

BAB II

LANDASAN TEORI

2.1. *Programmable Logic Controller (PLC)*

PLC merupakan suatu piranti elektronik yang dirancang untuk beroperasi secara *digital* dengan menggunakan memori sebagai media penyimpanan instruksi-instruksi *internal* untuk menjalankan fungsi-fungsi logika, seperti pencacah, fungsi urutan proses, fungsi pewaktu, fungsi aritmatika, dan fungsi lainnya dengan cara memprogramnya. Program-program dibuat kemudian dimasukkan dalam PLC melalui *programmer*/monitor. Pembuatan program dapat dilakukan melalui komputer sehingga dapat mempercepat hasil pekerjaan. PLC dapat digunakan untuk memonitor jalannya proses pengendalian yang sedang berlangsung, sehingga dapat dengan mudah dikenali urutan kerja (*work sequence*) proses pengendalian yang terjadi pada saat itu (Budyanto. M, 2003:1).

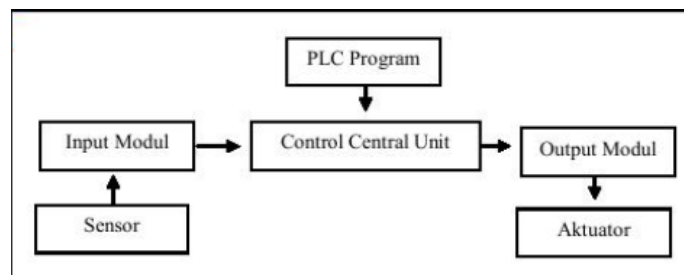
PLC pertama kali digunakan sekitar pada tahun 1960-an untuk menggantikan peralatan konvensional yang begitu banyak. Perkembangan PLC saat ini terus mengalami perkembangan sehingga bentuk dan ukurannya semakin kecil. Saat ini terdapat PLC yang dapat dimasukkan dalam saku karena bentuk dan ukurannya yang sangat kecil, dan dalam perkembangannya, di masa yang akan datang akan diperkenalkan PLC dalam bentuk dan ukuran sebesar kotak rokok.

Pada tahun 1980-an harga PLC masih terhitung mahal, namun saat ini dapat dengan mudah ditemukan dengan harga relatif murah. Beberapa perusahaan

komputer dan elektronik menjadikan PLC menjadi produk terbesar yang terjual saat itu. Pertumbuhan pemasaran PLC mencapai jumlah 80 juta dolar di tahun 1978 dan 1 milyar dolar pertahun hingga tahun 2000 dan angka ini terus berkembang, mengingat penggunaan yang semakin luas, terutama untuk proses pengontrolan di industri, pada alat-alat kedokteran, alat-alat rumah tangga.

Pabrik pembuat PLC mendesain sedemikian rupa sehingga pengguna dapat dengan mudah menguasai fungsi-fungsi dan logika-logika hanya dalam beberapa jam saja. Fungsi-fungsi dasar yang banyak digunakan antara lain : kontak-kontak logika, pewaktu (*timer*), pencacah (*counter*), dan sebagainya. Bagi yang mempunyai latar belakang logika-logika *digital* akan dengan mudah menguasainya dalam beberapa jam saja, berlainan halnya dengan orang yang tidak memiliki latar belakang ini akan memakan waktu agak lama untuk menguasai fungsi dan logika-logika kendali PLC.

PLC atau biasa disebut *Programmable Controller* (PC) adalah suatu perangkat yang dapat digunakan dengan mudah diprogram untuk mengontrol peralatan. PLC sederhana mempunyai komponen utama berupa *Central Control Unit* (CCU), *Unit I/O*, *Programing Console*, *Rack* atau *Mounting Assembly* dan catu daya, sistem komponen dari PLC adalah seperti gambar di bawah ini.



Gambar 2.1 Sistem Komponen PLC

Central Control Unit (CCU) merupakan *unit* pusat pengolahan data yang digunakan untuk melakukan proses pengolahan data dalam PLC. CCU merupakan sebuah *microprocessor*, jenis *processor* yang dipergunakan tergantung pada produsen pembuat PLC, untuk PLC FESTO DIDACTIC FPC 100 menggunakan *microcontroller* 8031.

Fungsi dari sebuah modul *input* adalah untuk mengubah sinyal *input* dari sensor ke PLC untuk diproses dibagian *Central Control Unit*, sedangkan modul *output* adalah kebalikannya, mengubah sinyal PLC ke dalam sinyal yang sesuai untuk menggerakkan aktuator.

Fungsi terpenting dari sebuah modul *input* adalah sebagai berikut :

- a. Mendeteksi sinyal masukan.
- b. Mengatur tegangan kontrol untuk batas tegangan logika masukan yang diijinkan.
- c. Melindungi peralatan elektronik yang *sensitive* terhadap tegangan luar.
- d. Menampilkan sinyal masukan tersebut.

Fungsi terpenting dari sebuah modul *output* adalah sebagai berikut :

- a. Mengatur tegangan kontrol untuk batas tegangan logika keluaran yang diijinkan.
- b. Melindungi peralatan elektronik yang *sensitive* terhadap tegangan luar.
- c. Memberikan penguatan sinyal *output* sebelum dikeluarkan sehingga cukup kuat untuk menggerakkan aktuator.

- d. Memberikan perlindungan terhadap arus hubungan singkat (*short-circuit*) dan pembebanan lebih (*overload*).

Beberapa kelebihan yang dimiliki oleh PLC disbanding dengan kontrol relay konvensional, adalah :

1. *Fleksibel*

Sebelum ditemukannya PLC, setiap mesin mempunyai alat kontrol atau pengendali tersendiri, dimisalkan terdapat 15 buah mesin, maka alat pengendali yang diperlukan juga terdapat 15 buah. Lain halnya sekarang ini dengan adanya PLC maka untuk beberapa mesin hanya memerlukan 1 buah PLC saja.

2. Deteksi dan koreksi kesalahan lebih mudah

Setelah desain program kontrol telah selesai dibuat, kemudian dimasukkan dalam PLC dengan cara memprogramnya, maka program tersebut dapat dengan mudah diubah dengan menggunakan keyboard hanya dalam beberapa menit saja. Setelah itu program kembali dapat dijalankan, jika masih terdapat kesalahan maka dapat dikoreksi dengan menggunakan diagram tangga (*ladder diagram*) sehingga koreksinya dapat dengan segera dilaksanakan.

3. Harga relative murah

Perkembangan teknologi memungkinkan untuk meningkatkan beberapa fungsi dengan bentuk ukuran yang semakin kecil. Tentunya hal

ini juga akan menurunkan harga pembuatan yang mahal. Salah satu fungsi yang terus ditingkatkan adalah modul I/O (masukan/keluaran). Saat ini kita mendapatkan PLC dengan jumlah masukan dan keluaran yang banyak dengan harga relatif murah.

4. Pengamatan visual (*visual observation*)

Operasi PLC saat menjalankan program yang telah dibuat dapat dilihat dengan teliti dengan menggunakan *layer CRT (Cathode Ray Tube)*, sehingga ini sangat memudahkan dalam proses pencarian, pengamatan, atau dalam pembenahan program. Dengan demikian proses pembenahan hanya membutuhkan waktu relatif singkat.

5. Kecepatan operasi (*speed of operation*)

Kecepatan operasi PLC sangatlah cepat. Kecepatan operasi ini adalah untuk mengaktifkan fungsi-fungsi logika hanya dalam waktu beberapa milidetik, dikarenakan menggunakan rangkaian elektronik sehingga operasinya sangatlah cepat, berlainan saat digunakan *relay magnetic*, yang mempunyai kecepatan operasinya lebih lambat.

6. Sederhana dan mudah

Lebih sederhana dan mudah dalam penggunaannya, memodifikasi lebih mudah tanpa tambahan biaya.

Beberapa kekurangan yang dimiliki oleh PLC disbanding dengan kontrol relay konvensional adalah sebagai berikut :

1. Dibutuhkan waktu untuk mengubah system konvensional yang telah ada.
2. Keadaan lingkungan. Untuk proses seperti pada lingkungan panas yang tinggi, *vibrasi* yang tinggi penggunaannya kurang cocok, karena dapat merusak PLC.

2.1.1 Konsep PLC

Konsep dari PLC sesuai dengan namanya, adalah sebagai berikut :

a. *Programmable*

Menunjukkan kemampuannya yang dapat dengan mudah diubah-ubah programnya sesuai program yang dibuat.

b. *Logic*

Menunjukkan kemampuan dalam memproses *input* secara aritmatik yaitu membandingkan, menjumlah, membagi dan sebagainya.

c. *Controller*

Kemampuan dalam mengontrol dan mengatur proses sehingga menghasilkan *output* yang diinginkan.

2.1.2 Fungsi PLC

Fungsi dari PLC dapat dibagi secara umum dan secara khusus. Secara umum fungsi PLC adalah sebagai berikut :

a. *Control Sequence*

PLC memproses *input* sinyal *biner* menjadi sinyal *output* yang digunakan untuk keperluan pemrosesan teknik dan yang secara berurutan (*sequence*). PLC menjaga agar semua *STEP* dalam proses *sequence* berlangsung dalam urutan yang tepat.

b. *Monitoring Plant*

PLC secara terus-menerus memonitor status suatu sistem (misalnya temperatur, tekanan, tingkat ketinggian) dan mengambil tindakan yang diperlukan sehubungan dengan proses yang dikontrol (misalnya nilai telah melebihi batas) atau menampilkan pesan tersebut pada operator.

Sedangkan fungsi PLC secara khusus adalah memberikan *input* ke *Computerized Numerical Control* (CNC). Beberapa PLC dapat memberikan *input* ke CNC untuk kepentingan pemrosesan lebih lanjut. CNC bila dibandingkan dengan PLC mempunyai ketelitian yang lebih tinggi dan lebih mahal harganya. CNC biasanya dipakai untuk proses *finishing*, membentuk benda kerja, digunakan pada unit press, moulding.

2.1.3 Kontrol Konvensional

Kontrol konvensional yang menggunakan relay atau kontraktor mempunyai keuntungan dan kerugian bila digunakan sebagai rangkaian kontrol bila dibandingkan kontrol dengan menggunakan PLC.

Relay sendiri merupakan kontrol elektronik, karena redapat koil atau kumparan yang akan menggerakkan kontak membuka atau menutup bila

kumparannya diberi arus listrik. Berikut ini adalah keuntungan dan kerugian menggunakan relay :

Keuntungan :

- a. Mudah diadaptasikan untuk tegangan yang berbeda.
- b. Tidak banyak dipengaruhi oleh temperature sekitarnya. Relay terus beroperasi pada temperaturus 353 K (80 derajat *celcius*) sampai 240 K (-33 derajat *celcius*).
- c. Tahanan yang relatif tinggi antara kontak kerja pada saat terbuka.
- d. Beberapa sirkuit terpisah dapat dihidupkan.
- e. Sirkuit yang mengontrol relay dan sirkuit yang membawa arus yang terhubung.
- f. Fisik terpisah satu sama lainnya.

Kerugian :

- a. Kontak dibatasi pada keausan dari bunga api atau dari oksidasi (material kontak yang terbaik adalah platina, emas, perak). Menghabiskan banyak tempat dibandingkan dengan transistor.
- b. Menimbulkan bunyi selama proses kontak.
- c. Kecepatan kontak yang terbatas 3 ms sampai 17 ms.
- d. Kontaminasi (debu) dapat mempengaruhi umur kontak.

2.1.4 PLC FESTO

Salah satu PLC yang dimiliki STIKOM dan digunakan untuk praktikum adalah PLC FESTO dari Jerman, seri FPC 101 B-LED dan FPC 101 AF. PLC ini mempunyai kelebihan dapat mengenal program dengan bahasa pemrograman tingkat tinggi (*high level language*), yaitu *Statement List* atau STL, selain menggunakan *Ladder Diagram* yang sudah umum dan menggunakan pemrograman *matriks* MAT. Bahkan untuk seri tertentu dapat diprogram dengan menggunakan bahasa *BASIC* atau *function chart* FUC (Indrijono Dwi, 1991:1).

PLC FPC 101B-LED memiliki spesifikasi sebagai berikut :

- a. Indicator untuk status dan *error*.
- b. Pemrograman yang mudah melalui PC dengan *Ladder Diagram* dan *Statement List*.
- c. Perlindungan *output* dari *short-circuit*.
- d. Pengaman polaritas *power* suplay.

Data teknik PLC FPC 101B-LED :

- a. 21 *input*.
- b. 14 *output*.
- c. 32 *timer*.
- d. 16 *counter*.

- e. *64 register.*
- f. *256 flag.*
- g. *12 Kbytes user memory.*
- h. *7,5 W untuk tiap output.*

Sensor-sensor yang digunakan di laboratorium PLC adalah :

- a. *Push button switch.*
- b. *Switch toggle.*
- c. *Sensor capacitive.*
- d. *Sensor inductive.*
- e. *Sensor optic.*
- f. *Limit switch.*

Aktuator yang digunakan pada laboratorium PLC adalah :

- a. *Single solenoid.*
- b. *Double solenoid.*
- c. *Indicator lamp.*
- d. *Buzzer.*

2.1.5 Bahasa Pemrograman

Terdapat banyak pilihan bahasa untuk membuat program dalam PLC. Masing-masing bahasa mempunyai keuntungan dan kerugian sendiri-sendiri tergantung dari sudut pandang kita sebagai *user*. *Ladder Diagram* adalah bahasa yang dimiliki oleh setiap PLC.

A. *Ladder Diagram* (LDR)

Ladder diagram menggambarkan program dalam bentuk grafik. *Diagram* ini dikembangkan dari kontak-kontak relay yang terstruktur dan menggambarkan aliran arus listrik. Dalam *ladder diagram* ini terdapat dua buah garis vertical. Garis vertical sebelah kiri dihubungkan dengan sumber tegangan negative catu daya pasif.

Di antara dua garis ini dipasang kontak-kontak yang menggambarkan kontrol dari *switch*, sensor atau *output*. Satu baris dari diagram disebut dengan satu *rung*. *Input* menggunakan symbol “[]” (kontak, *normally open*) dan “[/]” (negasi kontak, *normally closed*). *Output* mempunyai symbol “()” yang terletak paling kanan menempel garis vertical kanan. Selama pemrograman setiap symbol yang diberikan adalah PLC sesungguhnya atau merupakan alamat simbolik (misalnya S1, S2, S3, H1).

