

## Robot Scanner Gambar Menggunakan Intensitas Cahaya

David\*<sup>1</sup>, Sandy Kosasi<sup>2</sup>

<sup>1,2</sup>STMIK Pontianak

E-mail: \*<sup>1</sup>[david\\_liauw@yahoo.com](mailto:david_liauw@yahoo.com), <sup>2</sup>[sandykosasi@yahoo.co.id](mailto:sandykosasi@yahoo.co.id)

### Abstrak

Banyak teknologi robot yang dapat digunakan untuk meniru mesin-mesin dengan kemampuan digitalisasi. Salah satu teknologi robot yang terkenal sekarang adalah Lego Mindstorms NXT 2.0 dengan platform Mindstorm NXT-G. Pada Penelitian ini, dirakit dan dikodekan program sebuah robot scanner menggunakan intensitas cahaya. Metode yang digunakan adalah metode Mindstorms dengan robot Lego Minstorms NXT 2.0. Metode ini adalah metode sederhana yang terdiri dari empat tahapan yaitu mencari ide untuk robot, membangun robot, memprogram robot dan dokumentasi. Pemrograman robot dengan menggunakan bahasa pemrograman NXT-G digunakan untuk mendapatkan program yang bekerja terbaik. Pengujian V-model akan digunakan untuk menguji program utama. Robot yang dibuat dengan menggunakan Lego Mindstorms NXT 2.0 dapat melakukan scanning pada gambar dengan output pada Layar NXT Brick. Robot dapat membedakan warna berdasarkan intensitas cahaya yang dipantulkan. Sensor warna mendeteksi pantulan infra merah pada permukaan kertas yang berwarna dengan intensitas tertentu. Hasil pendeteksian disimpan dalam memory dan sebagai pusat kendali dari robot ini menggunakan NXT Brick yang sudah termasuk dalam paket LEGO MINDSTORMS NXT 2.0.

**Kata Kunci** — Robot Scanner Gambar, Lego Mindstorms NXT 2.0, Intensitas Cahaya, NXT-G

### Abstract

A lot of Robot technology that can be used to act like machines with digital ability. Nowadays, one of the famous robot technology is Lego Mindstorms NXT 2.0 with Mindstorm NXT-G platform. In this research, created and coded by program, a scanner robot uses light intensity. Method that being used is Mindstorms method with Lego Mindstorms NXT 2.0. robot. This method is a simple method with four steps. They are find the idea for the robot, construct the robot, program the robot and documentation. Robot programming with NXT-G programming language is use to get the best working program. V-Model test will be use to test the main program. Robot that being created with Lego Mindstorms NXT 2.0 can do scanning to an image with output from the NXT brick screen. Robot can differentiate colors base on light intensity that has been reflected. Color censor detect infrared reflection at the colored paper surface with certain intensity. Result of the detection is being saved into memory and as the control center from this robot, it uses NXT brick that has been included in the LEGO MINDSTORMS NXT 2.0. package.

**Keywords** — Image Scanner Robot, Lego Mindstorms NXT 2.0, Light Intensity, NXT-G

### 1. PENDAHULUAN

Lego Mindstorms NXT 2.0 adalah teknologi robot yang terkenal saat ini. Robot ini menggunakan platform Lego Mindstorms NXT 2.0 yang merupakan penyempurnaan dari robot Lego seri RIS (Robotic Invention System) dan NXT 1.0. Seri NXT merupakan seri robot yang

lebih ramah dan relatif mudah untuk digunakan tanpa harus merekayasa sirkuit maupun memasang motor sendiri seperti generasi pendahulunya. Lego Mindstorms NXT 2.0 dilengkapi dengan sensor ultrasonic yang dapat meningkatkan kinerja robot [1].

Lego Mindstorm NXT memungkinkan untuk pembangunan robot dengan menggunakan beberapa motor dan juga sensor. Namun daya pemrosesan, memori, dan kemampuan komunikasi LEGO NXT dapat dimanfaatkan untuk keperluan lainnya [2]. Melihat hal tersebut diatas, muncul gagasan untuk membuat robot NXT yang mempunyai kemampuan layaknya scanner seperti robot scanner gambar. Penulis merakit dan membuat program Robot Scanner Gambar berdasarkan Intensitas Cahaya Lego Mindstorms NXT 2.0. Robot Scanner Gambar Berdasarkan Intensitas Cahaya ini akan me-scan gambar berdasarkan intensitas cahaya. Menggunakan sensor-sensor yang tersedia dalam paket LEGO Mindstorm NXT, yaitu sensor cahaya untuk mendeteksi benda.

Penelitian menggunakan sensor kamera pada robot dengan pengolahan citra digital pada sensor kamera sebagai pengendali arah gerak robot line follower, penelitian ini bertujuan untuk membuat sistem (hardware dan software) yang dapat digunakan untuk mengendalikan arah gerak robot line follower dengan menggunakan sensor kamera [3].

