

PENGUNAAN PERANGKAT LUNAK *OPEN SOURCE* UNTUK SISTEM *OPEN ARCHITECTURE* PADA MESIN *MILLING CNC*

THE USE OF OPEN SOURCE SOFTWARE FOR OPEN ARCHITECTURE SYSTEM ON CNC MILLING MACHINE

Dalmasius Ganjar Subagio dan Tinton Dwi Atmaja

Pusat Penelitian Tenaga Listrik dan Mekatronik - LIPI
Komp. LIPI Bandung, Jl Sangkuriang, Gd 20, Lt 2, Bandung,
Jawa Barat 40135, Indonesia.
tinton_dwi@yahoo.com

Diterima: 28 Oktober 2011; Direvisi: 2 November 2011; Disetujui: 19 Desember 2011;
Terbit online: 22 Desember 2011.

Abstrak

Sistem kerja mesin *computer numerical control (CNC) milling* tidak dapat terlepas dari perangkat lunak sebagai penerjemah bahasa operator ke bahasa mesin. Perangkat lunak ini dituntut untuk mengikuti ketentuan *open architecture* yang antara lain mempunyai kapabilitas *portability*, *extendability*, *interoperability*, dan *scalability*. Proses manufaktur modern terkadang menggunakan mesin *CNC milling* untuk waktu yang lama bahkan melebihi waktu kerja yang ditentukan oleh produsen. Ketika mesin sudah melewati masa yang ditentukan dan produsen memutuskan *discontinue*, maka pengguna akan menemui masalah untuk mempertahankan kinerja mesin jika diharuskan tetap tergantung pada perangkat lunak bawaan dari produsen. Tulisan ini bertujuan menunjukkan bahwa penggunaan *open source software (OSS)* adalah jalan keluar untuk mempertahankan kinerja mesin. Dengan penggunaan *OSS*, pengguna tidak lagi tergantung pada perangkat lunak dari produsen karena *OSS* bersifat terbuka dan dapat dikembangkan secara mandiri. Dalam tulisan ini dipergunakan *USBCNC V.3.42* sebagai alternatif *OSS* pada mesin *CNC milling*. *OSS* ini menunjukkan hasil uji coba benda kerja yang sesuai dengan pola yang diinginkan. Hasil uji coba menunjukkan kinerja mesin dengan *OSS* serupa dengan kinerja mesin ketika menggunakan perangkat lunak dari produsen.

Kata kunci: Mesin *CNC milling*, *open architecture*, *open source software*.

Abstract

Computer numerical control (CNC) milling machine system can not be separated from the software required to follow the provisions of the Open Architecture capabilities that have portability, extendability, interoperability, and scalability. When a prescribed period of a CNC milling machine has passed and the manufacturer decided to discontinue it, then the user will have problems for maintaining the performance of the machine. This paper aims to show that the using of open source software (OSS) is the way out to maintain engine performance. With the use of OSS, users no longer depend on the software built by the manufacturer because OSS is open and can be developed independently. In this paper, USBCNC V.3.42 is used as an alternative OSS. The test result shows that the workpiece is in match with the desired pattern. The test result shows that the performance of machines using OSS has similar performance with the machine using software from the manufacturer.

Keywords: *CNC milling machine, open architecture, open source software.*

I. PENDAHULUAN

Teknologi di bidang manufaktur berkembang sangat pesat dalam beberapa tahun terakhir ini. Mesin *computer numerical control (CNC)* merupakan mesin yang tergolong canggih, sehingga banyak sekali digunakan dalam industri pemesinan untuk memproduksi komponen dengan tingkat kerumitan dan presisi yang tinggi. Selain itu, mesin *CNC* mempunyai konsistensi yang lebih efektif untuk pengerjaan dalam jumlah banyak. Penggunaan mesin *CNC* konvensional dalam proses pemotongan, pengeboran,

pengetapan dan proses pemesinan lainnya, tentu saja memberikan hasil yang tidak presisi dan memerlukan waktu cukup lama dikarenakan hasil produksi akan tergantung dari kemampuan operator dalam melakukan proses tersebut. Gambar 1 merupakan contoh mesin *CNC frais/milling* yang banyak terdapat di industri pemesinan, dimana mesin tersebut telah dilengkapi dengan kontrol penggerak untuk proses kerja pemesinan [1][2]. Mesin *CNC milling* tersebut mampu mengerjakan bentuk-bentuk yang tidak mungkin dikerjakan oleh



Gambar 1. Mesin CNC *frais/milling*.

mesin *frais* konvensional karena dilengkapi oleh kontrol penggerak yang dapat diperintah oleh program atau kode standar umum.