B. *Statement List* (STL)

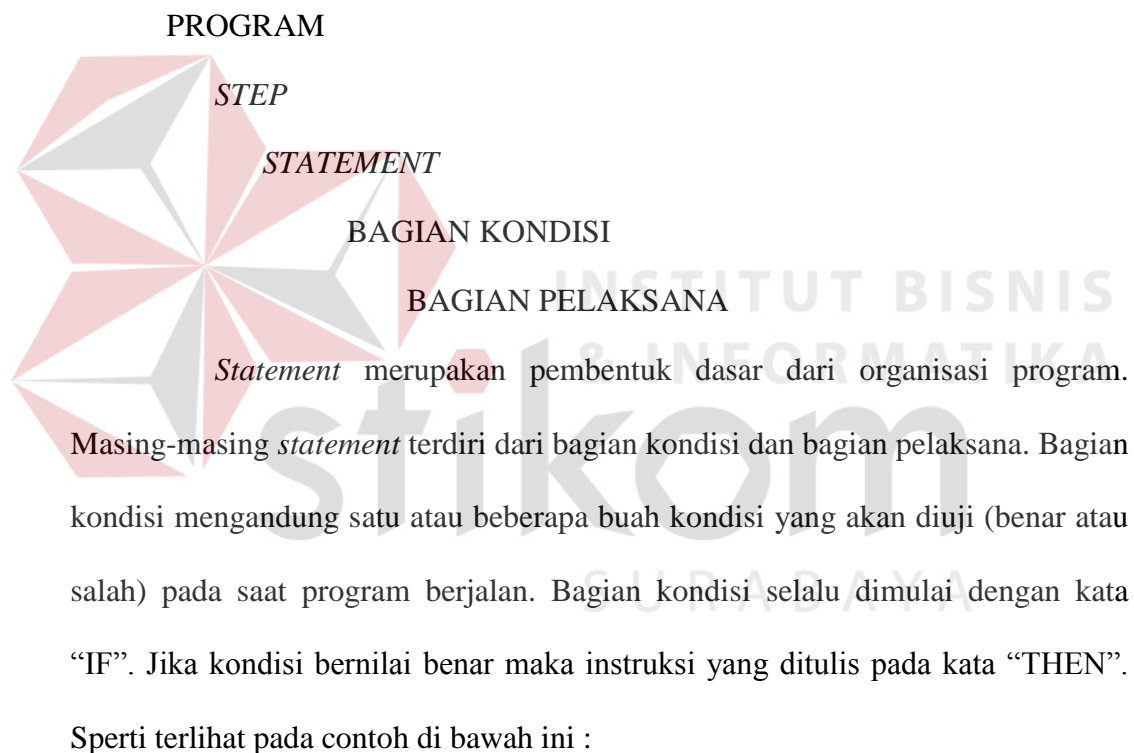
Statement list adalah bahasa pemrograman tingkat tinggi. Semua hubungan logika dan kontrol *sequence* dapat diprogram dengan menggunakan perintah dalam bahasa ini.

Perintah-perintah yang digunakan adalah mirip dengan bahasa tingkat tinggi seperti *pascal*. Terdapat kontrol untuk perulangan, *jump* dan sebagainya.

Misalnya :

IF	I1.0	“Jika input 1.0 aktif
THEN	SET T6	“maka aktifkan timer T6

Struktur dari *statement list* secara umum dapat dituliskan sebagai berikut :



IF	I6	“Jika input 6 memberikan sinyal
THEN	SET O1	“maka nyalakan output 1
IF	I6	“Jika input 6 memberikan sinyal
	AND I2	“dan input 2 memberikan sinyal

THEN RESET O5 “maka matikan ouput 5

SET 04 “nyalakan output 4

Program yang tidak menggunakan instruksi “STEP” dapat diproses dengan cara parallel. Tetapi STL menyediakan instruksi “STEP” yang membagi program menjadi bagian-bagian yang lebih kecil.

Dalam sebuah program dapat berisi sampai 256 “STEP”. Setiap “STEP” dapat diberi label atau tidak, dan hanya dibutuhkan jika setiap “STEP” tersebut merupakan target dari instruksi “JUMP”.

Bentuk paling sederhana dari instruksi “STEP” paling sedikit mengandung satu *statement*. Program akan menunggu pada “STEP” ini sampai kondisinya benar, yaitu bagian pelaksana akan dilaksanakan dulu baru setelah itu program akan berlanjut ke “STEP” berikutnya, dalam sebuah “STEP” dapat berisi beberapa *statement*.

Jika kondisi “IF” terakhir salah maka program tidak akan berlanjut ke “STEP” berikutnya dan akan kembali ke *statement* pertama dalam “STEP” tersebut. Dengan kata lain program akan menunggu sampai kondisi terakhir benar. Aturan pelaksanaan “STEP” adalah sebagai berikut :

- a. Jika kondisi dari sebuah *statement* terpenuhi maka bagian pelaksana akan dijalankan. Dan jika kondisi dari sebuah *statement* dalam suatu “STEP” tidak terpenuhi maka program akan berpindah ke *statement* berikutnya dalam “STEP” tersebut.

- b. Jika kondisi dari *statement* terakhir dalam suatu “STEP” terpenuhi maka bagian pelaksana akan dijalankan dan program berlanjut ke “STEP” berikutnya.
- c. Jika kondisi dari *statement* terakhir dalam sebuah “STEP” tidak terpenuhi maka program akan kembali ke *statement* pertama dari “STEP” yang sekarang.

Dalam STL juga terdapat instruksi “NOP” dapat diletakkan pada bagian kondisi atau bagian pelaksana dari sebuah *statement*. Bila digunakan dalam bagian kondisi, instruksi “NOP” selalu bernilai benar. Dengan kata lain “NOP” menyebabkan pelaksanaan tanpa suatu kondisi.

Jika digunakan dalam bagian pelaksana pengertian “NOP” adalah tidak melakukan sesuatu. Hal ini sering digunakan pada saat program harus menunggu untuk kondisi tertentu lalu pindah ke “STEP” berikutnya.

2.1.6 Timer

Banyak dari kontrol industri yang memerlukan pemrograman dengan waktu. Sebagai contoh, silinder 2 akan maju jika silinder 1 telah maju lebih dahulu tetapi hanya setelah lima detik. Hal seperti ini dikenal dengan *switch-on delay*. Penundaan sinyal *switch-on* pada *switching* rangkaian *power* sangat dibutuhkan demi alasan keamanan.

Timer dalam PLC direalisasikan dalam bentuk modul software yang didasarkan pada pembangkitan *timing* secara *digital* dari generator pulsa *microprocessor*. Lamanya waktu yang diperlukan ditetapkan dalam program

kontrol. Masing-masing *timer* dalam bahasa pemrograman STL terdiri dari beberapa bagian :

- a. *Timer Bit Status*, penulisannya “Tn” yang berfungsi menguji apakah *timer* sedang aktif atau tidak. Nilai *bit* berubah menjadi aktif pada saat *timer* dimulai dengan instruksi “SET”. Pada saat periode waktu yang diprogram selesai atau jika *timer* dihentikan dengan instruksi “RESET” status *bit* berubah menjadi tidak aktif.
- b. *Timer Preselect*, penulisannya “TPn” yang berfungsi sebagai *operand* 16 *bit* yang berisi nilai awal untuk sebuah *timer* ke-n.
- c. *Timer Word*, penulisannya “TWn” yang berfungsi sebagai *operand* 16 *bit* yang secara otomatis memiliki nilai yang sama dengan TP pada saat *timer* dimulai dengan instruksi “SET”. Isinya akan secara otomatis dikurangi oleh system pada *interval* yang teratur.

2.1.7 Counter

Banyak pula dari kontrol industri yang tidak lepas dari *counter*. *Counter* biasanya digunakan untuk menentukan perulangan *statement* dilakukan dalam jumlah tertentu. Misalnya silinder yang degerakkan maju mundur sebanyak 3 kali, untuk merealisasikannya maka digunakan sebuah *counter* yang dapat menghitung gerakan maju mundur dari silinder tersebut.

Counter juga direalisasikan dalam PLC dalam bentuk modul *software* yang didasarkan pada pembangkitan *counting* secara manual, dengan kata lain untuk menambah nilai dai *counter* harus dibangkitkan secara manual pada

program, hal ini kebalikannya *timer* yang dibangkitkan secara otomatis oleh pulsa *microprocessor*. Masing-masing *counter* dalam bahasa pemrograman STL terdiri dari beberapa bagian, yaitu :

- a. *Counter Bit Status*, penulisannya “Cn” yang berfungsi mengui apakah counter sedang aktif atau tidak. Nilai *bit* berubah menjadi aktif pada saat *counter* dimulai dengan instruksi “SET”. Pada saat *counter* sudah mencapai batas yang ditentukan atau *counter* dimatikan dengan instruksi “RESET” maka *bit* berubah menjadi tidak aktif.
- b. *Counter Preselect*, penulisannya “CPn” yang berfungsi sebagai *operand* 16 bit yang berisi nilai akhir sebuah *counter* ke-n.
- c. *Counter Word*, penulisannya “CWn” yang berfungsi sebagai *operand* 16 bit yang secara otomatis memiliki nilai nol pada saat *counter* dimulai. Untuk merubah nilai “CWn” umumnya menggunakan instruksi “INC Cn”.

2.2. Modular Production System (MPS)

Modular Production System (MPS) adalah salah satu peralatan *pneumatic* yang merupakan jenis dari PLC. MPS banyak dikenal dan digunakan pada industri-industri sebagai pengontrol produksi atau sebagai bahan pembelajaran mahasiswa teknik. MPS merupakan suatu peralatan yang dikendalikan oleh PLC. MPS memiliki 4 stasiun yaitu *Distribution Station*, *Testing Station*, *Processing Station*, *Handling Station* dimana setiap stasiun memiliki fungsi dan program untuk menjalankannya.

Peralatan MPS termasuk peralatan *pneumatic* dan *electro pneumatic* yang dikontrol oleh PLC. Dimana peralatan *pneumatic* adalah peralatan yang bekerja dan digerakkan dengan menggunakan udara yang bertekanan. Sedangkan *electro pneumatic* adalah peralatan yang bekerja dan digerakkan oleh udara bertekanan dimana untuk mengeluarkan udaranya digunakan katup yang bekerja menggunakan tegangan listrik, contohnya *solenoid valve*, *electrical input buttons*, *proximity switches* dan *relay*. Karena penggerak utama MPS adalah udara yang bertekanan, sehingga dibutuhkan pula sumber udara yang mampu menyediakan tekanan sesuai dengan yang dibutuhkan. Peralatan sumber udara tersebut berupa sebuah kompresor. Range tekanan yang dibutuhkan peralatan agar dapat bekerja yaitu berada pada range 2-10 bar.

Pada suatu sistem MPS memiliki *input*, *process*, *output*. *Input* yang diterima oleh MPS berasal dari sensor, *limit switch*, *push button* dan *selection switch*. Sedangkan proses yang dilakukan oleh *button* dan *selector switch*. Sedangkan proses yang dilakukan oleh MPS adalah sesuai dengan program yang dimasukkan ke MPS tersebut. *Output* pada MPS berupa solenoid dan motor yang bekerja sesuai dengan program yang dimasukkan.

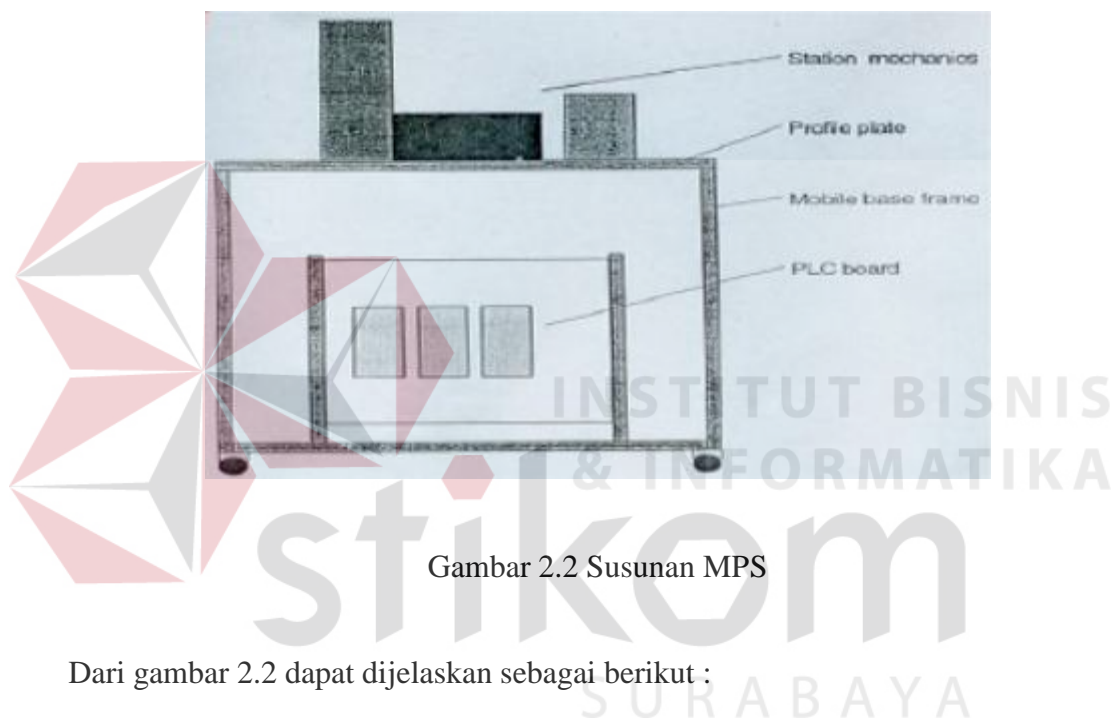
2.2.1 Fungsi MPS

Pada umumnya MPS mempunyai fungsi sebagai seperangkat peralatan yang mengontrol suatu produksi. Setiap bagian stasiun awal yaitu sebagai *feed station*, kemudian benda yang dikirim diseleksi pada *testing station*, yaitu untuk mengetahui jenis, ukuran serta warna barang yang tepat. Benda yang lolos seleksi diolah pada *processing station*. Pada *processing station* dilakukan pembentukan

benda yang nantinya akan diteruskan pada stasiun yang terakhir yaitu *handling station*.

2.2.2 Susunan MPS

MPS terdiri dari beberapa bagian, dimana susunannya dapat dilihat pada gambar 2.2. (Festo Didactic, 1995.)



