

PERANCANGAN ALGORITMA DAN SISTEM GERAKAN PADA ROBOSOCER R2C R9 (ROBOTIS GP)

THE DESIGN OF ALGORITHM AND MOTION SYSTEM FOR ROBOSOCER R2C R9 (ROBOTIS GP)

Kurnia Sanjaya¹, Saptadi Nugroho², Daniel Santoso³

Fakultas Teknik Elektro dan Komputer Jurusan Teknik Elektro
Universitas Kristen Satya Wacana – Salatiga

¹kurniasanjaya93@gmail.com, ²saptadi_nugroho@yahoo.com, ³danvicz@yahoo.com

Abstrak

Robot GP standar pabrik perlu dikonstruksi ulang untuk memenuhi spesifikasi ukuran robot pada KRSBI 2015 sehingga *motion* awal yang dimiliki oleh robot sama sekali tidak dapat digunakan dan *motion* yang dimiliki robot harus dibuat ulang. Oleh karena itu, proyek ini dirancang untuk membuat sistem kontrol dan algoritma gerakan pada robot GP. Algoritma gerakan yang ditanamkan pada robot merupakan algoritma *cut motion* dimana robot dapat menghentikan gerakan dan langsung menyambung gerakan lain tanpa menyamakan kaki. Untuk dapat melakukan *cut motion*, *motion* berulang (maju, mundur, geser, putar, dan geser putar) dipisah menjadi empat bagian *motion*, yaitu angkat kaki kanan, angkat kaki kiri, tapak kaki kanan, dan tapak kaki kiri, serta diberikan penanda yang membedakan saat robot menapakkan kaki kanan atau kaki kiri. Dengan cara ini, robot dapat mengetahui sedang melangkah kaki kanan atau kiri. Jika robot diberi perintah gerakan *motion* berulang lainnya, robot dapat langsung mengetahui harus mulai dari kaki kanan atau kaki kiri. Untuk menjaga keseimbangan robot dalam melakukan *motion* tersebut, robot dibantu oleh sensor *gyroscope*. Robot dapat melakukan *cut motion* dengan keberhasilan 100% dan peralihan *motion* berulang dengan keberhasilan mencapai lebih dari 75%.

Kata Kunci: robot, gerakan, peralihan, *cut motion*, sensor

Abstract

The standard GP robot had to be reconstructed to fulfill the robot's minimal size specification in the Indonesian Robot Soccer Competition 2015 so GP robot could not use its motions at all and its motions had to be recreated. This project was designed to create a motion control system and a motion algorithm for the GP robot. The cut motion algorithm was used to stop the current motion and immediately continued with another motion. A continuous motion that was used in the cut motion algorithm was divided into four sub motions. They were lifting the right foot, lifting the left foot, stepping the right foot and stepping the left foot. There was a flag to differ whether the robot was stepping the right or the left foot. The robot would know which sub motion should be used for the motion transition using this flag when the robot had to change its current motion. To keep the robot in balance condition, the robot uses gyroscope sensor. The robot could cut the current motion with 100% success rate and above 75% to do the motion transition.

Keywords: robot, motion, transition, *cut motion*, sensor

Tanggal Terima Naskah : 05 Oktober 2015
Tanggal Persetujuan Naskah : 15 Januari 2016

1. PENDAHULUAN

Pada tahun 2014 R2C membeli robot sepak bola baru dengan merek robotis GP, namun robot ini tidak dimainkan saat berlomba karena robot ini masih kurang siap untuk diperlombakan oleh karena beberapa faktor. Salah satu faktor utamanya adalah *motion* robot yang masih belum seimbang. Robot GP ini saat keluaran awal dari pabrik sangat seimbang dan pergerakannya cepat, namun seiring dengan kebutuhan robot untuk ditambah komponen-komponen, seperti HP *android*, kamera, mikrokontroler, dan komponen lainnya untuk menunjang robot dapat bermain sepak bola dan untuk memenuhi spesifikasi ukuran robot untuk dapat diikutsertakan dalam lomba, robot dikonstruksi ulang. Hal ini membuat *motion* awal yang dimiliki oleh robot sama sekali tidak dapat digunakan, sehingga *motion* yang diperlukan oleh robot untuk bermain sepak bola harus dibuat ulang dari awal. KRSBI tahun 2015 ini memiliki peraturan dan kondisi lapangan yang baru. Lapangan yang digunakan untuk perlombaan sepak bola ini menjadi lebih besar, dari semula berukuran 6 m x 4 m menjadi berukuran 9 m x 6 m. Ukuran minimal robot juga diperbesar untuk menyesuaikan dengan kondisi lapangan. Ukuran tinggi robot yang tahun 2014 hanya 38 cm menjadi 52 cm sehingga menyebabkan pembuatan *motion* yang seimbang akan menjadi lebih susah.