Manfaat mesin CNC yang signifikan menciptakan kebutuhan atas kinerja mesin yang konsisten bahkan disaat mesin telah tidak lagi dikembangkan (*discontinue*). Ketika masa ini datang, pengembangan perangkat lunak akan menjadi masalah jika tetap memaksakan untuk menggunakan perangkat lunak berlisensi dari produsen asli. Perangkat lunak lama akan bermasalah dengan kinerjanya yang harus *uptodate* terhadap kebutuhan yang terus meningkat. Kadang dituntut untuk *interoperable* dengan perangkat yang lebih baru. Hal ini akan meningkatkan pos pembiayaan tidak terduga demi mempertahankan kinerja mesin seperti ketika masih menggunakan perangkat lunak keluaran produsen. Ketika pengguna memaksakan diri untuk menghubungi produsen, maka akan muncul faktor penundaan waktu produksi karena produsen memerlukan waktu untuk mendapatkan kembali perangkat lunak yang dipesan lengkap dengan kondisi *ter-update*.

Dalam tulisan ini akan di bahas mengenai pengembangan *open architecture CNC milling machine* terutama dalam pemilihan perangkat lunak program/kontrol penggerak guna mengurangi biaya pembuatan dan mengurangi ketergantungan dengan produsen dengan tidak mengabaikan fungsi dari mesin CNC *milling* tersebut. Perangkat lunak kontrol penggerak yang digunakan adalah *open source software* (OSS) jenis *freeware controller* yang dapat diunduh dan *di-update*. Fungsinya sama dengan perangkat lunak kontrol penggerak berbayar yang banyak digunakan pada industri manufaktur saat ini seperti: Siemen, Fanuc, Okuma dan lain-lain.

II. OSS PADA OPEN ARCHITECTURE MESIN MILLING CNC

Pada tahun 90-an, "Technical Committee of Open Systems" dari IEEE mendefinisikan *Open System* sebagai sebuah sistem terbuka yang

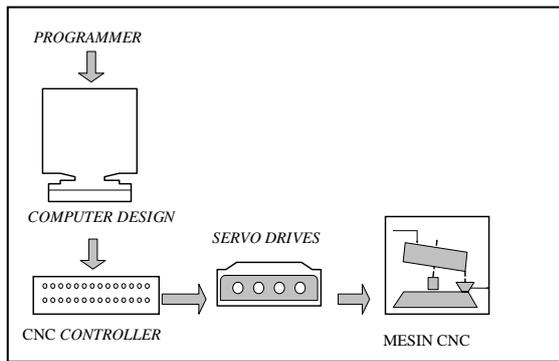
menyediakan kapabilitas yang memungkinkan aplikasi yang diimplementasikan dapat berjalan dalam berbagai platform dari berbagai vendor, *interoperate* dengan sistem aplikasi lainnya dan menyediakan gaya yang konsisten dalam berinteraksi dengan pengguna [3]. Estimasi kapabilitas dari keterbukaan sistem *open architecture* dikemukakan dalam empat hal berikut [4][5].

- *Portability*; modul aplikasi dapat digunakan pada *platform* yang berbeda tanpa ada perubahan dengan tetap mempertahankan kemampuan kerjanya.
- *Extensibility*; sejumlah modul aplikasi dapat berjalan dalam satu *platform* tanpa menimbulkan konflik.
- *Interoperability*; modul-modul dapat bekerja sama satu sama lain dalam suatu perlakuan yang konsisten dan dapat bertukar data dalam cara yang telah terdefiniskan.
- *Scalability*; fungsi, kinerja, dan ukuran fisik modul dapat beradaptasi sesuai dengan kebutuhan pengguna.

Banyak teknologi *open architecture* CNC dikembangkan berdasarkan pada Microsoft Windows, sebuah *32-bit embedded control system* yang cocok untuk manajemen tugas-tugas umum [6][7]. Dengan kemajuan terbaru dalam teknologi elektronik dan informasi, perangkat keras pengendali gerak telah menggunakan mikroprosesor dan memiliki *digital signal processor* (DSP), *field programmable gate array* (FPGA) atau aplikasi sirkuit terpadu lainnya sebagai pendukung. Pengembangan *open architecture CNC milling machine* juga mengadopsi struktur *master-slave* menggunakan PC dan modul *Numerical Control* (NC) yang masing-masing dirancang khusus sebagai *master* dan *slave*. Modul *Slave* NC menjalankan tugas-tugas kontrol *real-time*, seperti kontrol gerakan sumbu pada DSP dan FPGA, sedangkan PC *Master* menjalankan tugas-tugas non *real-time* seperti penjadwalan, pemantauan *input/output* dan *on-screen simulation* [8].

