# Implementasi Motor Stepper Bipolar Pada Arm Manipulator 6 DoF Dengan Pengendali Master Controller

Riska Wulandari1, Nama Penulis 22, Nama Penulis 33

1 Afiliasi Penulis 1

2 Afiliasi Penulis 2

3 Afiliasi Penulis 3

Email: email\_penulis\_pertama@afiliasi.ac.id (Hanya email penulis pertama)

|  |  |  |
| --- | --- | --- |
| **Informasi Artikel:** |  | **ABSTRAK** |
| *Received*:  DD MM YYYY  *Accepted*:  DD MM YYYY  *Available*:  DD MM YYYY | *Arm Manipulator* adalah sebuah robot yang menggunakan motor penggerak dengan bentuk seperti tangan untuk mempermudah pekerjaan manusia di bidang industry. Pada Proyek Akhir ini dibuat sebuah Arm Manipulator 6 DoF atau dengan kata lain adalah lengan robot yang memiliki 6 derajat kebebasan. Arm manipulator yang dibuat menggunakan penggerak motor stepper yang dikendalikan oleh joystick yang disebut dengan master controller. Pengendaliannya dilakukan secara remote/jarak jauh menggunakan komunikasi serial. Dalam hal ini, dilakukan pengujian bertujuan untuk mengetahui bagaimana implementasi stepper motor pada arm manipulator 6 DoF serta mengetahui keakurasian motor stepper berdasarkan pulsa yang diberikan. Hasil yang didapat dari pengujian tersebut yaitu error pada *joint* 1 sebesar 0,27%, *joint* 2 sebesar 0,4%, *joint* 3 sebesar 0,13%, *joint* 4 sebesar 0,09%, *joint* 5 sebesar 0,15%, dan *joint* 6 sebesar 0,03%. Dari pengujian tehadap keenam *joint* tersebut didapat rata-rata error yaitu sebesar 0,14%. |
| **Kata Kunci:** | ***ABSTRACT*** |
| Arm Manipulator  Master Controller  Motor Stepper | *Arm Manipulator is a robot that uses a motor with a shape like a hand to facilitate human work in the industrial sector. In this final project, a 6 DoF Arm Manipulator or in other words is a robotic arm that has 6 degrees of freedom. The arm manipulator is made using a stepper motor driven by a joystick called the master controller. Control is done remotely using serial communication. In this case, testing aims to determine how the implementation of the stepper motor on the 6 DoF arm manipulator and to determine the accuracy of the stepper motor based on the pulse given. The results obtained from the test are that the error at joint 1 is 0.27%, joint 2 is 0.4%, joint 3 is 0.13%, joint 4 is 0.09%, joint 5 is 0.15%, and joint 6 by 0.03%. From the testing of the joint, an average error of 0.14% was obtained.* |

jtrm.polman-bandung.ac.id

## Pendahuluan

Teknologi robot mengalami suatu kemajuan yang sangat pesat dewasa kini. Robot telah menggantikan peralatan-peralatan manual yang membutuhkan banyak tenaga manusia. Dalam dunia industri, robot merupakan salah satu alat bantu yang dalam kondisi tertentu sangat diperlukan. Terdapat beberapa kondisi tertentu dalam bidang industri yang tidak mungkin ditangani oleh manusia. Robot memiliki banyak kelebihan yang tidak dimiliki manusia, yaitu menghasilkan kualitas yang sama ketika mengerjakan suatu pekerjaan secara berulang–ulang, tidak mudah lelah, dan dapat diprogram ulang sehingga dapat difungsikan untuk beberapa tugas yang berbeda. Dan diantara robot yang sering digunakan dalam bidang industri adalah *Arm Manipulator*. Penerapan *Arm Manipulator* banyak ditemukan di industri, seperti mengelas, mengecat pada industri otomotif dan *packaging* dan *filling* pada industri kimia.

Pada tugas akhir ini, dibuatlah sebuah *Arm Manipulator* yang memungkinkan untuk membantu pekerjaan manusia. Arm Manipulator yang dibuat memiliki 6 degree of freedom (DOF), pada masing– masing DOF digerakan oleh sebuah motor stepper yang dikendalikan oleh mikrokontroler Arduino Mega. *Arm manipulator* dikendalikan oleh semacam pendantyang disebut dengan *master controller*. Pengendaliannya dilakukan secara remote/jarak jauh menggunakan komunikasi serial. *Master controller* memiliki kurang lebih 7 sensor yang terpasang padanya yang masing-masing mengendalikan aktuator yang berada pada *joint arm manipulator.* Pada *master controller* pun terdapat banyak tombol yang diperuntukkan untuk berbagai fitur seperti *scalling* pergerakan*, start, stop,* pergerakan translasi, dan lain-lain.