Gambar 2.2 Susunan MPS

Dari gambar 2.2 dapat dijelaskan sebagai berikut :

a. *Station Mechanics*

Station mechanics merupakan peralatan-peralatan *electro pneumatic* yang dikontrol oleh PLC. *Station Mechanics* terdiri dari peralatan-peralatan *input* dan *output* sensor, *solenoid*, silinder dan motor.

b. *Profile Plate*

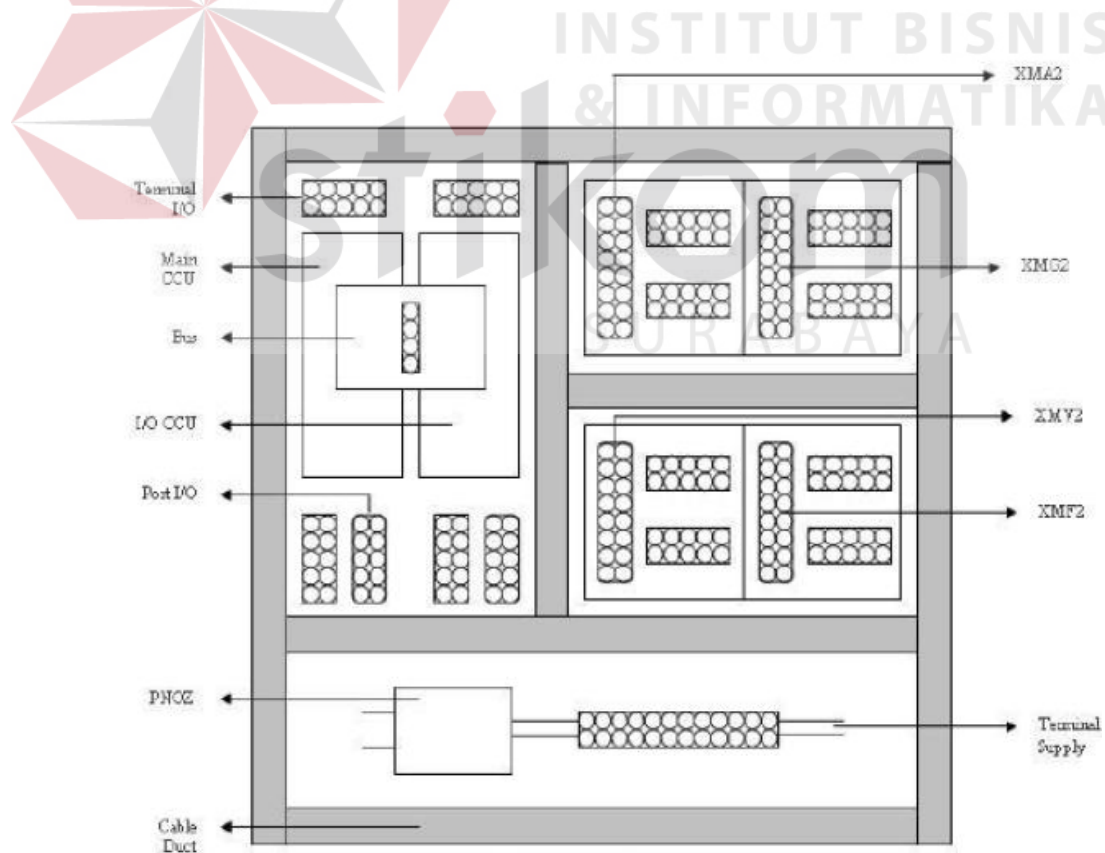
Profile plate adalah suatu papan aluminium dimana tempat peralatan mekanik terpasang. *Profile plate* mempunyai ukuran 700mm x 700mm.

c. *Mobile base frame*

Mobile base frame adalah suatu rak dua lapis yang dilengkapi dengan roda, sehingga sangat mudah untuk dipindah-pindahkan. Lapis yang atas digunakan untuk *profile plate*, sedangkan lapis bawah digunakan untuk *PLC board*.

d. *PLC board*

PLC board merupakan pusat kontrol dari keseluruhan MPS. *PLC board* terdiri dari sebuah *main central control unit (CCU)*, sebuah *I/O CCU*, port XMA2, XMG2, XMF2, XMV2, PNOZ dan beberapa terminal I/O. secara lengkap tampak pada gambar 2.3.



Gambar 2.3 *PLC Board*

Main CCU merupakan pengendali utama dari PLC board. *Main* CCU dibantu I/O CCU yang berfungsi sebagai tambahan I/O. kedua CCU ini dihubungkan oleh sebuah *bus* agar I/O masing-masing CCU dapat saling berkomunikasi.

Port XMA2 digunakan untuk menghubungkan I/O yang ada di PLC board dengan terminal I/O yang terdapat di *profile plat*. *Port* XMG2 digunakan untuk menghubungkan I/O yang ada pada PLC dengan *control console*.

Port XMV2 digunakan untuk berkomunikasi dengan *previous station*, sedangkan *port* XMF2 digunakan untuk berkomunikasi dengan *subsequent station*. PNOZ adalah peralatan yang digunakan untuk pengamanan. PNOZ biasa digunakan untuk *Emergency-Stop*. Saat *Emergency-Stop* tidak diaktifkan maka PNOZ akan melewati, maka PNOZ akan memutuskan jalannya arus tersebut. Secara umum, PNOZ berupa rangkaian elektronik yang dibentuk dari *relay* yang digunakan khusus untuk proses *emergency*.

2.2.3 PLC Pada MPS

PLC merupakan pengendali utama dari MPS yaitu *input* yang diterima akan diproses dan dikeluarkan ke *output*. Proses pengolahan *input* didalam PLC tergantung pada program yang dimasukkan ke dalam PLC tersebut.

PLC yang digunakan pada MPS adalah tipe FPC 101 B dimana spesifikasinya dapat dilihat pada tabel 2.2 dan tipe FPC 101 AF yang spesifikasinya dapat dilihat pada tabel 2.3. kedua jenis PLC tersebut berfungsi sebagai *main* CCU. PLC lain yang digunakan adalah tipe FPC 101EA. PLC ini

berfungsi sebagai tambahan I/O untuk *main* CCU. Untuk pembahasan tipe 101 EA berikutnya akan disebut sebagai I/O CCU. Spesifikasi tipe FPC 101 EA dapat dilihat pada tabel 2.1.

Tabel 2.1 Spesifikasi FPC 101 EA

Operand	Symbol	Number	Range
<i>Input</i>	I	21	10.0 - 12.7
<i>Input Words</i>	IW	3	10 - 12
<i>Outputs</i>	O	14	10.0 - 11.7
<i>Output Words</i>	OW	2	10 - 11

Tabel 2.2 Spesifikasi FPC 101 B

Operand	Symbol	Number	Range
<i>Inputs</i>	I	21	10.0 - 12.7
<i>Input Words</i>	IW	3	10 - 12
<i>Outputs</i>	O	14	10.0 - 11.7
<i>Outputs Words</i>	OW	2	10 - 11
<i>Flags</i>	F	256	0.0 - 15.15
<i>Flag Words</i>	FW	16	0 - 15
<i>Timers</i>	T	32	0 - 31
<i>Timer Words</i>	TW	32	0 - 31
<i>Counters</i>	C	16	0 - 15
<i>Counter Words</i>	CW	16	0 - 15
<i>Registers</i>	R	64	0 - 63

Tabel 2.3 Spesifikasi FPC 101 AF

Operand	Symbol	Number	Range
<i>Digital Input</i>	I	21	10.0 - 12.7
<i>Digital Output</i>	IW	3	10 - 12
<i>Analog input 0 to 20mA ± 10V</i>	II	4	0 - 3
	IU	4	0 - 3
<i>Analog Output ± 10V</i>	OU	2	0 and 1
<i>Flags</i>	F	256	0.0 - 15.15
<i>Flags Words</i>	FW	16	0 - 15
<i>Timers</i>	T	32	0 to 31
<i>Counters</i>	C	16	0 to 15
<i>Registers</i>	R	64	0 to 63

Karena tipe FPC 101 EA berupa I/O maka hanya mempunyai *input word* dan *output word* saja. Dengan demikian jumlah total *input word* dan *output*

word pada PLC adalah 6 *input word* (3 dari *main CCU* dan 3 dari *I/O CCU*) dan 4 *output word* (2 dari *main CCU* dan 2 dari *I/O CCU*).

Antara *main CCU* dan *I/O CCU* dihubungkan oleh sebuah *bus* yang berfungsi untuk menggabungkan *operand* (*input*, *output*, *flag*, *timer*, *counter*, *register*, dll) dari kedua *CCU* tersebut. Sehingga perubahan *operand-operand* yang terjadi pada *I/O CCU* dapat dipantau oleh *main CCU* dan demikian pula sebaliknya. Tiap-tiap *input word* dan *output word* yang dimiliki oleh *main CCU* dan *I/O CCU* mempunyai fungsi berbeda seperti terlihat pada tabel 2.4.

Tabel 2.4 Fungsi Input Word dan Output Word

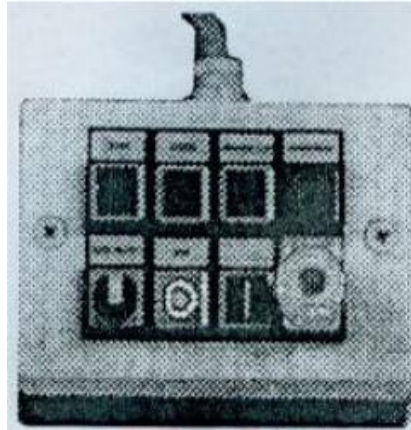
Operand	Fungsi
IW0 dan OW0	Menyimpan status <i>input</i> dan <i>output</i> dari peralatan (<i>solenoid</i> , <i>motor</i> , <i>sensor</i> , <i>limit switch</i>)
IW1 dan OW1	Menyimpan status <i>input</i> dan <i>output</i> dari <i>control console</i> (<i>status lamp</i> , <i>luminous push button</i> , <i>selector switch</i>)
IW10 dan OW10	Untuk komunikasi dengan <i>subsequence station</i> , baik kontrol komunikasi maupun informasi tentang karakteristik benda yang akan dikirim
IW11 dan OW11	Untuk komunikasi dengan <i>previous station</i> kontrol komunikasi maupun informasi tentang karakter benda yang dikirim

2.2.4 Control Console

Control console adalah peralatan yang berfungsi untuk mengatur kerja MPS. Pada dasar *control console* adalah alat *input* yang dijalankan secara *manual* oleh operator. Melalui *control console* seorang operator dapat melakukan proses, menghentikan proses, mengubah cara kerja MPS dari *automatic* menjadi *manual* dan sebaliknya, mengembalikan keadaan peralatan ke keadaan mula-mula (*reset* peralatan) dan memberikan keadaan *emergency-stop* pada MPS jika terjadi kecelakaan atau kesalahan yang fatal. Selain itu, melalui *control console* seorang

operator dapat memantau *input* apa saja yang saat itu diperlukan oleh MPS.

Control console dapat dilihat pada gambar 2.4.



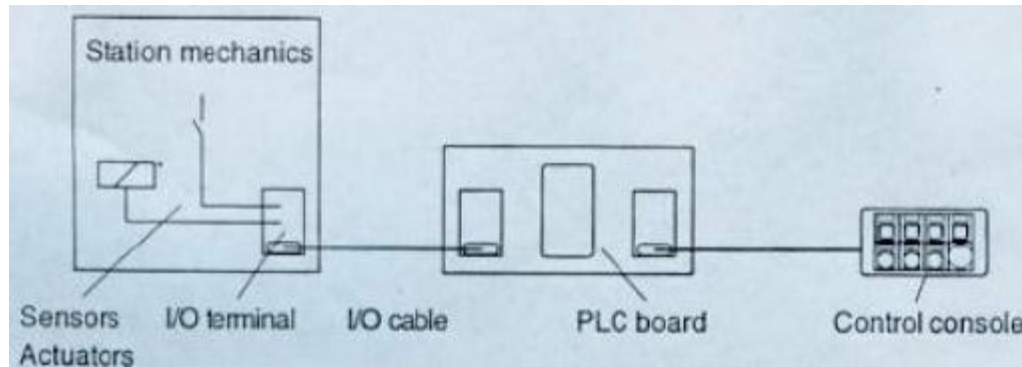
Gambar 2.4 Control Console

Control console terdiri dari beberapa *push button*, *selector switch*, indikator lampu, *emergency-stop* dan *controller ON*. Fungsi dari bagian-bagian *control console* dapat dilihat pada tabel 2.5.

Tabel 2.5 Fungsi dari bagian-bagian *control console*

Jenis	Nama	Fungsi
Lampu	<i>Network</i>	Menunjukkan hubungan dengan stasiun sebelumnya dan stasiun sesudahnya.
<i>Push Button</i>	<i>Reset</i>	Mengembalikan keadaan awal peralatan.
<i>Push Button</i>	<i>Start</i>	Memulai proses MPS.
<i>Push Button</i>	<i>Stop</i>	Menghentikan proses setelah mengerjakan langkah terakhir.
<i>Dual Selector Switch</i>	<i>Manual</i> atau <i>Automatic</i>	Mengubah langkah kerja MPS yaitu per- langkah atau terus menerus.
<i>Push Button</i>	<i>Emergency-stop</i>	Menghentikan semua peralatan karena terjadi kecelakaan atau kesalahan yang fatal.
<i>Push Button</i>	<i>Controller ON</i>	Menyambungkan kembali arus yang terputus akibat <i>emergency</i> .

Gambar 2.5 akan menjelaskan hubungan antara *station mechanics*, PLC board dan *control console*. (Festo Didactic, 1995 : A-6).



Gambar 2.5 Diagram hubungan antara *station mechanics*, *PLC board* dan *control console*

2.2.5 Macam MPS

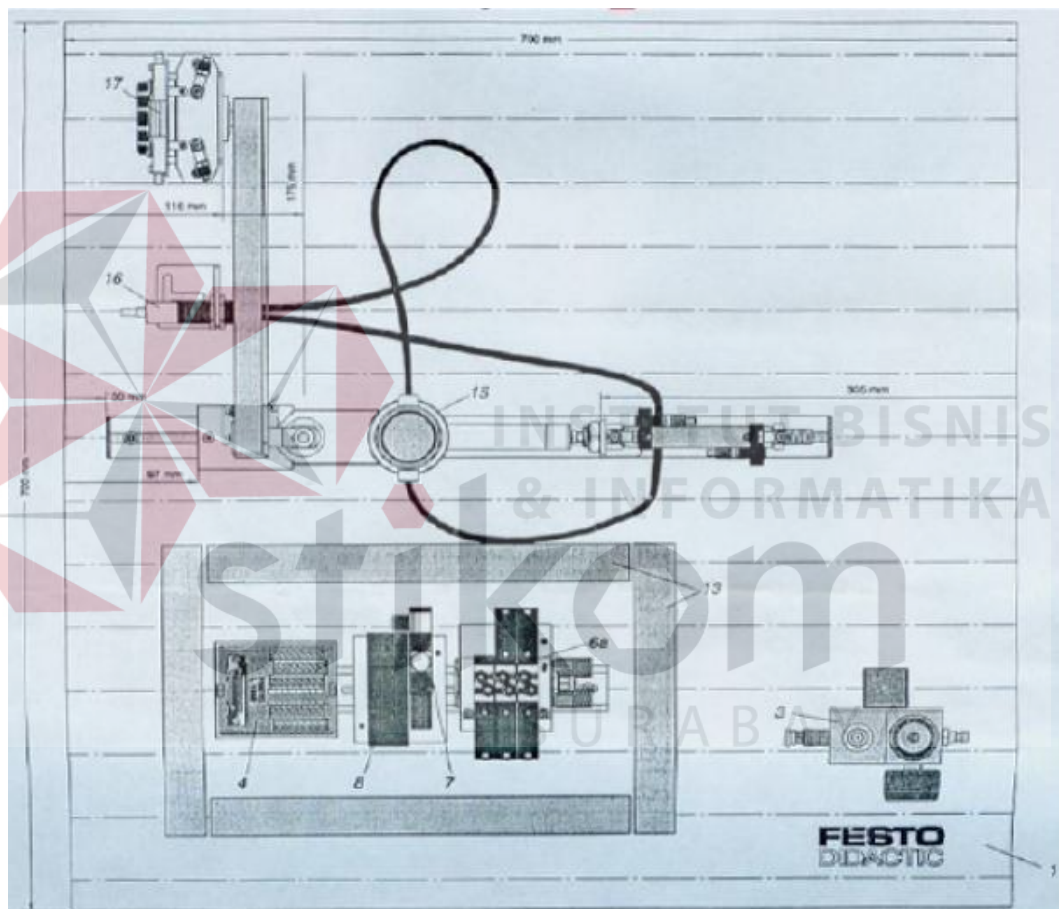
Laboratorium PLC STIKOM memiliki 4 stasiun MPS, yaitu *distributing station*, *testing station*, *processing station* dan *handling station*. Masing-masing stasiun mempunyai alur kerja dan fungsi sendiri-sendiri.

