

## 4. PENGUJIAN SISTEM

Pengujian sistem yang akan dilakukan disini meliputi beberapa hal yaitu:

1. Pemakaian histogram dalam pembedaan warna.
2. *Tracking* warna.
3. Pengujian pergerakan mouse dengan 3 metode.
4. Pengujian *event-event* khusus.

Kelima pengujian ini akan dibahas dalam sub bab yang terpisah untuk lebih jelasnya.

### 4.1. Pengujian Pemakaian Histogram Dalam Pembedaan Warna.

Pemakaian histogram dalam pembedaan warna sangat diperlukan untuk mendapatkan informasi untuk membedakan obyek dari latarnya. Karena hal ini merupakan tahap awal yang sangat penting dan dipakai disemua metode maka tingkat keberhasilannya perlu diuji.

Yang akan diuji adalah apakah pengambilan sampel melalui ROI berhasil, juga dilakukan pengujian apakah benar histogram dapat dipakai untuk membedakan obyek dan latarnya dengan memanfaatkan perbedaan warna obyek dan latar. Pengujian akan dilakukan dengan mencoba mengambil sampel warna ujung bolpoint yang berwarna merah kemudian warna sampel, hue, saturation dan value dari sampel ditampilkan pada layar. Setelah itu diuji lagi dengan mengambil sampel warna kulit dan dilihat apakah dengan histogram dapat membedakan antara obyek tangan dengan latarnya.

Pengujian akan dilakukan menggunakan kamera web cam intel dengan resolusi 160x120 pixel. Saat dilakukan pengujian satu hal yang harus diingat adalah bahwa warna image sample harus benar-benar berbeda dengan warna latarnya. Hal ini penting untuk meningkatkan keberhasilan dalam proses *tracking*. Cara pengambilannya adalah dengan mendekatkan benda yang akan di-*tracking*

ke kamera sampai warna benda yang akan di-*tracking* berada dalam kotak ROI yang berwarna merah. Seperti pada gambar 4.1.



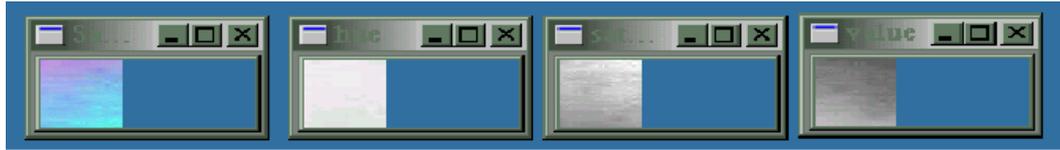
Gambar 4. 1 Penempatan Warna Obyek pada ROI

Setelah itu tombol init histogram ditekan. Seperti pada contoh gambar 4.2.

Gambar 4. 2 Letak Tombol Init Histogram

Hasil dari init histogram adalah sebuah histogram array dari hue, saturation dan valuenya. Gambar masing-masing dari channel image sampel ditunjukkan pada gambar 4.3. Dari kiri ke kanan adalah gambar :

1. Image sampel warna dari obyek.
2. Hue dari image sampel.
3. Saturation dari image sampel.
4. Value dari image sampel.



Gambar 4. 3 Image Sampel dan HSV-nya

Dari gambar-gambar ini kita ketahui bahwa pengambilan sampel lewat ROI dapat dilakukan. Setelah pengujian ROI berhasil dilakukan lagi pengambilan warna sampel kulit untuk pengujian pembedaan obyek tangan dengan latarnya.

Setelah gambar sampel tangan diambil kemudian dicari histogram untuk hue saturation dan value. Setelah dicari histogramnya maka dengan histogram yang ada program dapat membedakan warna yang sesuai dengan sampel dan yang tidak. Yang sesuai dengan sampel dianggap sebagai obyek sedangkan yang tidak dianggap sebagai latar. Obyek akan diberi warna putih sedangkan latar akan diberi warna hitam.