Dalam rangka sedikit sumbangsih untuk kemajuan teknologi pada industry di Indonesia dan dengan tujuan pembelajaran bagi penulis serta generasi selanjutnya. Dirancang lah proyek akhir ini yang merupakan lengan robot 6 DOF yang dikendalikan oleh master controller. Dikontrol menggunakan 2 Arduino Mega yang dikomunikasikan secara serial dengan tampilan *interface* berupa LCD *display* untuk menampilkan posisi dari lengan robot.

## METODOLOGI PENELITIAN

Pertama penulis membuat identifikasi masalah, kemudian membuat tuntutan hasil perancangan yang akan dicapai, setelah itu membuat konsep sistem perancangan. Setelah terkonsep secara keseluruhan, jika sistem yang dibuat belum sesuai, maka dilakukan pengkonsepan ulang, jika sistem sudah sesuai, maka dilakukan instalasi dan pengujian sistem. Jika sistem yang sudah di uji belum sesuai, maka dilakukan pengujian ulang, jika sistem yang sudah diuji sesuai, maka sistem sudah baik untuk di implementasikan.

### Alat dan Bahan

Teknologi yang digunakan para proyek akhir ini adalah:

1. Interfacing : LCD Display 20x4
2. Proses : Microcontroller - Arduino Mega 2560

Perangkat keras:

1. Motor Stepper : a. Nema 8 (8HS11-0204S)

b. Nema 11 (28HD1411-02)

c. Nema 17 (17HS4401)

d. Nema 23 (5718HB2401)

1. Driver Motor Stepper : a. A4988 Driver Stepper Module

b. TB6600 Driver Stepper Module

1. Input Digital : a. Push Button

b. Saklar Toggle

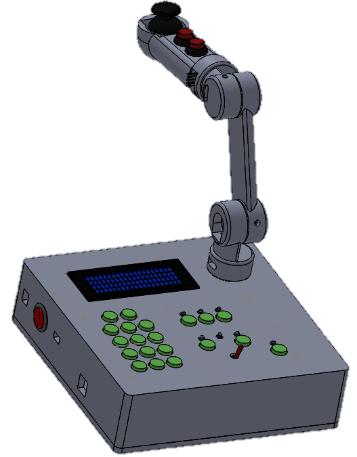
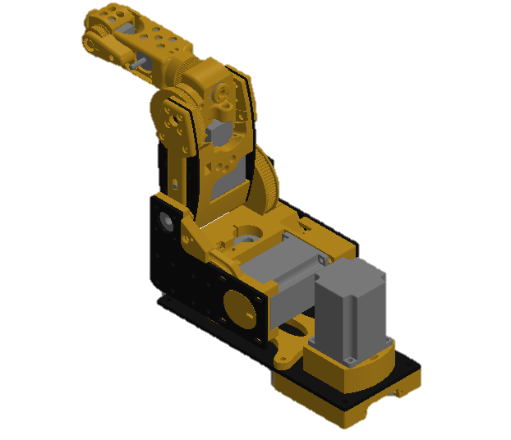
1. Output Digital : LED
2. Sensor : Potensiometer 100K Ohm

Perangkat Lunak:

