

# Implementasi *Software Proficy HMI/SCADA – CIMPLICITY 6.1* Untuk *Visualisasi Human Machine Interface (HMI)* Pada Mesin SHOTBLAST Divisi Cold Rolling Mill (CRM) PT.Krakatau Steel (Persero) Tbk

Auliyaurochman<sup>1</sup>, Agus Indra Gunawan<sup>2</sup>, Paulus Susetyo Wardhana<sup>2</sup>, M. Andi Wicaksono<sup>3</sup>

<sup>1</sup>Penulis, Mahasiswa Jurusan Teknik Elektronika PENS - ITS

<sup>2</sup>Dosen Pembimbing, Staf Pengajar di Jurusan Teknik Elektronika PENS – ITS

<sup>3</sup>Pembimbing Perusahaan, Karyawan di PT. Krakatau Steel (Persero) Tbk

Politeknik Elektronika Negeri Surabaya

Electronics Engineering Polytechnic Institute of Surabaya

Institut Teknologi Sepuluh Nopember Surabaya

Kampus ITS Sukolilo, Surabaya 60111, INDONESIA

Telp: +62 (31) 594 7280; Fax: +62 (31) 594 6114

email : [aulia\\_elzeva@yahoo.com](mailto:aulia_elzeva@yahoo.com)

**Abstrak** – *System* pengontrolan yang terdapat di mesin *Shotblast* mulai dari persiapan *work rolls* yang ditempatkan diatas *car* sampai proses pemberian kekasaran pada *work rolls* oleh mesin *rotoblast* pada mesin *shotblast* dan sampai proses kerja mesin selesai, sampai saat ini masih dijalankan secara manual melalui *push button* dari *panel thumb switch* yang keadaannya sudah *obsolete*.

Perbaikan dari kontrol kerja mesin *Shotblast*, Implementasi dari *Software Proficy HMI/SCADA – CIMPLICITY 6.1* dapat digunakan sebagai *Human Machine Interface (HMI)* pada *system* pengontrolan mesin *Shotblast* yang akan lebih mudah dalam proses pengontrolannya, serta pengguna dapat berinteraksi, mensupervisi dan mendapatkan data secara *real time*, dan apabila mesin mengalami *error* atau *fault*, maka dapat digunakan untuk data – data sebagai penanganan dari pihak perawatan mesin.

Hasil Implementasi dari *Software Proficy HMI/SCADA – CIMPLICITY 6.1* ialah untuk *push button* pengontrol mesin *shotblast* ialah model *toggle*, *trigger*, *setpoint*, perlu *scalling* nilai saat pengiriman ataupun penerimaan nilai dari *Human Machine Interface* ke *PLC* atau sebaliknya. Metode *scalling* dibuat dengan menggunakan *Script* pada *Software Proficy HMI/SCADA – CIMPLICITY 6.1*, dari 20 kali proses, terdapat 1 kali *error* dikarenakan *TRIP*, hal ini terjadi *Counter* tidak berjalan, dan *Car* berhenti sewaktu proses berlangsung.

**Kata Kunci:** *Shotblast*, *Human Machine Interface (HMI)*, *work rolls*, *rotoblast*, *Software Proficy HMI/SCADA – CIMPLICITY 6.1*, *Scalling*, *Script*, *TRIP*.

## I. PENDAHULUAN

Mesin *Shotblast* merupakan mesin yang berfungsi untuk memberikan kekasaran (*roughness*) pada permukaan *work rolls* dengan cara menembak – nembakkan pasir besi ke permukaan *work rolls* tersebut.

Untuk menjalankan mesin *Shotblast* mulai dari persiapan *work rolls* yang ditempatkan diatas *car* sampai proses pemberian kekasaran pada *work rolls* oleh mesin *rotoblast* pada mesin *shotblast* hingga sampai proses kerja mesin selesai, sampai saat ini masih dijalankan secara manual melalui *push button* dari *panel thumb switch* yang keadaannya sudah *obsolete*.

