

PERANCANGAN SISTEM TELEMETRI DAN KENDALI JARAK JAUH PERANTI VARIABLE SPEED DRIVE VIA PROTOKOL MODBUS BERBASIS APLIKASI VIRTUINO PADA ANDROID

Beny Firman¹, Teguh Hariyadi², Muhammad Suyanto³

^{1,2,3}Jurusan Teknik Elektro, Fakultas Teknologi Industri

Institut Sains & Teknologi AKPRIND Yogyakarta

Jalan Kalisahak 28 Komplek Balapan Tromol Pos 45 Yogyakarta 55222, D.I. Yogyakarta, Indonesia

¹benyfirman@akprind.ac.id, ²hariyaditeguh99@gmail.com, ³musyan@gmail.com

Abstrak

Motor induksi 3 fasa banyak digunakan pada industri, motor tersebut memiliki kekurangan yaitu kecepatan yang konstan, kekurangan ini dapat diatasi menggunakan VSD (Variable Speed Drive) sehingga kecepatan motor dapat dikendalikan, umumnya VSD dikendalikan menggunakan PLC pada sistem DCS (Distributed Control System) yang terhubung melalui protokol Modbus. Untuk membangun sistem DCS umumnya diperlukan PLC serta PC sebagai antarmuka yang diletakkan pada satu ruang kontrol, sehingga sistem ini kurang fleksibel karena operator terpaksa pada satu ruang kontrol, selain itu untuk membangun sistem tersebut diperlukan biaya yang tinggi. Pada penelitian ini PLC dan PC digantikan dengan platform arduino dan android, sehingga biaya untuk membangun sistem lebih murah dan penggunaan android membuat operator lebih fleksibel, selain itu sifat arduino yang open source menciptakan peluang untuk pengembangan lebih lanjut, arduino terhubung ke VSD melalui protokol modbus menggunakan modul RS485 dan terhubung ke jaringan menggunakan modul ethernet. Arduino dapat berkomunikasi dengan VSD melalui protokol modbus dengan baudrate 4800 bps, format 8-E-1, kode fungsi 03 untuk telemetry dan kode fungsi 16 untuk telecontrol, arduino dapat terhubung ke modul ethernet melalui protokol SPI dengan dengan OP-Code 0F/F0 dan 32 bit data per frame, nilai error rata-rata antara kecepatan tertampil pada virtuino dan terukur adalah sebesar 1,02% untuk kondisi forward serta 1% untuk kondisi reverse.

Kata Kunci: Arduino, Modbus, VSD (Variable Speed Drive).

Pendahuluan

Motor induksi 3 fasa merupakan motor listrik yang paling banyak digunakan pada industri (Naima Hamad, 2016). Alasan banyak dipergunakannya motor induksi 3 fasa antara lain karena konstruksinya yang sederhana, mudah dalam perawatan serta harga yang relative murah. Akan tetapi motor induksi 3 fasa memiliki kelemahan dari segi pengendalian kecepatan karena pada umumnya motor induksi 3 fasa bekerja pada kecepatan yang konstan sesuai dengan kecepatan nominal pada name plate motor tersebut. Selain itu, untuk dapat merubah arah putar motor induksi 3 fasa membutuhkan MC (*magnetic contactor*) untuk menukar salah satu fasa yang masuk ke motor.

Pada era kemajuan teknologi industry saat ini, motor induksi tiga fasa telah dapat dikendalikan kecepatan putarnya, Pengendalian kecepatan putar tersebut menggunakan perangkat VSD (*Variable Speed Drive*), aplikasi VSD pada motor induksi dapat meningkatkan variasi kecepatan (Muhamad Aiman MuhamadAzmi, 2014). Selain memudahkan pengendalian kecepatan putar motor induksi tiga fasa, VSD juga dapat dipergunakan untuk merubah arah putar motor tanpa membalik salah satu sumber fasa yang masuk ke motor, peranti VSD pun dapat dikendalikan secara local maupun remote.