1. Arduino IDE
2. Solidworks 2020
3. Eagle PCB Design

### Prinsip Kerja

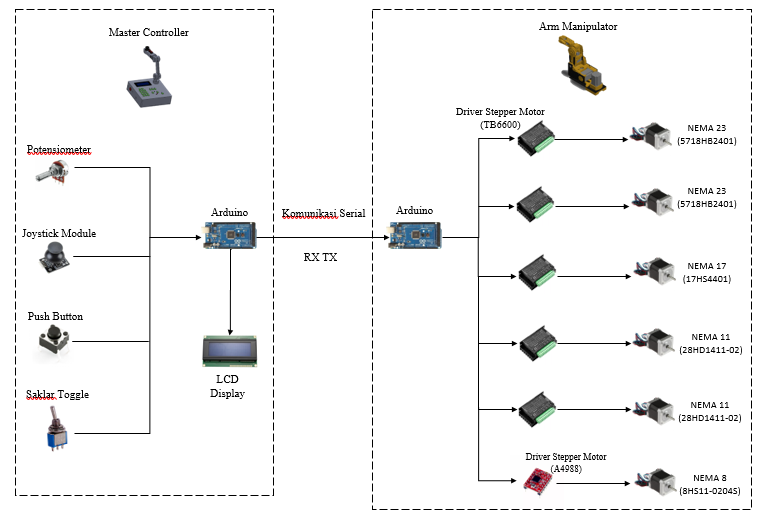
Secara hardware, sistem proyek akhir ini terdiri dari *arm manipulator* *6* *DOF* dan *master controller*. *Arm manipulator 6 DOF* merupakan robot berbentuk lengan yang memiliki 6 sendi pergerakan (*joint*). Sedangkan *master controller* adalah sebuah pendant yang mengendalikan pergerakan dari *arm manipulator*. Pergerakan *joint* pada *manipulator* dihasilkan dari pembacaan sensor *joystick* yang terpasang pada *master controller*. Satu *joint* pada sensor *joystick* akan mengendalikan satu *joint* pada *arm manipulator*. *Master controller* akan menghasilkan data derajat dari pembacaan sensor. Data sensor tersebut akan dikirim ke *Arduino slave* untuk dikonversi menjadi data *pulse* untuk menggerakkan stepper motor.



*Gambar 2. 17 Arm Manipulator 6 Dof dan Master Controller*

Secara teknis lengan robot dikendalikan oleh Master *controller* dan data hasil pembacaan yang sudah berupa besaran sudut akan ditampilkan pada display LCD sekaligus mengirimkan data secara serial kepada *Slave Controller Arduino mega* untuk pemprosessan data selanjutnya.

*User* menggerakkan lengan robot secara manual dengan menggerakkan *joystick* yang terdapat pada *master controller.* Kemudian pergerakkan yang terbaca oleh potensiometer berupa data diolah oleh program dan ditampilkan kedalam bentuk sudut.



*Gambar 2. 18 Gambaran Umum Sistem Arm Manipulator dan Mater Controller*

Terlihat pada Gambar 2.18 diatas *master controller* dibangun atas berbagai inputan komponen dan sensor, adapun inputan komponennya terdiri dari Potensiometer, modul *Joystick, Push button*, dan saklar Toggle. Data yang dihasilkan kemudian akan di proses oleh Arduino (master) sebagai controller. Lalu data akan di tampilkan melalui LCD sekaligus dikirim melalui komunikasi serial UART.

Data yang dikirim berupa sudut akan diterima oleh Arduino (slave) dan akan diubah menjadi pulsa digital untuk kemudian driver motor akan menggerakkan tiap-tiap aktuator motor stepper yang berada pada setiap *joint arm manipulator*/lengan robot.

### Konversi Data Sudut Ke Data Step

Digunakan beberapa motor stepper pada setiap *joint arm manipulator* untuk menghasilkan pergerakan dan dibutuhkan motor driver yang berguna untuk membangkitkan pulsa-pulsa periodik pada motor stepper. Seluruh setting *microstep* pada driver TB6600 menggunakan *Pulse/rev* sebanyak 1600. Alasan dipilihnya *Pulse/rev* sebesar 1600 dikarenakan berdasarkan perhitungan ,

Dimana 360 merupakan sudut satu putaran penuh dan 1600 merupakan banyaknya *pulse/rev* yang digunakan. Sehingga, diketahui bahwa gerak per pulsanya yaitu sebesar 0.225°. Ketika diberi 1 pulsa maka motor akan bergerak sebesar 0.225° yang membuat pergerakan motor stepper menjadi lebih halus. *Current* (arus) yang diatur pada TB6600 berberda-beda pada tiap *joint*-nya, pengaturan arus pada TB6600 berpacu pada spesifikasi dari masing-masing motor stepper yang digunakan.

