

## Sistem Penggambar Arsiran Bangun Datar Otomatis Berbasis *Android*

Christopher Henry Priyono<sup>1</sup>, Harlianto Tanudjaja<sup>2</sup> Dan Yohanes Calvinus<sup>1</sup>

**Abstract:** Beside Windows operating system on computer, *Android* is an Operating System that popularly used, especially in mobile devices. One advantage of *Android* Operating System is its open source license that gives chance to every people to develop it further. Many applications have been built for *Android* platform, including entertainment, sport, finance, etc. In spite of this, *Android* usage for controlling automation process is still rarely found. However, many process can be automated by using *Android* Device as its controller, one of those is drawing process. In this thesis, an automation system for drawing two-dimensional geometry is designed by using *Android* Device as an input device and for controlling the drawing process wirelessly. This system also uses microcontroller to control drawing actuator movement. The purpose of this design is to introduce *Android* Device for controlling drawing process wirelessly to draw two-dimensional geometry in an A4-sized paper. Parts that are designed consist of *Android* Application, microcontroller code, and drawing actuator. Microcontroller code is created by using CodeVisionAVR 2.03.4. *Android* Application is developed using Eclipse IDE with *Android* plug-ins. Construction of drawing set has dimension of 364 x 364 mm. This system has been tested by giving geometry input to *Android* Device. All user interactive features in *Android* application successfully work, including move, rotate, scale, edit points, margin, add shape, next shape, previous shape, delete shape, shape selection, raster line generation, and raster line slope. Wireless progress monitoring system in *Android* application also works well. Overall, the designed system has successfully accomplished its task. *Android* Device successfully interacts with user, sends draw command to drawing set, monitors drawing process and displays it to user. Drawing set successfully draw lines that construct geometries on the paper, with biggest error rate of 14% when drawing circle.

**Keywords:** *Android*, automation, draw.

**Abstrak:** Selain sistem operasi Windows pada komputer, *Android* merupakan sistem operasi yang banyak digunakan, terutama pada mobile devices. Salah satu keunggulan sistem operasi *Android* adalah sifatnya yang open source sehingga terbuka bagi pihak manapun yang ingin mengembangkannya. Berbagai jenis aplikasi telah banyak dibuat di atas *Android* platform, mulai dari entertainment, sport, finance, dan lain-lain. Meskipun demikian, penggunaan *Android* Devices di bidang otomasi masih dirasa kurang. Padahal cukup banyak proses yang dapat diotomatisasi dengan memanfaatkan *Android*, salah satunya adalah proses menggambar. Dalam tugas akhir ini dirancang suatu sistem otomasi dengan fungsi sebagai penggambar bangun datar otomatis. Gambar dibuat dengan pola arsiran yang menggunakan *Android* Device sebagai alat input sekaligus pengontrol proses secara wireless. Sistem ini juga menggunakan mikrokontroler sebagai alat pengendali pergerakan aktuator penggambar. Tujuan rancangan ini adalah memperkenalkan *Android* Device untuk mengontrol secara wireless proses penggambaran bangun datar dengan metode arsiran pada kertas berukuran A4. Modul yang dirancang meliputi program pada *Android* Device, program mikrokontroler, dan aktuator penggambar. Program mikrokontroler dibuat menggunakan CodeVisionAVR 2.03.4. Program *Android* dibuat menggunakan Eclipse IDE dengan *Android* plug-ins. Rancang bangun perangkat penggambar dibuat dengan ukuran 364 x 364 mm. Pengujian dilakukan dengan memberikan input gambar bangun datar melalui layar sentuh *Android* Device. Seluruh fitur-fitur user interactive pada program *Android* yang meliputi operasi move, rotate, scale, edit points, margin, add shape, next shape, previous shape, delete shape, pemilihan jenis bangun datar, generate garis arsir, dan kemiringan garis arsir berjalan dengan baik. Wireless progress monitoring system yang dibuat pada program *Android* berjalan dengan baik. Secara keseluruhan, sistem yang dirancang berhasil bekerja dengan baik. Program pada *Android* Device berhasil berinteraksi dengan user, mengirimkan perintah penggambaran garis arsir ke perangkat penggambar, dan memantau proses penggambaran serta menampilkannya kepada user. Perangkat penggambar berhasil menggambar garis-garis arsir yang membentuk bangun datar pada kertas dengan error terbesar 14%, yaitu pada penggambaran bangun lingkaran.

**Kata kunci:** *Android*, otomasi, gambar.

### PENDAHULUAN

Selain sistem operasi *Windows* pada komputer, *Android* merupakan sistem operasi yang banyak digunakan, terutama pada *mobile devices*. Salah satu keunggulan sistem operasi *Android* adalah sifatnya yang open source sehingga terbuka bagi pihak manapun yang ingin mengembangkannya. Berbagai jenis aplikasi telah banyak dibuat di atas *Android* platform, mulai dari games, entertainment, sport, finance, dan lain-lain. Meskipun demikian, penggunaan *Android* Devices di bidang otomasi masih dirasa kurang. Padahal cukup banyak proses yang dapat diotomatisasi dengan memanfaatkan *Android*, salah satunya adalah proses menggambar.

Alat ini dibuat suatu sistem otomasi dengan fungsi sebagai penggambar bangun datar otomatis. Gambar dibuat dengan pola arsiran yang menggunakan *Android* Device sebagai alat input sekaligus pengontrol proses secara wireless. Sistem ini juga menggunakan mikrokontroler sebagai alat pengendali pergerakan aktuator penggambar. Data yang digunakan untuk mendukung proses pembuatan Tugas Akhir ini diperoleh melalui survei yang dilakukan sebanyak dua kali. Survei yang pertama dilakukan pada manusia yang sedang mengarsir beberapa bangun datar di kertas. Survei yang kedua dilakukan pada printer yang sedang mencetak gambar bangun datar sederhana.

Survei pertama pada manusia yang sedang mengarsir di atas kertas menghasilkan kesimpulan sebagai berikut. Garis-garis arsiran yang dihasilkan tidak seluruhnya sejajar. Jarak antar setiap garis arsiran juga berbeda-beda. Dibutuhkan alat ukur untuk menghasilkan garis-garis arsiran yang benar-benar sejajar dan jaraknya sama satu dengan yang lain.

Survei kedua pada printer yang sedang mencetak gambar bangun datar menghasilkan kesimpulan sebagai berikut. Pencetakan dimulai dari bagian paling atas kertas dan bergerak ke bawah. Pencetakan dilakukan dengan

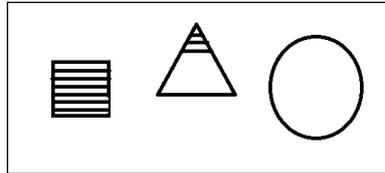
<sup>1</sup> Jurusan Teknik Elektro Universitas Tarumanagara Jakarta

<sup>2</sup> Jurusan Teknik Elektro Universitas Atmajaya Jakarta

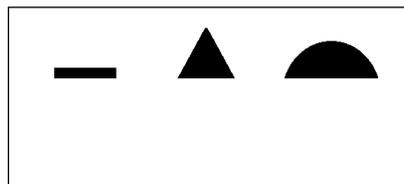
menggambar sekumpulan garis horizontal, mirip seperti mengarsir namun dengan tingkat kerapatan antar garis yang sangat tinggi.

Perbedaan utama printer dan manusia yang mengarsir bangun datar adalah pada urutan pengarsirannya. Manusia tidak diharuskan mengikuti urutan tertentu dalam pengarsiran, sementara printer mencetak garis demi garis dari bagian paling atas kertas hingga paling bawah. Printer tidak peduli apakah ia mencetak bagian dari beberapa bangun datar sekaligus dalam satu garis horizontal. Selain itu, manusia juga dapat mengarsir dengan berbagai kemiringan garis arsiran, sementara printer hanya garis horizontal.

Hasil dari sebagian proses pengarsiran gambar yang terdiri dari beberapa bangun datar yang dilakukan manusia dapat dilihat pada Gambar 1. Hasil sebagian proses pencetakan yang dilakukan printer untuk gambar yang sama dapat dilihat pada Gambar 2.



■ **Gambar 1.** Sebagian Proses Pengarsiran Gambar Bangun Datar oleh Manusia



■ **Gambar 2.** Sebagian Proses Pencetakan Gambar Bangun Datar oleh Printer

### KAJIAN PUSTAKA

Sistem yang dirancang berfungsi untuk menggambar bangun datar pada kertas berukuran A4 menggunakan pola arsiran. Bangun datar dimasukkan *user* melalui *touchscreen* milik *Android Device*. *User* terlebih dahulu memilih jenis bangun datar dari 8 jenis yang disediakan pada program *Android*. Bangun datar tersebut kemudian dapat disesuaikan posisinya pada area gambar, diperbesar/diperkecil, dirotasi, atau digeser titik-titik sudutnya untuk menghasilkan bentuk baru. *User* juga dapat menambahkan bangun datar lainnya ke area gambar, mengedit masing-masing bangun datar pada area, atau menghapusnya. Selain itu kemiringan garis arsiran masing-masing bangun datar juga dapat ditentukan. Setelah *user* selesai memasukkan semua bangun datar, data ditransmisikan dari *Android Device* ke perangkat penggambar secara *wireless* menggunakan *Bluetooth*. Data ini meliputi koordinat seluruh garis arsiran yang menyusun sekumpulan bangun datar tersebut beserta kemiringan garis arsirannya.