DCS (*Distributed Control System*) merupakan peralatan instrumen yang berfungsi untuk mengontrol dan memonitor jalannya suatu proses pada plant dengan jarak yang jauh (Kusumo, 2010). Teknologi DCS yang sudah ada masih memiliki kekurangan yaitu membutuhkan biaya yang tinggi pada saat membangun sistem tersebut, karena pada umumnya pada sistem DCS memerlukan PLC (*Programmable Logic Controller*), computer sebagai HMI (*Human Machine Interface*) dan software HMI, umumnya perangkat PLC dan VSD terhubung melalui komunikasi Modbus.

Modbus merupakan standar protocol komunikasi pada bidang industri, serta banyak dipergunakan untuk menghubungkan perangkat industri. Modbus memungkinkan perangkat berkomunikasi dalam bus yang sama, hanya terdapat satu master device dan dapat memiliki slave device hingga 247 pada bus yang sama (Kuang, 2014).

Transaksi data pataprotokol Modbus selalu diawali dengan query dari Master, dan Slave memberikan respons dengan mengirimkan data atau melakukan aksi sesuai perintah dari Master. Master hanya melakukan satu komunikasi dalam satu waktu. Slave hanya akan melakukan komunikasi jika ada perintah (query) dari Master dan tidak bisa melakukan komunikasi antar Slave, dalam jaringan komunikasi modbus harus memiliki format data dan kecepatan transfer (baudrate) yang sama (AgusTiyono, 2007).

Pada penelitian ini PLC akan digantikand engan platform Arduino Uno, computer digantikand engan android serta software HMI digantikan dengan aplikasi virtuino sehingga diharapkan dapat membuat sistem telecontrol dan telemetry yang lebih ekonomis dari sistem sebelumnya. Selain itu sifat arduino yang open source sehingga dapat dipergunakan oleh setiap orang dan dapat di modifikasi secara bebas dari segi software maupun hardware sehingga dapat menciptakan peluang untuk pengembangan lebih lanjut.

Metodologi

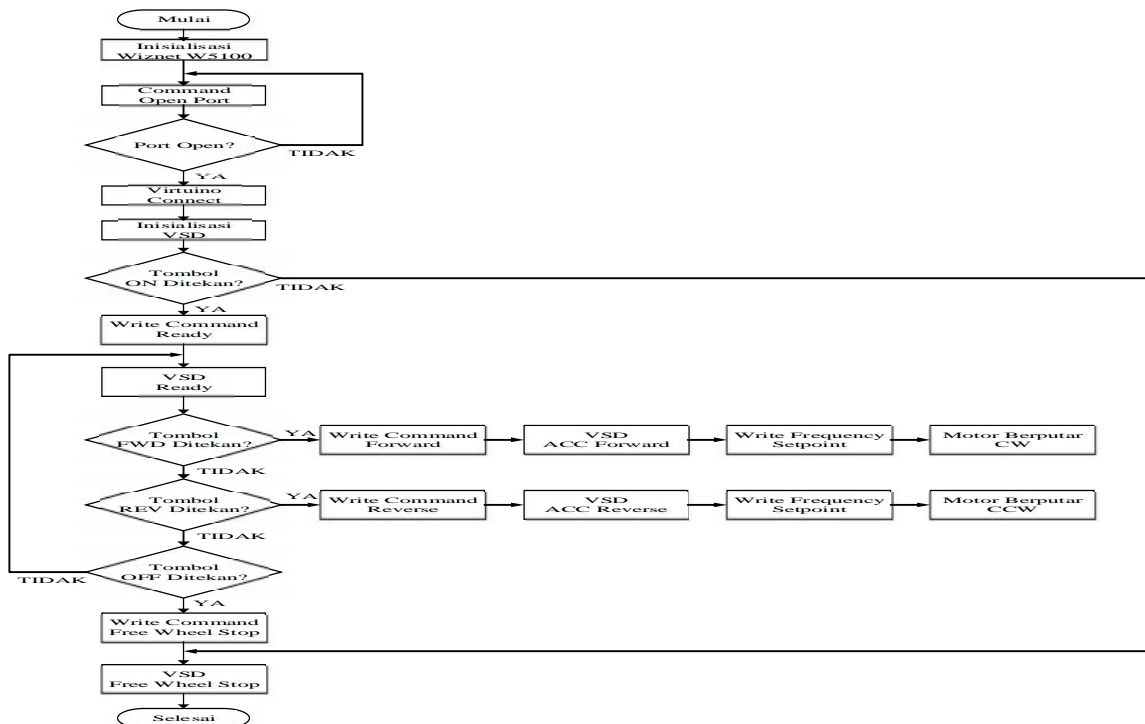
Pada bagian ini menjelaskan urutan proses-proses metodepenelitian yang digunakanmeliputi:

Alat dan Bahan

Pada bagian ini, menjelaskan semua alat yang digunakan dalam penelitian dan bahan-bahannya.

Tabel 1. Daftar alat yang dipergunakan

No	Nama Alat	Spesifikasi
1	Laptop	HP Pavilion dv2, Win 7, AMD Athlon Neo 1,6Ghz
2	USB Logic Analyzer	8 Channel 24 Mhz
3	Panel VSD	Altival12 0,37kW 0,5 HP 200-240V
4	Motor Induksi 3 fasa	Western Electric 0,37kW, CosQ 0,75, 220/380V
5	Multitester	Krisbow KW06-272
6	Router	TL-MR3420 300Mpbs
7	Tachometer	Krisbow KWD6-563, Range 2-20.000 Rpm
8	USB to RS485 Converter	
9	Obeng Set	Jackly



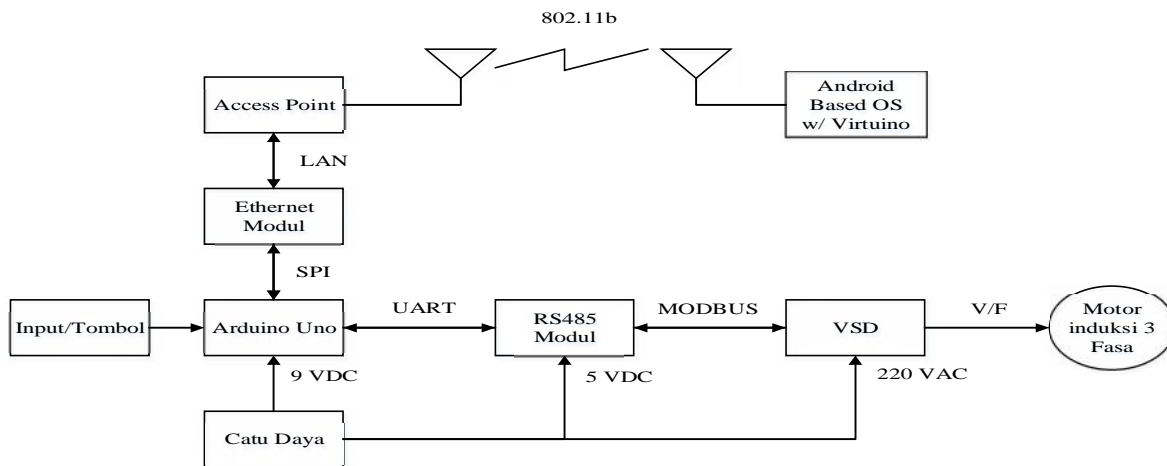
Gambar 1. Diagram alur program

Tabel 2. Daftar bahan yang diperlukan

No	Nama Bahan	Spesifikasi
1	Smartphone	Samsung J1 Ace Android V4.4.4 Kitkat Ram 0,75gb, Rom 4gb
2	Arduino	Arduino Uno R3 Atmega 328 7-12 Volt
3	Ethernet Shield	Hanrun HR911105A Wiznet W5100
4	RS485 Shield	Funduino V5 Max485
5	Kabel Jumper	20 cm
6	Power Supply	9 Volt DC, 1 Ampere

Perancangan Sistem

Pada bagian ini akan menjelaskan perancangan dari sistem yang dibuat, perancangan yang pertama adalah dari segi software atau perangkat lunak yaitu bahasa pemrograman yang akan di upload ke platform arduino, perancangan yang kedua dari segi hardware atau perangkat keras yaitu menghubungkan beberapa komponen supaya menjadi satu kesatuan sistem.