Karena pembedaan obyek dan latar hanya dengan memakai perbedaan warna maka bila ada benda pada latar yang memiliki warna yang mirip dengan obyek akan dianggap sebagai obyek juga. Hal ini membuat error yang kadang cukup besar dalam proses *tracking*. Oleh karena itu benda-benda yang mempunyai warna yang mirip dengan obyek sebaiknya dihindarkan.

Gambar hasil pembedaan obyek tangan dari latarnya ditunjukkan oleh gambar 4.4 dan 4.5.



Gambar 4. 4 Obyek Tangan



Gambar 4. 5 Hasil Pembedaan Obyek Tangan dan Latarnya

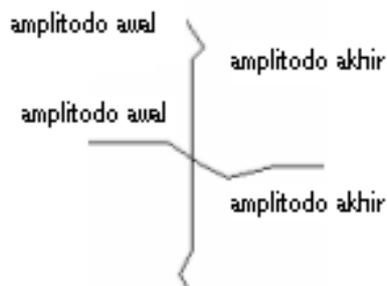
Bila dilihat dari hasilnya, nampak bahwa kurang baik, gambar tangan yang ada masih banyak cacatnya dan disekitarnya masih terdapat banyak noise. Namun bila hanya untuk melakukan *tracking* masih dapat digunakan.

#### 4.2. Tracking Warna.

Secara sekilas metode *tracking* warna sebenarnya hanya diperlukan untuk metode *tracking* dan metode *tracking* berlanjut. Namun pada metode ROI *tracking* warna juga diperlukan untuk menentukan didaerah ROI mana koordinat benda sebenarnya berada. Karena itu berarti sama seperti pembedaan warna dengan ROI *tracking* warna merupakan bagian yang penting dalam setiap metode dan memang layak dan harus diuji keberhasilannya.

*Tracking* warna dilakukan dengan kamera intel web cam intel dengan resolusi 160x120 pixel dengan jarak 1-2 m dari kamera. Kamera diletakkan diatas layar monitor dengan posisi tepat menghadap pemakai atau ke obyek yang akan di-*tracking*. Warna obyek yang dipakai adalah warna merah karena warna latar cenderung kebiruan. Warna lain juga sebetulnya dapat dipilih namun pada pengujin ini dipakai warna merah.

*Tracking* warna yang dilakukan disini adalah dengan cara mencoba apakah cursor mouse yang telah diprogram untuk mengikuti hasil dari koordinat *tracking* dapat bergerak mengikuti gerakan obyek berupa ujung sebuah bolpoint yang berwarna merah. Gambar 4.6 adalah gambar hasil dari pergerakan mouse yang digerakkan untuk menggambar sebuah salib sumbu dan digambar pada adobe photoshop .



Gambar 4. 6 Hasil Pergerakan Mouse Dari Proses *Tracking*

Dari hasilnya dapat dilihat bahwa hasilnya cukup memadai, walaupun tidak terlalu menyerupai sebuah salib sumbu. Tabel akan menunjukkan hasil-hasil penting dari pengujian untuk *tracking* warna yaitu:

1. dari gambar salib sumbu pada gambar 4.6 diuji besar penyimpangan dengan adobe photoshop dengan cara memperbesar zoom sampai 100% kemudian mengukur besar amplitudo tiap garis pada salib sumbu.
2. dilakukan 10 kali pengujian pada tahap 1 untuk dilihat seberapa besar kemungkinan penyimpangan
3. uji berapa besar perbedaan skala antara pergeseran pada layar dengan pergeseran tangan yang sebenarnya. Ini dilakukan dengan mengukur pergeseran tangan saat menggambar sebuah garis di adobe photoshop sebesar 1 cm. Pengukuran pergeseran tangan dilakukan dengan memakai penggaris 50 cm.