Setelah data koordinat garis arsiran pertama beserta kemiringannya diterima oleh perangkat penggambar, meja gambar akan berputar untuk menyesuaikan kemiringan garis yang diinginkan. Aktuator penggambar kemudian akan bergerak ke koordinat awal ( $x_1, y_1$ ), menurunkan pena hingga menyentuh kertas, kemudian bergerak ke koordinat akhir ( $x_2, y_2$ ) untuk menghasilkan gambar garis pada kertas. Pena kemudian diangkat lagi hingga tidak menyentuh kertas. Setelah itu perangkat penggambar mengirimkan konfirmasi ke *Android Device* secara *wireless*. Konfirmasi ini menandakan bahwa garis arsiran pertama telah selesai digambar. *Android Device* selanjutnya mengirimkan data koordinat garis arsiran kedua. Proses ini terus diulangi sampai seluruh garis arsiran digambar pada kertas.

### Diagram Blok Rancangan

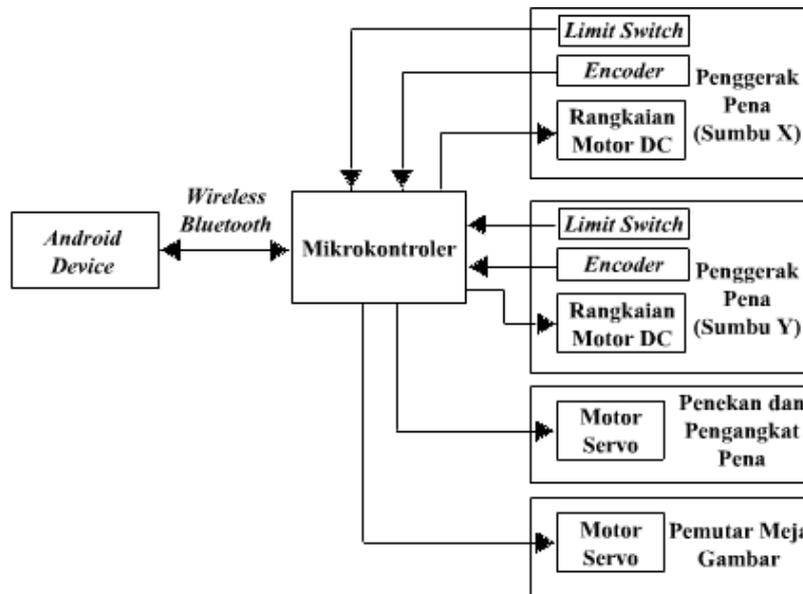
Sistem yang dirancang berfungsi untuk menggambar sekumpulan garis arsiran yang membentuk bangun datar. Bangun datar dimasukkan *user* melalui *Android Device*. Diagram blok dari sistem yang dirancang dapat dilihat pada Gambar 3.

*Limit switch* pada penggerak pena sumbu  $x$  dan  $y$  digunakan untuk mendeteksi bahwa pena telah kembali ke posisi awal. *Encoder* yang dipasang pada perpanjangan sumbu motor DC berfungsi untuk mendeteksi sejauh mana pena telah bergerak. Rangkaian motor DC terdiri dari *DC motor driver* yang telah dihubungkan dengan motor DC. Penggerak pena sumbu  $x$  dan  $y$ , penekan dan pengangkat pena, serta pemutar meja gambar merupakan satu kesatuan dalam modul aktuator penggambar.

### Android

*Android* adalah sebuah sistem operasi berbasis Linux yang didesain dengan sasaran utama berupa perangkat *mobile* yang dilengkapi layar sentuh, seperti *smartphone* dan *tablet PC*. Sistem operasi ini pada

awalnya dikembangkan oleh *Android, Inc.* yang didanai oleh Google. Pada akhirnya *Android, Inc.* dibeli oleh Google pada tahun 2005. *Android* memperkenalkan dirinya pada tahun 2007 bersamaan dengan Open Handset Alliance (OHA), sebuah konsorsium berbagai perusahaan *hardware, software,* dan telekomunikasi. Konsorsium ini mendedikasikan dirinya untuk mengembangkan lebih lanjut *open standards* untuk perangkat *mobile.* Handphone pertama yang berbasis *Android* pertama kali dijual pada Oktober 2008.



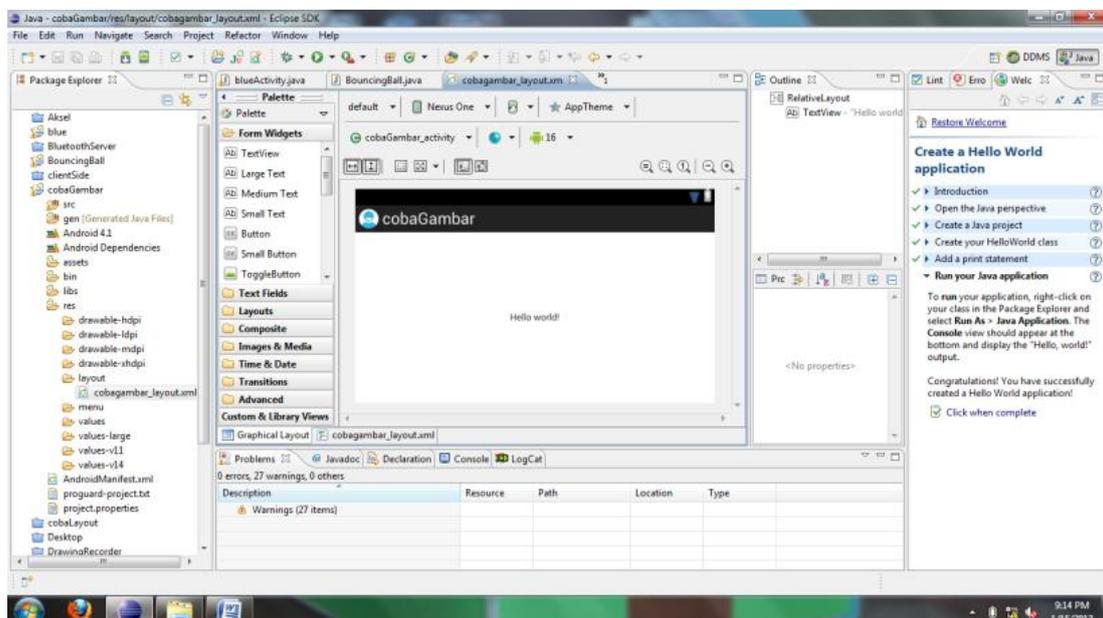
■ Gambar 3. Diagram Blok Sistem yang Dirancang

Pada bulan Oktober 2012, kira-kira 700.000 aplikasi telah tersedia untuk perangkat berbasis *Android* [1]. Sekitar 25 miliar kali *download* telah dilakukan melalui Google Play [2]. Hal ini menjadikan *Android* sebagai *smartphone platform* yang paling banyak digunakan di dunia [3], mengalahkan *Symbian* pada 2010 [4].

## Eclipse

Eclipse adalah *Software Development Environment* yang dapat digunakan untuk berbagai macam bahasa pemrograman. Eclipse dapat digunakan untuk mengembangkan aplikasi dengan bahasa pemrograman Java, serta berbagai bahasa pemrograman lainnya dengan bantuan *plug-ins*, seperti Ada, C, C++, COBOL, Fortran, Haskell, PHP, dan lain-lain. Salah satu *plug-ins* memungkinkan pengembangan aplikasi *Android* pada program Eclipse menggunakan bahasa pemrograman Java.

Eclipse *Software Development Kit (SDK)* bersifat gratis dan *open source*. Namun program Eclipse tidak dirilis di bawah *GNU General Public License* seperti *software open source* pada umumnya, namun menggunakan *license* tersendiri yang dinamakan *Eclipse Public License*. Tampilan program Eclipse yang sudah ditambahkan *plug-ins* untuk membuat aplikasi *Android* dapat dilihat pada Gambar 4.



■ Gambar 4. Tampilan IDE Program Eclipse

## Mikrokontroler

Mikrokontroler adalah suatu komputer kecil yang terdapat pada sebuah *integrated circuit* tunggal. Mikrokontroler terdiri dari inti prosesor, memori, dan *peripheral* input/output yang dapat diprogram. *Smithsonian Institution* menyatakan bahwa Gary Boone dan Michael Cochran sukses membuat mikrokontroler pertama pada tahun 1971. Mikrokontroler ini dinamakan TMS 1000 yang dijual secara komersial pada tahun 1974. Mikrokontroler TMS 1000 mengkombinasikan *read-only memory*, *read/write memory*, prosesor, dan *clock* pada sebuah chip tunggal dan ditargetkan untuk *embedded systems*.