a. *Distribution Station*

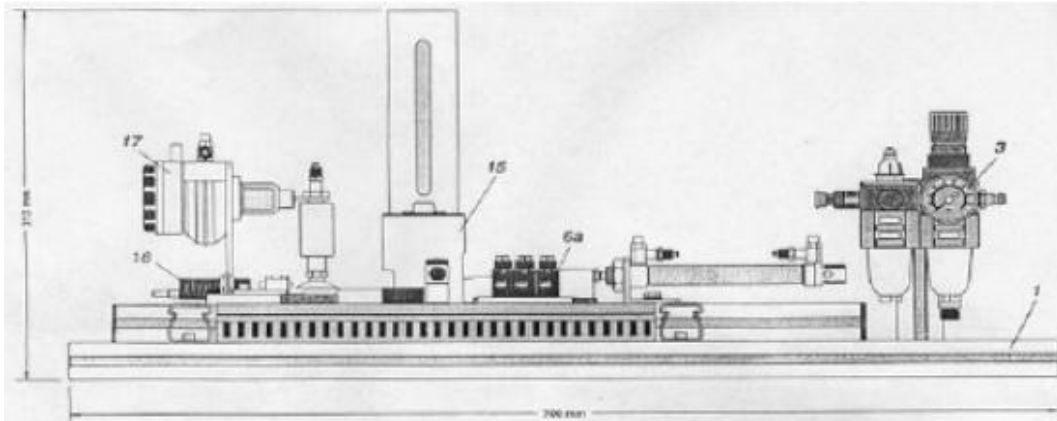
Distribution station adalah stasiun awal dari sebuah MPS yaitu sebagai stasiun pemberi atau *feed station*. Stasiun ini didefinisikan sebagai unit yang bekerja dari fungsi *magazine* dan sebagai stasiun pemesanan dan *feeding unit*. *Feeding unit* digunakan untuk pemesanan benda berdasarkan karakter dan klasifikasinya (dimensi dan berat). *Feeding unit* biasa digunakan yaitu sebagai contoh *magazine* dengan *separating device*, *vibratory bowl feeder*, *slope conveyor* dan *hopper* dengan *separating device*. *Distribution station* bertujuan untuk memindahkan benda dari *magazine* agar dapat digunakan untuk proses berikutnya. *Distribution*

station memiliki *profile plate*, PLC board dengan *feed magazine* modul dan transfer modul.

Distribution station terdiri dari peralatan *pneumatic* dan elektrolis. Peralatan-peralatan tersebut akan dijelaskan pada gambar 2.6 dan gambar 2.7.



Gambar 2.6 *Distribution Station* Tampak Atas



Gambar 2.7 *Distributing Station* Tampak Samping

Keterangan bagian-bagian *distributing station* yang terdapat pada gambar 2.6. dan 2.7 akan dijelaskan pada tabel 2.6.

Tabel 2.6 Daftar bagian-bagian dari *distributing station*

Item	Description
1	Profile plate, 700 mm X 700 mm
3	Service unit
4	I/O terminal
6a	Valve block
7	Vacuum generator
8	Pneumatic/electronic switch
13	Cable duct
15	Gravity feed magazine
16	Optoelectronic sensor
17	Swivel drive

Distributing station mempunyai beberapa peralatan yaitu antara lain :

- *Feed Magazine Module*

Feed Magazine Module yaitu memindahkan benda dari *magazine*. Silinder *double-acting* menekan benda keluar dari *feed magazine*. Posisi terakhir setiap proses menimbulkan efek bagi sensor yang ada. *Magazine*

berfungsi sebagai *monitoring* dan benda selalu sedia apabila sewaktu-waktu ada proses pengiriman ke *magazine*.

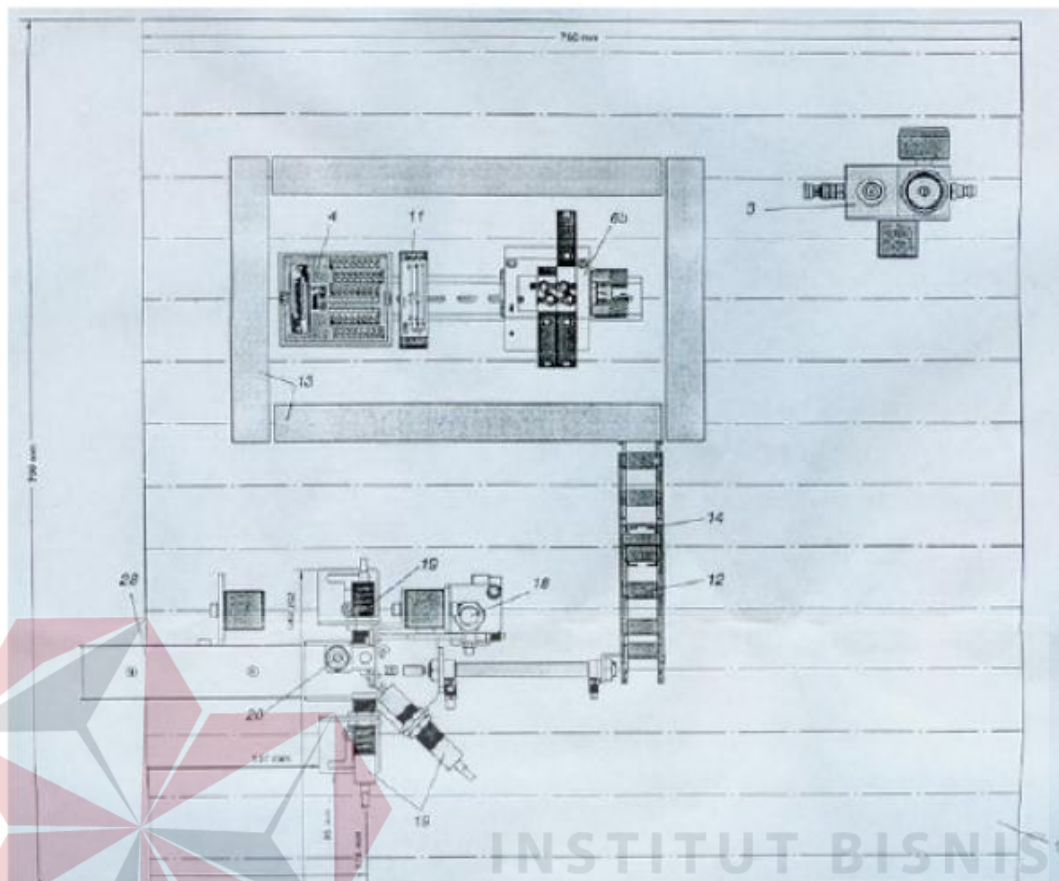
- *Transfer Module*

Transfer module adalah peralatan *pneumatic handling*. Benda diambil oleh *vacuum section cup* dan bisa dipindahkan mulai dari 0° - 180° melalui *swivel drive*. Posisi terakhir proses akan menimbulkan efek bagi sensor yang ada.

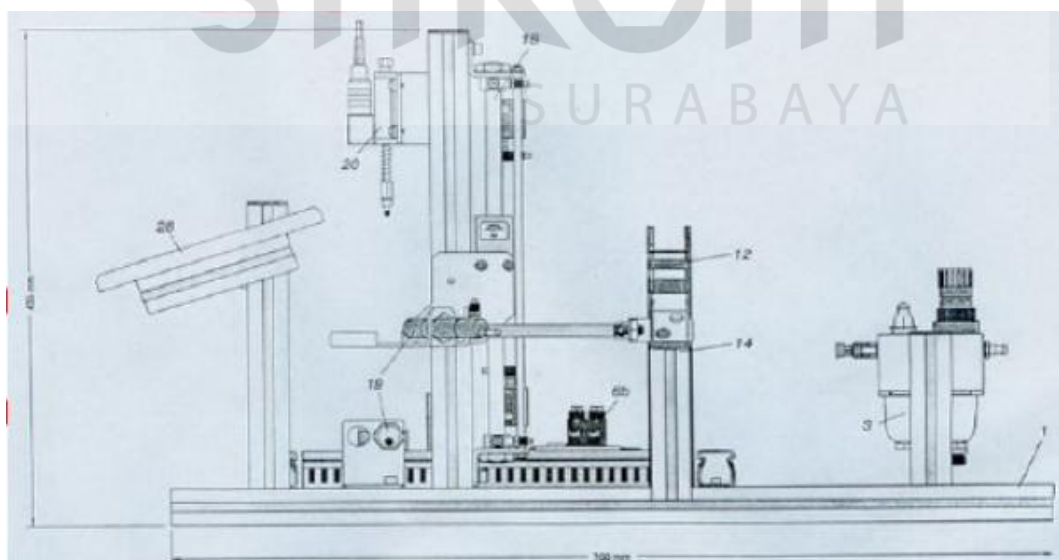
b. *Testing Station*

Testing station merupakan stasiun kedua dari keempat stasiun MPS. Stasiun ini menerima barang dari *distribution station*, kemudian mendeteksi barang tersebut untuk diketahui jenis serta ukurannya. Kemudian mengirimkan objek tersebut beserta data jenis dan ukurannya ke *processing station*.

Testing station terdiri dari peralatan *pneumatic*, elektrolik dan sensor. Peralatan-peralatan tersebut akan dijelaskan pada gambar 2.8 dan 2.9.



Gambar 2.8 *Testing Station Tampak Atas*



Gambar 2.9 *Testing Station Tampak Samping*

Tabel 2.7 Daftar bagian-bagian dari *Testing Station*

Item	Description
1	Profile plate, 700mm X 700mm
3	Service unit
4	I/O terminal
6b	Valve block
11	Impedance converter
12	Flexible cable duct
13	Cable duct
14	Profile rail
18	Lifting module
19	Detection module, capacitive, inductive, optical
20	Measuring module, position sensor
28	Slide module

Keterangan gambar 2.8 dan gambar 2.9 dapat dijelaskan pada tabel

2.7. *Testing station* mempunyai dua fungsi, yaitu untuk mendapatkan data karakteristik dari suatu benda dengan mendeteksi jenis, warna dan tinggi benda. Sedangkan fungsi keduanya yaitu membuang benda yang dinyatakan tidak layak atau meneruskan benda ke *processing station* merupakan stasiun yang akan memberikan informasi tentang karakteristik dari benda yang diterimanya.

Testing station mempunyai beberapa peralatan yaitu antara lain :

- *Detection Module*

Peralatan ini digunakan untuk mengetahui karakteristik jenis dan warna digunakan tiga buah *proximity sensor* yang mengeluarkan *output digital*. Ketiga sensor itu adalah *inductive proximity sensor* (mendeteksi benda dari logam), *capacitive proximity sensor* (mendeteksi semua jenis benda), *optical proximity sensor* (mendeteksi benda warna merah dan benda logam).

- *Lifting Module*

Peralatan ini digunakan untuk mengangkat benda dari *detection module* ke *measuring module*. *Lifting cylinder* dan *ejecting cylinder* digunakan sebagai penggerakanya. Benda yang diangkat yaitu setelah dideteksi dengan *magnetic* atau *iinductive proximity sensor*.

- *Measuring Module*

Peralatan ini terdiri dari sendor analog yang digunakan untuk mengukur ketebalan benda. Benda diangkat dengan *lifting module* untuk diukur ketebalannya pada modul ini.

- *Slide Module*

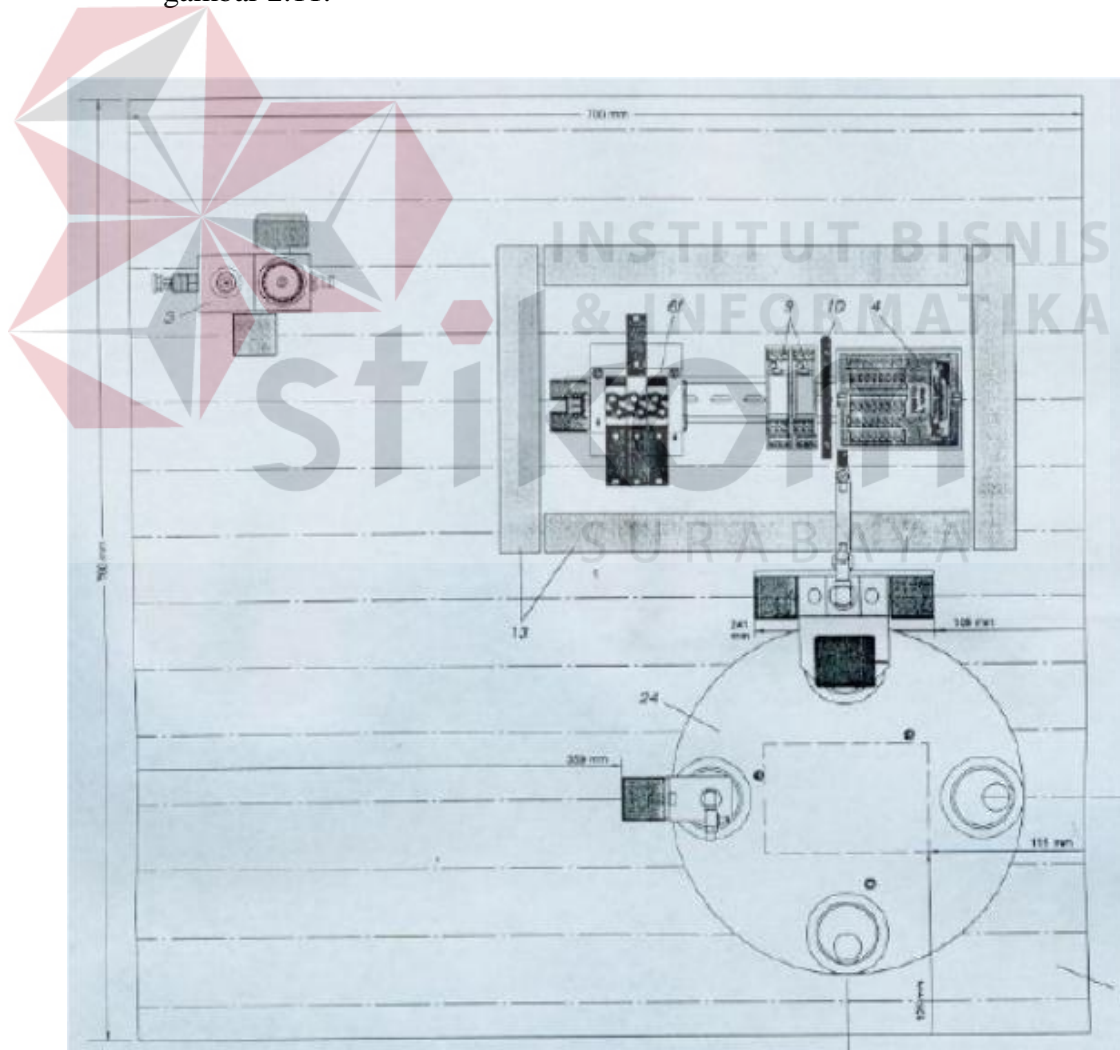
Peralatan ini digunakan untuk mengantarkan benda ke stasiun berikutnya. Benda yang telah diproses pada *measuring module* kemudian didorong dan melewati *slide module* sebagai pengantarnya ke stasiun berikutnya.