Gambar 1. Diagram alur perancangan sistem

Analisis dan pembahasan

Pada bagian ini berisi tentang analisis dari penelitian yang penulis lakukan, dan pembahasan tentang penelitian yang telah di kerjakan.

Pengujian catu daya

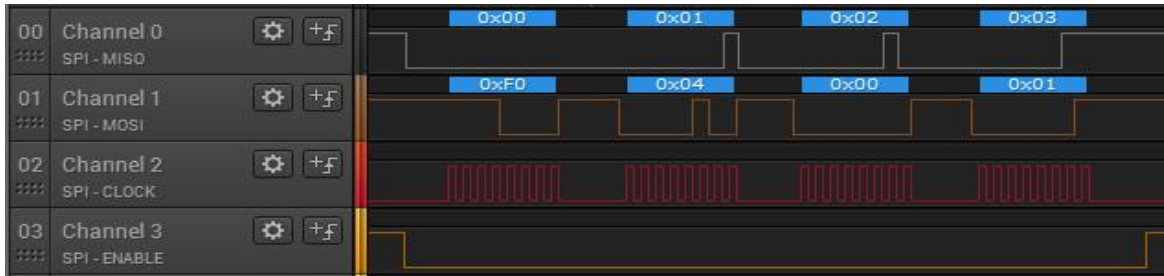
Pengujian catu daya dengan cara mengukur nilai tegangan keluaran dari power supply 9V DC serta mengukur arus yang dikonsumsi oleh kontroler.

Tabel 3. Pengujian catu daya

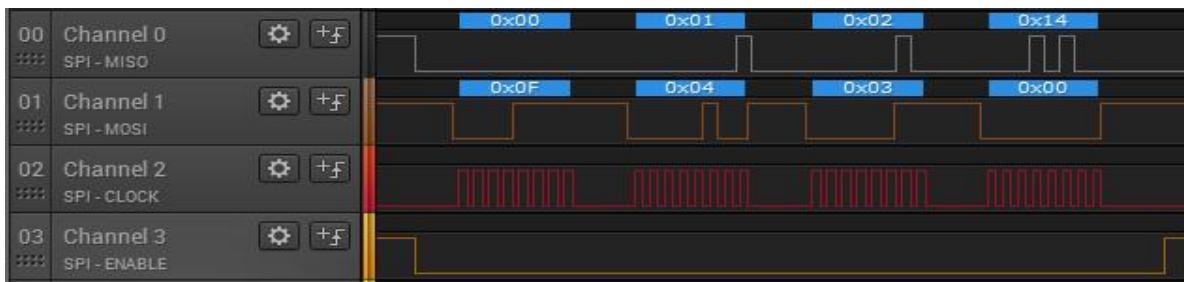
Pengujian	Ke-1	Ke-2	Ke-3	Ke-4	Ke-5	Rata-rata
TeganganPengukuran (V)	8,95	8,95	8,96	8,96	8,96	8,956
TeganganSpesifikasi (V)	9	9	9	9	9	9,00
Error Tegangan (%)	0,56	0,56	0,44	0,44	0,44	0,488
KonsumsiArus (mA)	220	230	230	230	220	226,00

Pengujian Protokol SPI

Pengujian protokol SPI pada penelitian ini dilakukan dengan menggunakan usb *logic analyzer* dan *software* saleae logic yang telah di set untuk men-decode SPI, Atmega328 pada arduino bertindak sebagai *master* dan chip wiznet w5100 pada modul *ethernet* sebagai *slave*