Sistem pergerakan yang dimiliki robot ini juga masih tidak *realtime*, karena robot belum dapat memotong gerakan yang sedang dilakukannya. Jadi, semisal robot sedang diperintahkan untuk maju dimana setiap satu gerakan maju membutuhkan waktu dua detik dan kemudian ada kondisi dimana robot harus geser kanan, robot akan menyelesaikan gerakan maju terlebih dahulu, kemudian baru bisa diberi perintah geser kanan. Untuk mengatasi permasalahan tersebut, diperlukan perancangan sebuah sistem kontrol dan algoritma supaya robot GP ini dapat bermain sepak bola dan dapat bersaing dengan tim lain dari perguruan tinggi di Indonesia dalam Kontes Robot Sepak Bola Indonesia (KRSBI) yang akan datang.

2. KONSEP DASAR

2.1 Servo Controller

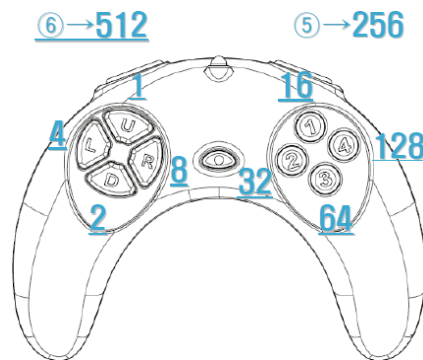
Pada robot *humanoid* GP digunakan servo *controller* tipe CM530. Servo *controller* ini memiliki enam *port external*, 1 *port* komunikasi, dan 5 *port Dynamixel*. *Port external* hanya terdiri dari *pin* ADC dan GPIO yang mendukung penggunaan sensor analog. *Port* komunikasi yang digunakan untuk komunikasi *wireless* seperti menggunakan zig-110A, BT-110A, modul penerima IR, dan lain-lain. *Port dynamixel* digunakan untuk mengontrol servo yang digunakan pada robot.



Gambar 1. CM530[1]

2.2 Kontrol Dasar Robot

Robotis menyediakan *controller* RC-100A sebagai *controller* dasar robot GP ini.



Gambar 2. *Controller* RC-100A [2]

RC-100A memiliki 10 tombol perintah yang masing-masing tombol mewakili sebuah nilai seperti yang terlihat pada gambar 2. Penekanan sebuah tombol atau kombinasi dari dua tombol atau lebih akan mengirimkan sebuah paket data ke CM-530. Paket data yang dikirimkan memiliki format seperti pada gambar 3.



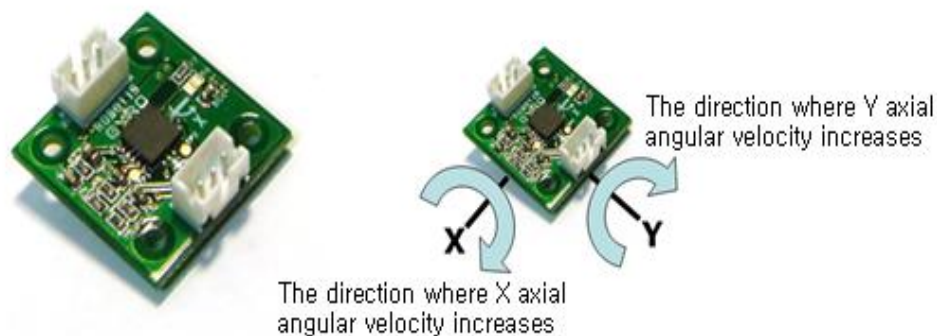
Gambar 3. Format paket data yang dikirimkan ke CM-530 [2]

Setiap segmen dari paket data memiliki panjang 8 bit. Segmen pertama dan kedua selalu bernilai FF dan 55. Delapan bit MSB penjumlahan nilai dari 1 tombol atau lebih yang ditekan akan dimasukkan ke data H sedangkan delapan bit LSB akan dimasukkan ke data L. Negasi data H dan negasi data L adalah 1's *complement* dari data H dan data L.