Tujuan penelitian adalah menghasilkan sebuah robot yang dapat melakukan Scan terhadap gambar berdasarkan perbedaan intensitas cahaya dengan menggunakan satu set Lego Mindstorm NXT 2.0. Obyek dalam penelitian ini adalah gambar yang berukuran sebesar kertas A4 atau lebih kecil. Hasil dari scanning akan ditampilkan pada layar NXT Brick dan dikirim ke komputer. Sensor yang digunakan dalam merakit robot adalah sensor cahaya dan warna RGB yang terdapat dalam paket Lego Mindstorms NXT 2.0[4][5]. Sensor ini adalah alat yang memberikan penglihatan pada robot (sensor ultrasonic salah satunya). Sensor ini memiliki 3 fungsi utama. Pertama, sensor dapat membedakan tempat gelap dan terang berdasarkan intensitas cahaya. Kedua, sensor dapat mendeteksi 6 warna yang berbeda dan yang terakhir dapat membaca intensitas cahaya permukaan suatu benda. Sensor ini juga dapat digunakan sebagai lampu warna.

Penelitian ini merujuk pada penelitian Mobile Robot pendeteksi warna bola dengan sensor kamera CMUcam3[6]. Penelitian ini menghasilkan Mobile Robot yang dapat mendeteksi warna secara otomatis dan dapat dikendalikan secara otomatis tanpa terkena gangguan dari luar. Penelitian lainnya adalah perancangan robot pendeteksi objek berdasarkan warna dengan sensor kamera sebagai media pembelajaran[7]. Penelitian ini bertujuan untuk mengetahui unjuk kerja dan tingkat kelayakan media pembelajaran berupa robot pendeteksi objek berdasarkan warna dengan sensor kamera serta untuk mengetahui peningkatan hasil belajar peserta didik dengan menggunakan media pembelajaran robot pendeteksi objek. Peningkatan prestasi belajar peserta didik yang dimaksud adalah pengetahuan peserta didik dalam pembelajaran robot vision. Penelitian mengenai robot scanner gambar menggunakan sensor warna dan sensor cahaya yang dipengaruhi oleh ontensitas cahaya.

## 2. METODE PENELITIAN

Bentuk Penelitian yang digunakan adalah studi literatur dan riset eksperimental. Perancangan tidak harus menyerupai bentuk asli dari alat yang ditiru, hanya langkah atau cara kerja dari alat yang tersebut. Pengumpulan data sekunder akan digunakan pada penelitian ini. Data sekunder merupakan sumber data penelitian yang diperoleh peneliti secara tidak langsung melalui media perantara (diperoleh dan dicatat pihak lain). Data sekunder umumnya berupa bukti, catatan atau laporan historis yang telah tersusun dalam arsip yang dipublikasikan dan tidak dipublikasikan.

Teknik pengumpulan data yang dilakukan penulis adalah studi literatur. Studi literatur dilakukan dengan mengumpulkan teory dari buku, media, pakar ataupun dari hasil penelitian orang lain yang bertujuan yang berhubungan dengan robot Lego Mindstorms NXT 2.0, baik dari segi perangkat keras dan perangkat lunak. Metode yang diterapkan adalah metode Mindstorms.

---

Empat tahap utama dalam metode Mindstorms yang memandu proyek pembuatan robot NXT, yaitu: Mendapatkan ide untuk robot, Membangun robot, Memprogram robot, dan Mendokumentasikan robot[1]. Dengan menggunakan metode ini pembuatan robot dapat dilakukan lebih efektif terutama dalam konstruksi dan pemrograman robot. Pengujian robot lebih ditekankan pada perangkat lunak (software)[1].

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Mendapatkan Ide untuk Robot

Sebuah cara yang baik untuk mengumpulkan ide-ide adalah dengan mengamati contoh-contoh proyek yang terdapat di situs Mindstorms, berbagai kreasi NXT yang dibuat oleh penggemar LEGO lainnya. Penulis mendapat dua ide untuk membangun robot. Ide pertama adalah membangun sebuah robot *printer*. Robot akan dibuat dengan kemampuan menggambar gambar yang dimasukkan. Tetapi ide ini tidak direalisasikan karena untuk memasukkan gambar ke komputer memerlukan program bantu lain. Penambahan komponen berupa pen atau spidol juga diperlukan untuk membangun robot *printer*.