Hasil pembacaan *input analog* akan tersimpan ke register-register yang telah ditentukan. Demikian pula jika hendak mengeluarkan suatu nilai ke *output analog*, nilai yang dikeluarkan adalah nilai yang tersimpan pada *register* tersebut. Pada pengukuran yang dilakukan pada modul ini, hasilnya tersimpan pada *register* 50 (R50) dimana sebagai selector *enable* adalah *flag* 14.0 (f14.0).

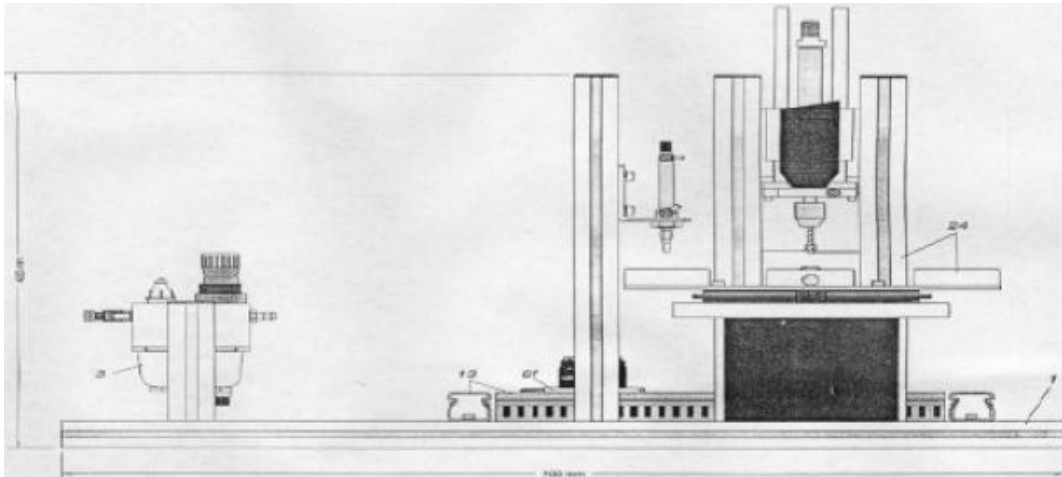
c. *Processing Station*

Processing station adalah stasiun ketiga dari empat stasiun MPS. *Processing station* menerima benda dan data karakteristiknya dari *testing station* dan mengolah benda tersebut untuk ditambahkan informasi baru, untuk kemudian data baru tersebut dikirimkan ke *handling station*. *Processing station* terdiri dari peralatan *pneumatic*, elektrolik dan sensor.

Peralatan-peralatan tersebut akan dijelaskan pada gambar 2.10 dan gambar 2.11.



Gambar 2.10 *Processing Station* Tampak Atas



Gambar 2.11 *Processing Station* Tampak Samping

Keterangan bagian-bagian *processing station* yang terdapat pada gambar 2.10 dan gambar 2.11 akan dijelaskan pada tabel 2.8.

Tabel 2.8 Daftar bagian-bagian dari *Processing Station*

Item	Description
1	Profile plate, 700mm X 700mm
3	Service unit
4	I/O terminal
6f	Valve block
9	Relay
10	Fuse cartridge
13	Cable duct
24	Processing module : drilling module, drilled hole, checking module, and indexing table module

Processing station mempunyai beberapa peralatan, antara lain :

- *Rotary Index table module*

Rotary index table module atau meja putar digerakkan oleh sebuah motor. Posisi perputaran meja diatur oleh tonjolan-tonjolan logam yang terdapat pada posisi tertentu, sehingga dapat dideteksi oleh *inductive sensor* (sensor logam). Meja putar ini mempunyai empat

buah wadah objek. Pada tiap-tiap wadah tersebut, terdapat lubang yang berfungsi untuk mendeteksi ada atau tidaknya benda oleh sebuah *optical sensor* (sensor warna).

- *Drilling module*

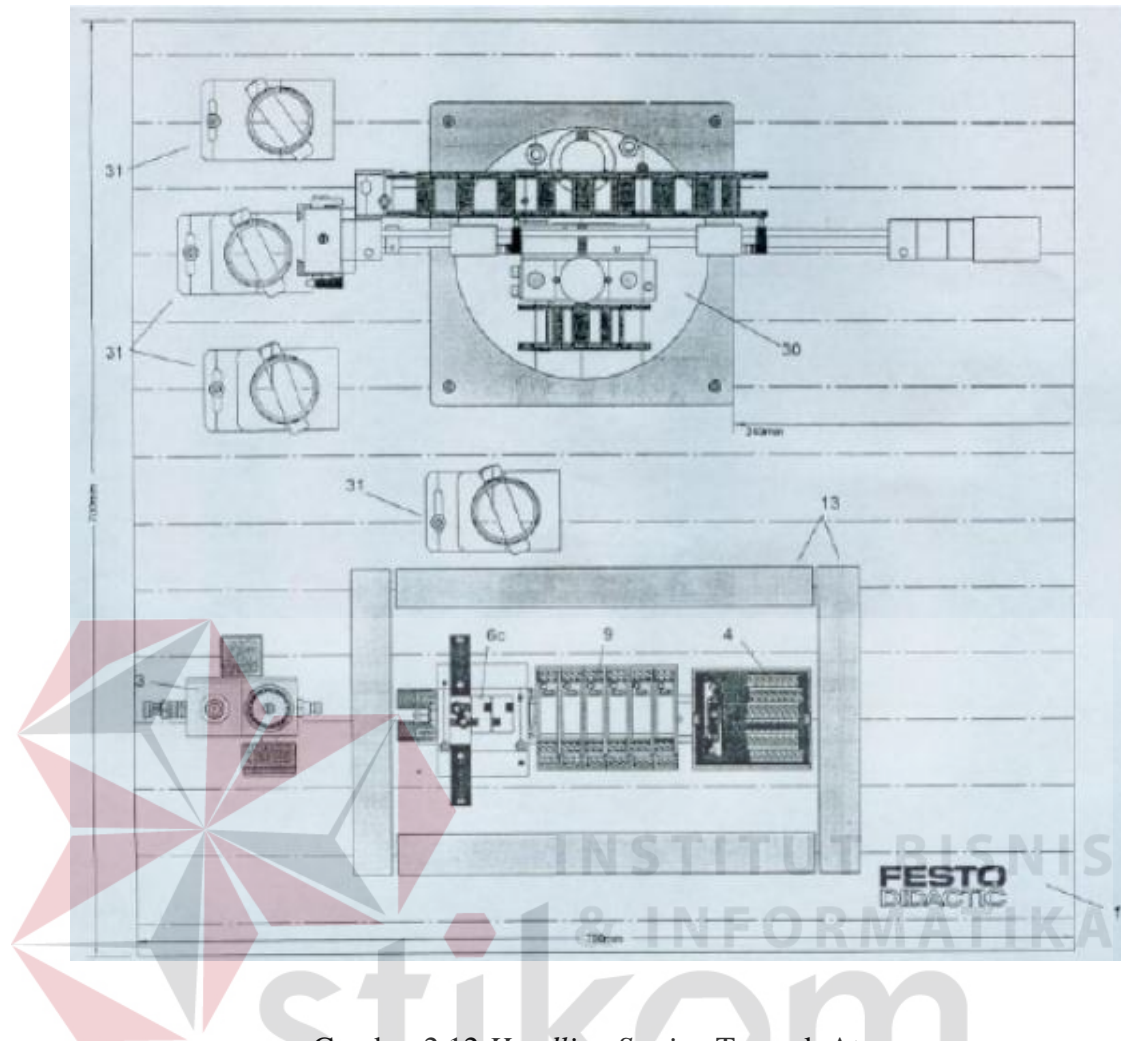
Drilling module berfungsi untuk melubangi objek menggunakan bor pada ukuran kedalaman tertentu. Sebelum pengeboran dilakukan, benda harus dijepit terlebih dahulu oleh sebuah silinder agar memiliki posisi yang stabil. Bor digerakkan oleh sebuah motor dan digerakkan naik-turun. Kedalaman lubang hasil pengeboran ditentukan oleh *inductive sensor*.

- *Drilling hole checking module*

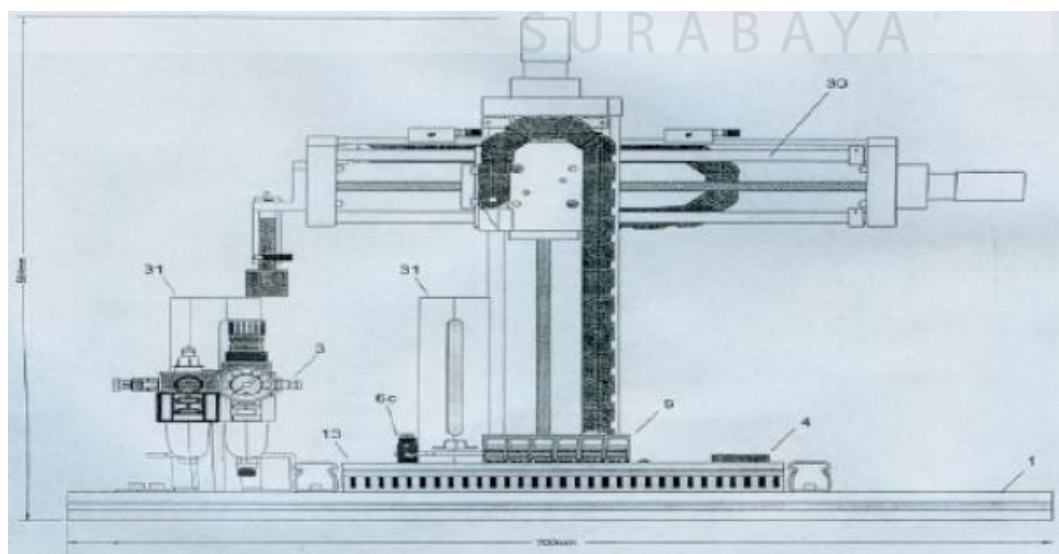
Drilling hole checking module atau modul pendeteksi lubang berfungsi untuk mengecek lubang yang telah dilakukan pada modul sebelumnya. Jika ukuran kedalaman lubang pada benda memenuhi syarat maka benda dianggap valid dan jika tidak benda dianggap barang *reject*. (Auckland University, 2007).

d. *Handling Station*

Handling station adalah stasiun terakhir MPS. Stasiun ini berfungsi untuk memindahkan benda dari *processing station* untuk kemudian ditempatkan pada tempatnya masing-masing sesuai dengan jenis, warna dan tinggi benda tersebut. Peralatan ini dapat dilihat pada gambar 2.12 dan gambar 2.13 (Festo Didactic, 1995 : 12).



Gambar 2.12 *Handling Station* Tampak Atas



Gambar 2.13 *Handling Station* Tampak Samping

Keterangan bagian-bagian *handling station* yang terdapat pada gambar 2.12 dan gambar 2.13 akan dijelaskan pada tabel 2.9.

Tabel 2.9 Daftar bagian-bagian dari *Processing Station*

Item	Description
1	<i>Profile plate, 700mm X 700mm</i>
3	<i>Service unit</i>
4	<i>I/O Terminal</i>
6c	<i>Valve block</i>
9	<i>Relay</i>
13	<i>Cable duct</i>
30	<i>Handling device</i>
31	<i>Magazine module</i>

Handling station mempunyai beberapa peralatan, antara lain :

- *3-Axis Gantry With Pneumatic Gripper*

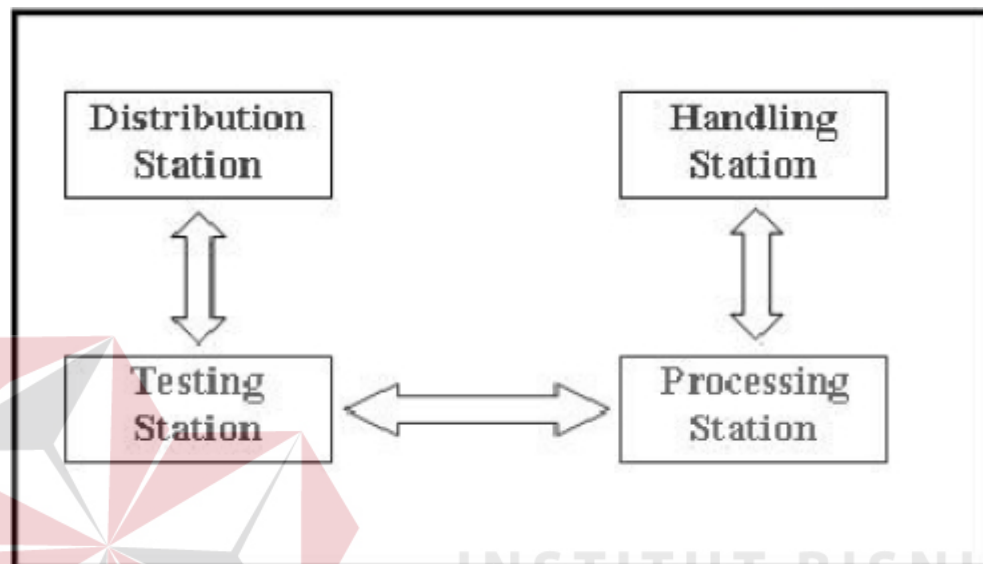
3-Axis gantry berupa sebuah lengan yang dapat bergerak. Peralatan ini memiliki 3 macam gerakan, yaitu maju dan mundur, naik dan turun serta kiri dan kanan. Pergerakan dari lengan tersebut digerakkan oleh 3 buah motor. *Pneumatic gripper* berfungsi untuk mengambil benda yang telah diolah pada *processing station* dengan cara menjepit benda tersebut.

- *Magazine module*

Magazine module adalah tempat dimana untuk meletakkan objek yang sesuai dengan karakteristik objek tersebut.

2.2.6 Hubungan Antar MPS

Agar dapat berkomunikasi maka tiap-tiap MPS harus saling berhubungan, seperti terlihat pada gambar 2.14 di bawah ini.



Gambar 2.14 Hubungan antar MPS

Dari gambar di atas dapat dijelaskan bahwa MPS hanya berhubungan dengan stasiun tetangganya. Untuk dapat berkomunikasi diperlukan *port* komunikasi pada masing-masing MPS.

2.2.7 Bit-bit Komunikasi MPS

Untuk dapat berkomunikasi, tiap MPS harus memberikan sinyal status berupa status siap (*ready*), *busy*, *error*, dan data yang terkirim atau dikirim. Sinyal status yang digunakan yaitu berupa bit-bit komunikasi yang sudah terstruktur pada MPS tersebut.

Bit-bit komunikasi terdapat pada suatu *output word* (OW) dan *input word* (IW). Untuk dapat berkomunikasi dengan *previous station* digunakan OW11 dan IW11. Sedangkan untuk berkomunikasi dengan *subsequent station* digunakan OW10 dan IW10. Bit-bit yang digunakan adalah sebagai berikut :

- *Transferring a part*

Bit ini pada MPS diberi identitas yaitu P_RDY. P_RDY berfungsi sebagai pemberi tanda bahwa MPS tersebut sudah siap untuk menerima benda dari MPS sebelumnya (*previous station*). Bit P_RDY harus di-SET terlebih dahulu pada MPS berikutnya (*subsequent station*) dan MPS sebelumnya akan menerima tanda bahwa MPS berikutnya sudah siap untuk menerima benda.