CM-530 akan membandingkan paket data yang diterima dengan paket data yang sudah didefinisikan saat *Roboplus Task* diprogram. Dalam perlombaan *controller* RC-100A ini tidak digunakan. Sebagai ganti *controller* RC-100A, paket data yang dikirim oleh RC-100A dikirim melalui mikrokontroler berdasarkan perintah dari *smartphone* android.

2.3 Sensor Gyroscope

Sensor *gyroscope* merupakan perangkat yang memanfaatkan gravitasi bumi untuk membantu menentukan orientasinya [3]. Sensor *gyroscope* yang digunakan pada robot GP adalah sensor tipe GS-12.

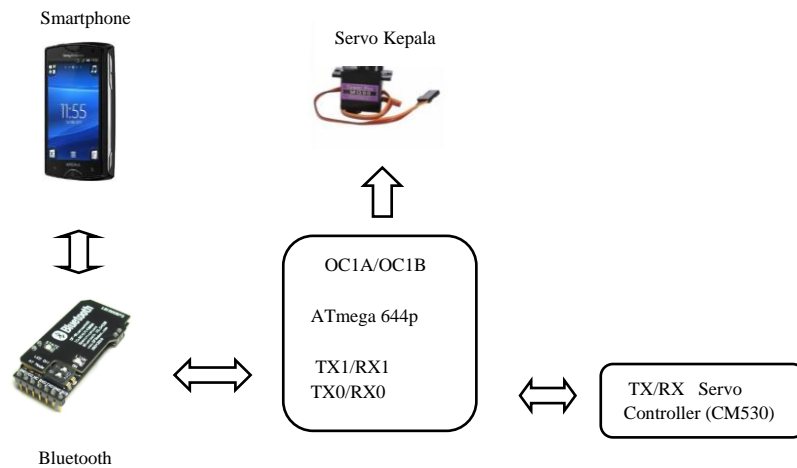


Gambar 4. Sensor *gyroscope* GS-12[4]

3. PERANCANGAN

3.1 Sistem Kontrol

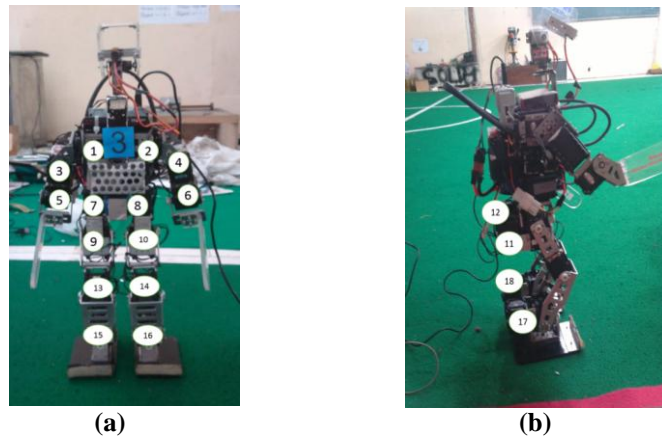
Sistem kontrol pergerakan pada robot dibagi menjadi dua bagian utama, yaitu sistem kontrol utama dan sistem kontrol servo yang mengatur pergerakan servo pada robot. Sistem kontrol utama terdiri dari *smartphone* dan mikrokontroler ATmega 644p. *Smartphone* berfungsi sebagai *processor* utama yang menentukan gerakan yang akan dilakukan oleh robot, mikrokontroler ATmega 644p berfungsi sebagai sistem kontrol yang mengatur kesinambungan antara *smartphone* dan servo *controller* (CM530). CM530 menerima perintah gerakan dari *smartphone* yang sudah diterjemahkan oleh mikrokontroler ATmega 644p menjadi bahasa yang dimengerti oleh CM530, kemudian CM530 mengatur pergerakan servo pada robot.



