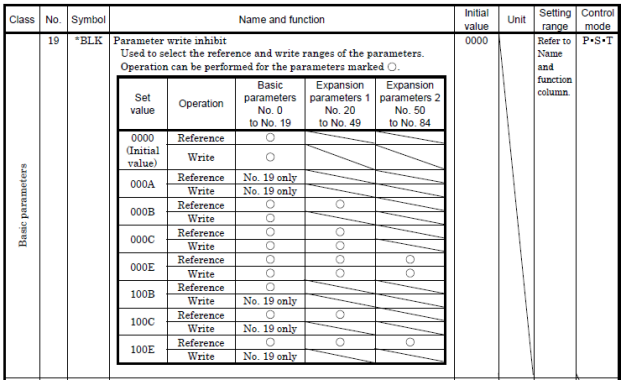
Jika diinginkan jumlah *pulse* per putaran sama dengan 360 maka nilai CDV dan CMX adalah:

$$\frac{CDV}{CMX} = \frac{\text{Jumlah Puls /Putaran}}{\text{Resolusi Motor}}$$

$$= \frac{360}{131072}$$

$$= \frac{45}{16384}$$

Tabel 3.5 Parameter 19 *Driver* Motor *Servo* MR-J2S-40A



Class	No.	Symbol	Name and function	Initial value	Unit	Setting range	Control mode
Basic parameters	19	*BLK	Parameter write inhibit Used to select the reference and write ranges of the parameters. Operation can be performed for the parameters marked ○.	0000		Refer to Name and function column.	P•S•T

Set value	Operation	Basic parameters No. 0 to No. 19	Expansion parameters 1 No. 20 to No. 49	Expansion parameters 2 No. 50 to No. 84
0000 (Initial value)	Reference	○		
	Write	○		
000A	Reference	No. 19 only		
	Write	No. 19 only		
000B	Reference	○	○	
	Write	○		
000C	Reference	○	○	
	Write	○	○	
000E	Reference	○	○	○
	Write	○	○	○
100B	Reference	○		
	Write	No. 19 only		
100C	Reference	○	○	
	Write	No. 19 only		
100E	Reference	○	○	○
	Write	No. 19 only		

Dari tabel 3.5 dapat dilihat bahwa parameter 19 digunakan untuk mengaktifkan status parameter yang belum terlihat di *display driver* motor *servo*. Nilai *initial value* ditetapkan sebagai patokan untuk men*setting* parameter 19.

Tabel 3.6 Parameter 21 *Driver* Motor *Servo* MR-J2S-40A

Class	No.	Symbol	Name and function	Initial value	Unit	Setting range	Control mode
	21	*OP3	Function selection 3 (Command pulse selection) Used to select the input form of the pulse train input signal. (Refer to section 3.4.1) 0 0 _ _ Command pulse train input form 0: Forward/reverse rotation pulse train 1: Signed pulse train 2: A • B-phase pulse train Pulse train logic selection 0: Positive logic 1: Negative logic	0000		Refer to Name and function column.	P

Dari gambar 3.7 dapat dilihat bahwa Parameter 41 digunakan untuk mengaktifkan motor *servo* AC, LSP dan LSN secara otomatis saat diberi *power supply*. Nilai *initial value* ditetapkan sebagai patokan untuk men*setting* parameter 41.

3.3 Motor *Stepper*

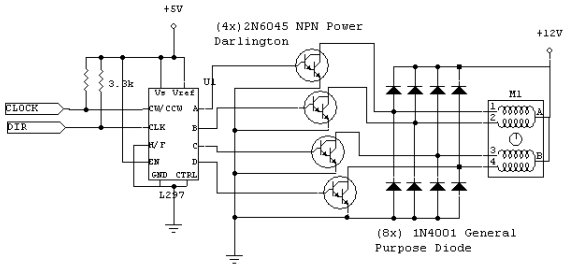
Motor *stepper* adalah motor DC yang khusus berputar dalam suatu derajat yang tetap yang disebut langkah (*step*). Satu *step* motor *stepper* bernilai antara 0,9° - 1,8°. Pada motor *stepper* terdapat *rotor* dan *stator*. *Rotor* adalah permanen magnet sedangkan *stator* adalah *elektromagnet*. *Rotor* akan bergerak ketika *stator* diberi aliran listrik. Aliran listrik yang dihasilkan stator dapat membangkitkan medan magnet dan membuat *rotor* menyesuaikan dengan kutub magnet yang dimilikinya. Tampilan motor *stepper* dapat dilihat pada gambar 3.6.

Tabel 3.7 Parameter 41 *Driver* Motor *Servo* MR-J2S-40A

41	*DIA	Input signal automatic ON selection Used to set automatic Servo-on (SON) ▪ forward rotation stroke end (LSP) ▪ reveres rotation stroke end (LSN). 0 _ _ _ Servo-on (SON) input selection 0: Switched on/off by external input. 1: Switched on automatically in servo amplifier. (No need of external wiring) Forward rotation stroke end (LSP) input selection 0: Switched on/off by external input. 1: Switched on automatically in servo amplifier. (No need of external wiring) Reverse rotation stroke end (LSN) input selection 0: Switched on/off by external input. 1: Switched on automatically in servo amplifier. (No need of external wiring)	0000		Refer to Name and function column.	P▪S▪T P▪S

Gambar 3.6 Bentuk fisik motor *stepper*

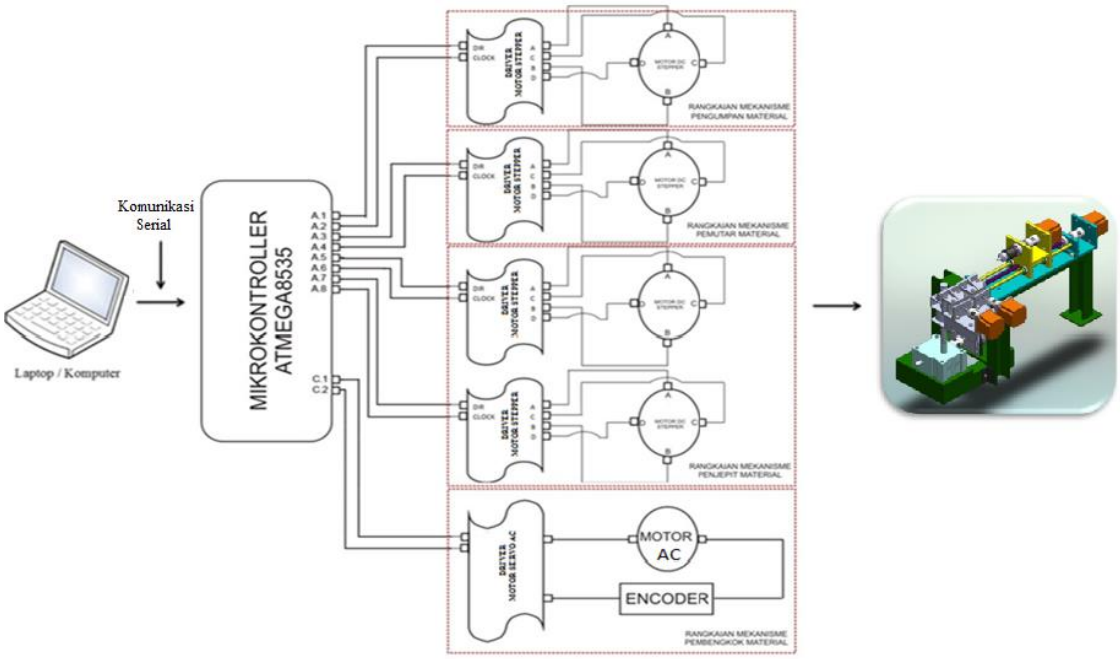
Rangkaian *driver* motor *stepper* adalah *driver* yang berfungsi untuk menggerakkan motor *stepper*. *Driver* motor *stepper* menggunakan *IC L297* yang memiliki 2 kaki input dan 4 kaki *output*. Kaki input pada *IC L297* ter*dir*i dari 2 kaki input yaitu, kaki *dir* (penentu arah putaran) dan kaki *clock* (pemberi sinyal pada kaki *dir*). Kaki *output* pada *IC L297* ter*dir*i dari 4 kaki *output* yaitu, A, B, C, dan D. Skematis rangkaian *driver* motor *stepper* dapat dilihat pada gambar 3.7.