Tabel 2.10 Status P_RDY

Nilai	Keterangan
0 (nol)	<i>Subsequent station</i> belum siap menerima benda
1 (satu)	<i>Subsequent station</i> siap menerima benda

- *Requesting part-specific data*

Bit ini pada MPS diberi identitas yaitu D_REQ. D_REQ berfungsi sebagai pengatur perpindahan data (informasi karakteristik benda) dari *previous station* ke *subsequent station*. Bit D_REQ harus di-SET terlebih dahulu pada *subsequent station* dan *previous station* akan menerima tanda bahwa *subsequent station* sudah menerima data. (Festo Didactic, 1995 : A8).

Tabel 2.11 Status D_REQ

Nilai	Keterangan
0 (nol)	<i>Subsequent station</i> belum siap menerima data benda
1 (satu)	<i>Subsequent station</i> siap menerima data benda

- *Transferring data*

Bit ini pada MPS diberi identitas yaitu EN. EN berfungsi sebagai memberi sinyal kepada *subsequent station* untuk mengambil data dari *previous station*. Informasi bit dari *previous station* dapat dibaca oleh *subsequent station* jika bit ini sudah di-SET oleh *previous station*. Bit ini digunakan setelah menerima sinyal bit D_REQ.

Tabel 2.12 Status EN

Nilai	Keterangan
0 (nol)	Data belum siap diambil
1 (satu)	Data siap untuk diambil

- *Readness of subsequent station*

Bit ini pada MPS diberi identitas yaitu S_RDY. S_RDY di-SET oleh *subsequent station* yang dikirimkan kepada *previous station* yang memberikan tanda bahwa *subsequent station* sudah aktif. (Festo Didactic, 1995 : A10).

Tabel 2.13 Status S_RDY

Nilai	Keterangan
0 (nol)	<i>Subsequent station</i> tidak aktif atau error
1 (satu)	<i>Subsequent station</i> telah aktif

- *Part-specific data*

MPS memiliki bit-bit data yang bisa digunakan yaitu bit D0-D5. Bit ini digunakan untuk memberikan informasi mengenai karakteristik benda yang dikirimkan ke *subsequent station*. Bit D0-D5 biasanya akan dikirimkan jika bit EN sudah diaktifkan oleh *previous station*.

Bit D0-D5 dibagi menjadi beberapa bagian , yaitu :

a. Bit-bit dari *testing station*

Pada *testing station* digunakan 2 bit data, yaitu D1 dan D2. Bit D1 dan D2 adalah bit yang digunakan untuk warna dan jenis benda.

Tabel 2.14 Bit Data D1 dan D2

Nilai D1	Nilai D2	Keterangan benda
1	0	Merah
0	1	Hitam
1	1	Logam

b. Bit dari *previous station*

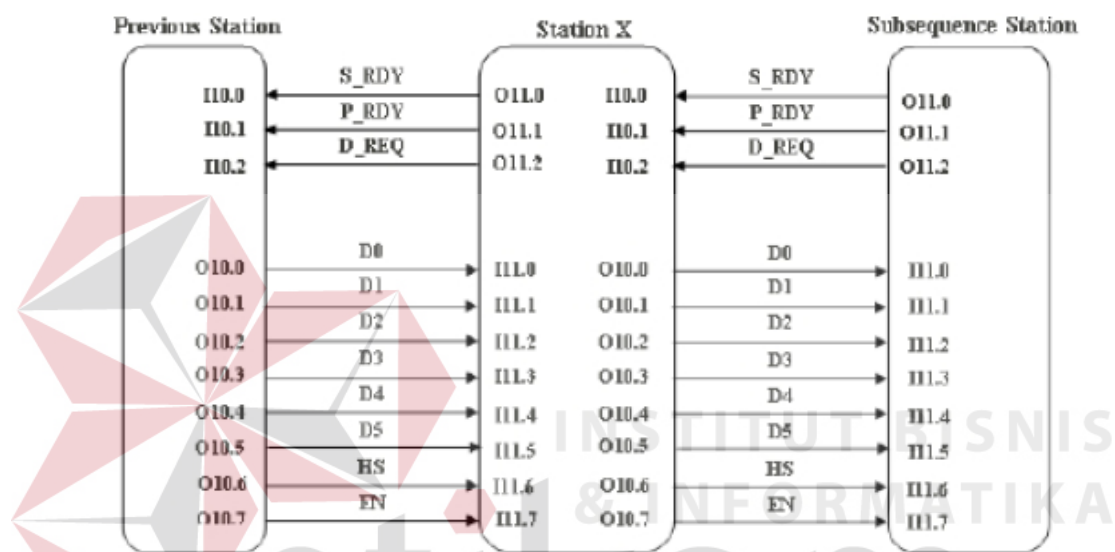
Pada *processing station* digunakan bit D3 yaitu untuk mendeteksi lubang dari hasil pengeboran.

Tabel 2.15 Bit Data D3

Nilai D3	Keterangan hole
0 (nol)	<i>Undrilled hole</i>
1 (datu)	<i>Drilled hole OK</i>

- c. Bit D0, D4 dan D5 tidak digunakan.

Untuk memperjelas teori bit-bit di atas, gambar 2.15 menunjukkan bit-bit komunikasi yang terdapat pada stasiun MPS. Pengalamatan bit-bit *input / output* yang digunakan dapat dilihat pada gambar 2.15. (Festo Didactic, 1995 : A13).



Gambar 2.15 Alur Bit I/O komunikasi MPS

2.3. Komunikasi Serial

Komunikasi data berarti pengiriman data antara dua komputer, antara sebuah komputer dengan terminal, atau antara terminal dengan terminal yang lain.

Komunikasi data dapat dilakukan dengan dua cara : paralel dan serial. Dalam *transfer* data paralel, sering 8 atau lebih jalur (konduktor kabel) digunakan untuk mentransfer data ke suatu *device* yang berjarak hanya beberapa kaki saja. Meskipun dalam kasus-kasus seperti ini banyak data bisa ditransfer dalam waktu singkat dengan menggunakan banyak kabel yang disusun paralel, tetapi jaraknya

tidak bisa jauh dan biayanya relatif mahal. Untuk mengirim data ke suatu *device* yang terletak sejauh beberapa meter, digunakan metode serial. Dalam komunikasi serial, data dikirim satu *byte* atau lebih. Kelebihan metode serial ini adalah selain dapat digunakan dalam jarak jauh, juga memerlukan biaya yang lebih murah jika dibandingkan dengan metode paralel.

Komunikasi data serial menggunakan dua metode, asinkron dan sinkron. Metode sinkron mengirim suatu blok data (karakter) pada suatu waktu sedangkan asinkron mengirim suatu *byte* tunggal pada suatu waktu. Ada kemungkinan untuk membuat *software* untuk digunakan dengan metode di atas. Tetapi programnya mungkin bisa menjadi panjang dan rumit.

2.3.1 Transmisi Half dan Full Duplex

Jika data bisa ditransmisikan dan diterima, itu disebut transmisi *duplex*. Beberapa dengan transmisi *simplex* seperti printer produksi dulu atau jadul, yang mana komputer hanya mengirim data. Transmisi *duplex* bisa *half* atau *full duplex*, tergantung pada *transfer* datanya bisa sekaligus atau tidak. Jika data dapat dikirimkan satu jalan pada suatu waktu, ini disebut *half duplex*. Dan bila data bisa melewati dua jalan pada waktu yang sama, itulah *full duplex*. Tentu saja *full duplex* membutuhkan dua kabel konduktor (selain *ground*), satu untuk transmisi dan satu untuk penerimaan, agar bisa mengirim dan menerima data secara sekaligus.

2.3.2 Komunikasi serial asinkron dan data framing

Data yang masuk pada akhir penerimaan dari jalur data dalam *transfer* data serial semuanya dalam program 0 dan 1, sangat sulit untuk memahami data kecuali pengirim dan penerima menyepakati seperangkat peraturan, sebuah protokol, tentang bagaimana data dipaketkan dan berapa banyak *bit constitute* sebuah karakter, dan kapan data mulai dan berakhir.

Komunikasi data serial asinkron digunakan secara luas untuk transmisi berorientasi karakter, dan *transfer* data berorientasi blok yang menggunakan metode sinkron. Dalam metode asinkron, setiap karakter diletakkan antara *start bit* dan *stop bit*. *Start bit* selalu satu *bit* tetapi *stop bit* bisa satu atau dua *bit*. *Start bit* selalu '0' (*low*) dan *stop bit* adalah '1' (*high*).

Dalam komunikasi serial asinkron, *chip-chip peripheral* dan *modem* bisa diprogram untuk data selebar 5, 6, 7, atau 8 *bit*. Ini sebagai tambahan dari jumlah *stop bit*, 1 atau 2. Sementara dalam system yang lebih lama karakter-karakter ASCII adalah 7 *bit* tapi dengan adanya karakter ASCII *extended*, dibutuhkan 8 *bit* untuk setiap karakter. *Keyboard non-ASCII* kecil menggunakan karakter-karakter 5 dan 6 *bit*.

Dalam beberapa system lama, disebabkan kelambatan peralatan mekanik yang menerima. 2 *stop bit* digunakan untuk memberikan peralatan tersebut cukup waktu untuk mengorganisasi dirinya sendiri sebelum transmisi dari *byte* berikutnya. Tetapi pada PC *modern* penggunaan 1 *stop bit* adalah umum. Dengan asumsi bahwa kita mengirim *text* dari karakter ASCII menggunakan 2 *stop bit* sehingga total untuk tiap karakter 11 *bit* yaitu 8 *bit* untuk program

ASCII-nya, dan 1 dan 2 *bit* masing-masing untuk *start bit* dan *stop bit*. Oleh karena itu, untuk setiap karakter 8 *bit* ada 3 *bit* ekstra, atau lebih dari 25%.

Dalam beberapa system untuk menangani integritas data, *bit parity* dari *byte* karakter dimasukkan dalam data *frame*. Ini berarti bahwa untuk setiap karakter (7 *bit* atau 8 *bit*, tergantung pada sistem) kita punya *bit parity* tunggal sebagai tambahan dari *start bit* dan *stop bit*. *Bit parity* adalah ganjil atau genap. Dalam *bit parity* ganjil jumlah total *bit* data, termasuk *bit parity* adalah ganjil dari 1-an. Serupa dengan itu, dalam sebuah *bit parity* genap jumlah total *bit* data termasuk *bit parity* adalah genap.

2.4. Penelitian Sebelumnya

Telah terdapat banyak penelitian yang menggunakan peralatan MPS, yaitu Susanto (2001), Febriadi (2005), Dirgayusari (2007), Wiseso (2008), dan Kurnia (2008). Salah satu fitur utama dari MPS adalah komunikasi yang terjalin antar MPS, dan secara teknis dilakukan oleh PLC pada tiap MPS. Komunikasi yang terjadi pada semua penelitian tersebut adalah menggunakan input dan output PLC sebanyak 16 bit input dan 16 bit output.

Salah satu fitur utama PLC adalah ketersediaan input dan output yang memadai. Jika untuk berkomunikasi dengan PLC lain membutuhkan demikian banyak input dan output, maka terjadi pemborosan fitur utama.

Penelitian ini dilakukan secara fokus untuk mengatasi pemborosan yang terjadi untuk melakukan komunikasi. Metode yang akan digunakan adalah metode serial yang merupakan penggabungan dari hasil penelitian Wijayapurwana

(2004) dan Abddurrachman (2005). Wijayapurwana (2004) melakukan transmisi data satu arah dari PLC menuju *microcontroller*, sedangkan Abdurracman (2005) melakukan transmisi data satu arah dari *microcontroller* menuju PLC.

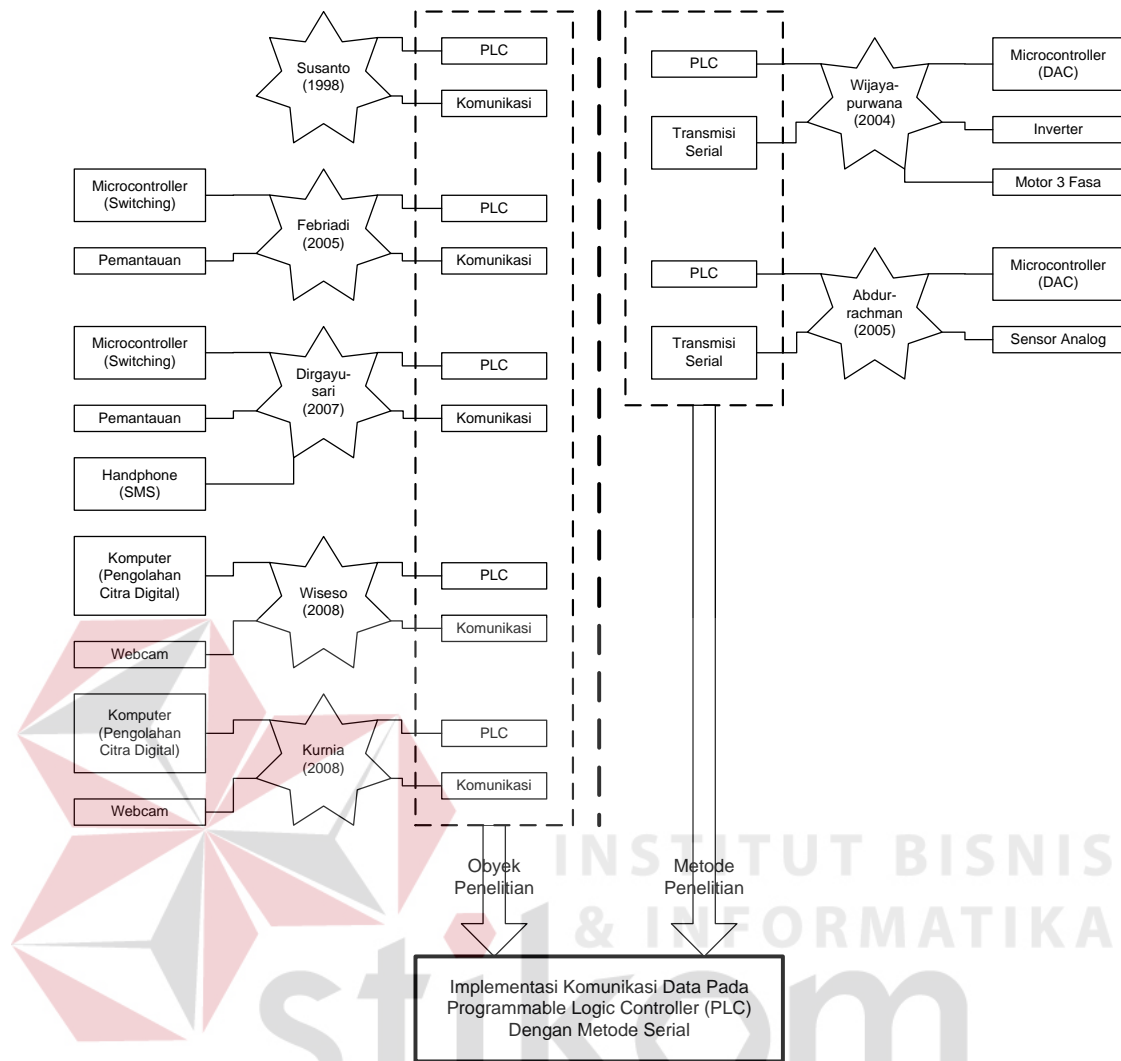
Metode serial dipilih dengan pertimbangan metode ini hanya menggunakan 1 bit input untuk penerimaan data dan 1 bit output untuk pengiriman data. Metode serial ini diharapkan dapat menghemat penggunaan input dan output PLC, semula 8 bit input dan 8 bit output menjadi 1 bit input dan 1 bit output. Jadi penghematan yang dilakukan mencapai 8 kali lipat untuk komunikasi antara sebuah PLC dengan sebuah PLC lain.