Gambar 5. Blok diagram sistem

3.2 Konstruksi Robot

Robot GP ini menggunakan bahan perpaduan antara aluminium dan *acrylic* untuk *body*. Perpaduan kedua bahan ini dimaksudkan agar robot memiliki bobot yang ringan sehingga dalam bergerak diperoleh kecepatan dan keseimbangan yang baik. Robot ini memiliki 18 *degree of freedom*, dengan rincian enam di setiap kaki, dan tiga di setiap lengan. Servo yang digunakan pada robot adalah servo seri AX-12A, AX-18A, dan MX-28. AX-12A digunakan pada bagian lengannya. AX-18A digunakan pada bagian kaki dengan id 7, 8, 11, 12, 17, dan 18. MX-28 digunakan pada bagian kaki dengan id 9, 10, 13,14,15, dan 16.

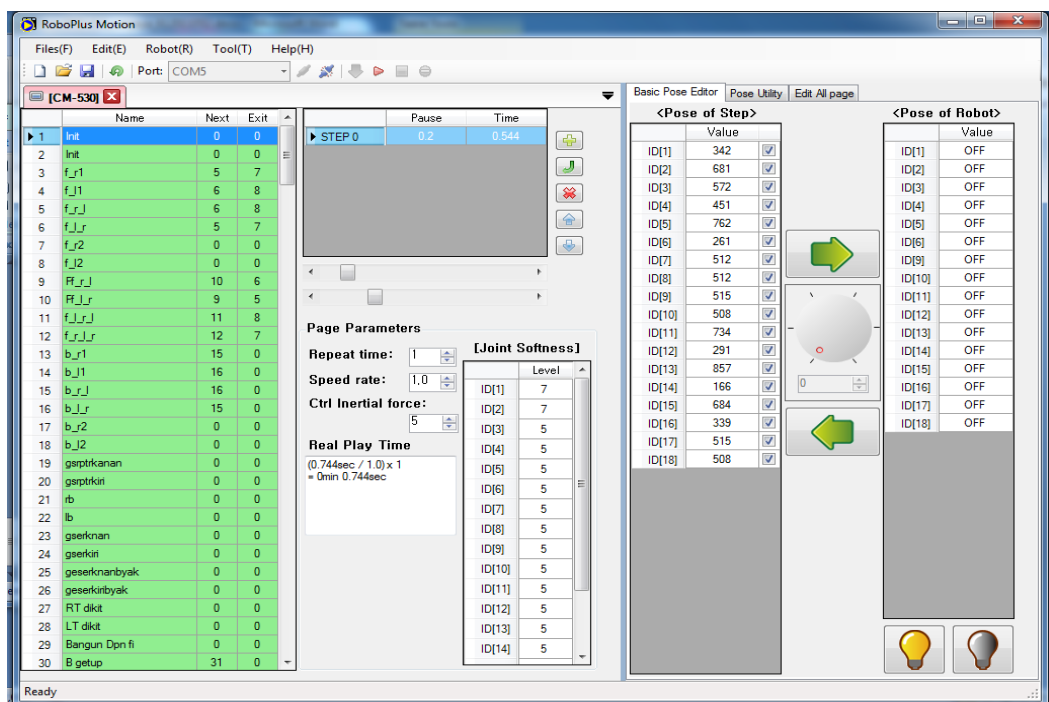


Gambar 6. (a)Perancangan mekanik robot dari depan (b)Perancangan mekanik robot dari samping