Gambar 1. Write command pada protokol SPI



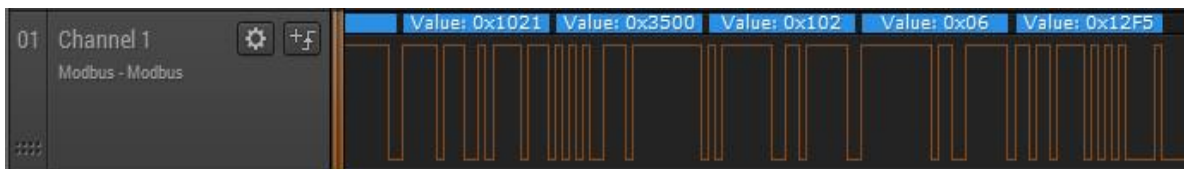
Gambar 2. Read value pada protokol SPI

Pada Gambar 1 memperlihatkan bahwa master mengirim OP-Code F0 yang berarti master akan menulis value, alamat register yang akan ditulis adalah register 0400 yang merupakan S0_MR (Socket 0 Mode Register), nilai yang ditulis adalah 01 yang mempunyai arti bahwa mode yang digunakan adalah TCP.

Pada Gambar 2 memperlihatkan bahwa master mengirim OP-Code 0F yang berarti master akan membaca nilai suatu register, alamat register yang nilainya akan dibaca adalah 0403 yang merupakan S0_SR (Socket 0 Status Register), nilai yang dibaca adalah 14 yang berarti bahwa socket 0 pada kondisi listen.

Pengujian Protokol Modbus

Pengujian protokol Modbus pada penelitian ini dilakukan dengan menggunakan usb *logic analyzer* dan *software* saleae logic, pada komunikasi serial RS485 jalur A dan B masing-masing dihubungkan dengan probe usb *logic analyzer*, kemudian mengambil sampling komunikasi yang terjadi menggunakan *software* saleae logic yang telah di set untuk men-decode protokol modbus, arduino bertindak sebagai *master* sedangkan VSD sebagai *slave* yang telah di set dengan *slave id* 1, baudrate 4800 bps dan format 8-E-1.



Gambar 3. Request 01 10 2135 0001 02 0006 12F5



Gambar 4. Respon 01 10 2135 0001 1BFB



Gambar 5. Request 01 03 0C81 0001 D772



Gambar 6. Respon 01 02 0231 78F0



Gambar 7. Responses Exception

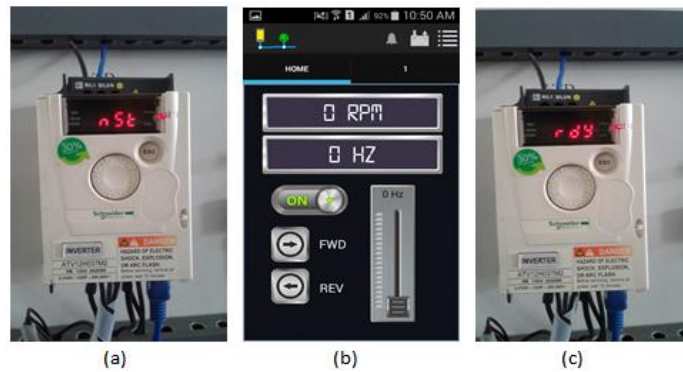
Pada Gambar 3 memperlihatkan bahwa slave id 01, function code 10 (16 = write multiple register), start address register 2135 (8501 = control word frequency), length 0001, byte count 02. value 0006 dan CRC 12F5. Pada Gambar 4 memperlihatkan bahwa slave id 01, function code 10(16 = write multiple register), start address register 2135 (8501 = control word frequency), number register written 0001, CRC 1BFB. Pada Gambar 5 memperlihatkan bahwa slave id 01, function code 03 (03 = read holding register), start address register 0C81 (3201 = status word frequency), length 0001 dan CRC D772.

Pada Gambar 6 memperlihatkan bahwa slave id 01, function code 03 (03 = read holding register), byte count 02, value 0231 dan CRC 78F0. Pada Gambar 7 exception respon didapatkan dengan cara menulis suatu nilai ke register status word frequency yang sesuai spesifikasi hanya dapat dibaca nilainya sehingga terjadilah exception responses, penulisan data dilakukan dengan bantuan software simply modbus master V8.0.6.