Modul Creative Vision MA-16 merupakan *minimum system* IC Mikrokontroler ATmega16. Modul mikrokontroler MA-16 digunakan karena memiliki kualitas soldering yang cukup baik. Hal ini dapat meminimalisasi terjadinya berbagai *error* yang tidak diinginkan akibat masalah pada *hardware* mikrokontroler. Selain itu, modul ini juga telah dilengkapi dengan *built-in downloader*, sehingga memungkinkan program untuk *di-download* dari PC ke IC Mikrokontroler tanpa menggunakan komponen tambahan.

ATMega16 digunakan karena memiliki 32 pin I/O yang mencukupi untuk koneksi dengan berbagai *peripheral* dalam sistem ini, memiliki fitur *Universal Asynchronous Receiver Transmitter* (UART) untuk berkomunikasi dengan modul *Bluetooth-to-serial*, memiliki fitur *timer* yang dapat menghasilkan sinyal *Pulse Width Modulation*(PWM) untuk menggerakkan servo, serta memiliki fitur *external interrupt* yang dapat menerima sinyal perubahan posisi dari *encoder*.

## Bluetooth

*Bluetooth* adalah sebuah standar teknologi nirkabel. Standar ini dibuat dengan tujuan pertukaran data jarak pendek antar perangkat, baik itu *mobile* maupun *non-mobile*. *Bluetooth* dibuat oleh sebuah vendor telekomunikasi bernama Ericsson pada tahun 1994. *Bluetooth* banyak digunakan pada produk, seperti telepon genggam, *tablet PC*, *media player*, Lego Mindstorms NXT, PlayStation 3, PS Vita, Nintendo Wii, dan beberapa *headset*, *modem*, dan jam tangan. Modul *Bluetooth-to-Serial* dipilih karena dapat mengubah protokol komunikasi *Bluetooth* menjadi komunikasi serial yang kompatibel dengan mikrokontroler

## Catu Daya

Catu daya adalah sebuah alat yang mensuplai daya elektrik kepada beban elektrik. Setiap catu daya harus memperoleh energi yang disuplainya kepada beban dari suatu sumber energi. Sumber energi ini dapat berupa perangkat penyimpanan energi (seperti baterai), sistem elektromekanikal (seperti generator), sel surya, dan sebagainya tergantung jenis catu dayanya.

Salah satu jenis catu daya yang cukup populer adalah *switched-mode power supply* (SMPS). Pada SMPS, tegangan listrik AC langsung disearahkan untuk menghasilkan tegangan DC. Tegangan DC ini kemudian dinyalakan dan dimatikan dengan kecepatan yang tinggi, sehingga menghasilkan listrik AC yang akan melalui transformator. *Switching* terjadi dengan frekuensi yang tinggi (antara 10 kHz – 1 MHz), sehingga memungkinkan penggunaan transformator dan kapasitor filter yang lebih kecil, ringan, dan murah dibandingkan catu daya standar. Setelah melalui transformator, listrik AC berfrekuensi tinggi ini disearahkan untuk menghasilkan tegangan keluaran DC.

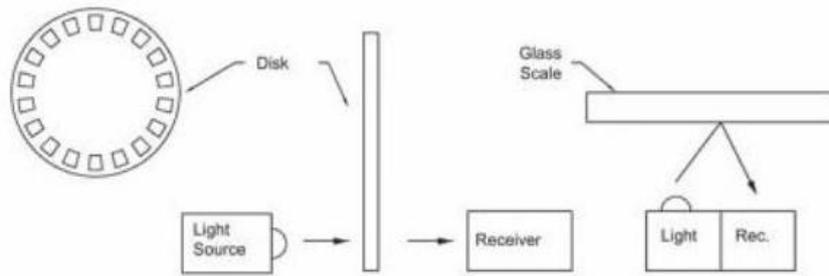
SMPS biasanya memiliki fitur pengaman seperti pembatas arus untuk melindungi perangkat dan pengguna. SMPS juga memiliki batas absolut minimum arus yang dapat dikeluarkan. Jadi catu daya SMPS tersebut hanya dapat digunakan untuk mensuplai daya di atas batas tertentu, dan tidak dapat berfungsi di bawah batas tersebut.

## Rotary Encoder

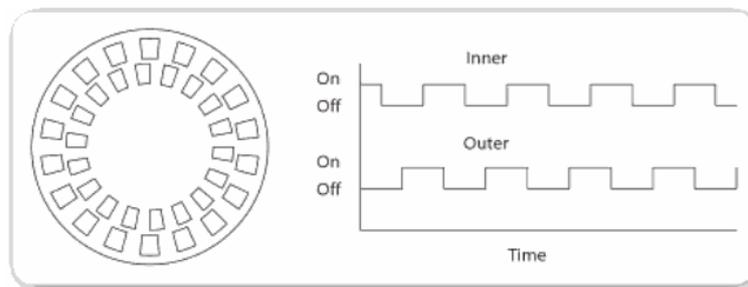
*Encoder* adalah semua *device* yang digunakan untuk memberikan *positioning feedback* kepada suatu sistem kontrol. *Rotary encoder* adalah sebuah tipe sensor yang dapat mendeteksi posisi angular ketika dihubungkan dengan *shaft* motor. Dua jenis *encoder* yang cukup populer adalah *Rotary Encoder* dan *Linear Encoder*. *Linear Encoder* berfungsi untuk mendeteksi pergeseran(translasi), sedangkan *Rotary encoder* berfungsi untuk mendeteksi perputaran(rotasi). Cara kerja kedua jenis *encoder* tersebut ditunjukkan pada Gambar 5. *Rotary encoder* menggunakan piringan (*disk*) yang berlubang, serta pemancar dan penerima cahaya pada sisi piringan yang berbeda. Namun dapat pula menggunakan piringan hitam putih dengan pemancar dan penerima pada sisi yang sama. Ketika piringan ini berputar, penerima akan menerima dan tidak menerima cahaya dari pemancar secara bergantian. Jumlah pergantian kondisi ini dihitung untuk mengetahui sejauh mana *encoder* telah berputar. *Linear encoder* menggunakan batangan panjang berlubang, serta pemancar dan penerima cahaya yang bersebelahan. Pemancar dan penerima ini diarahkan pada batangan. Ketika batangan ini bergeser, penerima akan menerima dan tidak menerima cahaya dari pemancar secara bergantian. Jumlah pergantian kondisi ini dihitung untuk mengetahui sejauh mana *encoder* telah bergeser.

*Encoder* juga dapat dibagi menjadi tipe *incremental* dan *absolute*. *Encoder* dengan tipe *incremental* dapat dilihat pada Gambar 6. *Incremental encoder* hanya dapat mendeteksi jumlah dan arah perputaran, tetapi tidak

dapat mendeteksi posisi angular aktual dari motor. Hal ini dapat menyebabkan error yang semakin lama semakin besar dari hasil akumulasi, sehingga harus di-*reset* ke sebuah *hard reference* secara berkala. *Absolute encoder* dapat mengatasi masalah ini.



■ Gambar 5. Cara Kerja Rotary Encoder(kiri) dan Linear Encoder(kanan)



■ Gambar 6. Incremental Rotary Encoder dengan Dua Sensor Untuk Mendeteksi Arah Putaran

### Motor DC

Motor DC adalah motor elektrik mekanis yang menggunakan arus searah sebagai sumber penggerakannya. Motor DC terdiri dari 2 komponen utama, yaitu stator dan rotor. Stator adalah bagian yang tidak bergerak, sementara rotor adalah bagian yang berputar. Secara umum, motor DC terbagi menjadi 2, yaitu *brush DC motor*, dan *brushless DC motor*. Pada *brush DC motor*, magnet permanen berada pada bagian stator. Sementara pada *brushless DC motor*, magnet permanen berada pada bagian rotor.

*Driver* motor DC berfungsi untuk mengatur arah perputaran motor DC, mengaktifkan atau menonaktifkan motor DC, serta mensuplai arus yang cukup untuk menggerakkan motor DC. *Driver* motor DC diperlukan karena arus dari mikrokontroler langsung tidak mencukupi untuk menggerakkan sebuah motor DC. Selain itu, tanpa menggunakan *driver*, arah perputaran Motor DC tidak dapat dibalik dari searah jarum jam menjadi berlawanan arah jarum jam, atau sebaliknya.

IC yang digunakan untuk driver motor DC adalah IC L298. IC L298 adalah suatu penguat yang dirancang khusus sebagai penggerak (*driver*) motor DC dengan *maximum amperage* yang cukup besar, yaitu 3 Ampere apabila kedua output diparalel. Modul yang digunakan memudahkan proses instalasi baik *wiring* maupun mekanis. Hal ini disebabkan karena *input* dan *output* sudah di-*extend* dalam bentuk *header pin* dan konektor kabel, serta terdapat lubang pemasangan sekrup pada keempat ujung *board* modul.

Modul ini memiliki 2 input untuk menentukan arah putar motor, 1 *input* untuk menentukan motor aktif/nonaktif, dan *output* untuk dihubungkan dengan sebuah motor DC. Tegangan suplai yang dibutuhkan adalah 5V untuk operasi logika TTL (*Transistor-to-Transistor Logic*). Tegangan maksimum untuk suplai Motor DC adalah 50V. Besar tegangan suplai Motor DC ini disesuaikan dengan spesifikasi tegangan dari Motor DC yang digunakan.