A. Alur Kerja Mesin Milling CNC

Alur kerja mesin *milling* ini dianalisis untuk menghasilkan alur sistem kendali yang diinginkan. Mesin CNC (*computer numerical control*) pada dasarnya merupakan gabungan antara mekanik dan elektronik dimana informasi pergerakan mesin didapatkan berdasarkan kode-kode yang dapat dibaca oleh mesin CNC tersebut. Informasi ini bersifat digital dan digunakan sebagai acuan untuk menggerakkan motor (*stepping motor* atau *servo motor*) sesuai dengan



Gambar 2. Alur kerja mesin CNC milling.

yang diinginkan melalui *driver*. Dari Gambar 2 dapat kita lihat bahwa modul *Computer Design* memiliki hubungan erat dengan operator dan mesin CNC. Operator hanya membuat suatu gambar menggunakan sistem *Computer Aided Design* (CAD) atau *Computer Aided Manufacturing* (CAM) dimana gambar tersebut berisi informasi mengenai bentuk dan ukuran yang dikehendaki oleh operator.

Modul *CNC Controller* merupakan perangkat keras yang berfungsi sebagai penerima kode-kode dari *Modul Computer Design* dan menterjemahkannya ke dalam instruksi atau perintah dalam bentuk bahasa yang dapat diterima oleh mesin CNC. Pada umumnya mesin CNC ini menggunakan sumbu X, Y dan Z untuk mesin frais (*milling 3 axis*). Hasil terjemahan dari Modul *CNC Controller* ini kemudian diterima oleh modul *Servo Drive*. Modul *driver* ini akan meneruskan instruksi kepada motor-motor penggerak mesin CNC sesuai dengan sumbu kerjanya.

Dalam prinsip kerja mesin CNC *milling* ini, *spindle motor current* akan dideteksi dan diolah untuk mendapatkan pergerakan pemotongan yang konstan dengan mengaplikasikan *fuzzy logic controller* (FLC) [9]. Teknologi *multi-agent* dipakai untuk meningkatkan performa distribusi sistem sebagai suatu bagian *intelligent mechatronic system* (IMS) generasi baru. Dasar keberhasilan IMS adalah kemampuan setiap *intelligent agent* untuk berkomunikasi dan bekerjasama antar agen [10][11][12]. Dengan IMS ini, dapat diperoleh *ultra-precision motion control*, proses manufaktur kecepatan tinggi, dan penyelesaian berkualitas tinggi dengan biaya rendah dengan *reliability* yang lebih baik [13].

B. Perangkat Lunak Mesin CNC milling

Berdasar pada definisi *the open systems architecture*, perangkat lunak dirancang dan diimplementasikan dengan *human machine*



Gambar 3. Contoh tampilan perangkat lunak berlisensi pada monitor mesin CNC milling [14].

interface (HMI) yang *user-friendly* sehingga memerlukan perhatian lebih pada penampakan visualnya. Perangkat lunak ini juga harus mudah dioperasikan, dan dapat memenuhi kebutuhan-kebutuhan operator yang beragam [16].

Perangkat lunak yang dapat digunakan dalam pengoperasian mesin CNC *milling* dapat berupa perangkat lunak yang berbayar atau perangkat lunak *open source*. Perangkat lunak yang berbayar atau *licensed software* biasanya merupakan bawaan dari mesin dan mendapat *update* dan pengembangan berkala dari produsennya. Gambar 3 menunjukkan tampilan perangkat lunak berlisensi GSK 983M yang ditanam pada mesin CNC *milling*.

Masalah akan terjadi ketika banyak mesin-mesin CNC *milling* yang sudah tidak diproduksi lagi membutuhkan pemeliharaan perangkat lunak. Oleh karena itu perangkat lunak *open source* menjadi sebuah alternatif yang cukup menjanjikan demi kelangsungan kinerja mesin.

Gambar 4 menunjukkan tampilan *open source software* yang digunakan sebagai *retrofit* perangkat lunak berbayar. OSS yang digunakan disini adalah USBCNC V.3.42 [15].