Untuk memecahkan masalah dimana *Thumb Switch* sebagai panel yang digunakan untuk menjalankan mesin *shotblast* yang keadaannya sudah *obsolete*. Maka terdapat beberapa metode yang dapat digunakan untuk memecahkan masalah tersebut, diantaranya ialah penerapan *system Human Machine Interface (HMI)* yang diimplementasikan dari *Software Proficy HMI/SCADA – Cimplicity 6.1*.

Dengan penerapan *Human Machine Interface (HMI)*, yang dibangun melalui *Software Proficy HMI/SCADA Cimplicity 6.1*, mesin *Shotblast* akan lebih mudah dalam proses pengontrolannya, serta memungkinkan pengguna dapat berinteraksi, mensupervisi dan mendapatkan data secara *real time* dan agar data tersebut dapat digunakan untuk jangka waktu tertentu, dan apabila mesin mengalami *error* atau *fault*, maka dapat digunakan untuk data – data sebagai penanganan dari pihak perawatan mesin.

## II. DASAR THEORY

### 2.1 MESIN SHOTBLAST

Mesin shot blasting merupakan mesin yang berfungsi untuk memberikan kekasaran (*roughness*) pada *work rolls*, *system* kerja dari mesin shot blasting ialah dengan cara menembak – nembakkan pasir besi ke permukaan *work rolls*, sehingga permukaan menjadi kasar.

Sehingga dengan fungsi utama yaitu memberikan kekasaran pada *work rolls* tersebut, maka *system* kerja dari mesin shot blasting tersebut ialah: pasir besi sisa dari proses sebelumnya akan diambil dengan mesin *exhauster* untuk diletakkan di suatu tandon, setelah pasir besi berada tandon, maka pasir besi akan dipisahkan dengan debu dengan

menggunakan mesin sweco separator, hasil dari mesin sweco separator ialah pasir besi yang tidak ada debunya, sehingga pasir besi tersebut akan diangkat dan diletakkan diatas mesin shot blasting menggunakan mesin elevator , setelah berada ditandon, maka pasir besi tersebut akan dibedakan partikelnya lagi mulai dari yang lembut, sedang maupun kasar, dan setelah dipisahkan, maka dari pengontrol mesin akan menentukan type pasir besi yang dipilih untuk memberikan kekasaran pada work rolls. Setelah pemilihan pasir besi selesai, maka yang selanjutnya ialah menentukan car mana yang digunakan untuk mengangkut work rolls, setelah car terpilih, maka mesin akan bekerja dengan automatic, setelah itu memasukkan data yaitu nilai phases atau banyaknya car berjalan didalam mein. Setelah memasukkan phases, maka car akan masuk ke adalam mesin shot blasting, dengan perhitungan delay dari limit switch, maka pintu dari mesin shot blasting tersebut akan menutup sendiri, dan rotoblast siap proses dengan referensi data dari pasir besi yang telah ditentukan sebelumnya, setelah jumlah phases terpenuhi, maka mesin rotoblast akan off sendiri, dan setelah itu pintu dari mesin shot blasting kan membuka, dan car pembawa work rolls akan keluar, selama work rolls melewati pintu dari mesin shot blasting, maka kipas yang bertujuan untuk menghilangkan sisa – sisa dari pasir besi dan debu yang masih melekat pada work rolls akan menyala, dan setelah car sampai pada tempat semula, baru car yang satunya dapat dijalankan, dan selama car belum sampai pada tempat semula, maka car yang satunya tidak akan dapat proses.