Tipe motor DC yang digunakan adalah tipe DME 44B9-08 memiliki kecepatan putar 3700 RPM pada tegangan maksimum 24V. Hal ini dibutuhkan agar dapat menggerakkan pena dari ujung ke ujung bidang gambar dalam waktu kurang dari 15 detik apabila diberi tegangan 12V.

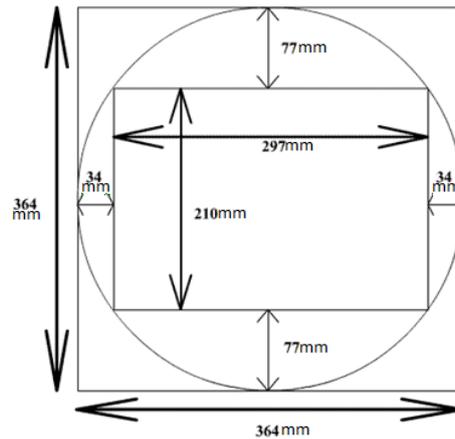
### Motor Servo

Motor servo adalah aktuator berputar yang memungkinkan kontrol yang presisi atas posisi sudut. Motor servo terdiri dari motor dan sensor untuk memberikan umpan balik posisi. Motor servo biasanya digunakan dalam bidang robotika, CNC, atau proses produksi otomatis.

Motor servo yang digunakan untuk alat ini adalah motor servo tipe Motor Servo Tower Pro Micro Servo 9g SG90. Motor servo tipe SG90 ini dipilih karena memiliki massa 9 gram, sehingga cukup ringan untuk ditempatkan di bagian atas modul aktuator penggambar untuk menekan dan mengangkat pena dari kertas. Selain itu, servo ini juga memiliki *stall torque* sebesar 1,6 kg.cm, sehingga cukup kuat untuk mengangkat pena serta memutar meja gambar.

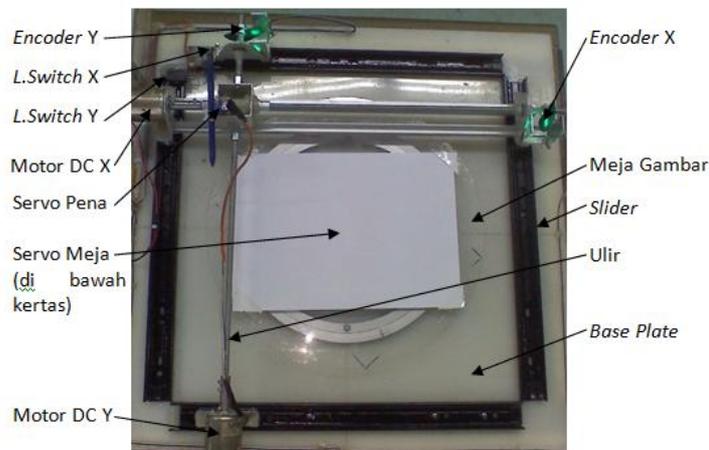
## REALISASI RANCANGAN

Modul aktuator penggambar dibuat untuk memiliki dimensi seperti terlihat pada Gambar 7.



■ **Gambar 7.** Dimensi Modul Aktuator Penggambar yang Dirancang

Bidang persegi panjang di bagian paling dalam adalah tempat untuk meletakkan kertas A4 sehingga dimensinya disesuaikan dengan ukuran kertas A4. Lingkaran pada bagian luar merupakan meja gambar yang dapat diputar 90 derajat searah dan berlawanan arah jarum jam. Hal ini digunakan untuk menyesuaikan dengan kemiringan garis arsiran yang diinginkan. Persegi pada bagian paling luar merupakan *base plate* tempat meja gambar beserta servo untuk memutarinya bertumpu. Modul aktuator penggambar setelah direalisasikan dapat dilihat pada Gambar 8.



■ **Gambar 8.** Modul Aktuator Penggambar Setelah Direalisasikan

*Base plate* yang digunakan berbahan dasar *nylon* dan memiliki dimensi 600 x 600 mm dengan ketebalan 5 mm. Meja gambar berbahan *acrylic* dengan diameter lingkaran 370 mm dan ketebalan 3 mm. *Slider* yang digunakan masing-masing memiliki panjang 450 mm dan lebar 35 mm. *Ulir* yang digunakan masing-masing memiliki panjang 500 mm dengan diameter 8 mm.

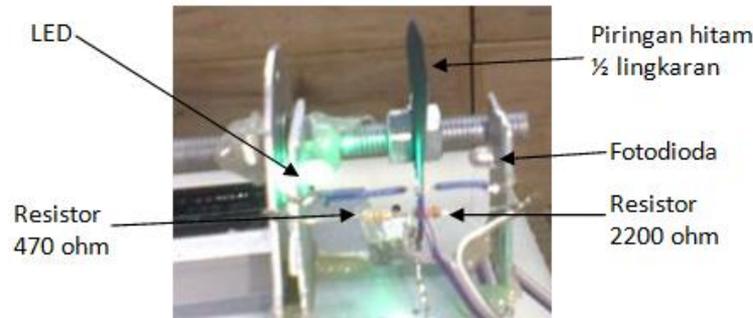
Motor servo untuk memutar meja gambar diletakkan di bawah meja gambar itu sendiri dengan bertumpu pada *base plate*. Titik tengah perputaran motor servo tersebut diposisikan agar kertas di atas meja berada pada posisi horizontal. Hal ini membuat kertas dapat diputar 90 derajat searah maupun berlawanan arah jarum jam sehingga seluruh kemiringan garis dapat dicapai.

Motor DC sumbu x dan y dihubungkan dengan ulir sumbu yang bersesuaian. Ketika motor DC sumbu x berputar pada satu arah, motor DC beserta ulir dan *encoder* sumbu y yang terletak pada *slider* terdorong ke arah x positif. Sebaliknya ketika motor DC sumbu x berputar pada arah yang berlawanan, ketiga komponen sumbu y tersebut akan terdorong ke arah x negatif. Hal yang serupa terjadi apabila motor DC sumbu y berputar. Pena beserta servo penggeraknya diletakkan di perpotongan ulir sumbu x dan y, sehingga ikut bergerak mengikuti pergeseran baik sumbu x maupun y.

Modul *encoder* digunakan untuk mendeteksi jumlah putaran motor DC. *Encoder* yang dipasang pada sistem berjumlah 2 buah, untuk motor DC sumbu x dan y. Modul *encoder* yang digunakan dibuat dari LED (*Light Emitting Diode*) hijau, fotodiode, resistor, dan piringan setengah lingkaran yang berwarna hitam

dan dijepit pada ulir yang berputar. Fotodiode dipasang dalam keadaan *reverse bias* agar lebar dari *depletion layer* bertambah. Hal ini mengurangi kapasitansi *junction*, sehingga *response time* juga berkurang.

Ketika fotodiode menerima berkas cahaya dari LED hijau, tegangan yang dikeluarkan pada pin *signal* yang terhubung dengan mikrokontroler mendekati 5V, sehingga mikrokontroler mendeteksi logika *high*. Sebaliknya ketika berkas cahaya terhalang piringan hitam, tegangan yang dikeluarkan pada pin *signal* mendekati 0V, sehingga mikrokontroler mendeteksi logika *low*. Kedua kondisi ini terus terjadi secara bergantian akibat piringan hitam setengah lingkaran yang berputar bersama ulir. Jumlah pergantian kondisi inilah yang dihitung oleh mikrokontroler untuk mendeteksi berapa putaran Motor DC telah berputar. Pada akhirnya besar pergeseran masing-masing sumbu dapat disimpulkan. Setiap pergantian kondisi merupakan 1 pulsa, sehingga satu putaran penuh terdiri dari 2 pulsa. Modul *encoder* tersebut setelah direalisasi dapat dilihat pada Gambar 9.



■ Gambar 9. Modul *Encoder* Setelah Direalisasi

*Limit switch* digunakan untuk mendeteksi apakah posisi pena sudah kembali ke titik awal. *Limit switch* yang dipasang berjumlah 2 buah, masing-masing diletakkan di ujung kiri dan atas pada *slider* sumbu x dan y. Hal ini disebabkan karena titik (0,0) terletak pada ujung kiri atas sesuai dengan koordinat layar komputer. Ketika ingin kembali ke titik awal tersebut, motor DC sumbu x dan y berputar ke arah yang membuat pena bergerak ke arah sumbu x dan y negatif. Pergerakan motor DC sumbu x akan dihentikan apabila *limit switch* sumbu x telah tertekan, begitu pula dengan sumbu y.