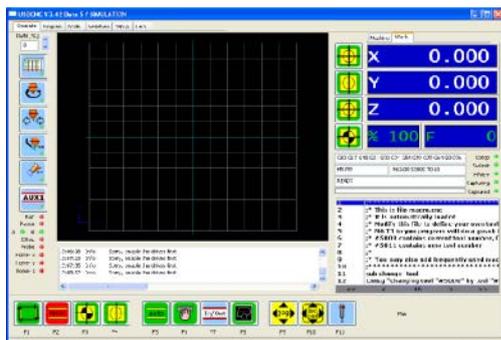
C. Standard G-Code

Penggunaan OSS tidak dapat dipisahkan dari penggunaan kode yang sudah terstandarisasi untuk membantu penentuan tingkat keakuratan hasil yang diperoleh. Tabel 1 menunjukkan G-Code sebagai kode standar yang digunakan untuk membuat program CNC *milling* secara populer melebihi penggunaan bahasa lainnya seperti STEP-NC [17].

Pada sistem ini, masih memungkinkan penggunaan STEP-NC karena perangkat lunak yang digunakan dapat menterjemahkan bahasa STEP-NC ke bahasa G-Code [18][19]. Fitur ini mendukung faktor *portability* sebagai salah satu kapabilitas sistem *open architecture* yang harus dipenuhi.

Tabel 1.
Keterangan standar *G Code*.

Kode	Keterangan	Kode	Keterangan
G00	Pindah posisi <i>axis</i> dengan kecepatan penuh	G83	<i>Counter bore</i> dengan waktu tunda
G01	Pindah posisi <i>axis</i> secara linear (<i>feed rate</i>)	G84	Pengetapan (<i>tapping</i>)
G02	Berputar searah jarum jam	G85	<i>Reamer</i>
G03	Berputar berlawanan arah jarum jam	G86	<i>Boring</i>
G04	Waktu tunda (<i>dwell</i>)	G90	<i>Absolute</i>
G17	Melingkar X-Y dipakai pada G02 dan G03	G91	<i>Incremental</i>
G18	Melingkar Y-Z dipakai pada G02 dan G03	G92	Koordinat referensi benda kerja
G19	Melingkar X-Z dipakai pada G02 dan G03	G98	Pengembalian pahat pada Z awal
G28	Mengembalikan keposisi otomatis	G99	Pahat pada jarak yg ditentukan (R)
G40	Pembatalan kompensasi diameter pahat	M02	<i>Spindle stop</i>
G41	Kompensasi diameter pahat kiri	M03	<i>Spindle</i> berputar searah jarum jam
G42	Kompensasi diameter pahat kanan	M04	<i>Spindle</i> berputar berlawanan arah jarum jam
G43	Kompensasi panjang arah Positif	M05	<i>Spindle</i> dan pendingin stop
G44	Kompensasi panjang arah Negatif	M06	Pergantian tool
G49	Pembatalan kompensasi panjang pahat	M08-M09	<i>On</i> dan <i>Off</i> pompa pendingin
G54 - G59	Sistem koordinat	M30	Akhir program
G80	Membatalkan <i>fixed cycle</i>	M98	Masuk ke sub program
G81	<i>Fixed cycle</i> untuk pengeboran (<i>drilling</i>)	M99	Keluar ke sub program



Gambar 4. Tampilan monitor USB CNC milling [15].



Gambar 5. Hasil rancang bangun mesin CNC milling di P2 Telimek-LIPI.

III. IMPLEMENTASI OSS DAN UJI COBA

Tahap berikutnya adalah melakukan implementasi dan uji coba OSS pada *open architecture CNC milling machine*. Pertama adalah menyiapkan perlengkapan ujicoba, membuat program CNC dan menguji coba mesin pada benda kerja.

A. Perlengkapan Uji Coba

Aparatus yang akan digunakan adalah mesin CNC milling hasil penelitian dan rancang bangun yang dilaksanakan di P2 Telimek – LIPI seperti ditunjukkan Gambar 5. Mesin CNC ini dengan menggunakan *freeware controller* USBCNC V.3.42 berbasis OSS. Mesin tersebut di buat dengan menggunakan komponen lokal dan beberapa komponen impor (Gambar 5).

Teori pendukung dalam proses rancang bangun mesin CNC milling ini diantaranya ilmu bahan dalam menentukan konstruksi mesin dan pemilihan benda kerja. Hal-hal yang harus diperhatikan dalam pemilihan bahan sebuah komponen mesin adalah fungsi, pembebanan dan umur, kemampuan untuk dibentuk dan diproduksi, ongkos produksi dan mudah didapat dipasaran. Pada umumnya pemilihan bahan tersebut dapat berdasarkan tabel bahan standar misalnya [20]:

1. Bahan untuk poros dan gardan digunakan baja karbon ST 60.
2. Bahan untuk poros engkol (*crankshaft*) digunakan baja kualitas tinggi atau besi cor khusus.
3. Bahan untuk pasak dan pegas digunakan ST 60.