Bila dilihat dari hasil pengujian ini maka dapat disimpulkan bahwa metode *tracking* warna hanya dapat dilakukan bila aplikasi yang dipakai cukup sederhana dan tidak terlalu menuntut ketelitian mengingat error/penyimpangan yang dihasilkan masih cukup besar. Hal ini disebabkan karena error hasil dari perbedaan warna dan terutama karena adanya perbandingan yang dilakukan saat mengubah dari nilai koordinat hasil *tracking* yang didapat dari kamera dengan resolusi kecil ke resolusi layar yang besar sehingga error/penyimpangan kecil dapat menyebabkan error yang besar. Namun sekalipun hasil *tracking* ini buruk tapi dengan metode pergerakan mouse yang tepat akan dapat mengurangi bahkan menghilangkan kelemahan ini.

Tabel 4. 1 tabel hasil uji *tracking* warna

| Jenis uji   | Kegunaan dilakukan pengujian ini   | Batasan nilai            | Hasil uji  | Kesimpulan  |
|---|--|--------------------------|------------|---|
| Penyimpangan dari garis lurus                         | Untuk mengetahui apakah kursor mouse benar-benar mengikuti gerak tangan          | 0-20 pixel               | 0-10 pixel | Dapat dipakai untuk aplikasi sederhana namun sulit untuk menggambar |
| Kemungkinan penyimpanan terjadi                       | Mengetahui kemungkinan terjadinya penyimpangan                                   | 0%-30%                   | 70%        | Penyimpangan pasti akan terjadi karena hasil dari perbandingan      |
| Skala gerak pada layar dengan gerak tangan (jarak 1m) | Mengetahui perbandingan skala pergerakan antara gerak tangan dengan kursor mouse | Tidak lebih dari 1:20 cm | 1:15 cm    | Cukup memadai   |

### 4.3. Pengujian Pergerakan Mouse Dengan 3 Metode.

Pangujian pada metode pergerakan mouse ini perlu dilakukan untuk mengetahui hal-hal penting berikut:

1. Penyimpangan pergerakan mouse saat obyek bergerak lurus (menguji kestabilan mouse).
2. Kemampuan melakukan gerak lingkaran (menguji keluwesan gerak mouse).
3. Penyimpangan pergerakan mouse saat posisi obyek diam (menguji kemantapan posisi mouse saat diam).
4. Pergeseran minimal yang dapat dilakukan (menguji resolusi mouse).
5. Kecepatan mouse bergerak dari satu titik ke titik lain.

Pengujian pergerakan mouse dilakukan dengan cara:

1. Menggambar salib sumbu dan melihat berapa besar pergeseran yang terjadi. Pengukuran dilakukan dengan adobe photoshop dengan perbesaran maksimal.
2. Menguji apakah setiap metode mampu menggambar suatu lingkaran.

3. Juga dilakukan uji dengan mendinginkan tangan/obyek selama lima detik dan dilakukan sebanyak 8 kali pada suatu posisi namun pada gambar hanya akan ditunjukkan hasil 1 kali percobaan. Pengukuran dilakukan dengan adobe photoshop dengan perbesaran maksimal.
4. Juga dilakukan uji dapatkah menggambar sebuah titik yang saling berdekatan semakin dekat titik maka semakin besar resolusinya. Pada uji ini tidak akan ditampilkan gambar 2 titik hasil penggambaran dari tiap metode karena terlalu kecil namun akan langsung disebutkan jarak 2 titik. Pengukuran dilakukan dengan adobe photoshop dengan perbesaran maksimal.
5. Selain itu juga akan dilihat kecepatan pergerakan mouse dari suatu titik ke titik lainnya pada monitor. Uji kecepatan akan dilakukan 6 kali dengan 3 arah dan tujuan berbedaa untuk setiap metode namun khusus untuk metode *tracking* berlanjut akan dilakukan 8 kali dengan 4 arah dan tujuan berbeda. Hal ini disebabkan karena untuk metode *tracking* berlanjut ada suatu hal khusus yang harus diuji yaitu uji dari satu tahap pergerakan.