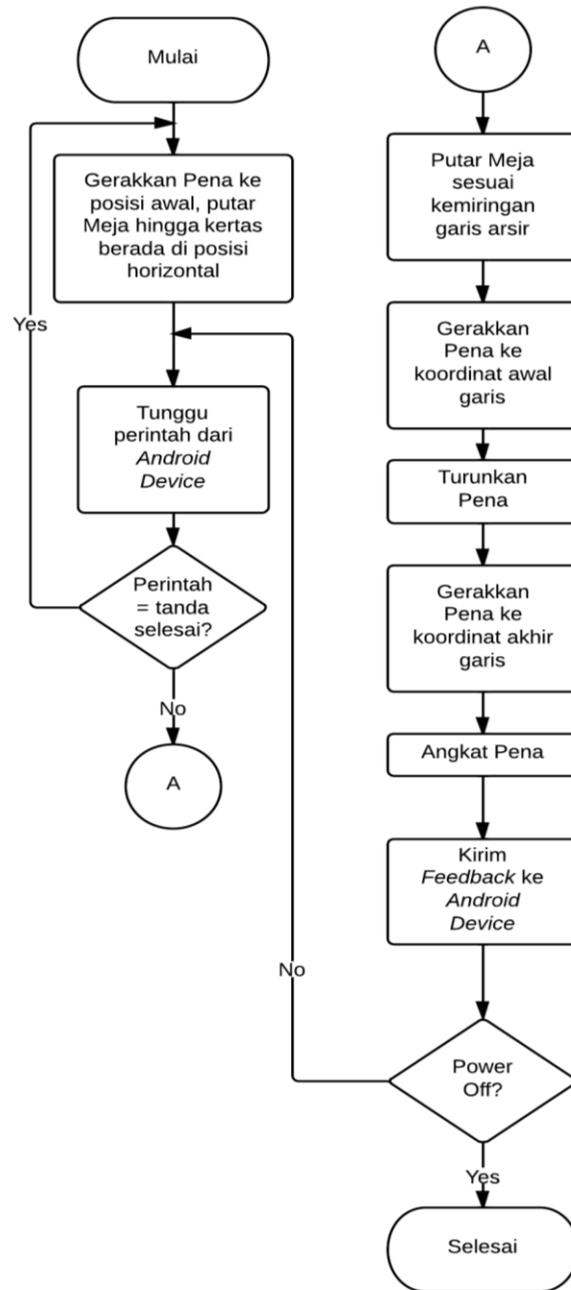
Pin NO (*Normally Open*) dari *limit switch* terhubung dengan GND (*Ground*). Sementara itu pin COM (*Common*) dari *limit switch* menghasilkan *signal* yang terhubung dengan pin mikrokontroler. Pin mikrokontroler ini telah diatur sebagai *input* dan diberi *internal pullup*. Ketika *limit switch* dalam keadaan tidak tertekan (NC), pin mikrokontroler dalam keadaan *floating*, sehingga tegangan yang terdeteksi mikrokontroler adalah 5V atau logika *high* karena adanya *pullup*. Sebaliknya ketika *limit switch* dalam keadaan tertekan, pin mikrokontroler terhubung ke GND, sehingga tegangan yang terdeteksi mikrokontroler adalah 0V atau logika *low*.

Program mikrokontroler berfungsi untuk menggerakkan aktuator penggambar. Pergerakan ini ditujukan untuk menggambar garis-garis arsiran pada kertas. Koordinat garis-garis arsiran ini diterima mikrokontroler dari *Android Device* melalui modul *Bluetooth-to-Serial*. *Flowchart* program mikrokontroler dapat dilihat pada Gambar 10. *Source code* program dapat dilihat pada Lampiran 1. Program dibuat menggunakan CodeVisionAVR 2.03.4.

Berikut adalah *pseudocode* program mikrokontroler.

```

MULAI
DO
  Kembalikan_aktuator_ke_posisi_awal();
  DO
    Tunggu_perintah_dari_Android();
    IF (perintah_tanda_selesai)
      EXIT LOOP
    ELSE
      putar_meja_ke_sudut(sudut);
      gerakkan_pena_ke(x1,y1);
      turunkan_pena();
      gerakkan_pena_ke(x2,y2);
      angkat_pena();
    END IF
  LOOP
LOOP
SELESAI
    
```



■ Gambar 10. Flowchart Program Mikrokontroler

Pertama-tama program mikrokontroler akan memerintahkan servo pemutar meja untuk kembali ke posisi tengah sehingga kertas pada posisi horizontal, memberi perintah kepada servo pengangkat dan penekan pena untuk mengangkat pena dari bidang gambar, serta menggerakkan pena ke titik awal hingga menyentuh *Limit Switch* masing-masing sumbu. Program kemudian menunggu perintah dari *Android Device*. Setelah perintah diterima, program akan memerintahkan servo pemutar meja untuk memutar meja gambar sesuai kemiringan garis arsiran, serta menggerakkan pena ke koordinat  $(x1,y1)$ . Servo pena selanjutnya diperintahkan untuk menurunkan pena hingga menyentuh kertas, kemudian pena digerakkan ke koordinat  $(x2,y2)$ . Pena pada akhirnya diangkat kembali oleh servo pena dan program menunggu perintah berikutnya. Proses ini terus diulangi hingga program menerima perintah dari *Android Device* bahwa semua garis arsir telah digambar. Apabila program menerima perintah tanda selesai ini, meja gambar akan diputar kembali hingga kertas dalam posisi horizontal dan pena dikembalikan ke posisi awal.

Program pada *Android Device* yang dirancang bernama *Shape Rasterizer*. Program *Shape Rasterizer* berfungsi sebagai penyedia GUI (*Graphical User Interface*). GUI ini memungkinkan *user* untuk menambahkan bangun datar yang ingin digambar, mengatur ukuran dan peletakkannya pada bidang gambar, mengedit posisi masing-masing titik sudutnya, serta mengatur kemiringan garis arsiran masing-masing bangun datar. Jenis bangun datar yang disediakan berjumlah 8, yaitu lingkaran, elips, segitiga, persegi, segilima, segienam, segitujuh, dan segidelapan. Program ini juga berfungsi untuk mengontrol proses penggambaran bangun oleh perangkat penggambar. Pengontrolan ini dilakukan secara *wireless* dengan mengirimkan koordinat-koordinat garis serta

menampilkan *current progress* penggambaran kepada *user* melalui layar *Android Device*. *Pseudocode* penanganan *gesture* dari *user* pada layar *Android* adalah sebagai berikut.

```

SUB EVENT ON_TOUCH()
  IF (MODE==ROTATE) AND (JUMLAH_JARI == 2)
    IF (SUDUT_DUA_JARI>SUDUT_SEBELUM)
      Putar_bangun_ccw();
    ELSE IF (SUDUT_DUA_JARI<SUDUT_SEBELUM)
      Putar_bangun_cw();
    END IF
  ELSE IF (MODE==SCALE) AND (JUMLAH_JARI == 2)
    IF (JARAK_DUA_JARI>JARAK_SEBELUM)
      Perbesar_bangun();
    ELSE IF (JARAK_DUA_JARI<JARAK_SEBELUM)
      Perkecil_bangun();
    END IF
  ELSE IF (MODE==MOVE) AND (JUMLAH_JARI == 1)
    IF (SENTUH_X > SENTUH_SEBELUM)
      geser_bangun_ke_kanan();
    ELSE IF (SENTUH_X < SENTUH_SEBELUM)
      geser_bangun_ke_kiri();
    END IF
    IF (SENTUH_Y > SENTUH_SEBELUM)
      geser_bangun_ke_bawah();
    ELSE IF (SENTUH_Y < SENTUH_SEBELUM)
      geser_bangun_ke_atas();
    END IF
  ELSE IF (MODE==EDIT_POINTS)
    tampilkan_titik_ungu();
    IF (tipe_gesture==TEKAN)
      FOR i=1 to JUMLAH_TITIK_SUDUT_PADA_BANGUN
        IF (koordinat_sentuh_di_dalam_lingkaran_ke(i))
          titik_yang_digeser=i;
        END IF
      END FOR
    ELSE IF (tipe_gesture==GESER)
      Titik_sudut_x[titik_yang_digeser]=sentuh_x;
      Titik_sudut_y[titik_yang_digeser]=sentuh_y;
    END IF
  END IF
  Refresh_tampilan_layar();
END SUB