Gambar 3.7 Skematis rangkaian *driver* motor *stepper*

3.4 Sistem Pengendali *Prototype* Mesin Pembengkok Batang Silinder

Sistem kendali *prototype* mesin pembengkok batang silinder adalah sistem yang berfungsi untuk menggerakkan empat motor *stepper* dan satu motor *servo* AC. Keempat motor *stepper* digunakan untuk menggerakkan mekanisme pengumpan, mekanisme pemutar, dan dua mekanisme penjepit. Motor *servo* AC digunakan untuk menggerakkan mekanisme pembengkok. Sistem kendali *prototype* mesin pembengkok batang silinder ter*dir*i dari PC(Personal Computer), mikrokontroller ATMega8535, *driver* motor *servo* AC MR-J2S-40A, *driver* motor *stepper*. Skematis sistem kendali *prototype* mesin pembengkok batang silinder dapat dilihat pada gambar 3.8.



Gambar 3.8 Skematis Pengendali *Prototype* Mesin Pembengkok Batang Silinder

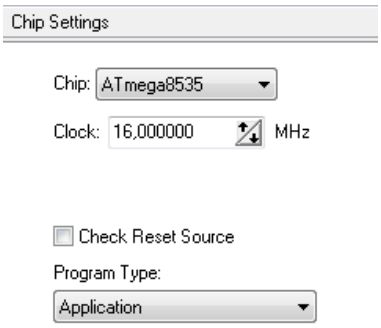
3.5 Program Pengendali *Prototype* Mesin Pembengkok Batang Silinder

Program pengendali sistem *prototype* mesin pembengkok batang silinder digunakan untuk menggerakkan empat motor *stepper* dan satu motor *servo* AC secara otomatis oleh komputer. *Software* yang digunakan untuk pengendali sistem *prototype* mesin pembengkok batang silinder adalah dan CodeVision AVR Visual Basic 6.0. Kedua *software* dipilih dikarenakan mudah untuk digunakan dan dimengerti. Kedua *software* yang digunakan untuk pengendali sistem *prototype* mesin pembengkok batang silinder akan dibahas di beberapa paragraf berikutnya.

3.5.1 *CodeVision AVR*

CodeVision AVR digunakan sebagai program penghubung antara komputer dengan mikrokontroller. Program yang dibuat di *CodeVision AVR* adalah program yang dapat mengirim data dari mikrokontroller ke komputer dan menerima data dari komputer ke mikrokontroller. Data yang diterima mikrokontroller dari komputer berupa sinyal *digital*. Sinyal *digital* dikirim ke *driver* motor *servo* AC untuk menggerakkan motor *servo* AC.

Hal tersebut dapat dilakukan dengan menggunakan komunikasi antara mikrokontroller dengan komputer. Komunikasi yang digunakan yaitu komunikasi serial secara *asinkron* antara mikrokontroller dengan komputer. Sebelum program di *CodeVision AVR* dibuat, terdapat beberapa fitur yang harus diatur. Beberapa fitur tersebut yaitu : jenis Chip, *Clock*, dan PORT. Chip yang digunakan untuk program pengendali sistem *prototype* mesin pembengkok batang silinder adalah mikrokontroller ATMega 8535 dengan nilai *Clock* sebesar 16 MHz. Tampilan pengaturan Chip dan nilai *Clock* dapat dilihat pada gambar 3.9.



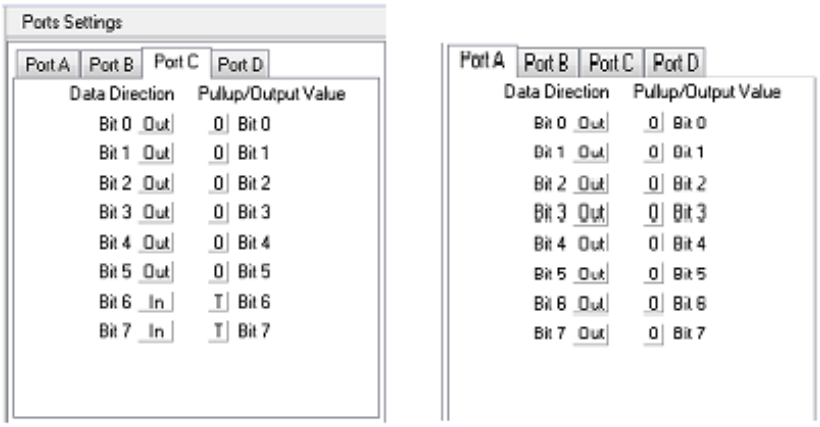
Gambar 3.9 Tampilan Pengaturan *Chip* dan *Clock*

Mikrokontroller ATMega8535 mempunyai empat buat PORT yaitu PORTA, PORTB, PORTC, PORTD. Keempat buah PORT memiliki delapan kaki. PORT yang digunakan untuk pengendali sistem *prototype* mesin pembengkok adalah PORTA dan PORTC. PORTA digunakan

untuk motor *stepper* dan PORTC digunakan untuk motor *servo*. Beberapa kaki di PORTA dan di PORTC diatur sebagai *output*. Tampilan pengaturan PORT dan beberapa kaki yang digunakan sebagai *output* dapat dilihat pada gambar 3.10.

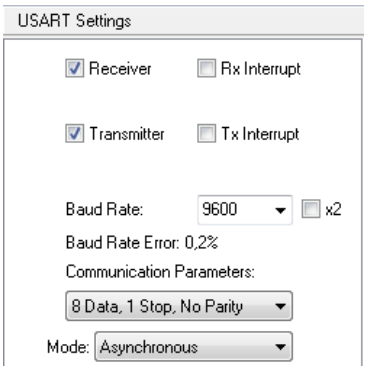
Pengiriman data yang digunakan untuk menggerakkan empat buah motor *stepper* dan satu motor *servo* AC dilakukan secara serial. Agar mikrokontroler dapat mengirim secara serial diperlukan pengaturan *USART*. *USART* diatur dengan cara memberi tanda centang pada kolom receiver dan transmitter. Tampilan pengaturan *USART* dapat dilihat paa gambar 3.11.

Agar cara kerja program penerimaan data dari komputer ke mikrokontroller dapat mudah dipahami dibuatlah diagram alir. Tampilan diagram alir penerimaan data dari komputer ke mikrokontroller dilihat pada gambar 3.12.



Gambar 3.10

Tampilan Pengaturan PORTC dan PORTA Yang Digunakan Pada Motor *Servo* dan Motor *Stepper*

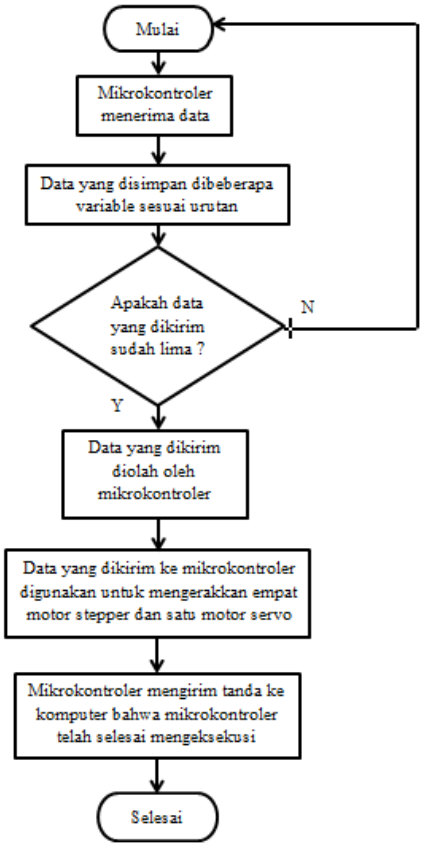


Gambar 3.11 Tampilan Pengaturan *USART*

3.5.2 Visual Basic 6.0

Visual Basic digunakan sebagai program untuk mengirim sejumlah data yang telah diolah dari komputer ke mikrokontroller. Data yang dikirim dari komputer ke mikrokontroller dilakukan menggunakan komunikasi serial. Beberapa data yang dikirim dari komputer menuju mikrokontroller ditampilkan di komputer agar komputer dapat mengetahui data yang telah diterima oleh mikrokontroller. Sebelum dilakukan komunikasi serial antara komputer dengan mikrokontroller beberapa *properties* milik *object* MSComm perlu diatur ulang. Beberapa *properties* milik *object* MSComm yang perlu diatur ulang dapat dilihat pada Tabel 3.8.