Penggunaan metode serial menuntut dibangunnya suatu protokol komunikasi yang mempunyai kemampuan untuk :

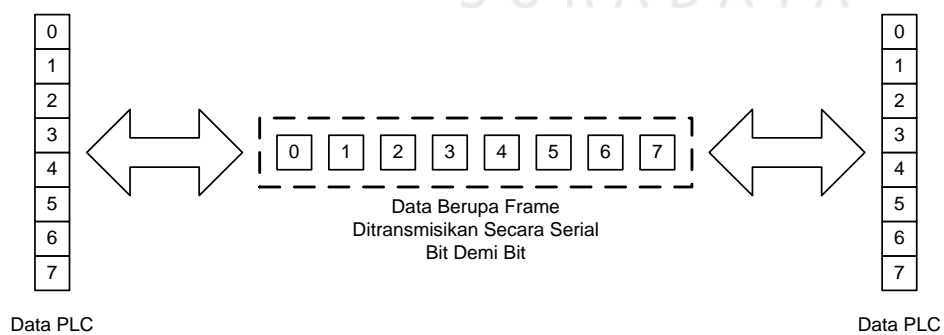
- a. Mengubah data yang ingin dikirim menjadi deretan data 1 bit.
- b. Mengirimkan deretan data tersebut bit demi bit melalui 1 bit output PLC.
- c. Menerima bit demi bit data melalui 1 bit input PLC.
- d. Mengubah deretan data tersebut menjadi data sesungguhnya.

Data yang digunakan untuk melakukan komunikasi direncanakan merupakan data dalam bentuk *frame* atau *packet*. Frame ini akan berisi *header*, data dalam bentuk terkode, dan *trailer*. Untuk membangun frame ini perlu dilakukan pengkodean terhadap sinyal-sinyak komunikasi yang digunakan pada MPS, kemudian menambahkan *header* di depan data dan *trailer* di belakang data.

Seluruh gagasan penelitian untuk tugas akhir ini tertuang pada diagram pada Gambar 2.16 dan Gambar 2.17 berikut.



Gambar 2.16 Gagasan Penelitian yang hendak dilakukan



Gambar 2.17 Metode Serial Yang Hendak Dibangun

Penelitian yang hendak dilakukan ini dilatarbelakangi oleh penelitian-penelitian yang telah dilakukan sebelumnya. Penelitian-penelitian tersebut adalah sebagai berikut.

2.4.1 Penelitian Susanto (2001)

Penelitian dilakukan dengan latar belakang bahwa terjadi ketidakstabilan komunikasi antara PLC yang terpasang pada MPS. Seringkali terjadi kesalahan komunikasi yang menyebabkan kemacetan kerja MPS.

MPS merupakan suatu peralatan simulasi tentang proses produksi yang terbagi menjadi 4 stasiun, yaitu *Distributing*, *Testing*, *Processing*, dan *Handling*. Setiap MPS dikendalikan oleh sebuah PLC. Empat MPS tersebut bekerja secara berurutan seperti proses produksi, yaitu penyediaan bahan baku (*Distributing*), pendeteksian jenis dan karakteristik bahan baku (*Testing*), pengolahan bahan baku (*Processing*), dan pemilahan hasil pengolahan (*Handling*).

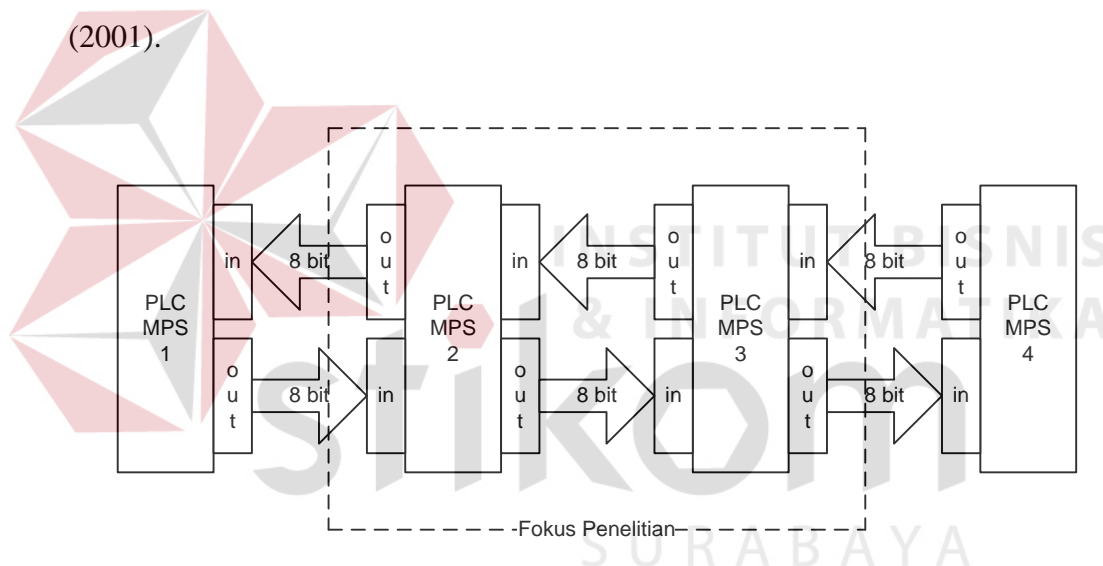
Bahan baku beserta datanya mengalir mulai dari *Distributing* menuju *Handling*. Sinkronisasi gerakan sistem antar MPS juga perlu dilakukan agar tidak terjadi benturan mekanik dan kecelakaan sistem. Untuk keperluan-keperluan ini diperlukan komunikasi antar MPS yang secara teknis dilakukan oleh PLC yang mengendalikan MPS tersebut.

Implementasi komunikasi yang telah ada sebelumnya sering tidak stabil, sehingga sering terjadi kegagalan komunikasi yang berakibat pada kemacetan kerja MPS. Berdasarkan masalah ini diteliti cara komunikasi yang dilakukan oleh PLC dan memberikan perbaikan-perbaikan sehingga tidak terjadi

lagi kegagalan komunikasi. Penelitian fokus hanya pada 2 MPS yang terletak di tengah, yaitu *Testing* dan *Processing*.

Komunikasi dilakukan dengan menggunakan input dan output PLC. Untuk berkomunikasi dengan sebuah PLC yang lain digunakan 8 bit input dan 8 bit output. Khusus untuk *Testing* dan *Processing*, karena berada di tengah, maka akan berkomunikasi dengan 2 PLC sekaligus, yaitu PLC sebelumnya dan PLC sesudahnya, sehingga kebutuhan komunikasi menjadi dua kali lipat, yaitu 16 bit input dan 16 bit output. Gambar 2.18 menunjukkan fokus penelitian Susanto

(2001).



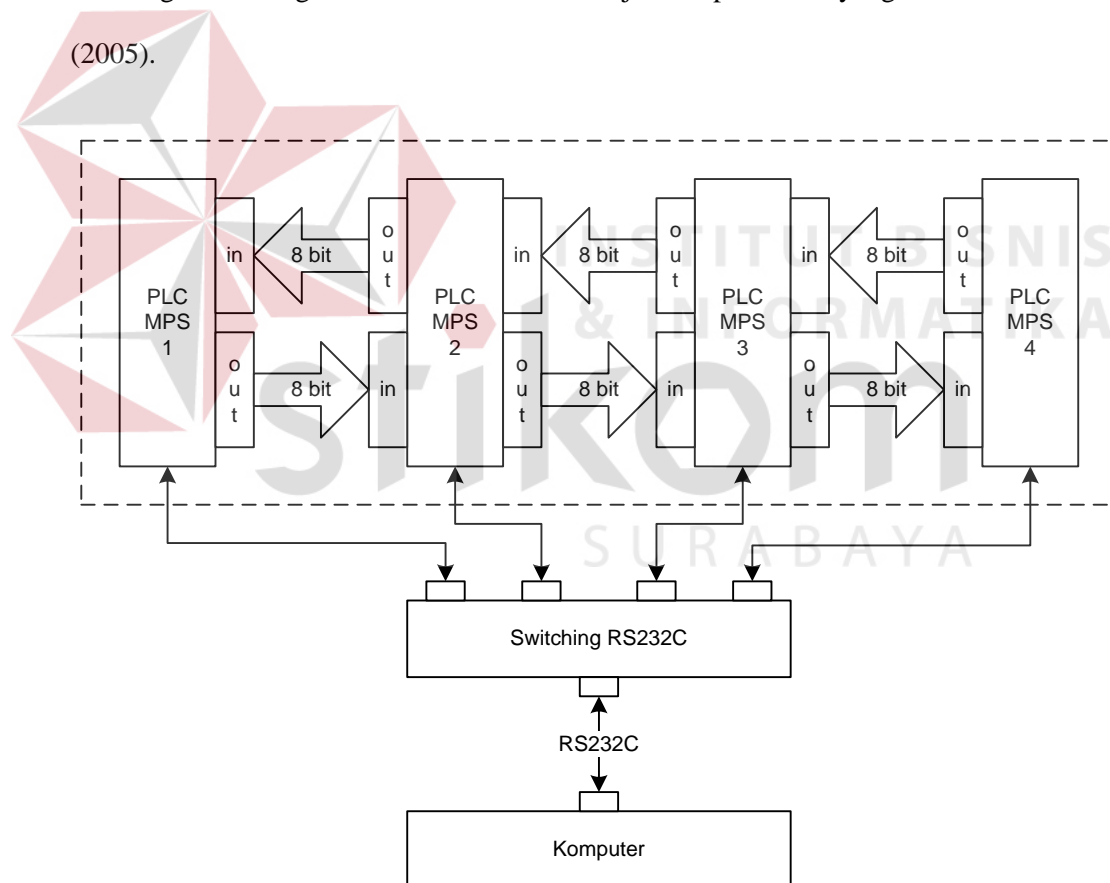
Gambar 2.18 Fokus Penelitian Susanto (2001)

2.4.2 Penelitian Febriadi (2005)

Komputer dan PLC pengendali MPS dihubungkan dengan RS232C. terdapat keadaan bahwa untuk melakukan pemantauan sebuah MPS diperlukan sebuah komputer. Hal ini disebabkan PLC hanya mempunyai sebuah port serial dan aplikasi yang ada pada computer hanya mampu berhubungan dengan sebuah PLC saja dalam satu waktu.

Berdasarkan keadaan tersebut diagagas untuk membuat sistem pemantauan semua MPS hanya satu komputer saja.

Penelitian ini menggunakan metode komunikasi antar PLC yang telah digunakan Susanto (2001) dan diterapkan untuk semua PLC pengendali MPS. Metode yang digunakan agar semua MPS dapat dipantau oleh satu komputer adalah dengan melakukan “switching” komunikasi serial RS232. Dalam satu saat komputer hanya akan memantau satu MPS, dan pada saat berikutnya komputer memantau MPS yang lain, dan seterusnya secara bergantian. Hasil pemantauan divisualisasikan pada komputer dalam bentuk gambar bergerak. Gambar 2.19 menunjukkan penelitian yang dilakukan Febriadi (2005).

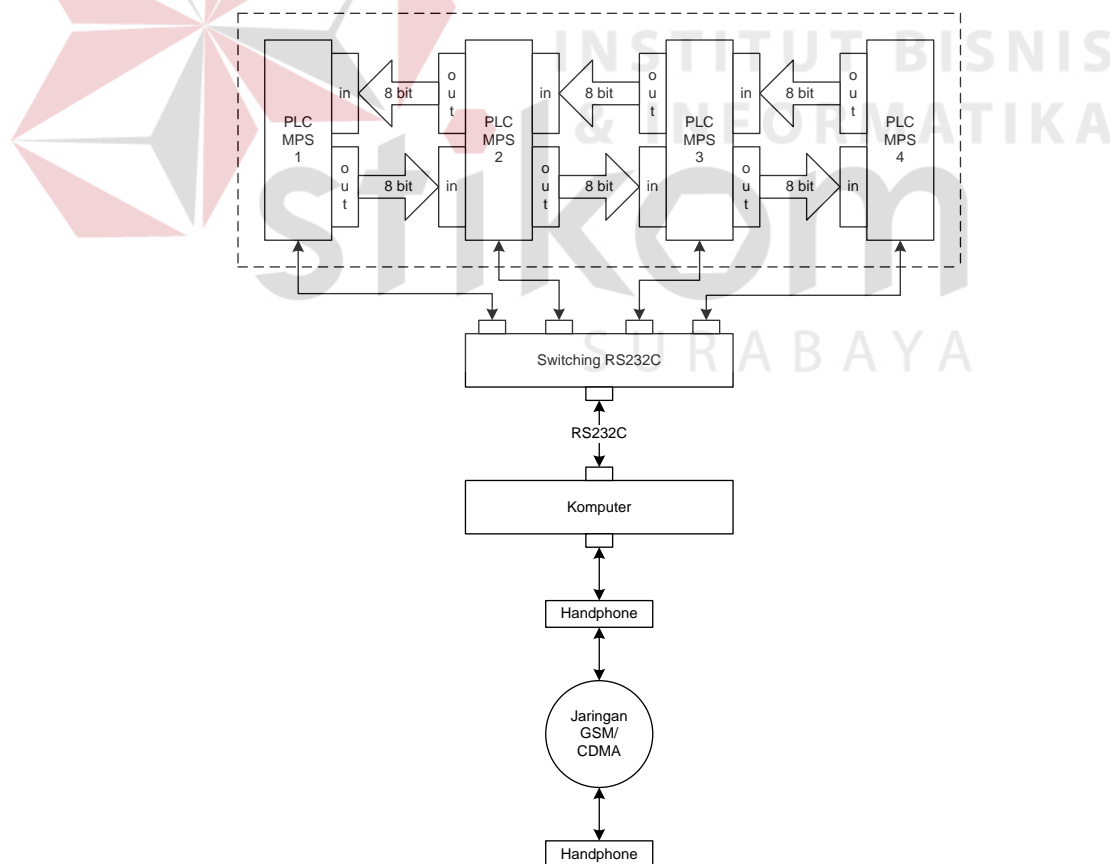


Gambar 2.19 Diagram Penelitian Febriadi (2005)

2.4.3 Penelitian Dirgayusari (2007)

Penelitian ini menggunakan semua MPS, memantau semua MPS dengan satu komputer, dan menambahkan “*warning system*” melalui *Short Message Service* (SMS). *Warning system* dilakukan bila terdeteksi kerusakan pada MPS, yang ditandai dengan kemacetan kerja MPS. Penerima SMS adalah orang yang mempunyai wewenang dan otoritas atas MPS.

