

## BAB IV

### IMPLEMENTASI DAN EVALUASI

#### 4.1 Implementasi Sistem

Tahap ini merupakan pembuatan perangkat lunak yang disesuaikan dengan rancangan atau desain sistem yang telah dibuat. Aplikasi yang dibuat akan diterapkan berdasarkan kebutuhan.

Implementasi sistem yang dilakukan untuk mencoba menerapkan aplikasi *Interactive Whiteboard* yang dapat dilihat pada Gambar 4.1.



Gambar 4.1 Implementasi Rancang Bangun Aplikasi *Interactive Whiteboard*

Aplikasi *interactive whiteboard* ini membutuhkan *bluetooth* sebagai media untuk menghubungkan antara *wiimote* dengan *pc*, setelah itu tampilan pada *pc* ditampilkan ke layar melalui proyektor yang sudah dihubungkan ke *pc*.

*Wiimote* berfungsi sebagai alat untuk mendapatkan posisi koordinat pada *IR Pen* yang digerakkan oleh pengguna untuk menulis di atas layar

Implementasi *interactive whiteboard* juga membutuhkan perangkat keras dan perangkat lunak dengan kondisi tertentu agar dapat berjalan dengan baik. Didalam pembahasan perangkat lunak dan keras, yang dibahas hanya seputar perangkat yang ada pada aplikasi *interactive whiteboard*. Berikut ini adalah kebutuhan perangkat keras dan perangkat lunak yang digunakan dalam tahap pembuatan aplikasi maupun tahap penerapan aplikasi.

#### 4.1.1 Pembuatan Perangkat Lunak

##### A. Menulis Dan Menghapus

Prosedur – prosedur yang dipakai untuk menulis adalah sebagai berikut :

- a. Prosedur *mouse down*, prosedur ini pertama – tama melakukan pengecekan apakah tombol mouse yang ditekan adalah tombol sebelah kiri jika ya maka akan membuat figur baru dalam mewakili serangkaian garis yang saling terhubung.

```
private void Layer_MouseDown(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButton.Left)
    {
        pointPath.StartFigure();
    }
}
```

b. Prosedur *mouse move*, prosedur ini pertama – tama melakukan pengecekan apakah tombol mouse yang ditekan adalah tombol sebelah kiri jika ya maka akan menambahkan koordinat untuk garis pada figur yang telah dibuat pada prosedur *figure*.

```
private void Layer_MouseMove(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButton.Left)
    {
        try
        {
            pointPath.AddLine(e.X, e.Y, e.X, e.Y);
        }
        catch (Exception) { }
    }
    Layer.Invalidate();
}
```

c. Prosedur *paint*, pada prosedur ini pertama – tama diawali dengan menempelkan kanvas ke *variable graphics* yang telah dibuat pada waktu inisialiasi *form drawing* setelah itu menggambar di *graphics* tersebut dengan memberi parameter *myPen* dan *pointPath*.

```
private void Layer_Paint(object sender, PaintEventArgs e)
{
    try
    {
        g = Graphics.FromImage((Image)mainCanvas);

        g.DrawPath(myPen, pointPath);
    }
    catch (Exception) { }
}
```

## B. Mengganti Slide

Prosedur – prosedur yang dipakai untuk mengganti slide adalah sebagai berikut :

- a. Prosedur *next slide*, prosedur ini diawali dengan melakukan pengecekan apakah *index* yang menjadi acuan masih kurang dari jumlah canvas yang telah dibuat dengan menekan tombol *new*, jika ya maka akan menambah nilai dari *variable index* dan memanggil prosedur *changeSlide*.

```
private void btnNext_Click(object sender, EventArgs e)
{
    if (cur_idx < canvas.Count-1)
    {
        cur_idx++;
        changeSlide();
        btnSaveAs.Enabled = false;
        if (canProp[cur_idx].isSave)
            btnSaveAs.Enabled = true;
    }
}
```

- b. Prosedur *prev slide*, prosedur ini diawali dengan melakukan pengecekan apakah *index* yang menjadi acuan masih lebih dari nilai 0, jika ya maka akan mengurangi nilai dari *variable index* dan memanggil prosedur *changeSlide*.

```
private void btnPrev_Click(object sender, EventArgs e)
{
    if (cur_idx > 0)
    {
        cur_idx--;
        changeSlide();
        btnSaveAs.Enabled = false;
        if (canProp[cur_idx].isSave)
            btnSaveAs.Enabled = true;
    }
}
```