4. Bahan roda gigi bergantung kepada pembebanan dan ukuran komponen. Disini digunakan bahan seperti besi cor kelabu, besi cor baja, baja ST 42 – ST 70 kadang-kadang digunakan bahan non logam atau bahan sintetis.
5. Bahan untuk baja pemotong digunakan bahan baja perkakas yang di keraskan.
6. Bahan konstruksi ringan dapat digunakan besi cor kelabu.

Disamping ilmu bahan juga digunakan digunakan teori pendukung untuk menghitung gaya-gaya yang terjadi pada suatu komponen yang tidak di bahas dalam tulisan ini.

B. Pembuatan Program CNC

Arsitektur yang akan digunakan dalam proses kendali ini mengutamakan tiga proses utama yaitu *sampling*, *filtering*, dan *processing data* untuk menghasilkan sinyal kendali [21]. Susunan perintah akan dibuat dalam bentuk teks dalam bahasa *G-Code* yang disesuaikan dengan perangkat lunak yang dipergunakan.

Terdapat dua cara dalam membuat *listing program* yang berisi perintah yaitu menggunakan *notepad* atau menggunakan CAD/CAM. Kedua cara tersebut diawali dengan pengetikan susunan perintah dalam *G-Code*. Ketika menggunakan perangkat lunak bawaan, seperti Fanuc, contoh format program yang biasa digunakan sebagai berikut :

```
G90 G00 G54 X0. Y0.Z50.  
M03 S2000  
M08  
G00 Z5.  
G01 Z-0,5 F100  
--  
Line omitted  
--  
G00 Z50.  
G91 G28 X0. Y0. Z0.  
M30
```



Gambar 6. Pola kerja yang akan di proses.

Sedangkan jika menggunakan *software controller USBCNC V.3.42*, format penulisan adalah seperti berikut:

```
N106G0G90G54X.375Y-  
3.249Z50.S5000M3  
N110Z5.  
N112G1Z-.2F20.  
--  
Line omitted  
--  
N7550G0Z5.  
N7552Z50.  
N7554M5  
N7556G91G28X0.Y0.Z0.  
N7560M30
```

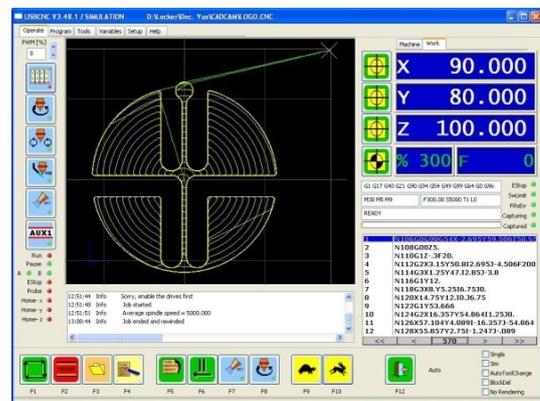
Ketika menggunakan *notepad*, susunan perintah dapat diketik langsung dalam program *Notepad* kemudian disimpan dengan nama *file*; misalnya: “percobaan” dengan format ekstensi ‘cnc’, contoh: “percobaan.cnc”. maka *file* tersebut akan dapat dieksekusi oleh mesin CNC *milling*.

Cara lain adalah dengan memasukkan susunan perintah tersebut kedalam CAD/CAM kemudian di simpan langsung dengan format *file* ‘cnc’, contoh: “percobaan.cnc”. maka *file* tersebut juga dapat dieksekusi langsung oleh mesin CNC *milling*.

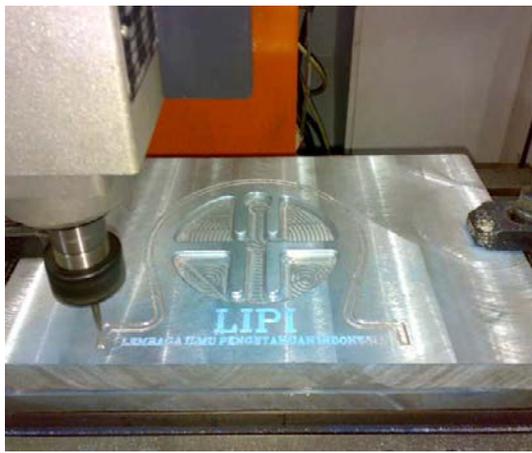
C. Uji Coba Mesin

Uji coba mesin dilakukan setelah mesin selesai dirangkai, baik konstruksi mekanik maupun bagian kontrol penggerak. Setelah mesin selesai terpasang kemudian dicoba untuk data program untuk mengoperasikan mesin.