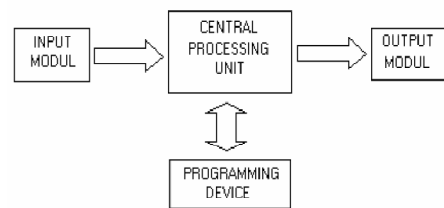
## 2.2 Programmable Logic Controller (PLC)

*Programmable Logic Controller (PLC)* adalah suatu peralatan elektronika digital yang dapat dilakukan pemrograman untuk menyimpan instruksi-instruksi dan melaksanakan fungsi khusus seperti logika, sekuensial, *timer*, *counter* dan aritmatika untuk kontrol mesin dan proses. Sebelum PLC telah banyak peralatan kontrol sekuensial, semacam *Cam Shaft* dan *Drum*. Ketika *relay* muncul, panel kontrol dengan *relay* menjadi kontrol sekuensial utama. Ketika transistor muncul, solid state *relay* diterapkan pada bidang yang *relay* elektromagnetik tidak cocok diterapkan, seperti kontrol dengan kecepatan tinggi. Sekarang sistem kontrol sudah meluas sampai keseluruhan pabrik dan sistem kontrol total dikombinasikan dengan kontrol *feedback*, pemrosesan data dan sistem monitor terpusat. Sistem kontrol logika konvensional tidak dapat melakukan beberapa kasus digital, dan *Programmable Logic Controller* diperlukan untuk itu. Sedangkan kini persaingan industri makin meningkat, efisiensi produksi secara umum dianggap sebagai kunci sukses.

PLC merupakan sistem yang dapat memanipulasi, mengeksekusi, dan atau memonitor keadaan proses pada laju yang amat cepat, dengan dasar data yang bisa diprogram dalam sistem berbasis mikroprosesor integral. PLC menerima masukan dan menghasilkan keluaran sinyal-sinyal listrik untuk mengendalikan suatu sistem. Dengan demikian besaran-besaran fisika dan kimia yang dikendalikan, sebelum diolah oleh PLC, akan diubah menjadi sinyal listrik baik analog maupun digital, yang merupakan data dasarnya. Karakter proses yang dikendalikan oleh PLC sendiri merupakan proses yang sifatnya bertahap, yakni proses itu berjalan urut untuk mencapai kondisi akhir yang diharapkan. Dengan kata lain proses itu terdiri beberapa subproses, dimana subproses tertentu akan berjalan sesudah subproses

sebelumnya terjadi. Istilah umum yang digunakan untuk proses yang berwatak demikian ialah proses sekuensial (*sequential process*). Sistem kontrol yang populer selain PLC, misalnya *Distributed Control System (DCS)*, mampu menangani proses-proses yang bersifat sekuensial dan juga kontinyu (*continuous process*) serta mencakup loop kendali yang relatif banyak.

Ada tiga komponen utama yang menyusun PLC yaitu, *Control Processing Unit (CPU)*, masukan/keluaran unit dan *Programming device*. Sedangkan komponen lain seperti *power supply*, *recorder player /tape atau disk*, *optional remote interconnection* dan *optional remote master computer*. CPU bekerja berdasarkan mikroprosesor yang bekerja menggantikan fungsi *relay*, *counter*, *timer* dan *sequencers*, sehingga *programmer* bisa membuat semua rangkaian yang menggunakan fungsi- fungsi *relay* diatas.



Gambar 2.1 Gambar Blok Diagram Programmable Controller

## 2.3 Omron Ethernet Card (TCP/IP)

Agar sebuah PC yang berisi software HMI/SCADA – Cimplicity dapat berkomunikasi dengan PLC Type CS1G-CPU44H-EVI, maka digunakan sebuah peripheral yang berguna sebagai perantara, yaitu Omron Ethernet Card (TCP/IP) Type CS1W ETN21. Pemilihan dari Ethernet dikarenakan system komunikasi menggunakan Ethernet (TCP/ IP) memiliki *networking area* yang cukup luas dan proses transfer data yang cukup cepat. Serta memungkinkan pengontrolan dari beberapa proses dalam satu system, yang tentunya lebih mudah, rapi dan hemat dalam segi pengkabelan/ *wiring-nya* dalam industry.