Penelitian ini menggunakan metode komunikasi antar PLC yang sama dengan Febriadi (2005). Komunikasi dibangun dengan menggunakan 16 bit input PLC dan 16 bit output PLC. Gambar 2.20 menunjukkan penelitian yang dilakukan Dirgayusari (2007).

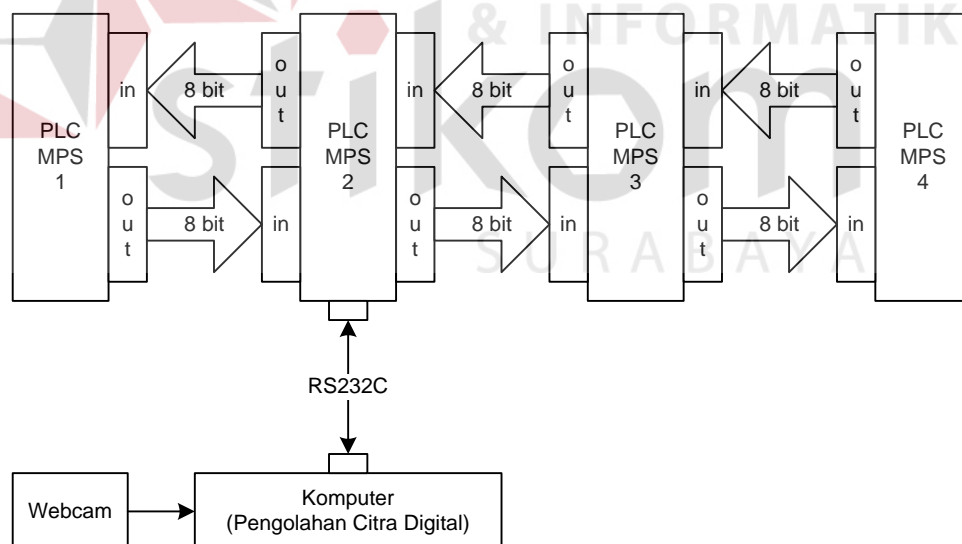


Gambar 2.20 Diagram Penelitian Dirgayusari (2007)

2.4.4 Penelitian Wiseso (2008) dan Kurnia (2008)

Penelitian dilakukan dengan fokus yang berbeda dengan Febriadi (2005) dan Dirgayusari (2007). Fokus penelitian terletak pada pengembangan sistem kerja dari MPS.

Wiseso (2008) mengamati benda kerja yang menjadi obyek dari MPS. Benda kerja yang ada berbentuk silinder dengan 3 ketinggian yang berbeda, yaitu pendek, sedang, dan tinggi. Benda kerja juga mempunyai perbedaan berdasarkan bahan dan warna, yaitu bahan bukan logam warna merah, bahan bukan logam warna hitam, dan bahan logam. Seluruh variasi benda kerja ini dapat dideteksi karakteristiknya oleh MPS (*Testing*) dengan menggunakan sensor benda, sensor logam, sensor warna merah, dan sensor tinggi.



Gambar 2.21 Diagram Penelitian Wiseso (2008) dan Kurnia(2008)

Wiseso (2008) mempunyai gagasan untuk menambah variasi benda kerja dengan memperbanyak variasi warna, yaitu biru, kuning, dan putih. Tiga

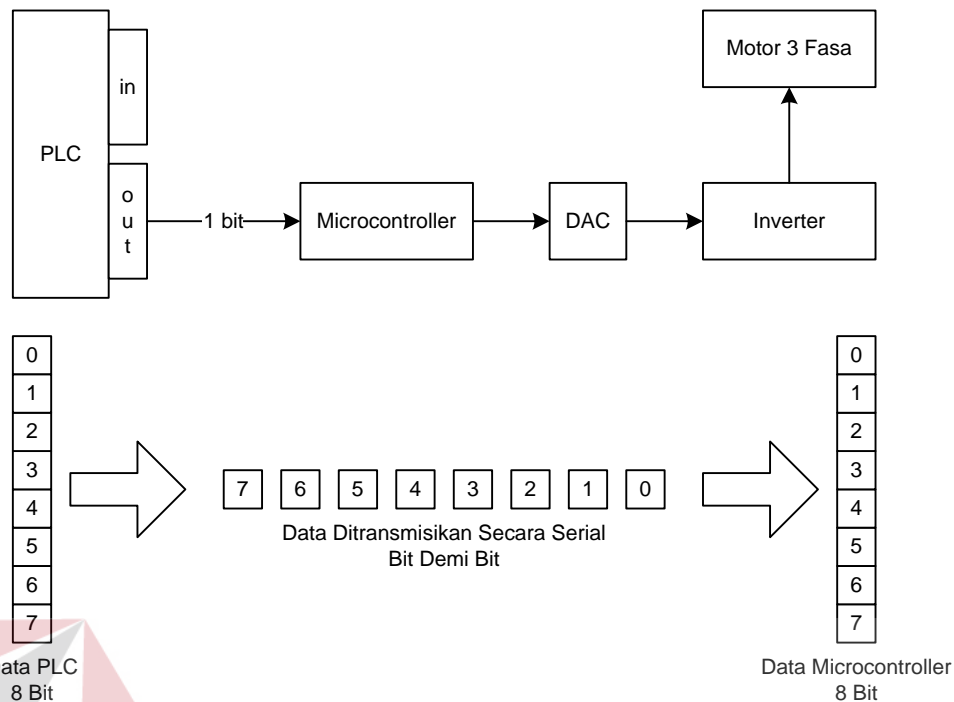
tambahan warna ini tidak dapat dideteksi dengan sensor yang ada. Digunakan webcam untuk menangkap citra benda, output citra diolah oleh komputer untuk disimpulkan warna benda tersebut. Akhirnya komputer memberi informasi kepada PLC mengenai warna benda yang dideteksi.

Kurnia (2008) mempunyai gagasan yang sama seperti Wiseso (2008), tetapi variasi benda ditambah dengan bentuk yang berbeda-beda. Dilakukan pengolahan citra dari hasil webcam untuk menyimpulkan bentuk benda kerja, dan menginformasikannya kepada PLC. Gambar 2.21 menunjukkan penelitian Wiseso (2008) dan Kurnia (2008).

Metode komunikasi yang terjadi antar PLC pengendali MPS menggunakan metode yang sama dengan Febriadi (2005), yaitu menggunakan 16 bit input PLC dan 16 bit output PLC.

2.4.5 Penelitian Wijayapurwana (2004)

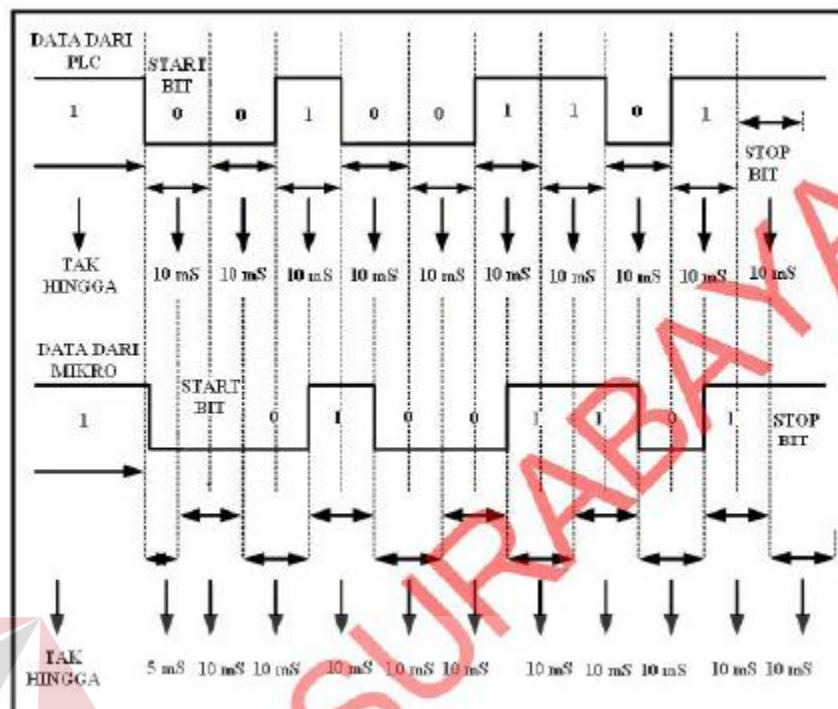
Penelitian ini melakukan pengaturan kecepatan motor tiga fasa menggunakan PLC. Motor tiga fasa digerakkan dengan menggunakan inverter. Kecepatan motor dapat diatur dengan memberikan tegangan antara 0 sampai 5 volt kepada inverter. Tegangan tersebut diatur oleh *microcontroller* melalui *Digital To Analog Converter* (DAC).



Gambar 2.22 Diagram Penelitian Wijayapurwana (2004)

Microcontroller mendapat data digital dari PLC. Pengiriman data digital sebesar 8 bit sekaligus akan memboroskan output PLC. Penelitian ini mengubah bentuk data 8-bit menjadi deretan data 1 bit. Deretan ini yang kemudian dikirimkan kepada *microcontroller*. Penelitian melibatkan perubahan bentuk data dari paralel menjadi serial untuk keperluan transmisi data. Gambar 2.22 menunjukkan diagram penelitian ini.

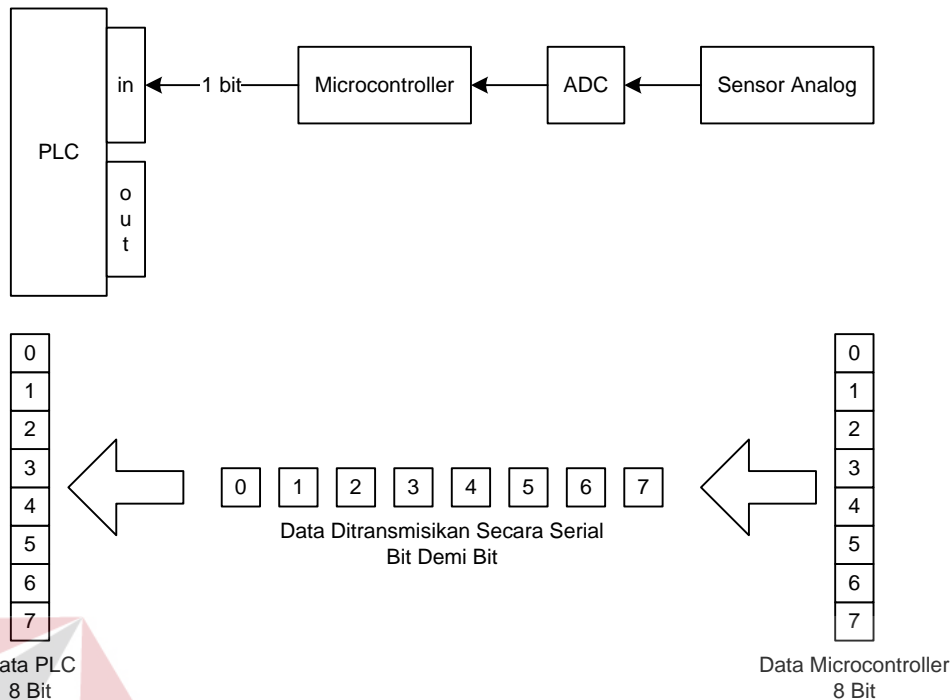
Wijayapurwana (2004) mendesain protokol komunikasi pada PLC agar bisa berkomunikasi dengan *microcontroller* menggunakan mode asinkron. Sehingga *start bit* dan *stop bit*. Desain protokol yang dibangun dapat dilihat pada gambar 2.23.



Gambar 2.23 Protokol Komunikasi antara PLC dan *microcontroller*

2.4.6 Penelitian Abdurrachman (2005)

Terdapat gagasan untuk menggunakan sensor analog pada PLC. PLC yang digunakan tidak memiliki fitur input analog dan output analog. Solusi yang dipilih adalah dengan menggunakan *microcontroller*. *Microcontroller* menggunakan *Analog To Digital Converter* (ADC) sebagai pengubah tegangan output sensor menjadi data digital. Hasil konversi yang sudah berupa data digital dapat langsung dikirimkan ke PLC.

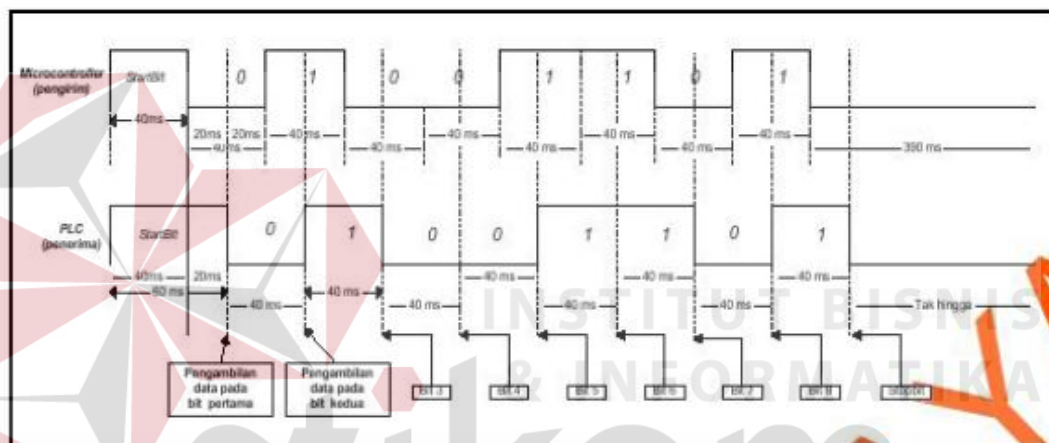


Gambar 2.24 Diagram Penelitian Abdurrachman (2005)

Metode termudah untuk pengiriman data ini adalah dengan menggunakan 8 bit output *microcontroller* yang dihubungkan dengan 8 bit input PLC. Penggunaan 8 bit input PLC untuk penerimaan data merupakan pemborosan, sehingga diperlukan metode pengiriman data yang lain. Dipilih metode yang digunakan oleh Wijayapurwana (2004), yaitu pengubahan bentuk data paralel (8 bit) menjadi serial untuk keperluan transmisi, kemudian di sisi penerima mengubahnya kembali menjadi bentuk paralel. Gambar 2.24 menunjukkan diagram penelitian ini.

Pada perancangan Abdurrachman (2005) protokol komunikasi antara *microcontroller* ke PLC diinginkan komunikasi yang dilakukan secara cepat, sehingga *timer* yang digunakan untuk mengirim satu bit data digunakan 10 ms. Sedangkan pada sisi penerima atau PLC untuk *start bit* 15 ms dan untuk

menerima data tiap bit 10 ms. Tetapi masalah terjadi karena *timer* PLC hanya bisa digunakan dengan kelipatan 10ms, sehingga untuk memakai 15 ms untuk *start bit* pada PLC tidak bisa digunakan. Maka untuk *start bit* PLC menggunakan 30 ms, dan 20 ms untuk tiap bit yang diterima. Tetapi masalah muncul lagi dikarenakan data yang diterima tidak valid sehingga diputuskan untuk membuat 40 ms untuk *start bit* PLC dan 30 ms untuk setiap bit PLC, dan hasilnya data yang diterima valid dan benar, seperti yang terlihat pada gambar 2.25.



Gambar 2.25 Protokol Komunikasi Abdurrachman (2005)