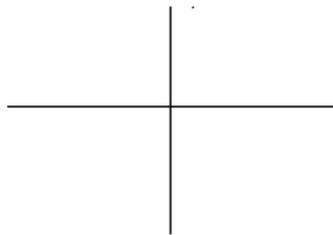
Keseluruhan pengujian ini akan dilakukan dengan menggunakan kamera yang sama dengan saat pengujian penggunaan histogram dan pengujian *tracking* yaitu kamera intel web cam dengan resolusi 160x120 pixel dengan jarak antara pemakai dan kamera sejauh kira-kira 1-2 meter. Untuk uji kecepatan pengukuran waktu yang dipergunakan adalah stop watch dengan ketelitian sampai 1/100 detik.

#### 4.3.1. Metode ROI

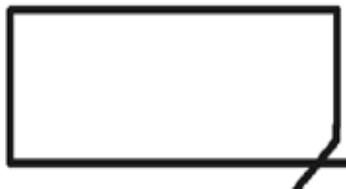
Hasil pengujian untuk metode ROI adalah:

1. Gambar 4.7a ini menunjukkan hasil dari metode ROI yang sangat sempurna dalam menggambarkan sumbu salib penyimpangannya adalah 0 pixel, ini membuktikan bahwa metode ini memiliki keunggulan dalam hal kestabilan dan pengontrolan pergerakan mouse.
2. Namun sayangnya, karena metode ROI bergerak seperti metode joystick maka untuk menghasilkan gerakan yang luwes dan berbelok-belok amatlah sulit untuk dilakukan. Hal ini terbukti pada gambar 4.7b yang seharusnya berbentuk lingkaran tetapi yang terjadi justru berbentuk persegi panjang.

3. Pada saat mouse dalam kondisi diam, gambar 4.7c menunjukkan tidak adanya pergerakan sama sekali penyimpangannya 0 pixel, hal ini sekali lagi membuktikan metode ini sangat tepat bila kestabilan dan kemantapan gerak sangat dibutuhkan.
4. untuk uji ini hasil yang diperoleh dari adobe photoshop, metode ini mampu menggambar 2 titik tepat bersebelahan berarti resolusinya sangat tinggi.
5. untuk kecepatan tabel 4.2 menunjukkan hasil yang mengecewakan. Bila dilihat dari tabel nampak bahwa kecepatan perpindahan mouse sangatlah lamban karena lebih dari 1 detik. Bahkan untuk menjangkau dari ujung ke ujung diperlukan waktu hampir 4 detik. Waktu yang sangat lama bila dibandingkan dengan mouse konvensional.



Gambar 4.7 a Hasil Uji Salib Sumbu



Gambar 4.7 b Hasil Uji Lingkaran



Gambar 4.7 c Hasil Uji Saat Tidak Ada Pergerakan

Tabel 4. 2 Kecepatan Mouse Metode ROI

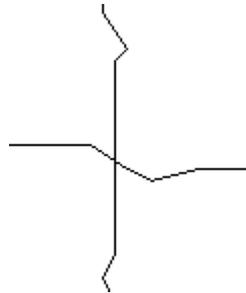
| No Percobaan | Asal Kursor Mouse | Akhir Kursor Mouse | Waktu (detik) |
|--------------|-------------------|--------------------|---------------|
| 1            | Pojok Kiri Atas   | Pojok Kanan Bawah  | 3.31          |
| 2            | Pojok Kiri Atas   | Pojok Kanan Bawah  | 3.40          |
| 3            | Kanan             | Kiri               | 2.68          |
| 4            | Kanan             | Kiri               | 2.32          |
| 5            | Atas              | Bawah              | 1.47          |
| 6            | Atas              | Bawah              | 1.38          |

#### 4.3.2. Metode *Tracking*

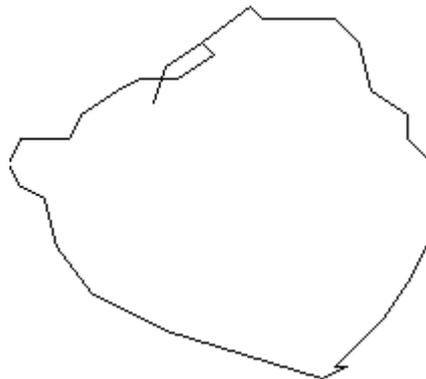
Hasil dari metode *tracking* ini akan sama dengan hasil pengujian *tracking* warna karena memang hasil dari *tracking* warna langsung dipakai untuk menggerakkan mouse.