Gambar 2.2 Omron Ethernet Card Type CS1W – ETN21

Metode akses media dari type ini menggunakan *CSMA/CD*, *CSMA/CD* adalah protokol layer kedua pada *OSI* model. Protokol ini didesain untuk menyediakan akses yang adil untuk sebuah channel yang digunakan bersama-sama sehingga semua stasiun mendapat kesempatan untuk menggunakan *network*. Setelah setiap paket transmisi pada semua station menggunakan *CSMA/CD* protokol untuk menentukan stasiun mana yang berikutnya untuk menggunakan *channel ethernet*. *CSMA/CD (Carrier Sense Multiple Access/Collision Detection)* merupakan modifikasi dari *CSMA (Carrier Sense Multiple Access)* di mana terjadi peningkatan performa dengan menghentikan transmisi jika

terjadi tabrakan (*collision*) dan mengurangi terjadinya tabrakan lagi pada percobaan transmisi signal berikutnya.

Dengan menggunakan kabel Unshielded twisted-pair (UDP) dengan kategori 5, 5e atau kabel Shielded twisted-pair (STP) kategori: 100  $\Omega$  at 5, 5e. maka besar baudrate disini maksimal sampai 100 Mbit/s (100Base – TX), dan jika menggunakan kabel Unshielded twisted-pair (UDP) dengan kategori: 3, 4, 5, 5e atau kabel Shielded twisted-pair (STP) dengan kategori: 100  $\Omega$  at 3, 4, 5, 5e. maka besar baudrate disini maksimal hanya sampai 10 Mbit/s (10Base-TX).

#### 1.4 SOFTWARE PROFICY HMI/SCADA – CIMPLICITY 6.1

Proficy HMI/SCADA – Cimplicity 6.1 merupakan software SCADA yang bervendor Ge-Fanuc yang dapat digunakan untuk membuat *Visualisasi Human Machine Interface (HMI)*. Sehingga dapat melakukan otomatisasi dalam pengoperasiannya.

### III. PERANCANGAN SYSTEM

Salah satu segi yang perlu diperhatikan dalam pembuatan *Human Machine Interface* untuk mesin *shotblasting* ialah apa saja yang dibutuhkan oleh *operator* dan bagian mana saja yang perlu dikontrol dan informasi apa saja yang perlu ditampilkan pada *Human Machine Interface*. Semua hal yang harus dikontrol dan informasi yang ditampilkan pada *Human Machine Interface* hanya sebatas *push button*, *selector*, dan *ammeter* yang ada saat ini pada *thumb switch* yang terdapat pada *pulpit operator*. Jadi semuanya yang ada *thumb switch* akan digantikan ke *screen Human Machine Interface* sebagai pengontrol dan mendapatkan informasi dari mesin *shotblasting*. Dan semua aktifitas yang ada pada mesin *shotblast* dapat ditampilkan dengan animasi pergerakan pada *screen Human Machine Interface*.



Gambar 3.1 *thumb switch*

Dari gambar tersebut dapat dilihat apa saja yang perlu dikontrol dan *indicator* apa saja yang ditampilkan pada *screen Human Machine Interface*, dari *thumb switch* tersebut, yang perlu dikontrol dan *indicator* yang ditampilkan meliputi:

#### Push Button Pengontrol Mesin Shotblast

- PP1 Selection
- Automatic Cycle
- Door
- Work Roller Rotation
- Shaker

- Read Abrasive
- Abrasive Selection
- Exhauster, Sweco Spearator,Elevator, Screw Conveyor, Roof Fans
- Rotoblast Start Manual
- Car Control Manual
- Selection Car
- Phases

#### Parameter Input:

- Rotoblast Speed Adjust
- Car Fast Adjust
- Car Slow Adjust

#### Parameter Output

- Rotoblast RPM
- Rotoblast Amperemeter
- Car Speed

Dari kebutuhan yang diinginkan oleh *operator* bahwa hampir sama dengan yang ada pada *thumb switch* maka seluruh fungsi dari *thumb switch* digantikan pada *screen Human Machine Interface*. Dan penambahan seluruh alamat *Input* dan *Output* juga ditampilkan pada *screen* pada *I/O*.