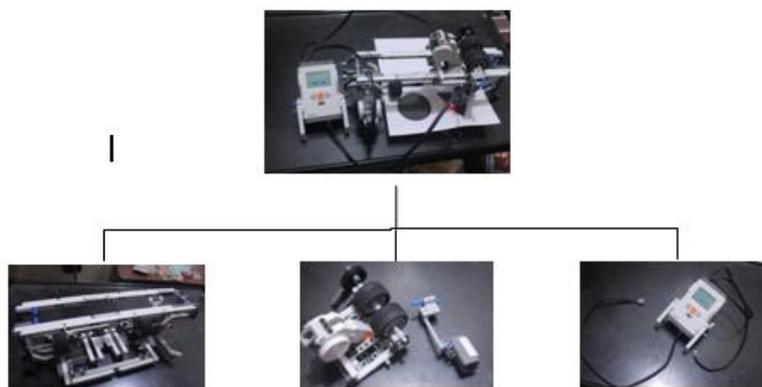
Ide kedua adalah membuat *scanner*. *Scanner* dapat bekerja tanpa program bantuan lain. Pembangunan *scanner* terinspirasi dari *printer* pada ide pertama. *Scanner* akan dibangun mirip dengan *printer*. Cara kerja *scanner* adalah menggunakan 2 buah motor agar dapat melakukan *scan* untuk posisi x dan y. Satu motor akan bergerak ke kiri dan ke kanan untuk melakukan *scan* pada posisi x dan satu motor berfungsi menggulung kertas seperti pada mesin *printer*. Ide kedua inilah yang akan direalisasikan pada penelitian ini.

PEAS ( *Performance Environment Actuators Sensors*) [8]:

<i>Agent</i>	: Robot <i>Scanner</i>
<i>Performance Measure</i>	: Gambar hasil <i>scan</i> pada layar NXT Brick
<i>Environment</i>	: Kertas gambar hitam putih
<i>Actuators</i>	: Servo Motor
<i>Sensor</i>	: Sensor Cahaya

#### 3.2. Membangun Robot

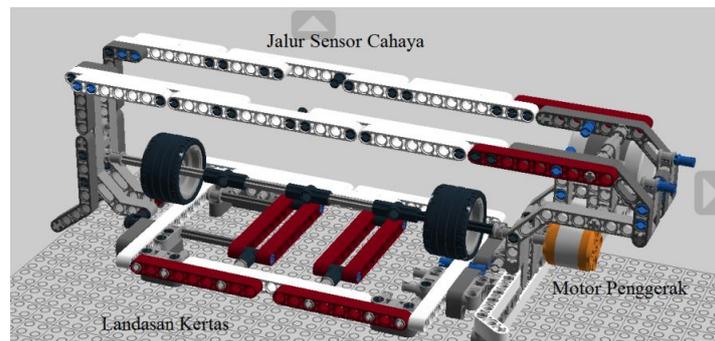
Pembangunan robot akan dilakukan dengan merakit beberapa *sub-assemblies* secara terpisah. Pembagian robot kedalam beberapa *sub-assemblies* bertujuan untuk mempermudah proses perakitan robot. Sistem pembagian *sub-assemblies* sangat memudahkan dalam memodifikasi robot tanpa harus membongkar robot secara keseluruhan, hanya cukup bagian atau modul (*sub-assemblies*) yang bersangkutan saja yang dibongkar. Membuat robot dalam beberapa modul sangat efektif [9]. Gambar 1 menunjukkan Sub-Assemblies Robot Scanner.



Gambar 1. Sub-Assemblies Robot Scanner

Proses *trial and error* akan diterapkan pada proses ini. Proses ini adalah proses bereksperimen dengan bagian-bagian robot yang berbeda dan teknik membangun hingga mencapai hasil yang diinginkan. Proses ini akan lebih mengarah pada percobaan pembangunan bagian-bagian berbeda pada robot untuk mendapatkan bentuk fisik robot yang diinginkan. Setelah membangun bentuk fisik robot, pengujian akan dilakukan untuk memastikan bahwa bagian yang telah dibangun dapat berfungsi sesuai tujuannya. Ide awal dapat berubah selama melakukan proses *trial and error*.

Lego Digital Designer (LDD) merupakan program bantu yang akan dipakai saat membangun robot. Program ini memungkinkan untuk merakit robot secara digital. Hal ini akan mempermudah dan mempercepat pekerjaan karena penulis tidak perlu bersusah payah membongkar dan merakit secara manual yang cukup memakan waktu. Gambar 2 menunjukkan desain pada program LDD.



Gambar 2. Desain pada program LDD

### 3.3. Memprogram Robot

Memprogram robot akan dimulai dengan menentukan bahasa pemrograman yang akan dipakai. Penentuan ini akan mempermudah proses pemrograman karena bahasa pemrograman disesuaikan dengan kemampuan programmer. Bahasa pemrograman yang digunakan oleh penulis adalah bahasa pemrograman NXT-G yang merupakan bahasa pemrograman yang sudah disediakan ketika kita membeli Lego Mindstorm NXT 2.0. Proses penulisan program, menguji, dan merevisi program akan terus dilakukan. Program yang dibuat akan disuntikkan kedalam NXT Brick dan langsung diuji. Jika program tidak bekerja dengan baik maka proses revisi dilakukan. Proses revisi dapat mempengaruhi kode program maupun bentuk fisik robot yang telah kita rakit. Penulis dapat saja melakukan modifikasi pada program dan *sub-assembly* yang tidak sesuai. Hal ini untuk mendapatkan program dan bentuk fisik yang bekerja secara optimal. Jika robot telah bekerja sesuai dengan keinginan maka tahap pemrograman robot telah selesai.