```

Berikut adalah *pseudocode* untuk menghasilkan garis arsir dari bangun datar.

```

PROCEDURE Generate_garis_arsir()
  IF (bangun==LINGKARAN)
    FOR j = UJUNG_ATAS to UJUNG_BAWAH step JARAK_ARSIR
      X_kiri=x_tengah-((r^2)-((y_tengah-j)^2));
      X_kanan=x_tengah+((r^2)-((y_tengah-j)^2));
      Jumlah_garis_arsir++;
      Garis_arsir_x1[jumlah_garis_arsir]=x_kiri;
      Garis_arsir_x2[jumlah_garis_arsir]=x_kanan;
      Garis_arsir_y1[jumlah_garis_arsir]=j;
      Garis_arsir_y2[jumlah_garis_arsir]=j;
    END FOR
  ELSE IF (bangun==ELLIPSE)
    cx=bangun_datar[i][18];
    cy=bangun_datar[i][19];

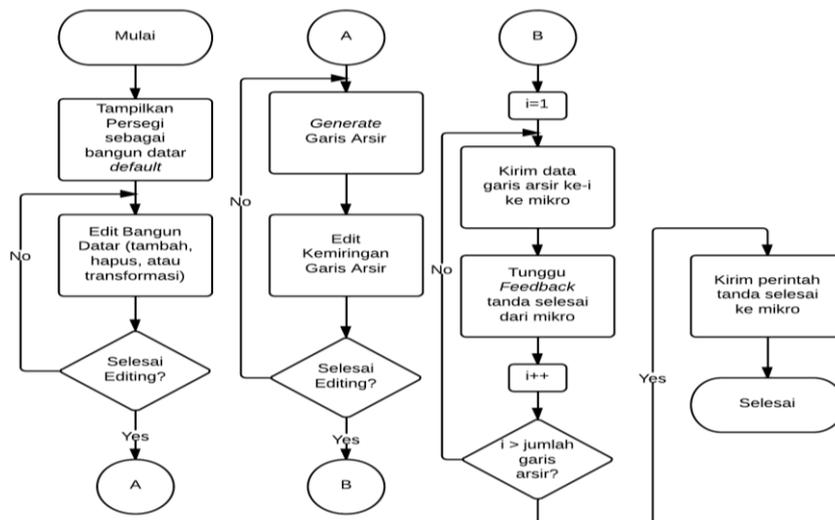
```

```

rx=cx-(bangun_datar[i][1]);
ry=cy-(bangun_datar[i][4]);
FOR j = UJUNG_ATAS to UJUNG_BAWAH step JARAK_ARSIR
  dx= sqrt((1-sqr((cy-j)/ry))*sqr(rx));
  batas_kiri=cx-dx;
  batas_kanan=cx+dx;
  jumlah_garis_arsir++;
  garis_arsir_x1[jumlah_garis_arsir]=batas_kiri;
  garis_arsir_x2[jumlah_garis_arsir]=batas_kanan;
  garis_arsir_y1[jumlah_garis_arsir]=j;
  garis_arsir_y2[jumlah_garis_arsir]=j;
END FOR
ELSE
  Ambil_titik_min();
  Ambil_titik_max();
  Kumpulkan_titik_di_kanan_titik_max_sampai_titik_min();
  Kumpulkan_titik_di_kiri_titik_max_sampai_titik_min();
  IF (sudut_kanan<sudut_kiri)
    Tukar_kumpulan_kanan_dan_kiri();
  END IF
  FOR j = UJUNG_ATAS to UJUNG_BAWAH step JARAK_ARSIR
    IF (j > y2_garis_batas_kiri)
      Garis_batas_kiri++;
    END IF
    IF (j > y2_garis_batas_kanan)
      Garis_batas_kanan++;
    END IF
    Batas_kiri=ambil_x_ketinggian_j_dari(garis_batas_kiri);
    Batas_kanan=ambil_x_ketinggian_j_dari(garis_batas_kanan);
    jumlah_garis_arsir++;
    garis_arsir_x1[jumlah_garis_arsir]=batas_kiri;
    garis_arsir_x2[jumlah_garis_arsir]=batas_kanan;
    garis_arsir_y1[jumlah_garis_arsir]=j;
    garis_arsir_y2[jumlah_garis_arsir]=j;
  END FOR
END IF
END PROCEDURE

```

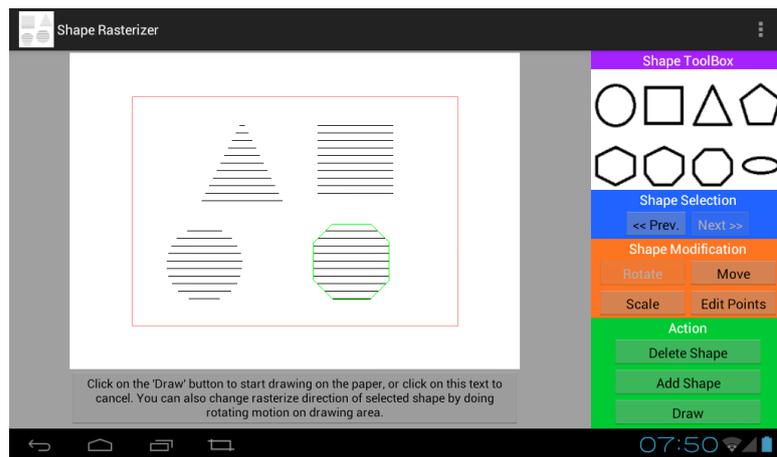
Flowchart dari program Android dapat dilihat pada Gambar 11. Source code program dapat dilihat pada Lampiran 2. Program dibuat menggunakan Eclipse IDE dengan Android plug-ins.



■ Gambar 11. Flowchart program pada Android Device

Program *Android* secara *default* akan menampilkan sebuah persegi pada bidang gambar saat dijalankan. Bangun persegi ini dapat digeser posisinya, dirotasi, diperbesar/diperkecil, dan diedit posisi masing-masing titik sudutnya oleh *user* melalui layar sentuh *Android Device*. Bangun datar lainnya juga dapat ditambahkan ke bidang gambar dengan mengklik tombol *Add Shape* dan memilih 1 dari 8 jenis bangun datar yang disediakan. Bangun datar yang tidak diinginkan juga dapat dihapus dari bidang gambar. Hal ini dilakukan dengan terlebih dahulu memilih bangun yang ingin dihapus dengan tombol *Prev. Shape* atau *Next Shape*, kemudian menekan tombol *Delete Shape*.

Tombol *Finish* diklik setelah semua bangun datar selesai diatur. Tulisan pada tombol *Finish* akan berubah menjadi *Draw*. Selain itu, garis-garis arsiran yang menyusun setiap bangun datar pada bidang gambar ditampilkan sebagai *preview*. *Preview* ini memungkinkan *user* untuk mengecek ulang sebelum garis-garis tersebut digambarkan pada kertas. Kemiringan garis arsiran masing-masing bangun dapat diubah dengan melakukan *gesture* rotasi melalui layar sentuh *Android Device*. Setelah selesai, tombol *Draw* diklik untuk memerintahkan perangkat penggambar melalui komunikasi *Bluetooth*, sehingga proses penggambaran garis-garis tersebut pada kertas dapat dimulai. Perangkat penggambar akan mengirimkan *feedback* setiap kali satu garis arsiran selesai digambar pada kertas, sehingga *progress* dari proses penggambaran dapat ditampilkan pada layar *Android Device*. Setelah semua garis selesai digambar, program akan mengirimkan perintah ke perangkat penggambar untuk mengembalikan pena ke posisi awal. Tampilan GUI program *Shape Rasterizer* setelah direalisasi dapat dilihat pada Gambar 12.



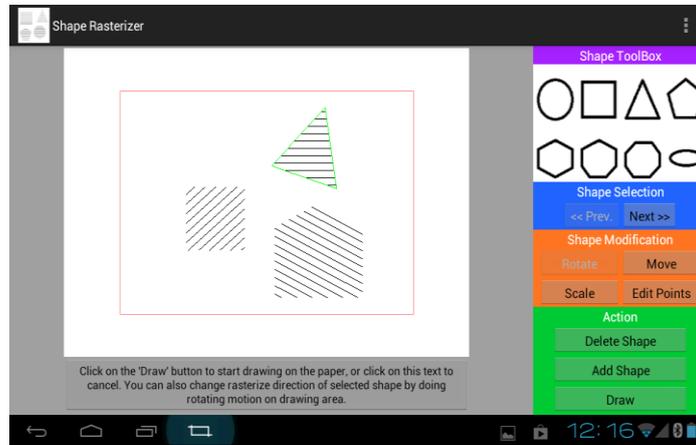
■ Gambar 12. Tampilan GUI Program *Shape Rasterizer*

Realisasi Sistem Keseluruhan dilakukan dengan menghubungkan aktuator penggambar dan modul *Bluetooth-to-Serial* dengan mikrokontroler. Hubungan motor DC pada aktuator penggambar dan mikrokontroler dijumpitani oleh modul *Motor Driver*.

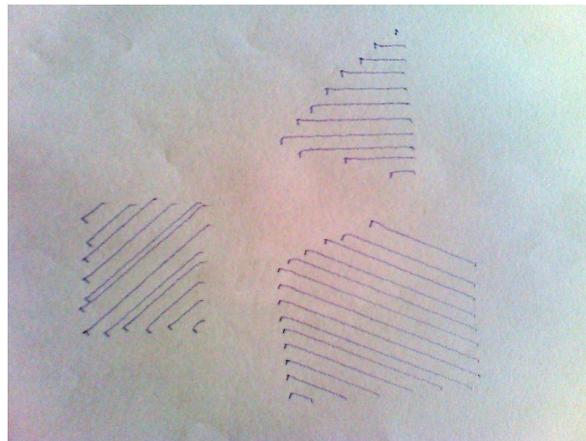
### PENGUJIAN DAN ANALISIS

Pengujian dilakukan menggunakan tiga buah bangun yang terdiri dari segitiga, persegi, dan segilima yang telah dikenai segala jenis transformasi dan perubahan kemiringan garis arsir. Pengujian ini bertujuan untuk menguji seluruh fungsi yang dapat dilakukan sistem. Hasil pengujian menunjukkan bahwa ketiga bangun tersebut dapat diarsir dengan baik pada kertas. Bangun yang digambarkan pada aplikasi *Android* dapat dilihat pada Gambar 13. Hasil arsiran di kertas dapat dilihat pada Gambar 14. Berdasarkan hasil tersebut dilakukan pengukuran dengan sistem koordinat. Koordinat ini merupakan seluruh titik ujung pada garis. Koordinat (0,0) dimulai pada ujung kiri atas dan dinyatakan dalam satuan milimeter. Pengukuran koordinat *output* dilakukan pada kertas hasil arsir menggunakan penggaris dengan tingkat keakuratan 1 mm. Pengukuran koordinat *input* dilakukan pada layar *Android* menggunakan penggaris dengan tingkat keakuratan 1 mm, kemudian dikalikan dengan faktor penskalaan terhadap ukuran A4 sebenarnya, yaitu 1:2,56. Data kemudian diproses untuk menghasilkan persentase *error*.