1. Gambar 4.8a menunjukkan gambar salib sumbu hasil dari penggambaran dengan metode *tracking*. Untuk penyimpangannya sama dengan uji *tracking* warna karena memang metode ini mengambil hasil langsung dari hasil *tracking* warna, penyimpangannya 5-10 pixel saat menggambar garis lurus.
2. Sedangkan gambar 4.8b adalah gambar lingkaran hasil dari penggambaran metode *tracking*. Metode ini dapat bergerak membentuk suatu lingkaran dapat dilakukan dengan baik.
3. Gambar 4.8c yang menampilkan kondisi pergerakan mouse yang terjadi saat seharusnya mouse dalam kondisi diam selama 5 detik. Pada gambar 4.8c dilakukan percobaan sampai 8 kali dan dilakukan di 8 tempat berbeda hasil salah satu percobaan terlihat pada gambar itu. Penyimpangannya sama dengan uji *tracking* warna karena memang metode ini mengambil hasil langsung dari hasil *tracking* warna, penyimpangannya 5-10 pixel.
4. Untuk uji ini hasil penggambaran 2 titik menunjukkan selisih sebesar 3 pixel menunjukkan resolusi yang tidak sebaik metode ROI.
5. Untuk kecepatan metode ini sudah tidak perlu diuji lagi karena memang sangat cepat. Tabel 4.3 akan membuktikannya. Waktu yang sedemikian kecil

ini disebabkan karena kursor mouse dapat melompat kearah manapun pemakai menghendakinya, jadi inilah kelebihan dari metode ini.



Gambar 4.8 a Hasil Uji Salib Sumbu



Gambar 4.8 b Hasil Uji Lingkaran



Gambar 4.8 c Hasil Uji Saat Tidak Ada Pergerakan

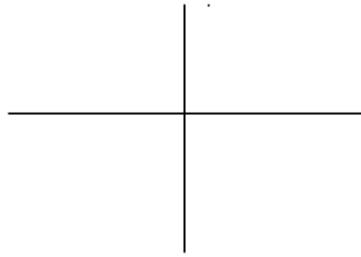
Tabel 4. 3 Kecepatan Mouse Metode *tracking*

| No Percobaan | Asal Kursor Mouse | Akhir Kursor Mouse | Waktu (Detik) |
|--------------|-------------------|--------------------|---------------|
| 1            | Pojok Kiri Atas   | Pojok Kanan Bawah  | 0.00          |
| 2            | Pojok Kiri Atas   | Pojok Kanan Bawah  | 0.01          |
| 3            | Kanan             | Kiri               | 0.01          |
| 4            | Kanan             | Kiri               | 0.00          |
| 5            | Atas              | Bawah              | 0.01          |
| 6            | Atas              | Bawah              | 0.00          |

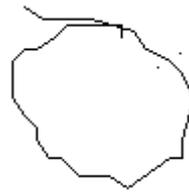
#### 4.3.3. Metode Tracking Berlanjut.

Dari hasil gambar ini kita ketahui bahwa metode ini dapat dikatakan sangat sempurna, namun kelemahannya adalah pada kecepatan dan kemudahannya. Inilah hasil pengujian dari metode ini:

1. Ia dapat menggambarkan salib sumbu dengan sangat baik sama seperti metode ROI (gambar 4.9a), dengan penyimpangan 0 pixel.
2. Selain itu ia juga dapat menggambar lingkaran dengan baik pula (gambar 4.9b).
3. Kestabilannya saat diam pun sama dengan metode ROI. Pergerakan hanya terjadi karena tangan pemakai tidak dapat mempertahankan posisi diamnya dengan benar namun bila obyek yang di-*tracking* tidak ditahan dengan tangan kestabilannya akan sama dengan ROI (gambar 4.9c).
4. Untuk uji ini hasil penggambaran 2 titik menunjukkan selisih sebesar 1 pixel menunjukkan resolusi yang tidak sebaik metode ROI namun lebih baik dari pada metode *tracking*.
5. Sekalipun metode ini merupakan metode yang lebih baik bukan berarti metode ini tanpa kelemahan. Kestabilan dan keluwesannya harus dibayar mahal dengan kecepatannya. Tabel 4.4 akan membuktikan hal itu. Bila dilihat dari tabel maka terdapat sesuatu yang aneh, saat metode ini dipakai untuk percobaan 1-6 semua menunjukkan waktu yang sangat lama, namun bila dipakai untuk mengakses jarak yang cukup dekat waktunya lebih cepat dibanding dengan metode ROI. Hal ini disebabkan karena metode ini cukup sulit dilakukan sehingga untuk mengakses jarak yang cukup jauh diperlukan berkali-kali pergeseran dan untuk melakukan berkali-kali pergeseran ini cukup sulit dan memakan waktu. Namun bila pergeseran dilakukan hanya sekali maka ini mudah untuk dilakukan dan waktu untuk melakukannya pun singkat. Ini dibuktikan dengan percobaan 7-8. Pada percobaan 7-8, yang dimaksud dengan titik terjauh 1 adalah titik terjauh yang dapat dilakukan tanpa melakukan pergeseran ulang.



Gambar 4.9 a Hasil Uji Salib Sumbu



Gambar 4.9 b Hasil Uji Lingkaran



Gambar 4.9 c Hasil Uji Saat Tidak Ada Pergerakan

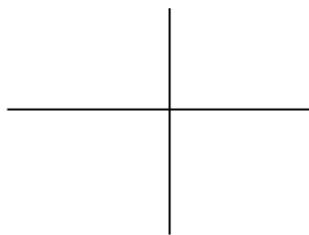
Tabel 4. 4 Kecepatan Mouse Metode *tracking* berlanjut

| No Percobaan | Asal Kursor Mouse | Akhir Kursor Mouse | Waktu (Detik) |
|--------------|-------------------|--------------------|---------------|
| 1            | Pojok Kiri Atas   | Pojok Kanan Bawah  | 10.23         |
| 2            | Pojok Kiri Atas   | Pojok Kanan Bawah  | 10.4          |
| 3            | Kanan             | Kiri               | 8.12          |
| 4            | Kanan             | Kiri               | 7.82          |
| 5            | Atas              | Bawah              | 7.01          |
| 6            | Atas              | Bawah              | 6.87          |
| 7            | Pojok Kiri Atas   | Titik Terjauh 1    | 0.99          |
| 8            | Pojok Kiri Atas   | Titik Terjauh 1    | 0.50          |

#### 4.3.4. Perbandingan Dengan Gerak Kursor Mouse Dengan Mouse.

Pada tugas akhir ini juga akan dicoba perbandingan pergerakan dan kecepatan antara ketiga metode itu dengan pergerakan dan kecepatan mouse asli.

1. Gambar 4.10 a dan c menunjukkan kestabilan dan control yang luar biasa dari mouse asli baik saat diam maupun saat bergerak. Kestabilannya dapat disamai oleh metode ROI dan *tracking* berlanjut, yaitu penyimpangannya 0 pixel. Hanya saja saat menggambarkan salib sumbu dengan mouse pemakai perlu lebih berusaha agar hasilnya dapat benar-benar lurus seperti gambar 4.10a.
2. Untuk keluwesan gambar 4.10 b membuktikan mouse asli dapat menggambar lingkaran dengan baik.
3. Untuk uji 2 titik mouse asli dapat menggambar 2 titik dengan selisih 0 pixel yang menunjukkan resolusi yang tinggi.
4. Untuk kecepatannya sangatlah cepat, hal ini ditunjukkan tabel 4.5 namun memang mungkin akan kalah bila dibandingkan metode *tracking* karena pada metode ini kursor mouse dapat langsung melompat kearah yang ditunjuk pemakai, namun hal ini tidak terlalu berpengaruh karena memang kelambatannya tidak akan terasa sama sekali karena terpaut hanya beberapa puluh mikro detik.