Sedangkan, pada joint 6 yaitu motor stepper Nema 8 8HS11-0204S menggunakan *driver* motor A4988. Untuk menggunakan *driver* ini, perlu diatur arus untuk membatasi arus melebihi *rated current* motor. Dilakukan perhitungan,

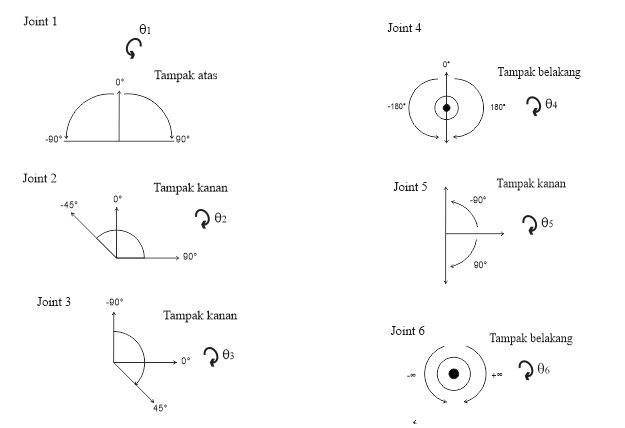
Diketahui:

*Rated current* motor stepper Nema 8 8HS11-0204S pada joint 6 yaitu 0.2 A.

Maka didapat Vref = 0.08 V.

*Microstepping* yang digunakan yaitu *eighth step* yang berarti 1600 *microstep/revolution.*

Agar stepper motor dapat bergerak diperlukan pulsa-pulsa yang diberikan kepada motor. *Arm Manipulator 6 DOF* ini bergerak mengikuti pergerakan *joystick* pada *master controller* menggunakan komunikasi serial Arduino dimana hanya menggunakan 2 pin yaitu RX dan TX. *Master controller* menerima data dari sensor berupa data *degree* (derajat), data tersebut akan dikirimkan ke *slave Arduino* agar dikonversi menjadi data pulsa dan dieksekusi untuk dapat menggerakkan motor stepper yang ada pada tiap joint. Hal pertama yang dilakukan untuk mengubah data *degree* ke data step yaitu dilakukan perhitungan untuk mendapat parameter *step* pada sudut tertentu. Berikut adalah gambar jangkauan sudut pergerakan dari setiap *joint*.



Gambar diatas merupakan sudut-sudut jangkauan pergerakan yang diinginkan. Pergerakan yang dihasilkan dari motor stepper akan ditransmisikan pada pulley penggerak. Perhitungan yang dilakukan untuk mengetahui parameter step pada sudut tertentu yaitu berdasarkan pulley yang digunakan. Perhitungan ini berfokus pada banyaknya gigi-gigi yang ada pada pulley. Berikut adalah analogi perhitungan yang dilakukan,

Keterangan:

A = jumlah gigi pada *pulley* penggerak

B = jumlah gigi pada *pulley* yang digerakkan

C = *pulse/rev* yang digunakan

X = *pulse/rev pulley* penggerak

Berikut adalah data *pulley* berdasarkan jumlah gigi yang digunakan pada tiap *joint*. Analogi perhitungan diatas merupakan analogi yang diambil dari perhitungan kecepatan roda gigi yang dirasa hamper sama penggunaanya pada pulley.

|  |  |  |
| --- | --- | --- |
| **Joint** | **Pulley Penggerak (gigi)** | **Pulley yang digerakkan (gigi)** |
| 1 | 20 | 96 |
| 2 | 20 | 80 |
| 3 | 20 | 100 |
| 4 | 20 | 56 |
| 5 | 20 | 43 |

*Tabel 2. 11 Jumlah gigi pada pulley yang digunakan*

Catatan : untuk *joint* 6 tidak menggunakan *pulley* karena merupakan *end effector*.

Pada tabel 2.11 diketahui bahwa tiap joint memanfaatkan *pulley* dengan jumlah gigi penggerak seluruhnya 20 joint dan jumlah gigi pada *pulley* yang digerakkan berbeda-beda pada tiap joint-nya.

Hasil dari perhitungan merupakan *step* untuk menggerakkan *pulley* untuk satu putaran atau 360°. Seluruh hasil tersebut dijadikan parameter pulse untuk menggerakan sudut tertentu dengan memasukkannya pada program.