Data arah putaran, data jumlah langkah (*step*), data waktu (delay) dan data posisi motor dikirim ke mikrokontroller. Mikrokontroller menerjemahkan data arah putaran, data jumlah langkah (*step*), waktu (delay) dan posisi motor menjadi sinyal digital. Sinyal digital digunakan untuk menggerakkan empat motor *stepper* dan satu motor *servo*. Data yang dikirim dari komputer ke mikrokontroller dilakukan menggunakan komunikasi serial. Beberapa data yang dikirim dari komputer ke mikrokontroller ditampilkan di komputer agar komputer dapat mengetahui data yang telah diterima oleh mikrokontroller.



Gambar 3.12 Diagram Alir Penerimaan Data

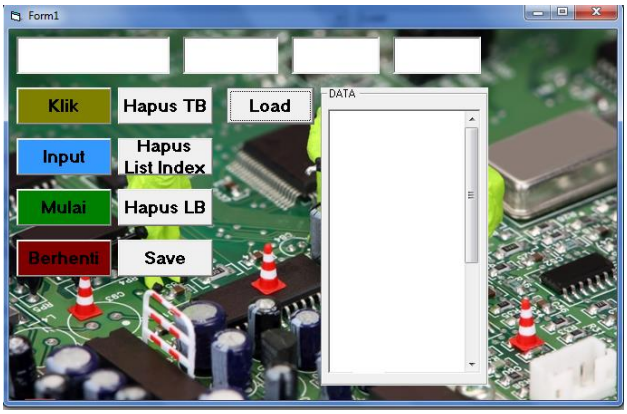
Tabel 3.8 Pengaturan *MSComm*

Objek	Properti	Harga
MSComm	Name	MSComm1
	Rthreshold	1
	Srhreshold	1
	Setting	9600,n,8,1

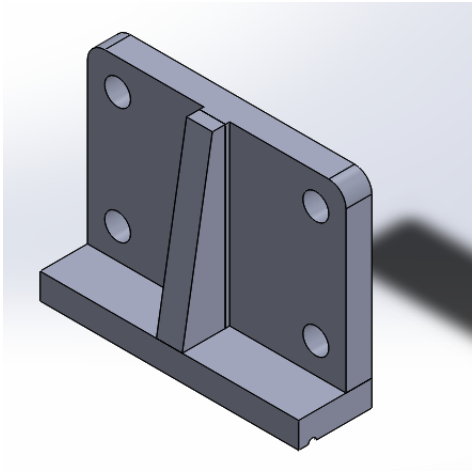
Tahapan pembuatan program pengendali sistem *prototype* mesin pembengkok batang silinder dimulai dengan mendesain Form. Form berisi beberapa object yang digunakan untuk mengirim sejumlah data ke mikrokontroller. Data-data yang dikirim berupa arah putaran, jumlah putaran (*step*), waktu (*delay*) dan posisi motor. Tampilan Form dapat dilihat pada gambar 3.13.

3.6 Modifikasi Dies Pada Mekanisme Penjepit

Mekanisme penjepit pada *Prototype* mesin pembengkok merupakan mekanisme untuk menjepit benda kerja saat material akan dibengkokkan. Pada mekanisme penjepit terdapat kendala pada saat benda kerja diputar oleh mekanisme pemutar. Modifikasi dies pada mekanisme penjepit bertujuan untuk mengoptimalkan kinerja mekanisme pemutar pada saat memutarkan benda kerja ke arah kiri. Dies pada mekanisme penjepit sebelum dimodifikasi dapat dilihat pada gambar 3.14.

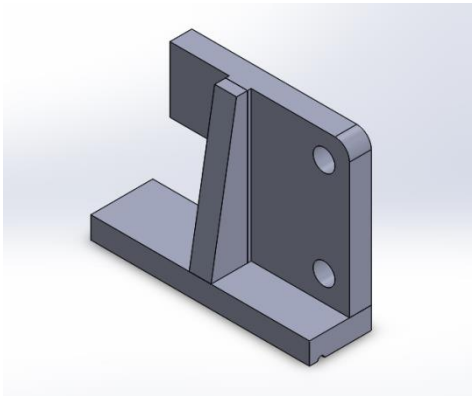


Gambar 3.13 Tampilan Form



Gambar 3.14 Dies Pada Mekanisme Penjepit Sebelum Dimodifikasi

Dies pada mekanisme penjepit setelah dimodifikasi mempunyai dua lubang yang masing-masing berdiameter 6 mm. Dies setelah dimodifikasi dapat dilihat pada gambar 3.15.



Gambar 3.15 Dies Setelah Dimodifikasi

BAB IV

PENGUJIAN DAN ANALISA HASIL PENGUJIAN *PROTOTYPE* MESIN PEMBENGKOK

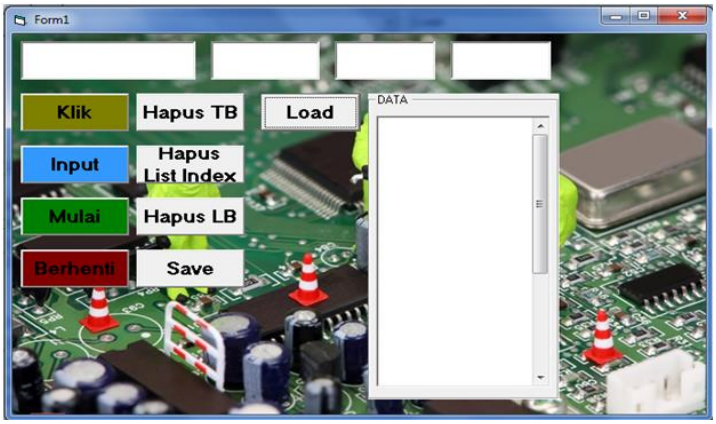
Pada bab ini dibahas tentang pengujian *prototype* mesin pembengkok yang dikendalikan oleh program dan analisa hasil pengujian *prototype* mesin pembengkok batang silinder.

4.1 Pengujian *Prototype* Mesin Pembengkok Batang Silinder dengan Menggunakan Program Visual Basic 6.0

Pengujian *prototype* mesin pembengkok batang silinder bertujuan untuk memastikan apakah sistem kontrol yang telah dibuat sesuai dengan tujuan pembuatan. Pengujian dilakukan dengan dua cara, pertama pengujian tanpa benda kerja, dan yang kedua pengujian dengan menggunakan dengan benda kerja.

Pengendalian *prototype* mesin pembengkok batang silinder menggunakan program yang dibuat pada *visual basic*. Pada *visual basic*, gerak laju mekanisme pengumpan, mekanisme pemutar, mekanisme penjepit, dan mekanisme pembengkok ditentukan dengan memasukkan data-data yang kemudian disimpan pada *listbox*. Data yang disimpan pada *listbox* kemudian dikirimkan ke mikrokontroler menggunakan jalur komunikasi secara serial.

Pada *form* pengendalian penggerak *prototype* mesin pembengkok batang silinder terdapat objek MSComm, CommDialog, 9 *commandbutton* untuk fungsi perintah gerakan**,** 1 *listbox* untuk menampilkan data gerakan, dan 4 *textbox* yang digunakan untuk memasukkan data gerakan. Tampilan *form* pengendalian penggerak *prototype* mesin pembengkok batang silinder dapat dilihat pada gambar 4.1.