3.3 Bagian Software

3.3.1 Roboplus Motion

Roboplus *motion* merupakan program yang digunakan untuk membuat *motion* dimana nantinya *motion* tersebut akan dipanggil di roboplus *task* untuk membuat robot bergerak sesuai dengan *motion* tersebut. *Motion* tersebut disimpan dalam *motion page*. Roboplus *motion* menyediakan 255 *motion page* dimana masing-masing *motion page* memiliki enam *step* yang dapat diisi gerakan-gerakan untuk membentuk kesatuan gerakan tertentu. Gerakan tersebut dibuat dengan menentukan nilai pada masing-masing servo di *pose of step*. Pada *motion page* terdapat pilihan untuk mengatur *motion page* lanjutan dan *motion page* akhir dari *motion page* tersebut dengan cara mengisi kolom *next* dan *exit* dengan nomor *motion page* yang dituju. Seperti terlihat pada gambar 7 jika robot diberi perintah untuk melakukan *motion page* nomor 3, maka robot akan melakukan urutan *motion page* nomor 3 kemudian nomor 5 kemudian nomor 6, kembali ke nomor 5 lagi. Jika perintah berhenti diberikan ke robot, maka robot akan melakukan *motion page* nomor 8 seperti yang terdapat pada kolom *exit* pada *motion page* terakhir robot diberikan perintah.

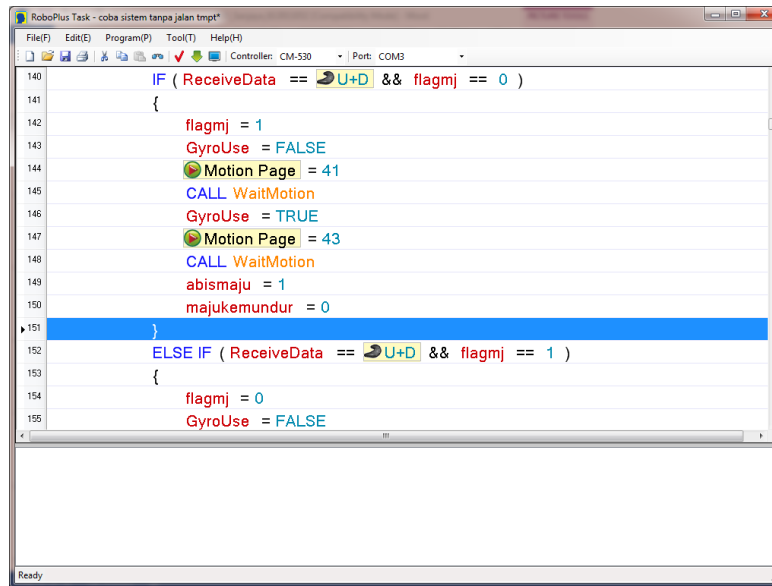


Gambar 7. User Interface Roboplus Motion

3.3.2 Roboplus Task

Roboplus *Task* merupakan program dari roboplus yang digunakan untuk meletakkan algoritma pergerakan robot dan untuk memanggil *motion* yang sudah dibuat di roboplus *motion*. Pada roboplus *task* terdapat dua fungsi yang berjalan secara terpisah. Fungsi pertama, yaitu fungsi *main* yang merupakan fungsi yang pertama kali dijalankan dan di dalamnya terdapat *code* untuk mengatur pergerakan servo pada robot. Jumlah variabel yang dapat digunakan di roboplus *task* adalah 50 variabel dan memori maksimalnya adalah 16124 byte. Fungsi lain yang jalannya terpisah dari jalannya fungsi *main* adalah fungsi *callback*. Pada fungsi *callback* tidak dapat diisi *code* yang membutuhkan banyak waktu dan hanya memiliki maksimal ukuran 512 byte, jadi tidak diperbolehkan adanya penggunaan perulangan pada fungsi *callback* ini. Pada fungsi ini

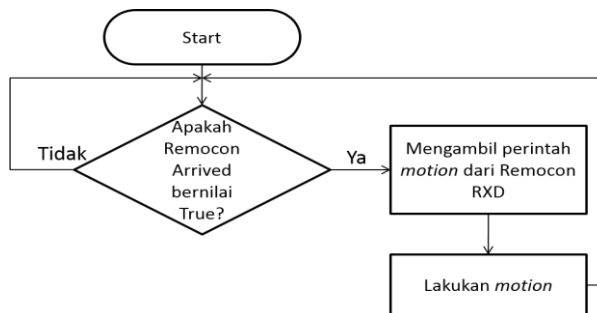
juga tidak diperbolehkan adanya label, *jump*, dan pemanggilan fungsi lainnya. Fungsi *callback* ini biasanya berisi *code* untuk mengatur sensor *gyroscope* yang ada. Sensor *gyroscope* digunakan untuk membuat robot lebih seimbang dalam melakukan gerakan-gerakan yang ada.