Hasil dari perhitungan tersebut dimasukkan ke dalam program utama untuk dijadikan parameter step sudut tertentu. Setelah didapat parameter sudut dan parameter *step*, konversi dapat dilakukan dengan menggunakan fungsi *mapping*. Fungsi *map* ini digunakan untuk mengkonversi suatu bilangan dalam rentang ke rentang bilangan lain, berikut merupakan program yang digunakan :

// ==== convert data degree ke step

currentStateJ1 = thetaConversions(T1, minAngleJ1, maxAngleJ1, batasJ1Min, batasJ1Plus);

currentStateJ2 = thetaConversions(T2, minAngleJ2, maxAngleJ2, batasJ2Min, batasJ2Plus);

currentStateJ3 = thetaConversions(T3, minAngleJ3, maxAngleJ3, batasJ3Min, batasJ3Plus);

currentStateJ4 = thetaConversions(T4, minAngleJ4, maxAngleJ4, batasJ4Min, batasJ4Plus);

currentStateJ5 = thetaConversions(T5, minAngleJ5, maxAngleJ5, batasJ5Min, batasJ5Plus);

currentStateJ6 = thetaConversions(T6, minAngleJ6, maxAngleJ6, batasJ6Min, batasJ6Plus);

// ==== Fungsi untuk converting degree ke step ====

int thetaConversions (int angle, int minAngle, int maxAngle, int minStep, int maxStep) {

int compare = 0;

compare = map(angle, minAngle, maxAngle, minStep, maxStep);

if (compare > maxStep) {

compare = 0;

}

else if ( compare < minStep) {

compare = 0;

}

return compare;

}

Pada program diatas, *thetaConversions* merupakan sebuah fungsi untuk memetakan nilai yang akan dikonversi. Di dalam kurung terdapat T1-T6 yang merupakan nilai hasil konversi dari data sudut ke data *step*. *MinAngle* dan *maxAngle* merupakan rentang nilai sudut yang akan dikonversi sesuai parameter yang telah dideklarasikan diatas, sedangkan batasMin dan BatasPlus merupakan rentang nilai step yang diinginkan.

## hasil

### Pengujian

Pengujian dilakukan untuk melihat bagaimana data *pulse* hasil perhitungan *pulley* dan keakurasian pada sudut yang ditentukan. Berdasarkan perhitungan *roda gigi* menggunakan jumlah gigi yang ada, hasil yang didapat yaitu berupa *step* untuk mencapai 1 putaran roda gigi atau 360° dengan analogi rumus berikut,

Keterangan:

A = jumlah gigi pada *pulley* penggerak

B = jumlah gigi pada *pulley* yang digerakkan

C = *pulse/rev* yang digunakan

x = *pulse/rev pulley* penggerak

Untuk mendapat hasil *step* pada sudut tertentu, diperlukan sedikit perhitungan tambahan. Berikut merupakan perhitungan tambahan untuk mendapatkan hasil *step* pada sudut tertentu.

### Hasil Pengujian

Parameter-parameter yang diujikan yaitu terlampir pada tabel 2.12 sampai dengan tabel 2.17 dimana penjelasannya sebagai berikut,

* *Pulse* yang diberikan merupakan masukan pada arm manipulator berupa *pulse* yang didapatkan dari hasil perhitungan.
* Sudut yang dihasilkan merupakan sudut *arm manipulator* yang seharusnya didapat dari masukan pulsa.
* Sudut aktual merupakan sudut yang dihasilkan pada *arm manipulator* dari hasil pengujian
* *Error* merupakan perbandingan kesalahan antara sudut yang dihasilkan dan sudut aktual.

Pengukuran pada pengujian didasarkan pada alat ukur busur digital. Busur digital yang digunakan dalam pengujian ini memiliki resolusi sebesar 0.05º dan memiliki kearuasian sebesar ±0.03º.

Joint 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan | Pulse yang diberikan | Sudut yang dihasilkan (°) | Sudut Aktual (°) | *Error*  (%) |
| 1 | 1920 | 90 | 89.95 | 0.06 |
| 2 | 960 | 45 | 45 | 0.00 |
| 3 | 640 | -30 | -30.05 | 0.17 |
| 4 | 1280 | -60 | -59.5 | 0.83 |
| Rata-Rata Error | | | | 0.27 |

Joint 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan | Pulse yang diberikan | Sudut yang dihasilkan (°) | Sudut Aktual (°) | *Error*  (%) |
| 1 | 1600 | 90 | 90 | 0.00 |
| 2 | 800 | 45 | 45.05 | 0.11 |
| 3 | 533 | -30 | -30.1 | 0.33 |
| 4 | 800 | -45 | -45.05 | 0.11 |
| Rata-Rata Error | | | | 0.14 |