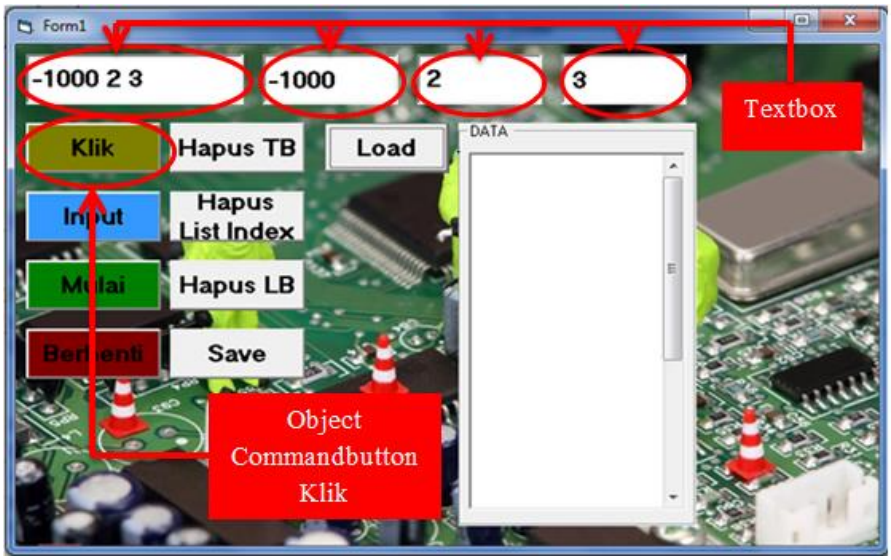


Gambar 4.1 Form Pengendali *Prototype* Mesin Pembengkok Batang Silinder

Pengujian ini bertujuan untuk memastikan apakah sistem kontrol yang telah *dir*ancang dapat bekerja sesuai dengan tujuan pembuatan. Pengujian yang dilakukan adalah menguji fungsi tombol **Klik**, tombol input, tombol mulai, tombol berhenti, tombol hapus textbox, tombol hapus listindex, tombol hapus listbox, tombol save dan tombol load.

4.1.1 Pengujian Object Commandbutton Klik

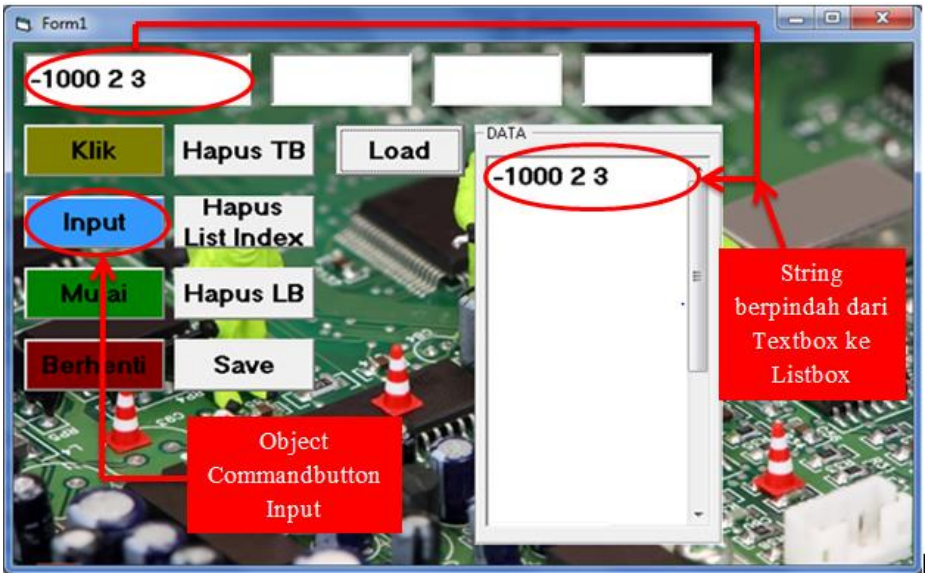
Pengujian object commandbutton **Klik** dilakukan dengan cara memasukkan string ke dalam textbox. Setelah string dimasukan ke dalam textbox lalu kursor diarahkan ke arah object commandbutton **Klik**. Tombol mouse sebelah kiri di-klik. Setelah tombol mouse sebelah kiri di-klik, string yang ada di textbox satu terurai ke textbox dua, textbox tiga, dan textbox empat dan salah satu motor penggerak pada *prototype* mesin pembengkok akan bergerak. Pengujian object commandbutton **Klik** dapat dilihat pada gambar 4.2.



Gambar 4.2 Pengujian Object Commandbutton Klik

4.1.2 Pengujian Object Commandbutton Input

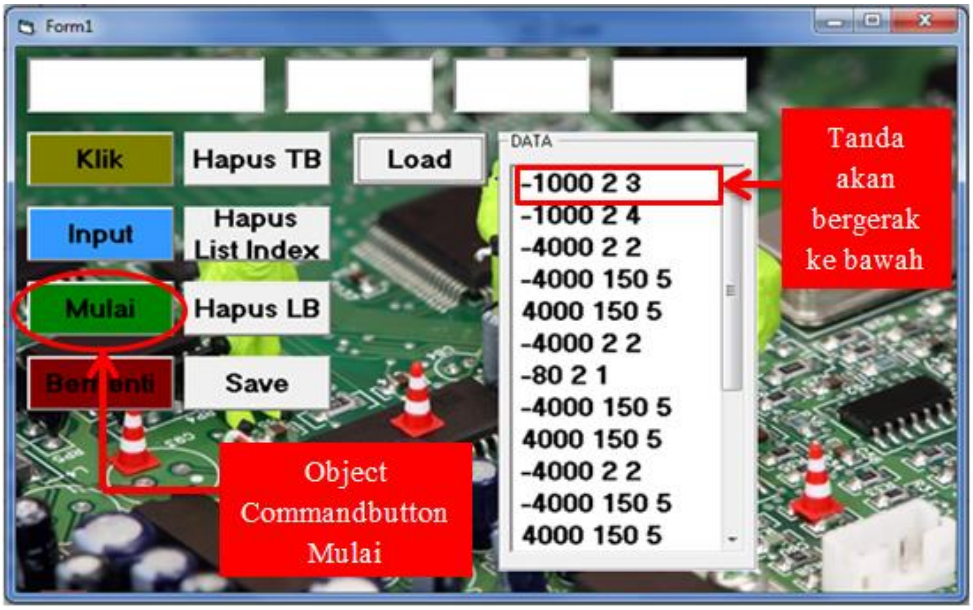
Pengujian object commandbutton **Input** dilakukan dengan cara string dimasukkan ke dalam textbox satu. Setelah string dimasukkan ke dalam textbox satu lalu kursor diarahkan ke arah object commandbutton **Input**. Setelah kursor diarahkan ke arah object commandbutton **Input** lalu tombol mouse sebelah kiri di-klik. Setelah tombol mouse sebelah kiri di-klik string yang telah dimasukkan di dalam textbox akan akan berpindah ke dalam listbox. Pengujian commandbutton **Input** dapat dilihat pada gambar 4.3.