Langkah pertama adalah menentukan pola yang akan digambarkan pada benda kerja. Gambar 6 menunjukkan pola yang akan di ukir pada benda kerja. Kemudian pola tersebut diterjemahkan kedalam *G-Code* dan dimasukkan kedalam *open source software*. Gambar 7 merupakan tampilan proses kerja mesin dari data



Gambar 7. Tampilan user interface OSS.



Gambar 8. Proses uji coba mesin.



Gambar 9. Benda kerja hasil uji coba.

yang diperoleh dari hasil CAD/CAM. Setelah mesin dieksekusi, maka pola pada Gambar 6 akan dieksekusi oleh OSS seperti Gambar 7. Kemudian diterjemahkan kedalam bahasa mesin yang dibaca oleh mesin CNC *milling* untuk melakukan kegiatan seperti ditunjukkan pada Gambar 8. Gambar 8 menunjukkan proses *milling* ketika program diuji-cobakan kepada benda kerja.

IV. HASIL UJI COBA DAN PEMBAHASAN

Berdasarkan hasil uji coba mesin dengan menggunakan kontrol USBCNC ternyata produk yang dihasilkan sesuai dengan gambar kerja yang direncanakan. Gambar 9 menunjukkan benda kerja hasil uji coba mesin CNC *milling* menggunakan OSS. Pendeteksian dan pengolahan sinyal *spindle motor current* serta penggunaan FLC telah mempertahankan kesinambungan proses pemotongan. Penggunaan teknologi *multi-agent* meningkatkan performa distribusi sistem sebagai suatu bagian IMS dapat dilihat dari kemudahan pengumpulan instruksi dalam bahasa mesin dan peningkatan kemampuan *monitoring*. Penggunaan *G-Code* tidak menimbulkan masalah karena kebanyakan perangkat lunak saat ini sudah terintegrasi dengan varian-varian gaya penulisan instruksi *G-Code*.

Teknologi *Open Architecture* memungkinkan pengembang dan operator mesin untuk mengimplementasikan dan mengintegrasikan perangkat keras dan perangkat lunak yang berbeda dari berbagai vendor dalam satu *platform* dengan tetap mempertahankan antarmuka pengguna yang efektif. Dalam hal ini, OSS sangat mendukung faktor *portability* dan *extendability*, bahkan untuk faktor *interoperability*, OSS dapat dengan mudah dimodifikasi untuk dapat saling bertukar data

antar modul. Faktor *scalability* tidak terlalu berpengaruh ketika dilihat dari sisi perangkat lunak, namun begitu skala besar dilihat dari biaya pengadaan perangkat lunaknya, maka OSS dapat menghemat biaya pengadaan secara signifikan. Efektivitas biaya OSS pada *open architecture* sangat tinggi selain dari biaya pengadaan/akuisisi yang lebih rendah daripada perangkat lunak berbayar, biaya *support* untuk pengembangan dan pengujian dapat dihemat jika dilakukan pengembangan personel IT mandiri. Mesin-mesin berbasis OSS akan mampu dipelihara secara mandiri tanpa tergantung pada layanan purna jual produsen.

Tingkat kemudahan pemakaian OSS akan tergantung pada pengembangan sumber daya manusianya. Operator pastinya terbiasa menggunakan perangkat lunak bawaan dari produsen. *User interface* dari OSS tentunya akan membuat operator yang terbiasa menggunakan perangkat lunak bawaan produsen membutuhkan waktu adaptasi. Proses adaptasi operator akan memakan waktu sehingga dalam skala besar akan menimbulkan hambatan tersendiri dalam proses produksi. Jalan keluarnya adalah melakukan manajemen perubahan dan pelatihan yang terencana sehingga penggunaan mesin tetap optimal. Keamanan pada OSS lebih menjanjikan daripada perangkat lunak berbayar. OSS mempunyai *patch* yang lebih sedikit daripada perangkat lunak berbayar. Lebih sedikit *patch* artinya lebih sedikit isu dan ancaman dari luar. Virus-virus yang dibuat untuk berjalan pada *platform* umum akan kesulitan untuk berjalan pada *platform open source*. Sehingga dapat disimpulkan bahwa OSS mempunyai resiko keamanan yang lebih rendah daripada perangkat lunak bawaan produsen. Tabel 2 menunjukkan parameter kinerja mesin yang menggunakan *open source software* dibandingkan dengan mesin yang menggunakan *software* berbayar.