### IV. HASIL PENGUJIAN

#### 4.1 KONEKSI

Untuk pengujian koneksi antara PC HMI dengan PLC *Omron type CSIG-CPU44H-EVI*, melalui *module ethernet card Omron TCP/IP type CS1W-ETN*, hal yang pertama ialah pastikan *Internet Protocol* dari kedua *module* ialah sama. Selanjutnya pastikan urutan jenis kabel yang dipakai ialah jenis *straight through*. Kemudian *ping* dan *netstat -n* dari PC HMI ke *Internet Protocol* dari PLC.

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2004 Microsoft Corp.
E:\Documents and Settings\Final project>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:
Reply from 192.168.1.2: bytes=32 time<ms TTL=255
Reply from 192.168.1.2: bytes=32 time<ms TTL=255
Reply from 192.168.1.2: bytes=32 time<ms TTL=255
Reply from 192.168.1.2: bytes=32 time<ms TTL=255

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
E:\Documents and Settings\Final project>netstat -n

Active Connections
Proto Local Address          Foreign Address        State
TCP        192.168.1.1049         192.168.1.2:9688      ESTABLISHED
E:\Documents and Settings\Final project>

```

Gambar 4.1 *ping* dan *netstat -n* dari PC HMI ke PLC

Jika tampilan seperti pada gambar diatas, maka komunikasi antara PC HMI dengan PLC berhasil, dan *module Ethernet card* dapat menerima data dari PC HMI, kemudian *module ethernet card* telah siap. Kemudian untuk pengujian koneksi antara *software Poficy HMI/SCADA – Cimplicity 6.1* dengan PLC ialah *run* program *software Poficy HMI/SCADA – Cimplicity 6.1*. dan kemudian jika pada *module ethernet card Omron TCP/IP type CS1W-ETN21* LED yang menyala ialah LED pada

- RUN
- SD
- RD
- LINK
- 100M
- TCP

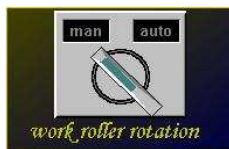


Gambar 4.2 pengujian koneksi antara PLC dan software Poficy HMI/SCADA – Cimplicity 6.1

Dari gambar 4.2 hasil pengujian koneksi antara PC HMI dengan PLC, terlihat bahwa LED yang menyala ialah LED pada informasi RUN, SD, RD, LINK, 100M, TCP. Jadi dapat disimpulkan bahwa antara PC HMI dengan PLC dapat berkomunikasi. Sehingga, antara software Poficy HMI/SCADA – Cimplicity 6.1 dengan PLC dapat terkoneksi dan berkomunikasi dengan baik.

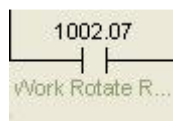
#### 4.2 PUSH BUTTON

Untuk pengujian *push button*, diambil *sample*, Sehingga pengujian *push button* yaitu diambil *sample* dari *push button work roller rotation*, sehingga hasilnya ialah:



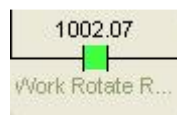
Gambar 4.3 Selector Work Roller

*Switch* ini digunakan untuk mengontrol *ladder* program PLC alamat 1002.07 yang digunakan untuk memilih putaran *work roller* secara *manual* dan *auto*. Sehingga kondisi awal ialah *selector* berada pada kondisi *manual*, dan kondisi dari *ladder* diagram dari program PLC ialah sebagai berikut:



Gambar 4.4 *ladder* diagram PLC alamat 1002.07 kondisi manual

Dan jika *selector* berada pada kondisi *auto*, maka kondisi dari *ladder* diagram dari program PLC ialah sebagai berikut:



Gambar 4.5 *ladder* diagram PLC alamat 1002.07 kondisi auto