Program Robot *Scanner* terdiri dari dua buah variable. Variable tersebut adalah variable Number1 dan Number2. Variable Number1 berfungsi sebagai penyimpanan nilai koordinat X sedangkan nilai Y akan disimpan pada variable Number2. Nilai variable Number1 dipengaruhi oleh pergerakan Motor A dan Number2 oleh pergerakan Motor B.

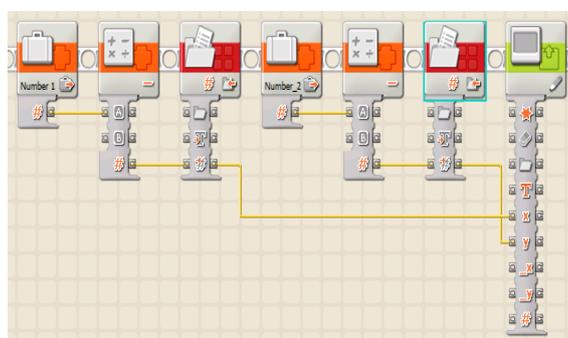
Program robot dimulai dengan *Switch* untuk tombol *Clear/Back*. Tombol yang ada pada NXT Brick ini berfungsi untuk menghentikan semua proses yang sedang berlangsung secara paksa jika ditekan. Layar NXT Brick akan *ter-reset* juga bila tombol ini ditekan. Apabila tidak ditekan maka proses akan dilanjutkan pada tahap berikutnya.

Looping program dilakukan pada tahapan selanjutnya. Looping ini akan berakhir sesuai dengan waktu yang telah ditentukan. Penentuan waktu loop berpengaruh terhadap lamanya proses *scanning*. Oleh karena itu dalam menentukan waktu looping perlu diperhatikan lamanya robot dalam melakukan *scanning* terhadap gambar. Nilai waktu yang ditentukan dalam loop ini adalah 300 atau sama dengan 5 menit.

Proses selanjutnya adalah membuat sebuah variable yang bernama Number1. Variable ini akan menjadi posisi X pada layar. Nilai awal yang diberikan adalah 0. Motor A akan

bergerah arah maju membawa Motor Scanner untuk melakukan proses scanning. Pada pergerakan motor A sebesar 17derajat maka nilai dari Number1 akan bertambah 1. Seiring pergerakan Motor Scanner, sensor cahaya bertugas membaca gambar. apabila mendapatkan putih maka tidak ada proses yang dilakukan tetapi jika mendapatkan warna hitam maka program akan memasukan nilai dari variable Number1 pada posisi X dan Number2 pada posisi Y selanjutnya akan ditampilkan pada layar NXT Brick. Nilai variable Number2 yang masih kosong pada pembacaan warna pertama tidak akan berpengaruh karena posisi Y akan ada pada saat scanning warna kali kedua. Block program pada proses penambahan nilai variable dapat dilihat pada gambar 3.

Proses penambahan nilai variable akan berpengaruh pada saat scanning kali kedua oleh Motor Scanner. Nilai dari variable Number1 dan Number2 masing-masing dikurangi 1 dan dimasukkan pada posisi X dan Y pada layar. Pengurangan dilakukan agar posisi X dan Y menjadi tepat. Pada saat berlangsungnya proses ini penyimpanan akan dilakukan pada file Image.txt. Layar NXT Brick akan menampilkan gambar hasil scan.



Gambar 3. Block program pada proses penambahan nilai variable

Setelah proses scan pertama oleh Motor Scanner selesai, motor B akan bekerja menaikkan kertas. Pada saat Motor B selesai menaikkan kertas maka variable Number2 yang kosong akan ditambahkan sebesar 1. Number2 yang berupa nilai dari posisi koordinat Y akan bertambah 1 pada layar. Artinya layar NXT Brick akan menampilkan titik hasil scan sensor cahaya pada baris berikutnya. Setelah Motor Scanner selesai melakukan scanning maka Motor A akan mengembalikan posisi Motor Scanner pada posisi awal. Pengembalian posisi ini bermaksud agar Motor Scanner dapat melakukan scanning pada baris berikutnya. Motor A akan bergerak sebesar 16 derajat arah mundur sebanyak 100 kali untuk mengembalikan posisi pada awal. Perintah mengembalikan Motor Scanner ke posisi awal dapat dilihat pada gambar 4.



Gambar 4. Perintah mengembalikan Motor Scanner ke posisi awal

Apabila Looping selama 5 menit yang telah ditentukan telah selesai, program akan menutup akses ke file Image.txt. dan menyimpan file tersebut kedalam memori pada NXT Brick. Layar NXT Brick akan menampilkan keseluruhan gambar yang telah selesai di-scan.