Gambar 4.10 a Hasil Uji Salib Sumbu



Gambar 4.10 b Hasil Uji Lingkaran



Gambar 4.10 c Hasil Uji Saat Tidak Ada Pergerakan

Tabel 4. 5 Kecepatan Mouse Asli

| No Percobaan | Asal Kursor Mouse | Akhir Kursor Mouse | Waktu (Detik) |
|--------------|-------------------|--------------------|---------------|
| 1            | Pojok Kiri Atas   | Pojok Kanan Bawah  | 0.02          |
| 2            | Pojok Kiri Atas   | Pojok Kanan Bawah  | 0.01          |
| 3            | Kanan             | Kiri               | 0.00          |
| 4            | Kanan             | Kiri               | 0.01          |
| 5            | Atas              | Bawah              | 0.00          |
| 6            | Atas              | Bawah              | 0.00          |

#### 4.4. Metode ROI Untuk *Event-Event* Khusus.

Pengujian *event-event* khusus sangatlah diperlukan mengingat *event-event* ini biasanya menentukan hasil dari suatu keputusan yang harus dibuat oleh pemakai. Semua program komputer pastilah memakai *event-event* khusus ini bila melakukan suatu keputusan atau mungkin pergerakan yang amat penting dan menentukan karena itu tingkat keberhasilan pengujian ini haruslah tinggi yaitu sekitar 90 %. Pengujian *event-event* khusus dilakukan dengan pengujian *klik* kiri , *klik* kanan, *drag and drop*, dan *double klik* sebanyak 10 kali.

Dengan metode ROI akan dilakukan pengujian sebanyak 10 kali percobaan terhadap tiap *event* . Hasilnya akan ditunjukkan oleh tabel 4.6.

Tabel 4. 6 Keberhasilan Metode ROI

| Jenis Even           | Banyak Percobaan | Banyak Keberhasilan | Tingkat Keberhasilan |
|----------------------|------------------|---------------------|----------------------|
| <i>Klik</i> kiri     | 10               | 10                  | 100%                 |
| <i>Klik</i> kanan    | 10               | 10                  | 100%                 |
| <i>Drag and drop</i> | 10               | 9                   | 90%                  |
| <i>Double klik</i>   | 10               | 10                  | 100%                 |

Bila dilihat dari tabel maka nampaknya metode ini dapat digunakan untuk membangkitkan *event-event* khusus pada mouse. Pada tugas akhir kali ini 2 metode menggunakan metode ini dalam mengakses *event-event* khusus pada mouse. Yang tidak menggunakan metode ini hanyalah metode *tracking*. Metode *tracking* tidak menggunakan metode ini karena memang cukup sulit untuk dilakukan bila pemakai belum terbiasa. Jadi metode ini memiliki tingkat keberhasilan yang sangat baik namun untuk pemakai yang tidak terbiasa pasti akan mengalami kesulitan.

#### **4.5. Pengujian Dengan Resolusi Kamera Yang Lebih Tinggi**

Dengan kamera resolusi kamera yang lebih tinggi akan diuji apakah hasil yang dicapai akan lebih baik lagi. Uji ini akan dilakukan akan sama dengan uji *tracking* warna ditambah dengan uji pada metode *tracking* saja. Karena resolusi kamera yang lebih tinggi tidak akan terlalu berpengaruh untuk metode ROI. Jadi uji yang akan dilakukan adalah:

1. Penyimpangan pergerakan mouse saat obyek bergerak lurus (menguji kestabilan mouse).
2. Penyimpangan pergerakan mouse saat posisi obyek diam (menguji kemantapan posisi mouse saat diam).
3. kemungkinan terjadi penyimpangan.
4. Kemampuan melakukan gerak lingkaran (menguji keluwesan gerak mouse).
5. Pergeseran minimal yang dapat dilakukan (menguji resolusi mouse).
6. skala pergeseran tangan dengan skala pergeseran pada layar.
7. Kecepatan mouse bergerak dari satu titik ke titik lain.