#### 4.3 PENGIRIMAN DAN PENERIMAAN NILAI

Untuk pengujian pengiriman dan penerimaan nilai, diambil *sample*, Sehingga pengujiannya yaitu diambil *sample* dari *rotoblast speed adjust* dan *Rotoblast Amperemeter*, sehingga hasilnya ialah:

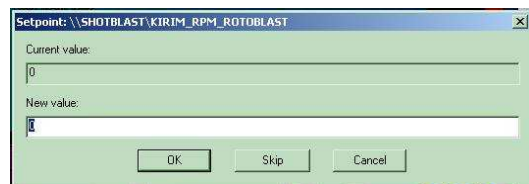
##### 4.3.1 Rotoblast Speed Adjust

Set point nilai antara 0 – 3000 dengan format data berupa data *integer*.



Gambar 4.6 Set Point Speed Rotoblast

Untuk memasukkan nilai kecepatan dari rotoblast itu sendiri, yaitu jika penambahan dapat dilakukan dengan cara mengeklik tombol UP, dan jika mengurangi maka dengan cara mengeklik tombol DOWN, selain itu dapat dengan cara langsung mengeklik nilai yang tertera di tombol, kemudian akan muncul *window* :



Gambar 4.7 Setpoint\\SHOTBLAST\KIRIM\_RPM\_ROTBLAST

Setelah *window* keluar seperti diatas, maka ketikkan nilai yang diinginkan di *New value*, kemudian klik OK, maka nilai kecepatan *rotoblast* akan mengeset ke nilai yang dimasukkan tersebut.

Nilai yang dimasukkan disini akan dikirim ke alamat *memory* dari PLC, ialah alamat DM400, sehingga data yang diperoleh ialah sebagai berikut:

Tabel 4.1 Data pengiriman *speed Rotoblast* ke DM400

| No | Data Masukan HMI | Data DM400 |
|----|------------------|------------|
| 1  | 1000             | 1000       |
| 2  | 1100             | 1100       |
| 3  | 1200             | 1200       |
| 4  | 1300             | 1300       |
| 5  | 1400             | 1400       |
| 6  | 1500             | 1500       |
| 7  | 1600             | 1600       |

|    |      |      |
|----|------|------|
| 8  | 1700 | 1700 |
| 9  | 1800 | 1800 |
| 10 | 1900 | 1900 |
| 11 | 2000 | 2000 |

Akan tetapi dengan data *integer* yang dikirim ke DM400 sebagai data yang akan diolah PLC sebagai *reference* untuk kecepatan dari *rotoblast* terjadi ketidak sesuaian kecepatan yang dimasukkan oleh HMI dengan kecepatan dari *rotoblast* yang terukur oleh *ammeter* yang terdapat pada *panel thumb switch*, sehingga perlu penyesuaian nilai masukan dari HMI agar nilai *speed* yang keluar di *thumb switch* sama dengan nilai masukan dari HMI. Penyesuaian disini menggunakan *system scalling* agar nilai masukan dari HMI sama dengan nilai kecepatan yang tertera pada *panel thumb switch*.

Tabel 4.2 Data masukan *speed Rotoblast* dari HMI dan yang terdapat pada *panel thumb switch*

| No | Data Masukan HMI | Nilai Thumb Switch |
|----|------------------|--------------------|
| 1  | 1000             | 1111               |
| 2  | 1500             | 1666               |
| 3  | 2000             | 2200               |
| 4  | 2500             | 2777               |
| 5  | 3000             | > 3000             |

Sehingga dari 5 data sample yang diambil, telah didapat nilai *scalling* ialah  $3000 / 3333 = 0.9$  atau  $9 / 10$  dari data masukan dari HMI, sehingga dari hasil data tersebut akan dikirim ke DM400 untuk diolah agar didapat nilai *speed* yang tertera pada *panel thumb switch* sama dengan masukan dari HMI. Proses *scalling* tersebut dibuat dengan menggunakan *script*, sehingga *script*nya sebagai berikut:

```

Sub Main()
  PrintSet "ROTOBLAST_SPEED_ADJUST", (9/10) * PointGet("KIRIH_RPM_ROTOLAST")
End Sub

```

Gambar 4.8 *Script* kirim\_rpm\_rotoblast

Sehingga nilai yang dikirim setelah proses *scalling* ialah sebagai berikut:

Tabel 4.3 Data pengiriman *speed Rotoblast* ke DM400 setelah *scalling*

| No | Data Masukan HMI | Data DM400 |
|----|------------------|------------|
| 1  | 1000             | 899        |
| 2  | 1100             | 989        |
| 3  | 1200             | 1079       |
| 4  | 1300             | 1169       |
| 5  | 1400             | 1259       |
| 6  | 1500             | 1349       |
| 7  | 1600             | 1439       |
| 8  | 1700             | 1529       |
| 9  | 1800             | 1619       |
| 10 | 1900             | 1709       |
| 11 | 2000             | 1799       |

Sehingga setelah proses *scalling*, nilai *reference speed* yang dikirim dari HMI dan nilai yang diinformasikan pada *panel thumb switch* sama.

Tabel 4.4 Nilai *reference speed Rotoblast* dari HMI dan informasi *speed rotoblast* pada *panel thumb switch*

| No | Reference speed HMI | Panel thumb switch |
|----|---------------------|--------------------|
| 1  | 1000                | 1000               |
| 2  | 1100                | 1100               |
| 3  | 1200                | 1200               |
| 4  | 1300                | 1300               |
| 5  | 1400                | 1400               |
| 6  | 1500                | 1500               |
| 7  | 1600                | 1600               |
| 8  | 1700                | 1700               |
| 9  | 1800                | 1800               |
| 10 | 1900                | 1900               |
| 11 | 2000                | 2000               |

Dari proses pengiriman nilai tersebut dapat dianalisa bahwa nilai yang diolah untuk *reference speed* untuk *rotoblast* harus *discalling* dahulu sebelum diolah didalam PLC, agar nilai *reference* yang dikirim ke *DC Drive* sama dengan kecepatan yang diterima atau ditampilkan pada *panel thumb switch*.

### 4.3.1 Car Speed

Nilai keluaran antara 0 – 100 dengan format data berupa data *integer*.



Gambar 4.9 Car Speed

Parameter ini berfungsi untuk menampilkan nilai keluaran dari *DC Drive* yang berupa nilai kecepatan dari jalannya *Car*, nilai disini dibatasi antara 0 sampai 100 m / min.

Parameter yang akan ditampilkan ke informasi ini diambil dari *memory PLC* dari alamat *DM502*, dan nilai data dari *DM502* bernilai antara 0 sampai 4000, sehingga perlu *scalling* agar data yang dikeluarkan antara 0 – 100 sesuai dengan yang ada di *panel thumb switch*.

Sehingga nilai yang akan dikeluarkan dalam informasi ini ialah hasil *scalling* 1 / 40 dari data *DM502*.

Proses *scalling* tersebut dibuat dengan menggunakan *script*, sehingga *scriptnya* sebagai berikut:

```

Sub Main()
PointSet "TERIMA_CAR_SPEED", (1/400) * PointGet("CAR_SPEED")

If PointGet("TERIMA_CAR_SPEED") > 10 Then
MsgBox "Awas... car Speed Melebihi Anbang Batas"
End If