Tabel 2.

Perbandingan penggunaan mesin dengan *open source software* dan mesin dengan *software* berbayar.

Parameter	Mesin menggunakan <i>open source software</i>	Mesin menggunakan <i>software</i> berbayar
Portability	Compatible	Compatible
Extendability	Compatible	Compatible
Interoperability	Compatible	Compatible
Scalability	Compatible	Compatible
Antarmuka pengguna	Butuh adaptasi	Mudah dipahami
Keamanan	Beresiko rendah	Beresiko tinggi
Biaya pengadaan/akuisisi perangkat lunak	Biaya rendah	Biaya tinggi
Biaya pemeliharaan perangkat lunak	Support yang hemat	Support yang berkesinambungan
Pemeliharaan mesin	Dapat dilakukan mandiri	Tergantung pada produsen

V. KESIMPULAN

Penggunaan *open source software* adalah alternatif yang telah teruji sebagai pengganti perangkat lunak berbayar bawaan dari produsen. *Open architecture CNC milling machine* mempunyai prinsip *portability*, *extendability*, *interoperability*, dan *scalability* dimana *Open Source Software* dapat menyajikannya dengan tidak mengurangi kinerja mesin *CNC milling*. OSS yang digunakan adalah USBCNC dengan menggunakan *G-Code* yang diterapkan pada prototipe mesin *CNC milling* 3 sumbu yang dikembangkan oleh Pusat Penelitian Tenaga Listrik dan Mekatronik. Hasil pengujian pada benda kerja aluminium menunjukkan pola yang dihasilkan sesuai dengan pola yang direncanakan. Sehingga secara kinerja, OSS dapat menterjemahkan perintah operator kepada mesin serupa dengan perangkat lunak berbayar.

Mengaplikasikan OSS pada *CNC milling machine* secara umum tetap dapat mendukung empat prinsip dasar *open architecture* dengan memunculkan keuntungan dan kerugian tersendiri. Keuntungan akan muncul dari segi biaya akuisisi, pemeliharaan, dan keamanan. Pemeliharaan mesin akan lebih mandiri tanpa tergantung pada layanan produsen terutama pada produk *discontinue*. Kelemahan akan muncul pada faktor adaptasi operator yang nantinya memerlukan manajemen perubahan dan pelatihan yang terencana.

UCAPAN TERIMAKASIH

Penulis berterima kasih kepada Dian Andriani dan Siti Fausiyah Rahman dari Chonnam National University atas sumbangan kajian internasionalnya. Juga kepada peneliti dari Gwangju Institute of Science and Technology, dan University of Adelaide yang telah mendukung banyak hal dalam tulisan ini. Penulis juga berterima kasih kepada Vita Susanti dan

semua peneliti Pusat Penelitian Tenaga Listrik dan Mekatronik – LIPI yang belum dapat disebutkan satu persatu yang telah banyak membantu terselesaikannya tulisan ini

DAFTAR PUSTAKA

- [1] Suryaputra, (2008, December) *Teknik Permesinan*. [Online]. Available: <http://suryaputra.edublogs.org/2008/12/20/teknik-frais-dasar/> [dikutip: 25 Oktober 2011]
- [2] Suzhou Sunda Machine Tools Co., Ltd. *Machine Tools - CNC Milling machine*. [Online]. Available: <http://www.sundamachinetools.com/cnc-milling-machine/> [dikutip: 25 Oktober 2011]
- [3] Institute of Electrical and Electronics Engineers, 1003.0-1995; *IEEE Guide to POSIX Open System Environment (OSE)*.: IEEE Computer Society, 1995.
- [4] Pengfei Li, Tao Gao, Jianping Wang, Hongzhao Liu, "Open Architecture of CNC System Research Based on CAD Graph-driven Technology," *Robotics and Computer-Integrated Manufacturing*, Volume 26, Issue 6. pp. 720–724, December 2010.
- [5] Günter Pritschow, Co-ordinator, Yusuf Altintas, Francesco Jovane, Yoram Koren, Mamoru Mitsuishi, Shozo Takata, Hendrik van Brussel, Manfred Weck, Kazuo Yamazaki, "Open Controller Architecture - Past, Present and Future," *CIRP Annals - Manufacturing Technology*, Volume 50, Issue 2, pp. 463-470, 2001.
- [6] Z. Wang, E. Gao and Y. Zhang, "A Windows CE-based open-architected CNC system," *Manufacturing Automation*, Volume 23, Issue 9, pp. 1-4, 2005.
- [7] S. Xiao, D. Li, Y. Lai, J. Wan and S. Feng, "An open architecture numerical control system based on Windows CE," in