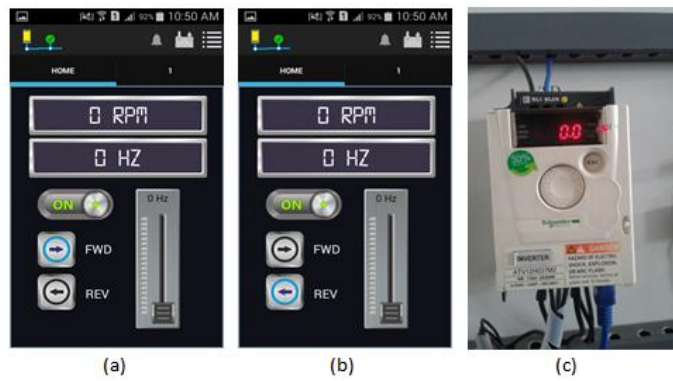
Dari exception responses memperlihatkan bahwa slave id 01, exception responses terjadi pada function code 10 (16 = write multiple register), exception responses ditandai dengan penambahan nilai 80 heksa pada function code lalu di ikuti dengan exception code, exception code 06 (slave device busy) hal ini dikarenakan nilai pada status word frequency selalu dibaca oleh peranti VSD, CRC CC02.

Pengujian Aplikasi Virtuino

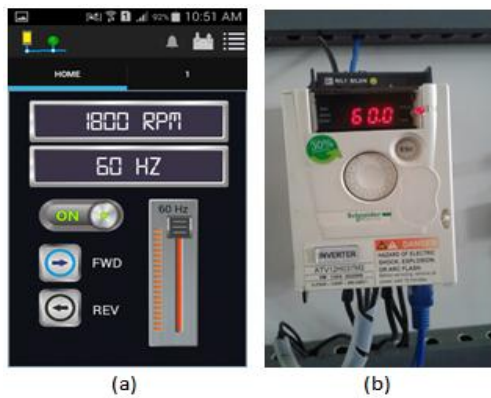
Pengujian aplikasi virtuino dilakukan dengan menekan tombol pada aplikasi virtuino lalu melihat apakah peranti *variable speed drive* merespon terhadap penekanan tombol pada virtuino



Gambar 8. (a) VSD Status NST (b) Tombol ON Ditekan (c) VSD Status RDY



Gambar 9. (a) Tombol FWD Ditekan (b) Tombol REV Ditekan (c) VSD Running 0 Hz



Gambar 10. (a) Slider Posisi 60 Hz (b) VSD Running 60 Hz

Pada Gambar 8 (a) memperlihatkan kondisi awal VSD sebelum tombol ON pada virtuino ditekan, Gambar 8 (b) memperlihatkan kondisi aplikasi virtuino saat tombol ON ditekan dan Gambar 8 (c) memperlihatkan kondisi VSD setelah tombol ON pada virtuino ditekan.

Pada Gambar 9 (a) memperlihatkan tombol FWD pada virtuino ditekan, Gambar 9 (b) memperlihatkan tombol REV pada virtuino ditekan dan Gambar 9 (c) memperlihatkan kondisi VSD pada saat salah satu tombol FWD/REV ditekan yaitu Running dengan frekuensi 0 Hz.

Pada Gambar 10 (a) memperlihatkan bahwa slider pada aplikasi virtuino dinaikkan ke nilai maksimum yaitu sebesar 60 Hz dan Gambar 10 (b) memperlihatkan kondisi VSD setelah slider pada aplikasi virtuino dinaikkan ke nilai 60 Hz, peranti VSD juga memperlihatkan nilai 60 Hz