### 3.4. Mendokumentasikan Robot

Langkah keempat dan terakhir dalam metode Mindstorms adalah untuk mendokumentasikan robot yang telah selesai dirakit dan diprogram. Dokumentasi dilakukan agar setiap proses perakitan dari awal hingga akhir tercatat dan dapat digunakan kembali di waktu yang akan datang. Dokumentasi juga bermanfaat bagi penggemar Lego lainnya di internet. Proses dokumentasi dimulai dengan menulis ringkasan fungsi robot, konstruksi, dan pemrograman. Kamera digital dapat digunakan untuk mengambil beberapa cuplikan gambar dari robot.

Proses dokumentasi dilakukan dengan meringkat fungsi robot. Peringkasan fungsi robot akan dicatat pada sebuah buku catatan. Perancangan robot sesuai dengan ide awal akan dicatat pada buku catatan sebagai dokumentasi. Kerangka cara kerja robot, fungsi dan sebagainya akan dicatat dan digambarkan secara sederhana. Proses pengerjaan, perakitan, dan pengujian juga terdapat pada buku catatan. Buku catatan berfungsi sebagai pedoman dan pengingat pada saat melakukan proses perakitan robot, buku ini juga berperan penting pada saat pembuatan laporan penelitian. Mendokumentasikan konstruksi robot akan dilakukan dengan bantuan kamera digital. Kamera yang digunakan dapat berupa kamera handphone atau kamera khusus. Perakitan badan robot akan diabadikan dengan kamera. Setiap pemasangan komponen akan didokumentasikan secara lengkap agar pembaca atau penggemar Lego Mindstorms dapat memahami proses yang dilakukan. Proses *scanning* yang dilakukan oleh robot juga akan divideokan. Hasil dokumentasi perakitan robot dapat dilihat pada lampiran A *Building Instruction*.

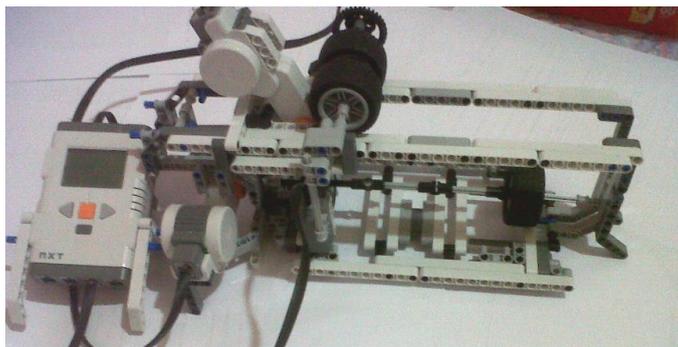
Pemrograman robot didokumentasikan dengan bantuan program yang sudah disediakan oleh *windows*. *Print screen* dan *Snipping Tools* dapat mengambil gambar yang ada pada layar monitor PC. Gambar-gambar block program disimpan dalam bentuk file JPG atau PNG dan properties dari block program akan ditampilkan juga. Pendokumentasian ini dapat mempermudah pembaca atau penggemar Lego dalam mengikuti proses pemrograman robot *scanner*. Hasil dokumentasi program dapat dilihat pada Lampiran B *Source Code*.

### 3.5. Pengujian Robot

Proses yang paling penting dalam perakitan robot adalah proses pengujian. Proses pengujian adalah proses yang dilakukan beriringan dengan proses perakitan dan proses pemrograman. Pada proses perakitan dan pemrograman robot dilakukan dengan sistem *trial and error*. Pengujian dengan sistem *trial dan error* inilah yang dilakukan untuk mendapatkan robot yang bekerja secara optimal. Pengujian robot akan difokuskan pada program robot. Program yang baik akan menghasilkan robot yang bekerja maksimal pula. Perubahan pada kode program mungkin terjadi pada saat proses pengujian. Bentuk robot juga akan disesuaikan dengan cara kerja program yang disuntikkan kedalam robot. Oleh karena itu pengujian dilakukan dengan menggunakan metode V-Model *software testing*.

Setiap Block program memiliki peranan masing-masing pada robot *scanner*. salah satu contohnya adalah satu Block program *Move* memiliki beberapa nilai seperti kecepatan motor, arah, derajat perputaran dan sebagainya. Oleh karena itu pengujian akan dilakukan pada setiap Block program. Hal ini dilakukan untuk mendapatkan data yang akurat.

Pengujian dilakukan dengan menyatukan semua *sub-assembly* robot menjadi satu robot *scanner* secara utuh. Setelah itu maka setiap komponen robot berupa Motor dan sensor akan dihubungkan ke NXT Brick. Motor A akan dihubungkan ke *port A*, Motor B akan dihubungkan ke *port B* dan Sensor Cahaya dihubungkan ke *port 3*. Gambar 5 di bawah adalah robot yang telah selesai disatukan.