Persentase *error* yang dihasilkan untuk bangun segitiga dapat dilihat pada Tabel 1. Persentase *error* yang dihasilkan untuk bangun persegi dapat dilihat pada Tabel 2. Persentase *error* yang dihasilkan untuk bangun segilima dapat dilihat pada Tabel 3. Persentase rata-rata *error*  $x_1$ ,  $y_1$ ,  $x_2$ ,  $y_2$  keseluruhan berturut-turut adalah 2,82%, 8,16%, 1,66%, dan 7,98%. Berdasarkan hasil tersebut terlihat bahwa *error rate* pada sumbu  $x$  lebih kecil daripada sumbu  $y$ .



■ **Gambar 13.** Gambar Bangun pada *Android* untuk Pengujian Sistem 3 Bangun



■ **Gambar 14.** Hasil Arsiran di Kertas pada Pengujian Sistem 3 Bangun

■ **Tabel 1.** Persentase *Error* yang Dihasilkan Untuk Bangun Segitiga

Garis ke-	$\Delta x1$	% err x1	$\Delta y1$	% err y1	$\Delta x2$	% err x2	$\Delta y2$	% err y2
1	4.56	2.407095	7.52	17.27941	3	1.5625	7.52	17.27941
2	2.68	1.453993	7.64	15.70724	2.44	1.254112	7.64	15.70724
3	2.8	1.5625	7.76	14.43452	2.44	1.254112	7.76	14.43452
4	0.92	0.528493	8.88	15.08152	1.44	0.740132	8.88	15.08152
5	1.04	0.61553	8	12.5	0.12	0.060877	8	12.5
6	1.4	0.841346	8.12	11.74769	0.88	0.446429	8.12	11.74769
7	1.28	0.793651	8.24	11.09914	0.88	0.446429	8.24	11.09914
8	2.16	1.383197	8.36	10.53427	0.68	0.340545	8.36	10.53427
9	3.84	2.34375	9.48	11.22159	1.68	0.841346	9.48	11.22159
10	4.2	2.34375	10.6	11.83036	0.68	0.340545	10.6	11.83036
11	4.56	2.34375	10.72	11.31757	3.24	1.602057	10.72	11.31757
Rata-rata		1.510641		12.97757		0.808098		12.97757

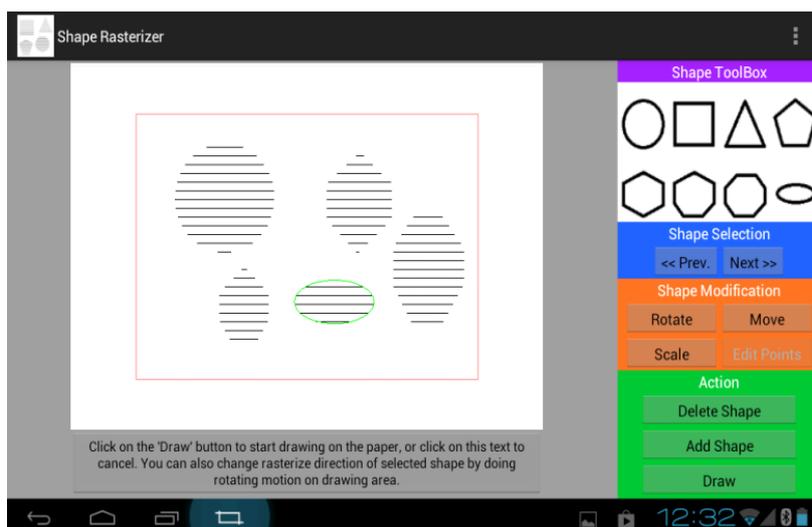
Pengujian dilanjutkan dengan 5 jenis bangun lainnya yang belum diuji, yaitu lingkaran, elips, segienam, segitujuh, dan segidelapan. Hasil pengujian menunjukkan bahwa kelima bangun datar berhasil diarsir dengan baik. Bangun yang digambarkan pada aplikasi *Android* dapat dilihat pada Gambar 15. Hasil arsiran di kertas dapat dilihat pada Gambar 16. Persentase *error* yang dihasilkan untuk bangun lingkaran dapat dilihat pada Tabel 4. Persentase *error* yang dihasilkan untuk bangun elips dapat dilihat pada Tabel 5. Persentase *error* yang dihasilkan untuk bangun segitujuh dapat dilihat pada Tabel 6. Persentase *error* yang dihasilkan untuk bangun segienam dapat dilihat pada Tabel 7. Persentase *error* yang dihasilkan untuk bangun segidelapan dapat dilihat pada Tabel 8.

■ **Tabel 2.** Persentase *Error* yang Dihasilkan Untuk Bangun Persegi

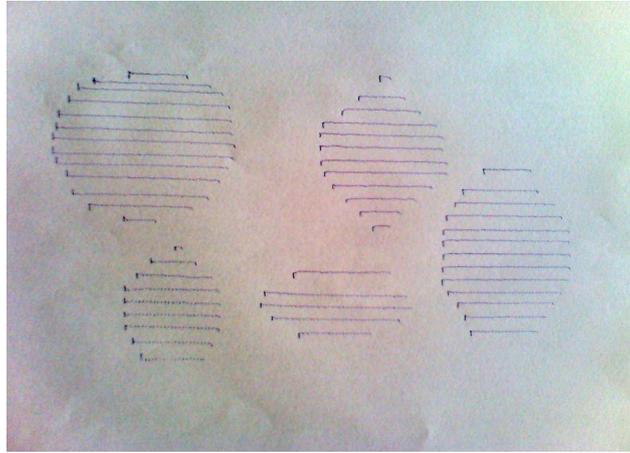
Garis ke-	$\Delta x1$	% err x1	$\Delta y1$	% err y1	$\Delta x2$	% err x2	$\Delta y2$	% err y2	% err sudut
1	1.6	1.785714	4.84	4.847756	2.28	2.34375	3.72	3.927365	11.86
2	0.6	0.669643	4.52	4.203869	0.96	0.914634	2.72	2.871622	8.7
3	1.6	1.785714	3.64	3.231534	0.64	0.568182	4.72	4.983108	1.52
4	2.6	2.901786	5.32	4.421543	3.32	2.759309	2.72	2.871622	5.8
5	2.6	2.901786	7	5.46875	2	1.5625	2.72	2.871622	9.86
6	4.6	5.133929	8.12	6.09976	6.12	4.597356	4.28	4.399671	5.8
7	5.16	5.598958	6.24	4.513889	4.12	3.094952	3.4	3.320313	6.94
8	6.84	6.850962	5.24	3.790509	7.12	5.348558	2.08	1.889535	7.14
9	8.52	7.924107	6.24	4.513889	7.12	5.348558	6.76	5.740489	1.49
10	8.2	7.118056	5.24	3.790509	7.12	5.348558	4.88	3.971354	5.63
11	8.88	7.226563	6.24	4.513889	8.12	6.09976	6.56	5.02451	1.56
12	8.56	6.556373	5.24	3.790509	8.12	6.09976	4.68	3.449292	25.32
Rata-rata		4.704466		4.432201		3.673823		3.776708	7.635

■ **Tabel 3.** Persentase *Error* yang Dihasilkan Untuk Bangun Segilima

Garis ke-	$\Delta x1$	% err x1	$\Delta y1$	% err y1	$\Delta x2$	% err x2	$\Delta y2$	% err y2	% err sudut
1	1.32	0.716146	9.08	8.248547	1.16	0.52689	12	9.375	15.22
2	2.64	1.494565	8.64	7.670455	1.16	0.52689	9.12	6.850962	4.55
3	3.52	2.052239	9.2	7.986111	2.16	0.981105	9.24	6.684028	2.27
4	5.4	3.245192	7.76	6.589674	2.16	0.981105	9.36	6.529018	9.09
5	5.28	3.27381	8.32	6.914894	1.16	0.52689	12.04	7.971398	16.67
6	4.16	2.663934	7.88	6.41276	1.16	0.52689	12.16	7.786885	14.58
7	4.16	2.663934	7	5.46875	1.16	0.52689	11.28	6.994048	14.58
8	5.16	3.304303	9.68	7.134434	0.16	0.072674	11.4	6.850962	11.11
9	4.16	2.663934	8.8	6.25	1.6	0.735294	11.52	6.716418	10.87
10	3.16	2.023566	9.92	6.798246	1.2	0.585937	12.52	7.29944	14.58
11	3.16	2.023566	10.04	6.647246	0.44	0.226151	12.52	7.29944	18.37
12	3.16	2.023566	12.72	8.014113	0.32	0.173611	11.52	6.716418	2.33
13	3.16	2.023566	12.84	7.836914	0.48	0.279851	11.52	6.716418	4.35
14	2.16	1.383197	11.96	7.078598	0.28	0.173611	11.52	6.716418	13.04
Rata-rata		2.253966		7.075053		0.488842		7.179061	10.83