Pengujian Kecepatan Putar Motor

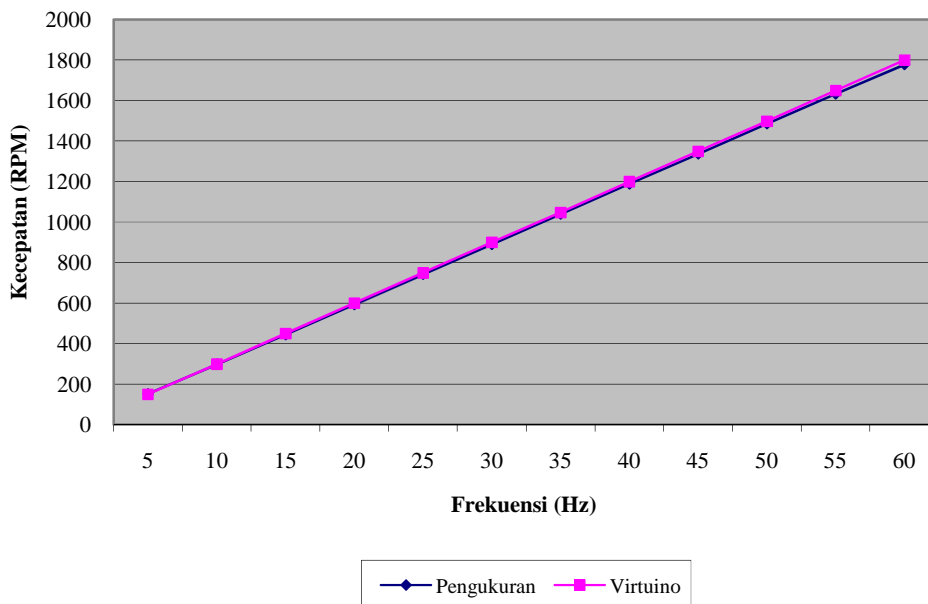
Pada bagian ini akan menampilkan selisih data yang di dapat dari pengukuran kecepatan putar motor induksi 3 fasa menggunakan tachometer dengan kecepatan yang ditampilkan oleh aplikasi virtuino.

➤ **Kondisi Forward**

Tabel 1. Kecepatan tertampil dan terukur kondisi forward

No	Set Frekuensi Virtuino (Hz)	Kecepatan Pengukuran (RPM)	Kecepatan Virtuino (RPM)	Error (%)
1	5	151,0	149	1,34
2	10	296,8	299	0,73
3	15	444,2	449	1,07
4	20	592,5	600	1,25
5	25	741,6	750	1,12
6	30	890,3	900	1,08
7	35	1039,4	1048	0,82
8	40	1188,0	1199	0,92
9	45	1336,9	1349	0,90
10	50	1485,8	1499	0,88
11	55	1632,7	1649	0,99
12	60	1777,8	1800	1,23

Perbandingan Kecepatan Tertampil dan Terukur



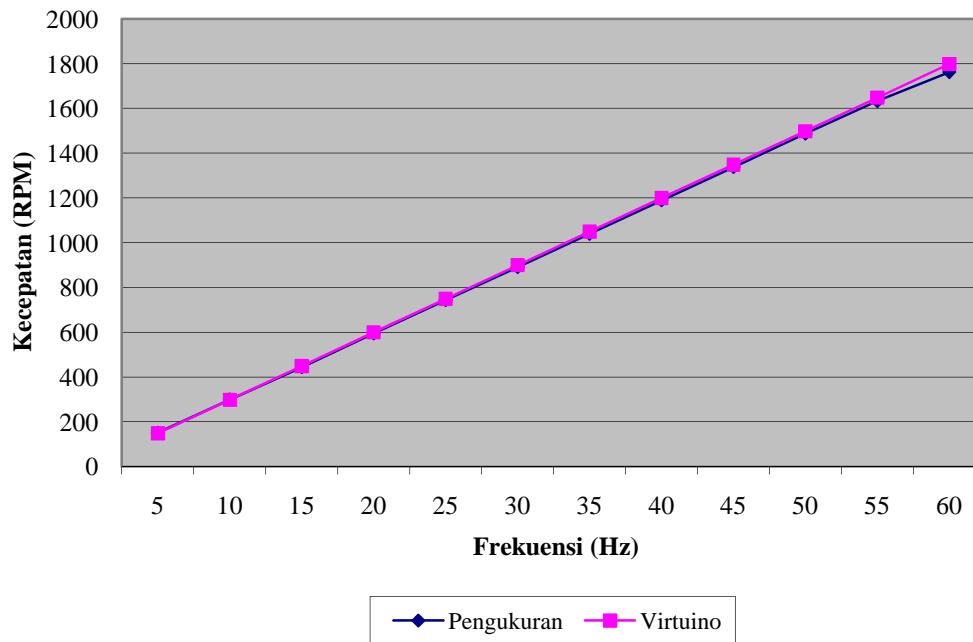
Gambar 11. Grafik perbandingan kecepatan pada kondisi forward