Joint 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan | Pulse yang diberikan | Sudut yang dihasilkan (°) | Sudut Aktual (°) | *Error*  (%) |
| 1 | 1000 | 45 | 45 | 0.00 |
| 2 | 555 | 25 | 24.95 | -0.20 |
| 3 | 1333 | -60 | -60.15 | 0.25 |
| 4 | 2000 | -90 | -90.05 | 0.06 |
| Rata-Rata Error | | | | 0.13 |

Joint 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan | Pulse yang diberikan | Sudut yang dihasilkan (°) | Sudut Aktual (°) | *Error*  (%) |
| 1 | 1867 | 150 | 150.1 | 0.07 |
| 2 | 1120 | 90 | 90 | 0.00 |
| 3 | 560 | -45 | 45.05 | 0.11 |
| 4 | 747 | -60 | 60.1 | 0.17 |
| Rata-Rata Error | | | | 0.09 |

Joint 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan | Pulse yang diberikan | Sudut yang dihasilkan (°) | Sudut Aktual (°) | *Error*  (%) |
| 1 | 860 | 90 | 90 | 0.00 |
| 2 | 430 | 45 | 45.15 | 0.33 |
| 3 | 287 | -30 | 29.95 | -0.17 |
| 4 | 573 | -60 | 60.05 | 0.08 |
| Rata-Rata Error | | | | 0.15 |

Joint 6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan | Pulse yang diberikan | Sudut yang dihasilkan (°) | Sudut Aktual (°) | *Error*  (%) |
| 1 | 133 | 30 | 30.00 | 0.00 |
| 2 | 266 | 60 | 60.05 | 0.08 |
| 3 | 800 | 180 | 180.00 | 0.00 |
| 4 | 933 | 210 | 210.05 | 0.02 |
| Rata-Rata Error | | | | 0.03 |

Rata-rata persen error

|  |  |
| --- | --- |
| Joint | Rata-Rata Error |
| 1 | 0.27 |
| 2 | 0.14 |
| 3 | 0.13 |
| 4 | 0.09 |
| 5 | 0.15 |
| 6 | 0.16 |
| Rata-rata error keseluruhan | 0.27 |

## kesimpulan

*Arm Manipulator 6 DoF* merupakan lengan robot yang memiliki 6 *joint* dan dikendalikan menggunakan sebuah *master controller*. *Arm Manipulator* ini digerakkkan dengan menggunakan mode *Joystick*. Motor stepper dipergunakan sebagai aktuator pada proyek ini. Pergerakan motor stepper pada tiap joint dapat digerakkan secara bersamaan dalam satu gerakan. Digunakan setting *microstep* 1600 *pulse/rev* pada tiap driver motor karena optimal diaplikasikan pada *arm manipulator 6 DoF*. Pengkonversian data *degree* menjadi data *step* memanfaatkan perhitungan rasio jumlah gigi pada pulley yang digunakan untuk dijadikan parameter. Keakurasian motor stepper yang diaplikasikan pada arm manipulator 6 DoF terhadap pulse yang diberikan menunjukkan bahwa motor stepper memiliki keakurasian sebesar 99,86% dengan rata-rata error terhadap seluruh joint yaitu sebesar 0,14%.

## Referensi

Bagian referensi menggunakan bibliography Mendeley dengan format IEEE Style, font Tahoma 11. Referensi yang ditampilkan adalah referensi yang disitasi di dalam artikel yang diusulkan. Jika pada word processor belum tersedia IEEE *styles* maka dapat melakukan pengunduhan pada halaman http://bibword.codeplex.com.

Prosiding:

[1] Q. X. Do, Y. S. Chan, and D. Roth, “Minimally supervised event causality identification,” in *EMNLP ’11 Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2011, pp. 294–303.

Jurnal:

[2] J. Atkinson and A. Rivas, “Discovering novel causal patterns from biomedical natural-language texts using Bayesian nets,” *IEEE Trans. Inf. Technol. Biomed.*, vol. 12, no. 6, pp. 714–722, 2008.

Buku:

[3] G. Milette and A. Stroud, *Professional Android Sensor Programming*. Wrox; 1 edition, 2012.

Karya tulis:

[4] K. Elissa, “Title of paper if known,” unpublished.