Gambar 5. Robot Scanner yang akan diuji

Pengujian yang pertama adalah menguji Motor A. Motor A adalah motor yang berfungsi menggerakkan *scanner* di jalur yang telah disediakan pada *Base*. Motor A bergerak membawa sensor cahaya dan menghasilkan nilai *scan* pada koordinat X (Horizontal) pada layar. Kinerja Motor A akan mempengaruhi ketepatan robot dalam melakukan *scanning*. Pengujian dilakukan dengan mengubah nilai *Power* dari Block *Move* Motor A. Program *Scanner* terdapat dua block *Move* yang digunakan untuk mengendalikan Motor A. Block program *Move* tersebut adalah Block *Move* yang mengendalikan Motor A untuk bergerak maju dan Block *Move* yang memerintahkan Motor A bergerak mundur. Block program *Move* yang mengendalikan Motor A bergerak mundur hanya berfungsi sebagai penggerak motor *scanner* agar kembali ke posisi semula. Karena fungsinya yang hanya sebagai pengembali posisi Motor A maka *Power* pada Block program *Move* ini akan dimaksimalkan atau bernilai 100. Hal ini dilakukan agar mempercepat proses pengembalian Motor *Scanner*.

Motor B adalah motor yang bertanggung jawab dalam menggerakkan kertas ke dalam *scanner*. Block program yang mengendalikan motor B pada *software scanner* ini hanya satu. Block program tersebut adalah block program *Move* motor B bergerak mundur. *Properties* pada block program ini akan diganti dan diuji demi mendapatkan nilai *properties* yang bekerja maksimal. Nilai *Duration* pada motor B berpengaruh terhadap tinggi gambar hasil *scan*. Nilai *Duration* yang terlalu besar akan menghasilkan gambar yang tertarik terlalu tinggi sedangkan nilai *Duration* yang terlalu rendah akan menghasilkan gambar yang kerdil. Oleh karena itu untuk menghasilkan gambar hasil *scan* yang proporsional diperlukan nilai *Duration* yang tepat.

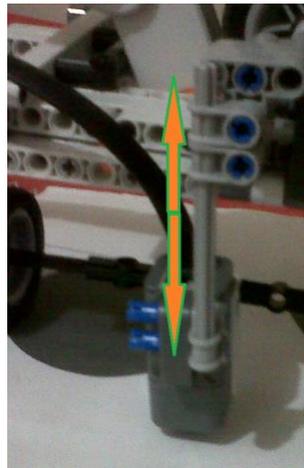
Tabel 1. Hasil pengujian Motor A

No.	Power	Hasil	No.	Power	Hasil
1	10	Motor A tidak mampu menggerakkan Motor <i>Scanner</i>	4	70	
2	30		5	90	
3	50		6	100	

Tabel 2. Hasil Pengujian motor B

No.	Duration	Hasil	No.	Duration	Hasil
1	1		3	5	
2	3		4	7	

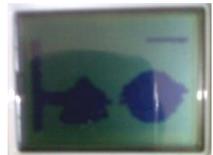
Pengujian dengan menempatkan sensor cahaya pada ketinggian tertentu terhadap kertas. Jarak ketinggian yang akan diuji adalah 0,5cm, 1cm, 1,5cm, 2cm, 2,5cm dan 3cm. Pengujian pada jarak 0 cm tidak dilakukan karena layar akan menampilkan layar hitam. Hal ini terjadi karena permukaan kertas menjadi gelap. Pengujian diatas ketinggian 3cm juga tidak dilakukan karena posisi sensor cahaya tidak dapat melebihi ketinggian 3cm. Pengaturan ketinggian pada Sensor Cahaya dapat dilihat pada gambar 6.



Gambar 6. Pengaturan ketinggian pada Sensor Cahaya.

Permukaan kertas sangat berpengaruh pada pengujian kali ini. Kertas yang akan di-*scan* tidak selalu mulus. Permukaan kertas akan bergelombang pada saat proses *scanning* berlangsung. Bergelombangnya permukaan kertas ini akan mengakibatkan kesalahan pembacaan intensitas cahaya oleh sensor cahaya. Hasil pengujian jarak ketinggian sensor cahaya terhadap permukaan kertas dapat dilihat pada Tabel 3.

Tabel 3. Hasil pengujian jarak ketinggian sensor cahaya terhadap permukaan kertas

No.	Jarak	Hasil	No.	Jarak	Hasil
1	0,5cm		4	2cm	
2	1cm		5	2,5cm	
3	1,5cm		6	3cm	

Pengujian kepekaan sensor cahaya terhadap warna adalah pengujian untuk mengetahui kemampuan robot dalam melakukan *scan*. Sensor cahaya dapat dengan mudah melakukan proses *scanning* pada gambar hitam putih. Gambar hitam putih memantulkan intensitas cahaya yang sangat berlawanan sehingga dapat dibedakan oleh sensor cahaya. Bertolak belakang dengan gambar hitam putih, gambar berwarna memantulkan intensitas cahaya yang bervariasi. Oleh karena itu diperlukan pengujian kepekaan sensor cahaya terhadap beberapa warna.