Gambar 4.3 Pengujian Object Commandbutton Input

4.1.3 Pengujian Object Commandbutton Mulai

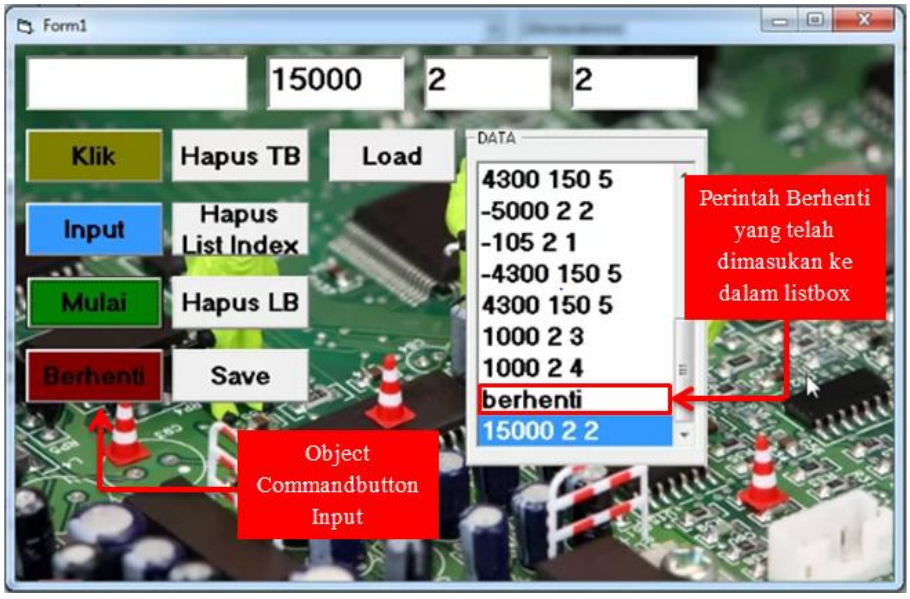
Pengujian object commandbutton **Mulai** dilakukan dengan cara kursor diarahkan ke object commandbutton **Mulai**. Setelah kursor diarahkan ke object commandbutton **Mulai** lalu tombol mouse sebelah kiri di-klik, setelah tombol mouse sebelah kiri di-klik string pertama yang telah dimasukkan ke dalam listbox akan dikirim ke mikrokontroler secara serial. Pengiriman string selanjutnya dilakukan jika mikrokontroler telah memberikan tanda ke komputer. Pengiriman string dilakukan sampai semua string yang ada di dalam listbox telah dikirimkan semua. Pengujian commandbutton mulai dapat dilihat pada gambar 4.4.



Gambar 4.4 Pengujian Object Commandbutton Mulai

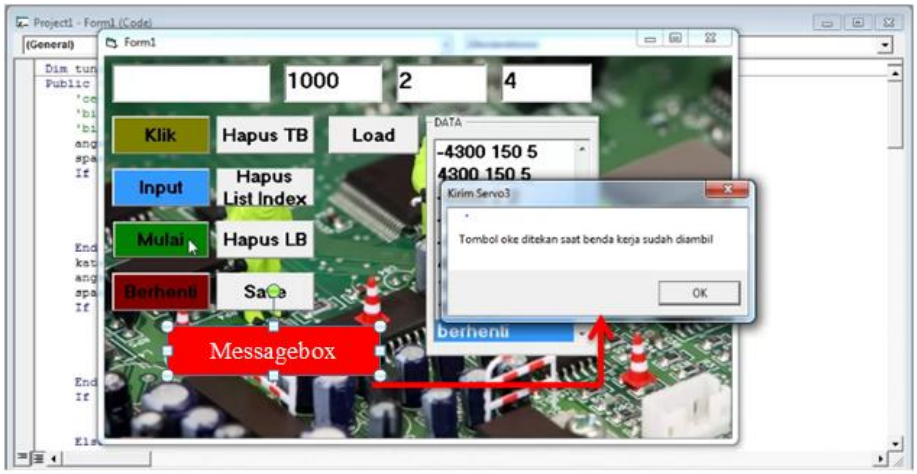
4.1.4 Pengujian Object Commandbutton Berhenti

Pengujian commandbutton **Berhenti** dilakukan dengan cara kursor diarahkan ke arah object commandbutton **Berhenti**. Setelah kursor diarahkan ke arah object commandbutton **Berhenti** lalu tombol mouse sebelah kiri di-klik, setelah tombol mouse sebelah kiri di-klik di object commandbutton **Berhenti** akan muncul kata berhenti di dalam textbox satu. Setelah muncul kata "**Berhenti**" di dalam textbox satu, kursor diarahkan ke arah object commandbutton **Input** lalu tombol mouse sebelah kiri di-klik. Setelah tombol mouse sebelah kiri di-klik di object commandbutton **Input** kata "**Berhenti**" yang ada di dalam textbox akan berpindah ke dalam listbox. Pengujian object commandbutton berhenti dapat dilihat pada gambar 4.5.



Gambar 4.5 Pengujian Object Commandbutton Berhenti

Gambar 4.5 menunjukan bahwa perintah "**Berhenti**" telah dimasukkan ke dalam listbox. Perintah "**Berhenti**" akan muncul ketika string yang ada diatas perintah "Be**rhenti**" telah dikerjakan. Perintah "**Berhenti**" muncul dengan tampilan messagebox yang berisi tulisan "tombol oke ditekan jika benda kerja telah diambil", tampilan messagebox dapat dilihat pada gambar 4.6. Gambar 4.6 menunjukan bahwa pada messagebox terdapat tombol "**Ok**" yang berungsi untuk menjalankan kembali proses pengujian. Setelah muncul messagebox lalu kursor diarahkan ke arah tombol "**Ok**", setelah kursor diarahkan ke tombol "**Ok**" lalu tombol mouse sebelah kiri di-klik. Setelah tombol mouse sebelah kiri di-klik di tombol "**Ok**" messagebox, proses pengujian akan berjalan kembali. Proses pengujian setelah tombol ok ditekan dapat dilihat pada gambar 4.7.



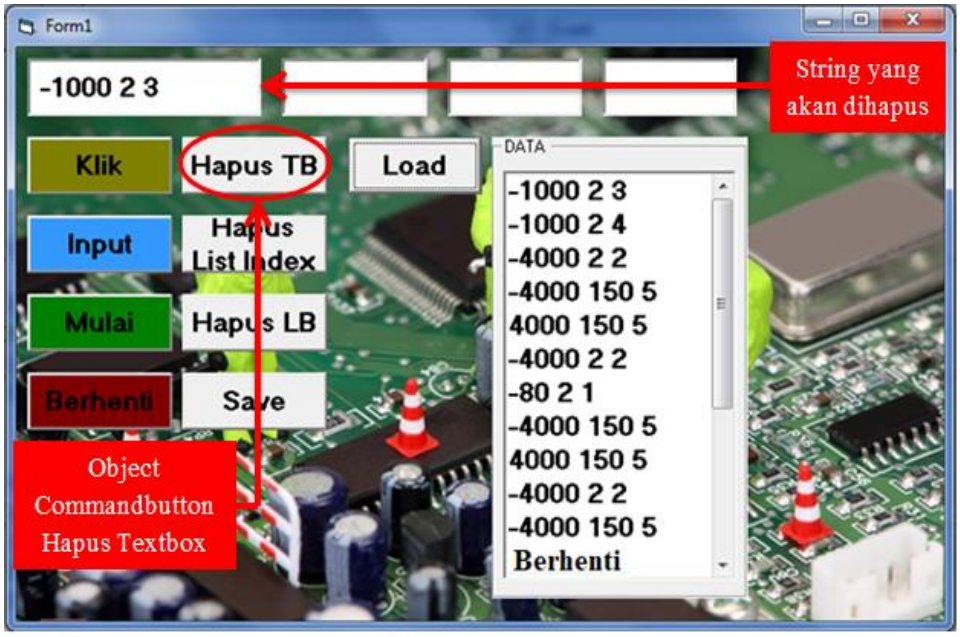
Gambar 4.6 Tampilan Messagebox Perintah “ Berhenti “

Gambar 4.7 Setelah Tombol “Ok“ Pada Messagebox Ditekan

4.1.5 Pengujian Object Commandbutton Hapus Textbox

Pengujian object commandbutton hapus textbox digunakan untuk menghapus string yang ada di dalam textbox. Pengujian object commandbutton hapus textbox dilakukan dengan cara kursor diarahkan ke arah object commandbutton hapus textbox lalu tombol mouse sebelah kiri di-klik kiri. Setelah tombol mouse sebelah kiri di-klik string yang ada di dalam textbox satu akan terhapus. Pengujian object commandbutton hapus textbox dapat dilihat pada gambar 4.8.

Gambar 4.8 menunjukan bahwa string yang ada pada textbox akan dihapus, lalu kursor diarahkan ke arah object commandbutton hapus textbox. Setelah kursor diarahkan ke arah object commandbutton hapus textbox lalu tombol mouse sebelah kiri di-klik. Setelah tombol mouse sebelah kiri di-klik string yang ada di dalam textbox akan terhapus. String yang telah dihapus dapat dilihat pada gambar 4.9.