Gambar 8. User interface roboplus task

3.3.3 Algoritma Cut Motion

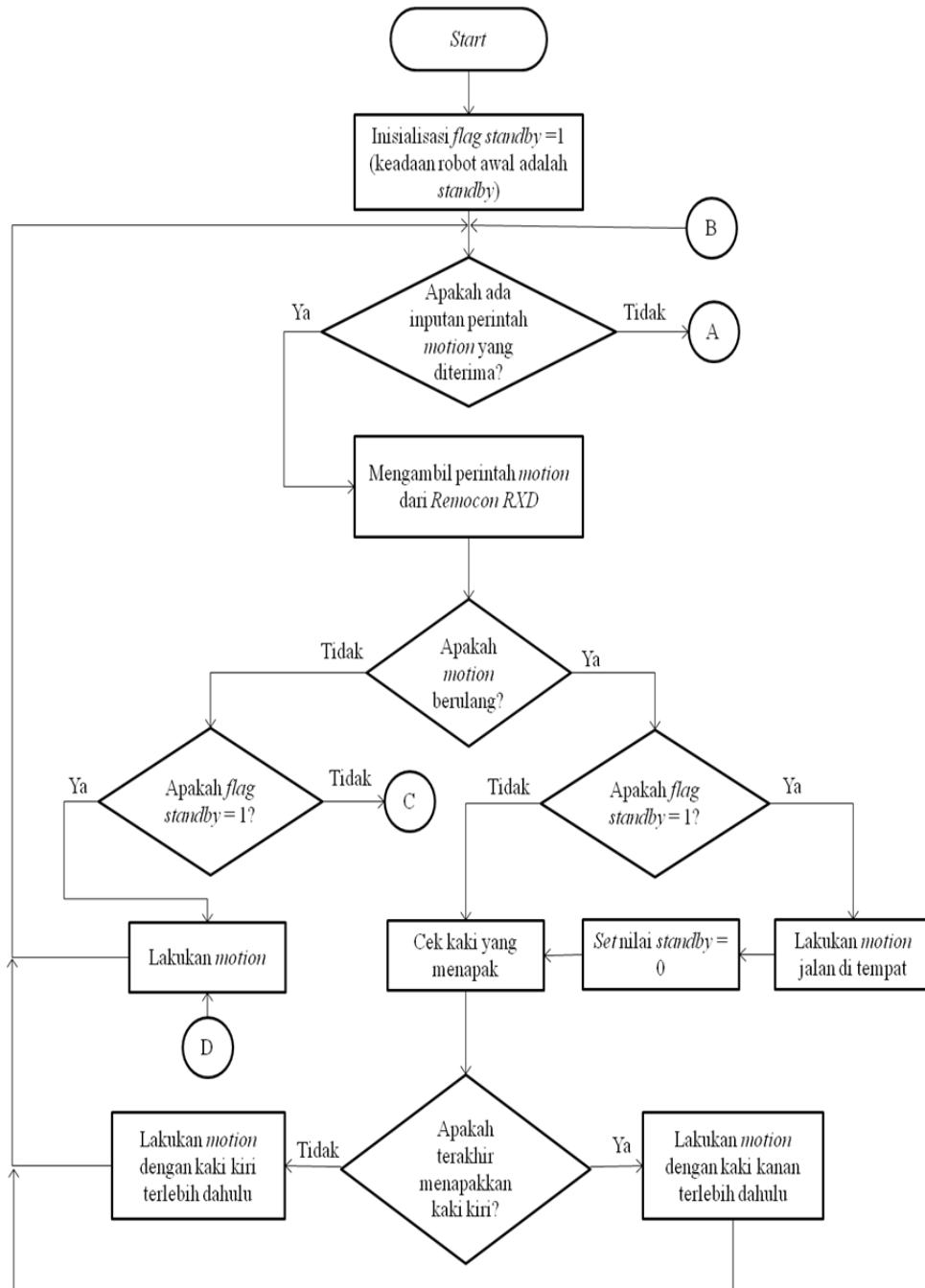
Pada bagian ini akan dijelaskan dua buah *flowchart* yang digunakan untuk menggambarkan algoritma robot saat bergerak. *Flowchart* pertama merupakan *flowchart* yang digunakan tahun lalu saat robot pertama kali dibeli. *Flowchart* kedua merupakan *flowchart* yang digunakan sekarang dengan algoritma *cut motion*.