Cara melakukan pengujian adalah sama dengan pengujian pada *tracking* warna dan pada pergerakan mouse yaitu:

1. Menggambar salib sumbu dan melihat berapa besar pergeseran yang terjadi. Pengukuran dilakukan dengan adobe photoshop dengan perbesaran maksimal.
2. Juga dilakukan uji dengan mendinginkan tangan/obyek selama lima detik dan dilakukan sebanyak 8 kali pada suatu posisi namun pada gambar hanya akan

ditunjukkan hasil 1 kali percobaan. Pengukuran dilakukan dengan adobe photoshop dengan perbesaran maksimal.

3. Melakukan uji 1 sebanyak 10 kali kemudian dari 10 kali percobaan berapa banyak yang menyimpang.
4. Menguji apakah setiap metode mampu menggambar suatu lingkaran.
5. Juga dilakukan uji dapatkah menggambar 2 buah titik yang saling berdekatan dan dihitung selisihnya. Pengukuran dilakukan dengan adobe photoshop dengan perbesaran maksimal.
6. Uji skala dilakukan sama dengan uji skala pada *tracking* warna. Ini dilakukan dengan mengukur pergeseran tangan saat menggambar sebuah garis di adobe photoshop sebesar 1 cm. Pengukuran pergeseran tangan dilakukan dengan memakai penggaris 50 cm.
7. Selain itu juga akan dilihat kecepatan pergerakan mouse dari suatu titik ke titik lainnya pada monitor seperti pada uji-uji kecepatan yang sebelumnya, dan pengukuran waktu dilakukan dengan stopwatch dengan ketelitian 1/100 detik.

Uji akan dilakukan dengan jarak antara kamera dengan pemakai sejauh 1-2 meter dan hasil-hasil pengujian akan ditampilkan pada tabel 4.7.

Dilihat dari hasil-hasil pengujiannya maka dapat disimpulkan bahwa dengan resolusi kamera yang lebih tinggi memang dapat dicapai hasil yang lebih baik. Semua *performance* dari software ini dapat lebih ditingkatkan, hanya saja penyimpangan yang terjadi memang lebih besar kemungkinannya karena dengan resolusi yang lebih tinggi maka kamera akan lebih peka terhadap suatu perubahan sehingga kemungkinan terjadi penyimpangan pun makin besar. Sekalipun demikian namun besar penyimpangan ternyata dapat ditekan sehingga penyimpangan yang terjadi tidak terlalu besar.

Tabel 4. 7 Hasil Pengujian Dengan Resolusi Kamera Yang Lebih Tinggi

| Jenis pengujian          | Hasil uji dengan resolusi 160x120 | Hasil uji dengan resolusi 320x240 | Satuan | Kesimpulan   |
|--------------------------|-----------------------------------|-----------------------------------|--------|--|
| Menggambar salib sumbu   | 5-10                              | 3-7                               | Pixel  | Penyimpangan yang terjadi lebih kecil karena perbandingan resolusi layar dengan sesungguhnya lebih kecil |
| Penyimpangan saat diam   | 5-10                              | 3-8                               | Pixel  | Penyimpangan yang terjadi lebih kecil karena perbandingan resolusi layar dengan sesungguhnya lebih kecil |
| Kemungkinan penyimpangan | 70                                | 90                                | %      | Karena skala lebih kecil maka penyimpangan lebih sering terjadi karena kepekaan kamera bertambah         |
| Menggambar lingkaran     | Dapat                             | Dapat                             | -      | Tetap dapat menggambar lingkaran bahkan lebih baik   |
| Selisih 2 buah titik     | 3                                 | 1                                 | Pixel  | Karena resolusinya meningkat maka mampu resolusi <i>tracking</i> -nya pun meningkat                      |
| Skala pergeseran         | 1:15                              | 1::10                             | Cm     | Skala perbandingan pun lebih kecil karena resolusi kamera meningkat                                      |
| Kecepatan mouse          | 0.00-0.01                         | 0.00-0.01                         | Detik  | Untuk kecepatan tetap sama karena tidak berpengaruh  |