■ **Gambar 15.** Gambar Bangun pada *Android* untuk Pengujian Sistem 5 Bangun



■ **Gambar 16.** Hasil Arsiran di Kertas pada Pengujian Sistem 5 Bangun

■ **Tabel 4.** Persentase *Error* yang Dihasilkan Untuk Bangun Lingkaran

Garis ke-	$\Delta x_1$	% err x1	$\Delta y_1$	% err y1	$\Delta x_2$	% err x2	$\Delta y_2$	% err y2
1	7.04	8.088235	1.64	3.371711	10.08	9.156977	1.64	3.371711
2	8.8	11.45833	2.76	5.133929	9.76	8.288043	2.76	5.133929
3	8.68	12.10938	4.88	8.288043	9.88	8.040365	4.88	8.288043
4	9.12	13.19444	6	9.375	10.44	8.322704	6	9.375
5	9.56	14.36298	5.12	7.407407	12	9.375	5.12	7.407407
6	10.56	15.86538	5.24	7.05819	12	9.375	5.24	7.05819
7	11.56	17.36779	6.36	8.014113	11	8.59375	6.36	8.014113
8	11.56	17.36779	6.48	7.670455	11	8.59375	6.48	7.670455
9	12.12	17.53472	8.6	9.598214	13	10.15625	8.6	9.598214
10	10.68	14.89955	8.72	9.206081	12.44	9.917092	8.72	9.206081
11	11.24	15.14009	7.84	7.852564	11.32	9.408245	7.84	7.852564
12	10.36	13.05444	8.96	8.536585	11.2	9.722222	8.96	8.536585
13	11.16	12.10938	9.08	8.248547	10.4	10.15625	9.08	8.248547
Rata-rata		14.0425		7.673911		9.161973		7.673911

■ **Tabel 5.** Persentase *Error* yang Dihasilkan Untuk Bangun Elipse

Garis ke-	$\Delta x_1$	% err x1	$\Delta y_1$	% err y1	$\Delta x_2$	% err x2	$\Delta y_2$	% err y2
1	8.48	5.711207	9	7.03125	11.32	6.141493	9	7.03125
2	13.36	9.319196	6.12	4.597356	13	6.770833	6.12	4.597356
3	15.36	10.71429	6.24	4.513889	11	5.729167	6.24	4.513889
4	12.92	8.854167	7.36	5.133929	13.44	7.094595	7.36	5.133929
5	14.16	9.067623	7.48	5.037716	9.64	5.457428	7.48	5.037716
Rata-rata		8.733296		5.262828		6.238703		5.262828

■ **Tabel 6.** Persentase *Error* yang Dihasilkan Untuk Bangun Segitujuh

Garis ke-	$\Delta x_1$	% err x1	$\Delta y_1$	% err y1	$\Delta x_2$	% err x2	$\Delta y_2$	% err y2
1	8.52	7.924107	9.32	7.746011	10.64	9.446023	9.32	7.746011
2	13.96	13.3003	8.44	6.728316	11.76	9.986413	8.44	6.728316
3	11.28	11.59539	8.56	6.556373	10.88	8.854167	8.56	6.556373
4	11.72	12.37331	9.68	7.134434	11.44	9.119898	9.68	7.134434
5	11.72	12.37331	9.8	6.960227	10.44	8.322704	9.8	6.960227
6	11.72	12.37331	10.92	7.483553	10.44	8.322704	10.92	7.483553
7	11.72	12.37331	11.04	7.309322	10.44	8.322704	11.04	7.309322
8	11.28	11.59539	11.16	7.146516	5.76	4.891304	11.16	7.146516
9	13.4	13.08594	10.28	6.374008	10.32	8.577128	10.28	6.374008
Rata-rata		11.88826		7.048751		8.427005		7.048751

■ Tabel 7. Persentase *Error* yang Dihasilkan Untuk Bangun Segienam

Garis ke-	$\Delta x1$	% err x1	$\Delta y1$	% err y1	$\Delta x2$	% err x2	$\Delta y2$	% err y2
1	12.76	7.020246	8.76	16.29464	13.88	7.427226	8.76	16.29464
2	13.08	7.513787	5.88	9.986413	14	7.291667	5.88	9.986413
3	12.96	7.670455	6	9.375	13.12	6.655844	6	9.375
4	15.84	9.667969	5.12	7.407407	13.24	6.546677	5.12	7.407407
5	13.28	8.234127	6.24	8.405172	10.24	5.063291	6.24	8.405172
6	13.28	8.234127	6.36	8.014113	10.24	5.063291	6.36	8.014113
7	13.28	8.234127	6.48	7.670455	9.24	4.568829	6.48	7.670455
8	11.28	6.994048	7.6	8.482143	9.24	4.568829	7.6	8.482143
9	13.4	8.052885	7.72	8.150338	11.68	5.849359	7.72	8.150338
10	14.52	8.465485	7.84	7.852564	12.56	6.455592	7.84	7.852564
11	14.64	8.288043	7.96	7.583841	13.44	7.094595	7.96	7.583841
12	14.76	8.120599	8.08	7.340116	11.32	6.141493	8.08	7.340116
Rata-rata		8.041325		8.880184		6.060558		8.880184

■ Tabel 8. Persentase *Error* yang Dihasilkan Untuk Bangun Segidelapan

Garis ke-	$\Delta x1$	% err x1	$\Delta y1$	% err y1	$\Delta x2$	% err x2	$\Delta y2$	% err y2
1	11.6	5.330882	10.6	11.83036	11.52	4.891304	10.6	11.83036
2	13.48	6.344127	7.72	8.150338	13.64	5.668218	7.72	8.150338
3	13.92	6.631098	7.84	7.852564	10.2	4.194079	7.84	7.852564
4	14.36	6.925154	7.96	7.583841	12.32	4.96134	7.96	7.583841
5	12.8	6.25	8.08	7.340116	12.88	5.133929	8.08	7.340116
6	12.8	6.25	9.2	7.986111	12.88	5.133929	9.2	7.986111
7	12.8	6.25	9.32	7.746011	12.88	5.133929	9.32	7.746011
8	12.8	6.25	9.44	7.52551	12.88	5.133929	9.44	7.52551
9	12.8	6.25	9.56	7.322304	10.32	4.155928	9.56	7.322304
10	14.36	6.925154	9.68	7.134434	10.76	4.378255	9.68	7.134434
11	12.92	6.154726	11.8	8.380682	11.2	4.605263	11.8	8.380682
12	12.48	5.873494	10.92	7.483553	8.08	3.393817	10.92	7.483553
13	13.04	6.063988	11.04	7.309322	9.52	4.04212	11.04	7.309322
Rata-rata		6.269125		7.972703		4.678926		7.972703

Berdasarkan data keseluruhan terlihat bahwa rata-rata *error* terbesar terjadi pada titik x1 garis arsiran bangun lingkaran, yaitu sebesar 14,0425%.

### KESIMPULAN

Seluruh fitur-fitur *user interactive* pada program *Android* yang meliputi operasi *move*, *rotate*, *scale*, *edit points*, *margin*, *add shape*, *next shape*, *previous shape*, *delete shape*, pemilihan jenis bangun datar, generate garis arsir, dan kemiringan garis arsir berjalan sesuai dengan yang diharapkan. *Wireless progress monitoring system* yang dibuat pada program *Android* berjalan sesuai dengan yang diharapkan. *Progress* penggambaran berhasil ditampilkan pada layar *Android Device*. Perangkat penggambar dapat mengerjakan seluruh perintah terdaftar yang diberikan kepadanya. *Android Device* berhasil berinteraksi dengan *user* melalui GUI, mengirimkan perintah penggambaran garis arsir ke perangkat penggambar, dan memantau proses pengarsiran serta menampilkannya pada *user*. Perangkat penggambar berhasil menggambar garis-garis arsir yang membentuk bangun datar pada kertas dengan *error* terbesar 14%, yaitu pada penggambaran bangun lingkaran.

### DAFTAR PUSTAKA

- [1] Z. Islam, "Google Play Matches Apple's iOS With 700,000 Apps," October 30, 2012. <http://www.tomsguide.com/us/Google-Play-Android-Apple-iOS,news-16235.html>.
- [2] J. Rosenberg, "Google Play hits 25 billion downloads," September 26, 2012. <http://officialandroid.blogspot.ca/2012/09/google-play-hits-25-billion-downloads.html>.
- [3] Canalys, "Google's Android becomes the world's leading smart phone platform," January 31, 2011. <http://www.canalys.com/newsroom/google%E2%80%99s-android-becomes-world%E2%80%99s-leading-smart-phone-platform>.
- [4] H. Victor, "Android steals Symbian's top smartphone OS crown," January 31, 2011. [http://www.phonearena.com/news/Android-steals-Symbians-Top-Smartphone-OS-crown\\_id16332](http://www.phonearena.com/news/Android-steals-Symbians-Top-Smartphone-OS-crown_id16332).