➤ **Kondisi Reverse**

Tabel 2. Kecepatan tertampil dan terukur kondisi reverse

No	Set Frekuensi Virtuino (Hz)	Kecepatan Pengukuran (RPM)	Kecepatan Virtuino (RPM)	Error (%)
1	5	152,5	149	2,35
2	10	299,7	299	0,23
3	15	445,4	450	1,02
4	20	594,6	600	0,90
5	25	743,8	749	0,69
6	30	892,5	900	0,83
7	35	1041,3	1050	0,83
8	40	1189,6	1200	0,87
9	45	1339,1	1349	0,73
10	50	1488,0	1499	0,73
11	55	1633,4	1648	0,88
12	60	1763,5	1799	1,97

Perbandingan Kecepatan Tertampil dan Terukur



Gambar 11. Grafik perbandingan kecepatan pada kondisi forward

Kesimpulan

Arduino uno dapat terhubung ke VSD altivar12 menggunakan chip max485 pada modul RS485 melalui protokol modbus dengan parameter serial yang sama yaitu format 8-E-1 dan baudrate sebesar 4800 bps. Komunikasi serial pada protokol modbus antara arduino uno dengan VSD altivar12 menggunakan *function code 3 (read holding register)* sebagai *telemetry* dan *function code 16 (write multiple register)* sebagai *telecontrol*. Sistem transaksi data protokol modbus pada *function code 3* menggunakan format 1 Byte *slave id*, 1 Byte *function code*, 2 Byte *start address register*, 2 Byte *length* di akhiri dengan 2 Byte CRC untuk query dan 1 Byte *slave id*, 1 Byte *function code*, 1 Byte *byte count*, 2 Byte *value* di akhiri 2 Byte CRC untuk respon. Sedangkan pada *function code 16* menggunakan format 1 Byte *slave id*, 1 Byte *function code*, 2 Byte *start address register*, 2 Byte *length*, 1 Byte *byte count*, 2 Byte *value* di akhiri 2 Byte CRC

untuk query dan 1 Byte *slave id*, 1 Byte *function code*, 2 Byte *length*, 2 Byte *byte written value* diakhiri 2 Byte CRC untuk respon.

DaftarPustaka

- Agus Tiyono, S. I. (2007). Sistem Telekontrol SCADA Dengan Fungsi Dasar Modbus Menggunakan Mikrokontroler AT89S51 Dan Komunikasi Serial RS485. *Jurnal Universitas Diponegoro*, 2.
- Kuang, Y.-C. (2014). Communication Between PLC And Arduino Based On Modbus Protocol. *IEEE Journal*, 370-371.
- Kusumo, F. C. (2010). Perancangan Dan Realisasi Sistem Kontrol Proses Menggunakan Distributed Control System Centum CS3000 Melalui Jaringan. *Jurnal Institut Teknologi Sepuluh Nopember*.
- Muhamad Aiman Muhamad Azmi, H. D. (2014). Three-Phase Variable Speed Drive Using ARM Cortex-M3. *IEEE Journal*.
- Naima Hamad, K. F. (2016). An Investigation Of Electrical Motor Parameters In A Sensorless Variable Speed Drive For Machine Fault Diagnosis. *IEEE Journal*.