End Sub

```

Gambar 4.10 Script terima\_MPM\_car

## V. PENUTUP

### 5.1 KESIMPULAN

Dari hasil implementasi *Human Machine Interface* ke *riil plant* yaitu mesin *shotblast* pada PT. Krakatau Steel (Persero) Tbk, maka dapat ditarik kesimpulan, meliputi:

- Untuk *push button* model *toggle*, digunakan jika tombol berfungsi mengaktifkan atau menonaktifkan mesin hanya pada satu tombol tersebut.
- Untuk *push button* model *trigger*, merupakan model tombol untuk memberi nilai 1 atau 0 sesaat sewaktu penekanan saja, dan jika penekanan

tersebut dilepas, maka nilai tersebut akan kembali seperti semula sebelum penekanan.

- Untuk *push button* model *setpoint*, merupakan model tombol yang digunakan untuk memberi nilai masukan 1 atau 0 sewaktu penekanan, dan jika tombol tersebut dilepas, maka nilai yang di set tersebut tetap pada alamat tersebut.
- Pengiriman nilai dari *Human Machine Interface* ke alamat *memory PLC* dapat terjadi jika *type* nilai data yang dikirim sama.
- Pada pengiriman nilai, perlu *scalling* nilai agar nilai yang dimasukkan buat *reference* sama besarnya dengan nilai yang terukur oleh *ammeter* dari *panel thumb switch*.
- Penerimaan nilai dari alamat *memory PLC* ke *Human Machine Interface* dapat terjadi jika *type* nilai data yang diterima sama.
- Pada penerimaan nilai, diperlukan *scalling* nilai agar nilai yang dikirim dari alamat *memory PLC* dapat ditampilkan oleh *Human Machine Interface* dan nilai yang ditampilkan pada *Human Machine Interface* sama dengan nilai yang ditampilkan oleh *ammeter* pada *panel thumb switch*.
- Dari 20 kali proses secara *auto*, terdapat 1 kali error pada jalannya *Car*, atau biasa disebut dengan *TRIP*.
- Metode *scalling* dibuat dengan menggunakan *Script*.

## 5.2 SARAN

Hal yang perlu diperhatikan dalam pengembangan *Human Machine Interface* ini lebih lanjut ialah penambahan *Data Base* untuk *report* yang diperlukan oleh *divisi*, dan perlu penambahan konfigurasi dari pada alamat *Internet Protocol* pada sisi *Software Proficy HMI/SCADA CIMPLICITY 6.1* jika *Human Machine Interface* ini ingin dimonitor dan di kontrol dengan mesin – mesin yang lain dalam ruang lingkup satu *divisi*.

## DAFTAR PUSTAKA

- [1] \_\_\_\_\_. *SHOTBLASTING INSTALLATION FOR WORK ROLLS*. Pangborn Europe S.p.A Caronno Pertusella (Varese), Italia,1984. Sheet: 6/7
- [2] \_\_\_\_\_. *SHOTBLASTING INSTALLATION FOR WORK ROLLS*. Pangborn Europe S.p.A Caronno Pertusella (Varese), Italia,1984. Sheet: 9/10
- [3] \_\_\_\_\_. *SHOTBLASTING INSTALLATION FOR WORK ROLLS*. Pangborn Europe S.p.A Caronno Pertusella (Varese), Italia,1984
- [4] \_\_\_\_\_. *Getting Started - Manual Guide Proficy HMI/SCADA – Cimplicity 6.1*. Ge-Fanuc, Germany. 2004
- [5] \_\_\_\_\_. *CIMPLICITY Monitoring and Control Products. Base System, Device Communications Manual*. GE Fanuc Automation, Germany. 2001
- [6] \_\_\_\_\_. *SYSMAC CS-Series Ethernet Unit CS1W-ETN21*. OMRON Corporation Industrial Automation Company, Japan. 2009
- [7][http://en.wikipedia.org/wiki/Carrier\\_sense\\_multiple\\_access\\_with\\_collision\\_detection](http://en.wikipedia.org/wiki/Carrier_sense_multiple_access_with_collision_detection)
- [8] <http://www.ge-ip.com/products/2819>