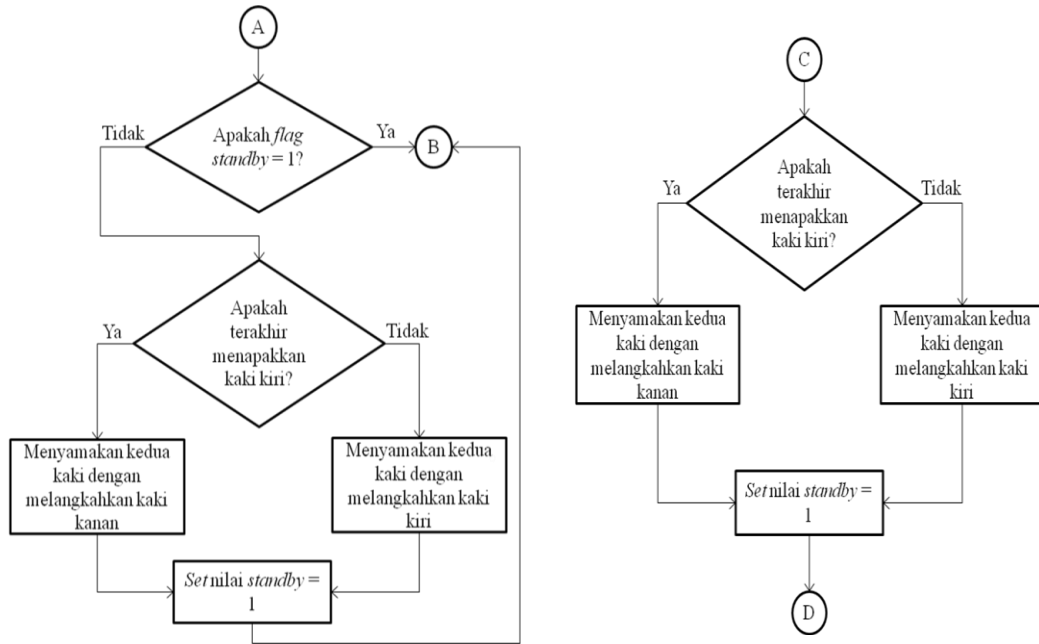
Gambar 9. Diagram alir algoritma bawaan robot

Pada Gambar 9 terlihat bahwa robot hanya menerima perintah *motion* yang ada kemudian melakukannya tanpa membedakan mana *motion* berulang mana *motion* sekali gerak. *Motion* berulang adalah *motion* yang dilakukan dengan mengulang pergerakan kaki kanan dan kaki kiri robot, seperti maju, geser, geser putar, putar dan mundur. *Motion* berulang dapat dipotong pergerakannya. *Motion* sekali gerak adalah *motion* yang terdiri dari satu gerakan utuh dan tidak dapat dipotong pergerakannya. Pada algoritma yang seperti ini *motion* berulang dibuat dengan memberikan nilai pada kolom *next* dan *exit* yang tersedia pada Roboplus *motion*. Robot hanya memanggil *motion page* awalnya saja kemudian dengan sendirinya robot akan berjalan sesuai dengan nilai *next* dan *exit* yang diberikan di Roboplus *motion*. Dengan diisinya nilai *next* dan *exit* maka tidak mungkin untuk dilakukan *cut motion* saat robot melakukan gerakan berulang karena robot hanya

akan menjalankan *motion page* yang ada pada kolom *exit* jika robot diberi perintah *motion* lain. Oleh sebab itu, nilai *next* dan *exit* pada RoboPlus *motion* dikosongi sehingga algoritma yang memungkinkan robot untuk melakukan *cut motion* dibuat. *Flowchart* algoritma *cut motion* dapat dilihat pada gambar 10.