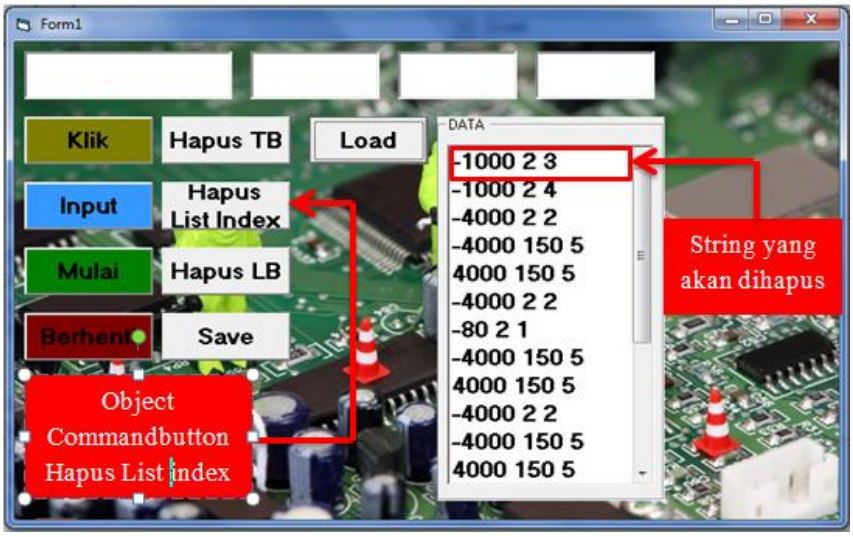
Gambar 4.8 Pengujian Object Commandbutton Hapus Textbox

Gambar 4.9 String yang Telah Dihapus di Dalam Textbox

4.1.6 Pengujian Object Commandbutton Hapus List Index

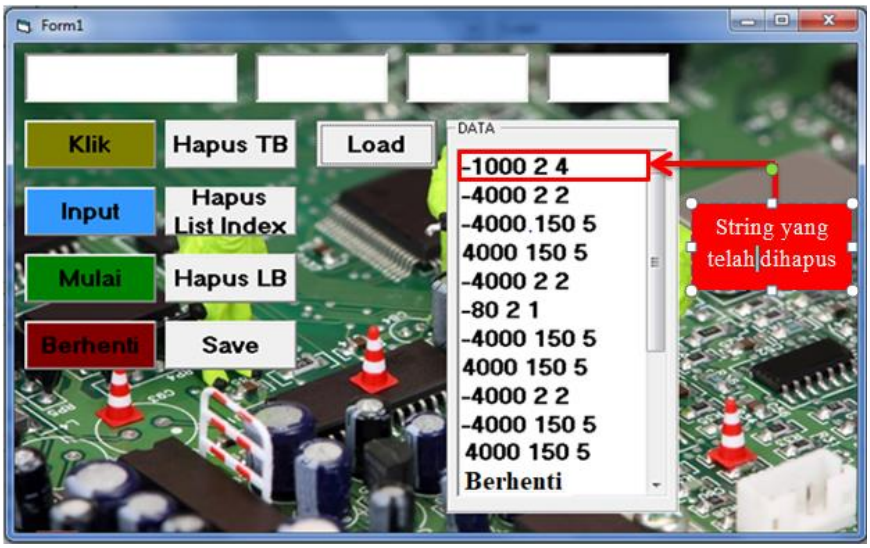
Pengujian object commandbutton hapus list index bertujuan untuk menghapus salah satu string yang telah dimasukan ke dalam listbox ketika terjadi kesalahan. Pengujian object commandbutton dilakukan dengan cara kursor diarahkan ke arah object commandbutton hapus list index. Setelah kursor diarahkan ke arah object commandbutton list index lalu

tombol mouse sebelah kiri di-klik. Pengujian object commandbutton hapus list index dapat dilihat pada gambar 4.10.



Gambar 4.10Pengujian Object Commandbutton Hapus List Index

Gambar 4.10 menunjukan salah satu string yang akan dihapus di dalam listbox. Salah satu string dihapus di dalam listbox ketika terjadi kesalahan dalam memasukan string. String yang telah dihapus dapat dilihat pada gambar 4.11.

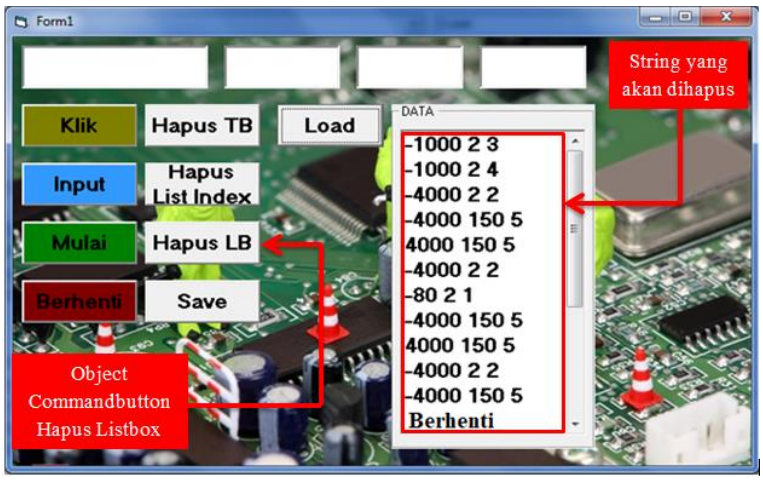


Gambar 4.11 String yang Telah Dihapus di Dalam Listbox

4.1.7 Object Commandbutton Hapus Listbox

Object commandbutton **Hapus listbox** adalah tombol yang digunakan untuk menghapus string yang ada di dalam listbox. Pengujian perintah **Hapus Listbox** dilakukan

dengan cara kursor diarahkan ke arah object commandbutton **Hapus Listbox**. Setelah kursor diarahkan ke object commandbutton **Hapus Listbox** lalu tombol mouse sebelah kiri di-klik. Setelah tombol mouse sebelah kiri di-klik di object commandbutton **Hapus Listbox** string yang ada di dalam listbox akan terhapus. Perintah hapus listbox dapat dilihat pada gambar 4.12.



Gambar 4.12 Pengujian Object Commandbutton Hapus Listbox

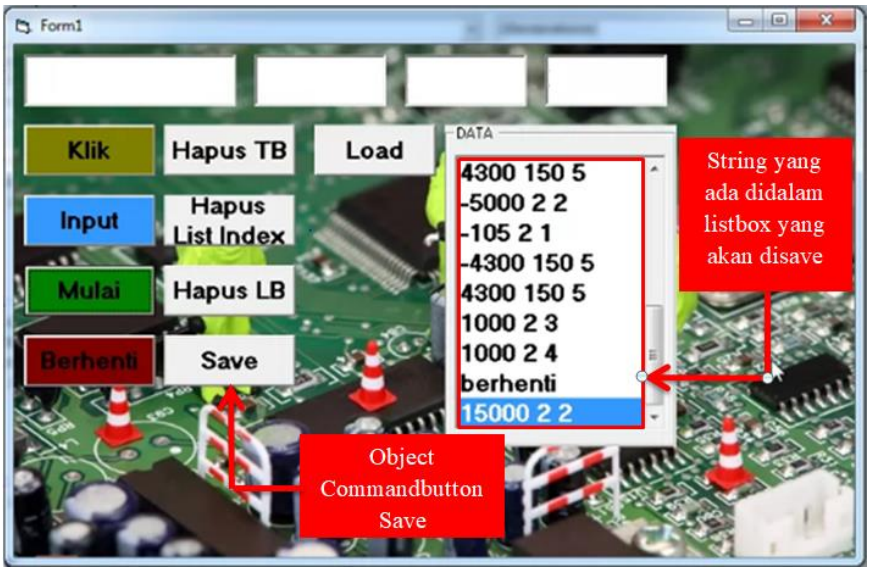
Gambar 4.12 menunjukkan bahwa string yang ada di dalam listbox akan dihapus. String yang telah dihapus di dalam listbox dapat dilihat pada gambar 4.13.