Proses pengujian dilakukan dengan men-*scan* gambar berwarna yang telah disediakan. Warna yang diujikan adalah *Standart Color* yang terdapat pada aplikasi *Microsoft Word*. Berikut adalah warna-warna yang termasuk *Standart Color* yang terdapat pada aplikasi *Microsoft Word* : a) Merah Tua (*Dark Red*); b) Merah (*Red*); c) Jingga (*Orange*); d) Kuning (*Yellow*); e) Hijau Muda (*Light Green*); f) Hijau (*Green*); g) Biru Muda (*Light Blue*); h) Biru (*Blue*); i) Biru tua (*Dark Blue*); dan j) Ungu (*Purple*). Gambar 7 Berikut ini adalah gambar yang akan diujikan.



Gambar 7. Gambar berwarna yang akan diuji

Pengujian dilakukan dengan mengubah nilai *Compare* yang ada pada block program sensor cahaya. Pengubahan nilai *compare* berfungsi untuk mengatur tingkat kepekaan sensor terhadap intensitas cahaya yang dipantulkan oleh permukaan kertas. Block program sensor cahaya dan nilai properties dapat dilihat pada gambar 8.



Gambar 8. Block program sensor cahaya dan nilai properties

Nilai *compare* pada block program diubah dengan interval 10. Pada nilai *compare* di bawah 30 sensor cahaya tidak dapat membaca semua warna termasuk hitam. Sedangkan pada nilai *compare* 60 dan di atasnya semua akan dianggap hitam. Hasil pengujian dapat dilihat pada tabel di bawah ini:

Tabel 4. Hasil pengujian kepekaan sensor cahaya terhadap gambar berwarna

No.	Nilai <i>Compare</i>	Hasil
1	20	Sensor cahaya tidak membaca warna pada kertas
2	30	
3	40	
4	50	
5	60	

Pengujian menunjukkan hasil kemampuan sensor cahaya ketika nilai *compare* diubah. Pada pengujian nilai *compare* 30 menghasilkan empat warna yang dibaca oleh sensor cahaya yaitu hijau, biru muda, biru dan biru tua. Sedangkan pada nilai 40 sensor cahaya dapat membaca 5 buah warna yaitu hijau, biru muda, biru, biru tua dan ungu. Kemampuan sensor cahaya berkurang saat nilai *compare* 50. Sensor tidak membedakan garis batas berwarna putih. Gambar hitam didapat saat nilai *compare* 60. Sensor cahaya akan menganggap semua permukaan berwarna gelap.

Pengujian robot *scanner* secara lengkap dilakukan pada tahap ini. Tahap ini berfungsi untuk menguji kinerja robot secara keseluruhan. Robot akan diuji dengan melakukan *scanning* terhadap berbagai macam gambar. Gambar yang disediakan berupa gambar hitam putih dan gambar berwarna. Pengujian Gambar huruf dapat dilihat pada gambar 9.



Gambar 9 Pengujian Gambar huruf

Gambar angka dan tulisan dengan ukuran 200 pixel atau lebih tinggi dapat menghasilkan gambar yang lebih jelas pada layar NXT Brick. Pada pengujian dengan pixel kurang dari 200 pixel, hasil *scan* tidak jelas. Huruf dan angka dengan pixel kurang dari 200 memiliki bagian hitam yang tipis sehingga sulit dibaca oleh sensor cahaya. Pengujian gambar bayangan muka dapat dilihat pada gambar 10.



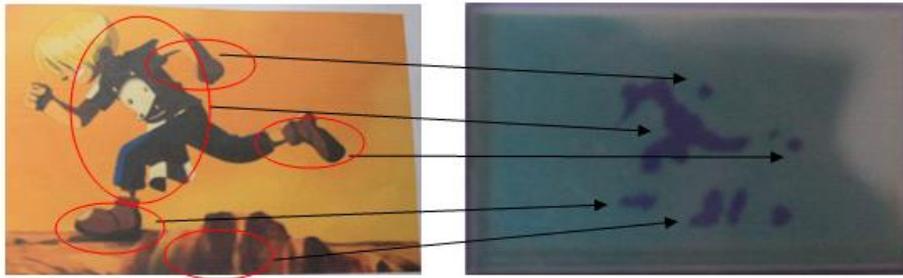
Gambar 10. Pengujian gambar bayangan muka

Berdasarkan hasil *scan* yang ditampilkan pada layar NXT Brick *scanner* tidak dapat membaca warna hitam berupa garis kecil. Hal ini dikarenakan intensitas cahaya yang dipantulkan oleh warna putih lebih kuat daripada warna hitam yang berupa garis. Pengujian Plat kendaraan dapat dilihat pada gambar 11.