- Proceeding of IEEE International Conference on Control and Automation, Guangzhou, PRC, June 2007, pp. 1237-1240.*
- [8] K. Ekkachai, U. Komin, W. Chaopramualkul, A. Tantaworrasilp, P. Kwansud, P. Seekhao, T. Leelasawassuk, K. Tanta-Ngai, K. Tungpimolrut, "Design and Development of an Open Architecture CNC Controller for Milling Machine Retrofitting," in *Proceeding of ICROS-SICE International Joint Conference 2009*, Fukuoka International Congress Center, Japan, August 18-21, 2009, pp. 5629-5632.
- [9] Dohyun Kim, Doyoung Jeon, "Fuzzy-Logic Control of Cutting Forces in CNC Milling Processes using Motor Currents as Indirect Force Sensors," *Precision Engineering*, Volume 35, Issue 1, pp. 143–152, Januari 2011.
- [10] G. Rzevsski, "On Conceptual Design of Intelligent Mechatronics System," *Mechatronics*, Volume 13, Issue 10, pp. 1029-1044, Desember 2003.
- [11] A. Coronato, G. De Pietro, L. Gallo, "An agent based platform for task distribution in virtual environments," *Journal of Systems Architecture*, Volume 54, Issue 9, pp. 877–882, September 2008.
- [12] X. Desforges, B. Archimede, "Multi-agent framework based on smart sensors/actuators for machine tools control and monitoring, 19," *Engineering Applications of Artificial Intelligence*, Volume 19, Issue 6, pp. 641–655, September 2006.
- [13] Luis Morales-Velazquez, Rene de Jesus Romero-Troncoso, Roque Alfredo Osornio-Rios, Gilberto Herrera-Ruiz, Eduardo Cabal-Yeppez, "Open-architecture system based on a reconfigurable hardware–software multi-agent platform for CNC machines," *Journal of Systems Architecture*, Volume 56, Issue 9, pp. 407–418, September 2010.
- [14] GSK CNC. (2007) CNC Machine and CNC Equipment, CNC Milling Controller GSK 983M. [Online]. Available: http://gskcnc.com/cnc_controller.php [dikutip: 25 Oktober 2011]
- [15] Damen CNC. (2009) USBCNC CPU V3 Software+Hardware Interface. [Online]. Available: <http://www.damencnc.com/damencnc.php?dir=cpt&idComp=195&langId=EN> [dikutip: 25 Oktober 2011]
- [16] Li Zhang, Ruqiang Yan, Robert X. Gao, and Kang B. Lee, "Open Architecture Software Design for Online Spindle Health Monitoring," in *Proceeding of Instrumentation and Measurement Technology Conference - IMTC 2007*, Warsaw, Poland, 1-3 May, 2007, pp. 1-6.
- [17] Fanuc LTD Japan, *Fanuc Operator's manual.*, 1988.
- [18] Tao Liu, Yongzhang Wang and Hongya Fu, "The Open Architecture CNC System HITCNC Based on STEP-NC," in *Proceeding of 6th World Congress on Intelligent Control and Automation*, Dalian, China, June 21 - 23, 2006, pp. 7983-7987.
- [19] M. Minhat, V. Vyatkin, X. Xu, S. Wong, Z. Al-Bayaa, "A novel open CNC architecture based on STEP-NC data model and IEC 61499 function blocks," *Robotics and Computer-Integrated Manufacturing*, Volume 25, Issue 3, pp. 560–569, June 2009.
- [20] Lawrence H. Vanvlack, *Ilmu dan Teknologi Bahan*, Ed. Surabaya, Indonesia: Elangga, 1990.
- [21] Roberto Augusto Gómez Loenzo, Pedro Daniel Alaniz Lumbreras, René de Jesús Romero Troncoso, Gilberto Herrera Ruiz, "An object-oriented architecture for sensorless cutting force feedback for CNC milling process monitoring and control," *Advances in Engineering Software*, Volume 41, Issue 5, pp. 754–761, Mei 2010.