Gambar 4.13 String yang Ada di Dalam Listbox di Hapus

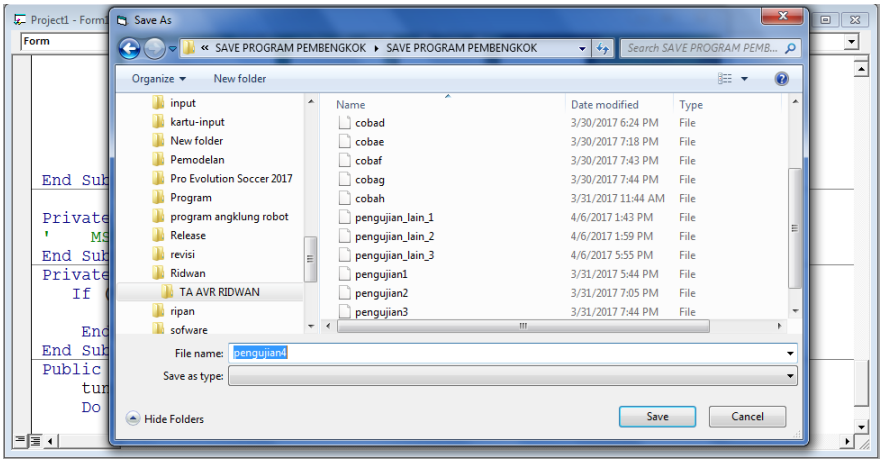
4.1.8 Pengujian Object Commandbutton Save

Object commandbutton save digunakan untuk menyimpan string yang ada dalam listbox setelah dilakukan pengujian. Pengujian object commandbutton **Save** dilakukan dengan

cara kursor diarahkan ke arah object commandbutton **Save**. Setelah kursor diarahkan ke arah object commandbutton **Save** tombol mouse sebelah kiri di-klik. Setelah tombol mouse sebelah kiri di-klik muncul tampilan dialog box save as. Setelah muncul tampilan dialog box save as, folder tempat menyimpan file dipilih. Dalam hal ini folder yang dipilih adalah folder TA AVR RIDWAN. Langkah berikutnya adalah textbox file name diisi. Dalam hal ini nama file yang dipilih adalah pengujian4. Tombol save di-klik, string yang ada di dalam listbox tersimpan. Pengujian object commandbutton **Save** dapat dilihat pada gambar 4.14 dan tampilan proses penyimpanan string di folder TA AVR RIDWAN dapat dilihat pada gambar 4.15.



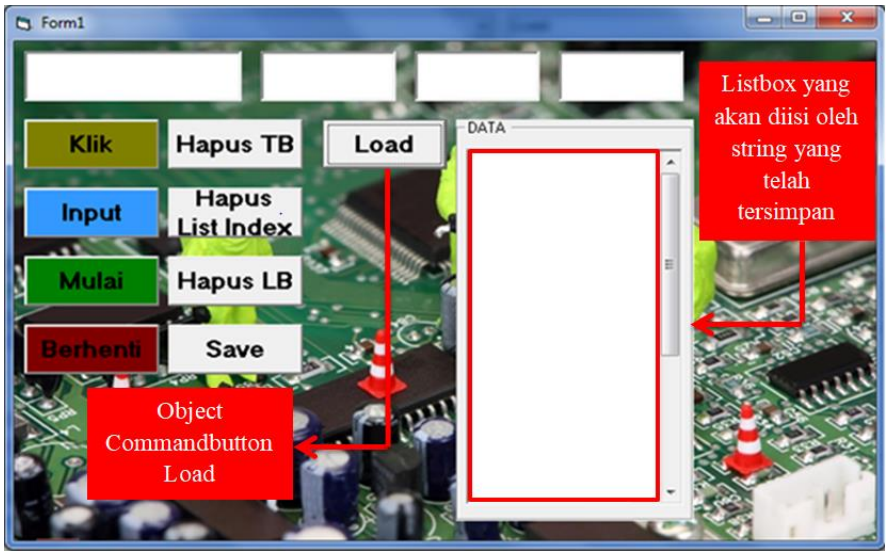
Gambar 4.14 Pengujian Object Commandbutton Save



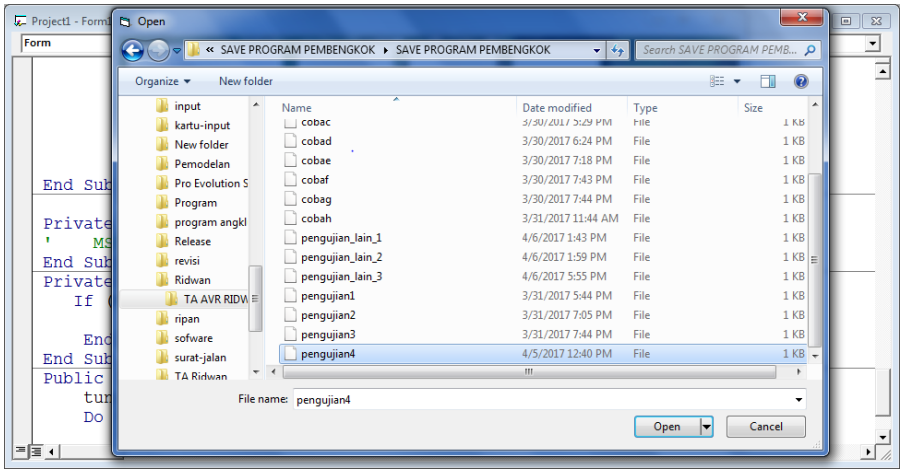
Gambar 4.15 String Telah Tersimpan di File Explorer

4.1.9 Pengujian Object Commandbutton Load

Object commandbutton **Load** digunakan untuk membuka file yang telah disimpan. Pengujian object commandbutton **Load** dilakukan dengan cara kursor diarahkan ke arah object commandbutton **Load.** Setelah kursor diarahkan ke arah object commandbutton **Load** lalu tombol mouse sebelah kiri di-klik. Setelah tombol mouse sebelah kiri di-klik muncul dialog box open, file yang akan dibuka dipilih. Dalam hal ini file yang akan dibuka ada di dalam folder TA AVR RIDWAN. Dalam hal ini file yang akan dibuka adalah file dengan nama pengujian4. Setelah file yang akan dibuka dipilih, tombol open di-klik. Isi file yang telah dipilih akan muncul di dalam object listbox. Pengujian object commandbutton **Load** dapat dilihat pada gambar 4.16 dan tampilan proses pembukaan file dapat dilihat pada gambar 4.17. Object listbox yang telah diisi string dapat dilihat pada gambar 4.18.



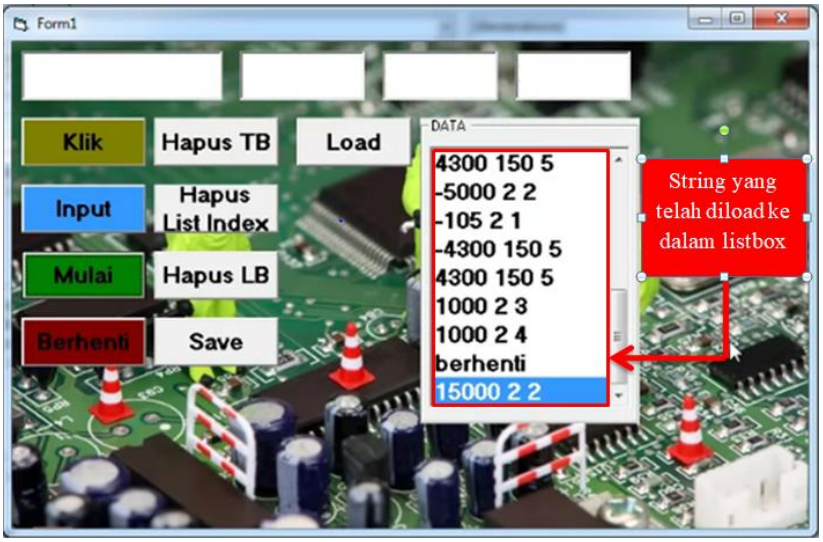
Gambar 4.16 Pengujian Object Commandbutton Load