Gambar 11. Pengujian Plat kendaraan

Berdasarkan hasil *scan* pada plat kendaraan bermotor hasil *scan* menunjukkan hasil berberda dengan pengujian sebelumnya. Pada pengujian gambar hitam putih warna hitam berupa garis tidak terbaca karena area warna putih yang lebih besar. Hal berlawanan terjadi pada pengujian kali ini. Warna putih berupa tulisan tipis tidak terbaca dengan sempurna atau tidak terbaca sama sekali karena area warna hitam yang lebih dominan akibatnya gambar tulisan putih menghasilkan titik yang tidak jelas pada layar NXT Brick. Pengujian gambar Berwarna dapat dilihat pada gambar 12.



Gambar 12. Pengujian gambar Berwarna

Sensor cahaya membaca beberapa warna saja. Warna yang memantulkan intensitas cahaya yang tinggi tidak akan dibaca oleh sensor cahaya sedangkan warna yang memantulkan intensitas cahaya rendah akan dibaca oleh sensor cahaya. Gambar di atas menunjukkan kemampuan sensor cahaya dalam melakukan *scanning*. Sensor cahaya membaca warna yang lebih gelap. Warna kuning yang dominan pada gambar tidak dibaca karena memantulkan intensitas cahaya yang lebih besar.

#### 4. KESIMPULAN

Berdasarkan seluruh penjelasan dan uraian laporan perancangan Robot Scanner Gambar Berdasarkan Perbedaan Intensitas Cahaya Berbasis Lego Mindstorms NXT 2.0 dapat diambil kesimpulan yang mungkin dapat dijadikan acuan dalam perancangan robot. Robot dapat bekerja sesuai dengan program yang telah dibuat dan dapat membedakan warna hitam putih dan beberapa warna lainnya. Tingkat kecepatan dalam proses *scanning* dapat diatur dengan mengubah nilai kecepatan motor servo A (penggerak Motor Scanner pada jalur scan) serta kecepatan motor servo B (penggerak kertas). Robot bekerja secara autonomous atau bekerja sendiri sesuai dengan program yang sudah di-upload sebelumnya. Robot tidak dapat melakukan *scanning* pada gambar tulisan dengan pixel kurang dari 200. Robot melakukan *scanning* pada gambar berwarna sesuai dengan intensitas cahaya yang dipantulkan oleh tiap warna.

#### 5. SARAN

Saran-saran yang dapat diberikan untuk pengembangan lebih lanjut, yaitu kemampuan scanner dapat di tingkatkan performanya menggunakan bahasa pemrograman lainnya dengan mengubah firmware NXT. Robot scanner gambar ini dapat dikembangkan lagi sebagai robot portrait yakni robot pelukis dimana hasil scan dapat dilanjutkan lagi menjadi sebuah lukisan yang digambar oleh robot.

#### DAFTAR PUSTAKA

- [1] Perdue, D. J., Valk, L., 2011, *The Unofficial Lego(R) Mindstorms(R) NXT 2.0 Inventor's Guide*, No Starch Press, Inc., San Francisco, California.
- [2] Kelly, J. F., 2010, *Lego Mindstorms NXT-G Programming Guide (Technology in Action)*, 2<sup>nd</sup> edition, Apress, United States of America.
- [3] Hariyanto, D., Hertanto, D. I., 2010, Identifikasi Warna Obyek Benda dengan Menggunakan Mikrokontroler dan Sensor Kamera, *Laporan Penelitian*, Fakultas Teknik Universitas Negeri Yogyakarta, Yogyakarta.

- 
- [4] Claque, K., Yabuki, H., 2002, *Lego Mindstorms: Dark Side Robots, Transport, and Creature*, Syngress, Inc., Rockland, America.
- [5] Gasperi, M., Hurbain, P., 2009, *Extreme NXT: Extending Lego Mindstorms NXT to the Next Level*, edisi kedua, Apress, Inc., United States of America.
- [6] Hariyanto, D., Wibowo, K. A., 2011, *Mobile Robot Pendeteksi Warna Bola dengan Sensor Kamera CMUCAM3, Laporan Pengembangan Tugas Akhir*, Fakultas Teknik, Universitas Negeri Yogyakarta.
- [7] Setiawan, R., 2012, Pengembangan Robot Pendeteksi Objek Berdasarkan Warna Dengan Sensor Kamera sebagai Media Pembelajaran, *Jurnal Skripsi*, Program Studi Pendidikan Teknik Mekatronika, Fakultas Teknik, Universitas Negeri Yogyakarta.
- [8] D. Forsyth dan J. Ponce, 2000, *Computer Vision: A Modern Approach*, Prentice-Hall, Upper Saddle River, New Jersey.
- [9] Bergren, C. M., 2003, *Anatomy of a Robot*, McGraw-Hill Companies, Inc., United States of America.
-