Gambar 10. Diagram alir *cut motion*



Gambar 10. Diagram alir *cut motion* (lanjutan)

Flag standby bernilai 1 berarti robot sedang dalam kondisi diam di tempat, dan sebaliknya jika *flag standby* bernilai 0 berarti robot tidak dalam kondisi diam di tempat. *Flowchart* bagian A dan C berfungsi untuk menyamakan kaki kanan dan kaki kiri robot. Perbedaan *flowchart* bagian A dan C, yaitu pada *flowchart* bagian A masih ada kemungkinan jika *flag standby* bernilai 1, namun pada *flowchart* bagian C *flag standby* sudah pasti bernilai 0.

4. PENGUJIAN SISTEM

4.1 Pengujian *Cut Motion*

Pengujian *cut motion* dilakukan sebanyak 30 kali pada masing – masing gerakan berulang yang dilakukan robot. Kriteria keberhasilan, yaitu jika robot dapat menghentikan pose robot di saat pengiriman perintah diberhentikan. Hasil pengujian dapat dilihat pada tabel 1.

Tabel 1. Tabel pengujian *cut motion*

	Maju	Mundur	Geser	Putar	Geser Putar
Rata-rata keberhasilan(%)	100	100	100	100	100

Hasil pengujian menunjukkan bahwa persentase keberhasilan robot dalam melakukan *cut motion* adalah 100%.

4.2 Pengujian Peralihan *Motion* Berulang

Pengujian ini dilakukan dengan memotong *motion* berulang kemudian menyambungkannya dengan *motion* berulang lainnya. Pengujian dilakukan sebanyak 30 kali pada setiap peralihan *motion* berulang ke *motion* berulang lainnya. Rata-rata hasil pengujian dapat dilihat pada tabel 2. Nilai pada baris ke-i, kolom ke-j menunjukkan rata-rata hasil pengujian peralihan *motion* berulang-i ke *motion* berulang-j.

Tabel 2. Tabel pengujian peralihan *motion* berulang

	Maju	Mundur	Geser	Putar	Geser Putar
Maju	-	83,33 %	93,33 %	96,67 %	86,67 %
Mundur	83,33 %	-	90 %	93,33 %	80 %
Geser	90 %	86,67 %	-	93,33 %	83,33 %
Putar	96,67 %	90 %	93,33 %	-	86,67 %
Geser Putar	83,33 %	80 %	83,33 %	86,67 %	-

Keberhasilan robot dalam melakukan peralihan *motion* dari sebuah *motion* berulang ke *motion* berulang lainnya sudah mencapai lebih dari 75%. Keberhasilan robot dalam melakukan peralihan *motion* sangat bergantung pada *motion* yang dimiliki robot. Hal ini terlihat pada kegagalan yang lebih sering terjadi pada peralihan *motion* mundur dan *motion* geser putar karena *motion* mundur dan geser putar masih kurang sempurna.

5. KESIMPULAN

Berdasarkan perancangan, perealisasiannya, dan pengujian dapat ditarik kesimpulan sebagai berikut:

1. Robot dapat melakukan *cut motion* dengan cara mengosongkan nilai *next* dan *exit state* pada roboplus *motion* dan memanggil *motion page* secara terpisah pada roboplus *task*. Keberhasilan robot dalam melakukan *cut motion* atau menghentikan *motion* adalah 100%.
2. Keberhasilan robot dalam melakukan peralihan *motion* dari sebuah *motion* berulang ke *motion* berulang lainnya sudah mencapai lebih dari 75%. Keberhasilan robot dalam melakukan peralihan *motion* sangat bergantung pada *motion* yang dimiliki robot.

REFERENSI

- [1]. Robotis. CM-530, <http://support.robotis.com/en/product/controller/cm530.htm>; diakses 18 Februari 2015.
- [2]. Robotis. RC-100A, http://support.robotis.com/en/product/auxdevice/communication/rc100_manual.htm; diakses 10 September 2015.
- [3]. Goodrich, Ryan. Accelerometer vs. Gyroscope: What's the Difference?, <http://www.livescience.com/40103-accelerometer-vs-gyroscope.html>; diakses 12 Maret 2015.
- [4]. Robotis. Gyro Sensor, http://support.robotis.com/en/product/auxdevice/sensor/dxl_gyro.htm; diakses 9 September 2015.