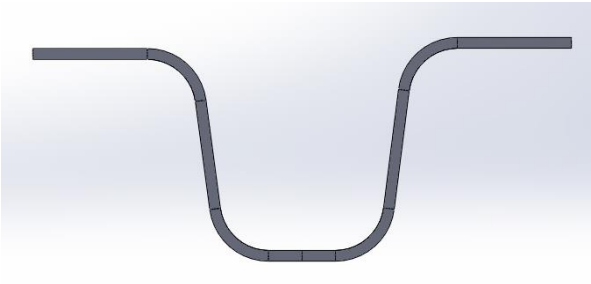
Gambar 4.17 Tampilan Proses Folder yang Akan Dibuka

4.2 Pengujian Pembengkokan

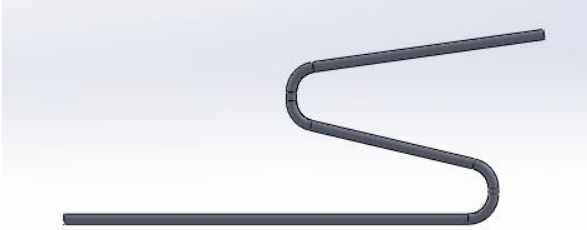
Pengujian pembengkokan bertujuan untuk mengetahui apakah dies pada mekanisme pembengkok berfungsi dengan baik atau tidak. Pengujian pembengkokan dilakukan dengan cara membuat benda kerja yang bentuknya bermacam-macam. Bentuk benda kerja yang dibuat di *prototype* mesin pembengkok berupa bentuk stang motor, bentuk huruf "S", bentuk anak tangga, dan bentuk mix. Bentuk-bentuk benda kerja yang akan dibuat dapat dilihat pada gambar 4.19, gambar 4.20, gambar 4.21, dan gambar 4.22.



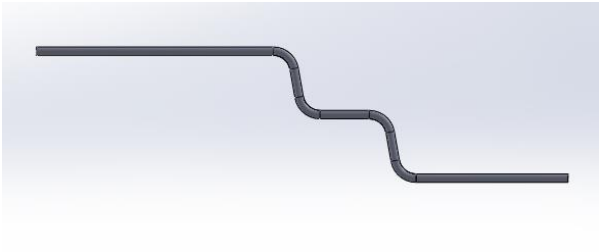
Gambar 4.18 String yang Telah Di Load (Dimasukkan) ke Dalam Listbox



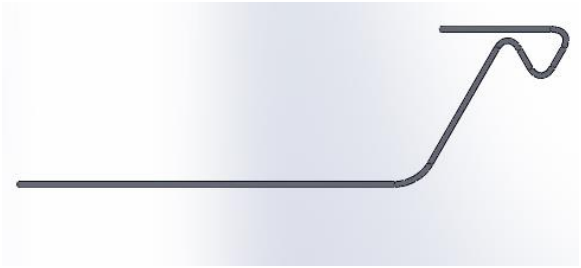
Gambar 4.19 Pengujian Dengan Bentuk Stang Motor



Gambar 4.20 Pengujian Dengan Bentuk Huruf " S "



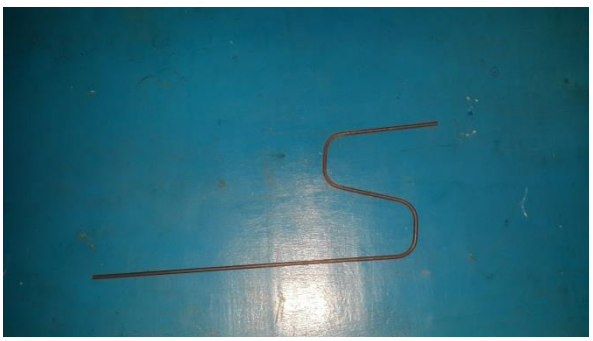
Gambar 4.21 Pengujian Dengan Bentuk Anak Tangga



Gambar 4.22 Pengujian Dengan Bentuk Mix

Gambar 4.19, gambar 4.20, gambar 4.21, dan gambar 4.2 menunjukan benda kerja yang akan dibuat di *prototype* mesin pembengkok batang silinder. Benda kerja tersebut dibuat pada *prototype* mesin pembengkok dengan menggunakan program yang telah dibuat. Program untuk membuat benda kerja tersebut dapat dilihat di lampiran. Benda kerja hasil proses pembengkokan dapat dilihat pada gambar 4.23, gambar 4.24, gambar 4.25, dan gambar 4.26.

Gambar 4.23 Bentuk Stang Motor



Gambar 4.24 Bentuk Huruf "S"

4.3 Analisa Hasil Pengujian

Data hasil pengujian *prototype* mesin pembengkok batang silinder perlu dianalisa. Analisa data hasil pengujian tersebut adalah sebagai berikut:

1. Semua object program pengendali *prototype* mesin pembengkok batang silinder berfungsi dengan baik, dan
2. *Prototype* mesin pembengkok batang silinder dapat membuat bentuk sesuai program yang telah dibuat.

Gambar 4.25 Bentuk Anak Tangga

Gambar 4.26 Bentuk Mix

BAB V

KESIMPULAN DAN SARAN

Pada bab ini dibahas tentang kesimpulan dan saran mengenai hasil pengujian *prototype* mesin pembengkok batang silinder yang telah diprogram untuk melakukan proses pembengkokan.

5.1 Kesimpulan Hasil Pengujian

Setelah dilakukan pengujian dan analisa *prototype* mesin pembengkok batang silinder, dapat disimpulkan bahwa program visual basic yang ditambahkan pada *prototype* mesin pembengkok berhasil untuk menggerakkan *prototype* mesin pembengkok batang silinder. *Prototype* mesin pembengkok dapat membuat berbagai macam bentuk dalam satu kali pemrograman.

5.2 Saran

Setelah dilakukan pengujian dan analisa *prototype* mesin pembengkok batang silinder dapat disampaikan beberapa saran. Beberapa saran tersebut adalah sebagai berikut:

1. Mekanisme penjepit material perlu dilakukan modifikasi. Modifikasi bertujuan agar saat mekanisme pemutar material berputar ke arah kiri benda kerja tidak menabrak mekanisme penjepit.
2. Mekanisme pengumpan perlu gerakan yang cepat untuk mempersingkat waktu pengujian. Untuk mempersingkat waktu pengujian motor *stepper* pada mekanisme pengumpan perlu diganti dengan menggunakan motor *servo* AC. Agar pada saat pengujian kecepatan motor penggerak pada mekanisme pengumpan dapat diatur.

DAFTAR PUSTAKA

- Dwipayana,Hendry. "*Pengendalian Mekanisme Pembengkok Pada Simulator Mesin Pembengkok Batang Silinder"*, Tugas Akhir Sarjana, Teknik Mesin FT UNPAS, 2014.
- Hariz, R. Ibrahim Firdous, *"Catatan & Dokumentasi Pribadi"*, Universitas Pasundan, Bandung, 2014.
- Makalah Bending, Teknik Mesin S-1, diperoleh dari situs internet : http://arissulistyo.blogspot.com/2014/04/makalah-bending-teknik-mesin-s-1.html. Diunduh pada 03 Februari 2017.
- Makalah Driver Motor Stepper, diperoleh dari situs internet : http://electronics-diy-motor-stepper.php. Diunduh pada tanggal 03 Februari 2017.
- http://www.geckodrive.com/support/motor-control-manuals/dc-servo-drives/ g320x-rev-10.html 28 Februari 2017
- http://hades.mech.northwestern.edu/index.php/Rotary_Encoder 5 April 2017
- Makalah Driver Motor Stepper, diperoleh dari situs internet : http://electronics-diy-motor-stepper.php. 15 April 2017