c. Prosedur *change slide*, pertama – tama membuat *pointPath* yang digunakan untuk acuan dalam menyimpan koordinat x dan y dalam membuat garis yang saling terhubung setelah itu memberikan isi *default* pada kanvas yang dipakai acuan dalam menggambar dengan *default bitmap* dan dilakukan pengecekan apakah index yang menjadi acuan dalam setiap *slide* masih lebih dari nilai 0, jika ya maka kanvas tersebut diisi dengan *bitmap* pada index acuan, kemudian menempelkan *bitmap tersebut ke picturebox* yang diberi nama *layer*.

```
private void changeSlide()
{
    pointPath = new System.Drawing.Drawing2D.GraphicsPath();
    mainCanvas = (Bitmap)canProp[cur_idx].defImage;
    if (canProp[cur_idx].cur_idx_store >= 0) mainCanvas =
    (Bitmap)canProp[cur_idx].childImage[canProp[cur_idx].cur_idx_st
    ore].Clone();
    Layer.Image = mainCanvas;
    layerInfo.Caption = "Page : " + (cur_idx + 1) + "/" +
    (canvas.Count );
}
```

### C. Menyimpan

Prosedur yang dipakai untuk menyimpan adalah sebagai berikut :

Prosedur simpan, prosedur ini menampilkan jendela untuk menyimpan jika memilih tombol *OK* maka akan memanggil prosedur *saveJpeg* dengan memberikan *parameter* berupa nama *file*, kanvas yang akan disimpan dan kualitas gambar.

```
private void btnSave_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
{
    if (!canProp[cur_idx].isSave)
```

```

{
    sfd.FileName = "Untitled" + incSave;
    if (sfd.ShowDialog() ==
System.Windows.Forms.DialogResult.OK)
    {
        string fName = sfd.FileName;
        string pName = System.IO.Path.GetDirectoryName(fName);
        if (System.IO.Directory.Exists(pName))
        {
            saveJpeg(fName + ".jpeg", mainCanvas, jpgQuality);
            canProp[cur_idx].getLastSavePath = fName;
            canProp[cur_idx].isSave = true;
            btnSaveAs.Enabled = true;
            incSave++;
        }
    }
}
else
{
    string fName = canProp[cur_idx].getLastSavePath;
    string pName = System.IO.Path.GetDirectoryName(fName);
    if (System.IO.Directory.Exists(pName))
    {
        saveJpeg(fName + ".jpeg", mainCanvas, jpgQuality);
    }
}
}
}

```

#### D. Membuka File

Prosedur yang dipakai untuk membuka *file* adalah sebagai berikut :

Prosedur *open file*, prosedur ini menampilkan jendela untuk membuka *file* jika memilih tombol *OK* maka akan menerima alamat dari *file* yang telah dipilih dan melakukan *streaming* dari *file* gambar tersebut dengan mengisi *parameter* alamat *file* dan posisi pada sudut atas dan kiri pada *drawImage* yang kemudian ditampilkan ke *kanvas*.

```

private void btnOpen_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
{
    if (ofd.ShowDialog() == System.Windows.Forms.DialogResult.OK)
    {
        if (System.IO.File.Exists(ofd.FileName))
        {
            string fName = ofd.FileName;
            string pName = System.IO.Path.GetDirectoryName(fName);
            pointPath = new
System.Drawing.Drawing2D.GraphicsPath();
            g.DrawImage(Image.FromFile(fName), new Point(0, 0));
            Layer.Image = (Image)mainCanvas;
        }
    }
}

```

### E. Kalibrasi

Prosedur yang dipakai untuk melakukan kalibrasi adalah sebagai berikut :

Prosedur *DoCalibration*, prosedur ini melakukan pengecekan pada *variable calibrateState* dan dilakukan sampai 4 kali dan pada setiap *case* memberikan nilai x dan y dengan nilai koordinat x dan y pada layar yang dipakai sebagai acuan untuk kalibrasi pada *case* yang kelima dilakukan penghitungan untuk menghubungkan antara bidang sesungguhnya di dunia ini dengan gambaran perspektif.

```

public void DoCalibration()
{
    if (frm == null)
        return;
    int x, y;
    switch (calibrateState)
    {
        case 1:

```

```

x = (int)(screenSize.Width * calibrationMargin);
y = (int)(screenSize.Height * calibrationMargin);
frm.ShowCalibration(x, y);
dstX[calibrateState - 1] = x;
dstY[calibrateState - 1] = y;
break;
case 2:
x = screenSize.Width - (int)(screenSize.Width *
calibrationMargin);
y = (int)(screenSize.Height * calibrationMargin);
frm.ShowCalibration(x, y);
dstX[calibrateState - 1] = x;
dstY[calibrateState - 1] = y;
break;
case 3:
x = (int)(screenSize.Width * calibrationMargin);
y = screenSize.Height - (int)(screenSize.Height *
calibrationMargin);
frm.ShowCalibration(x, y);
dstX[calibrateState - 1] = x;
dstY[calibrateState - 1] = y;
break;
case 4:
x = screenSize.Width - (int)(screenSize.Width *
calibrationMargin);
y = screenSize.Height - (int)(screenSize.Height *
calibrationMargin);
frm.ShowCalibration(x, y);
dstX[calibrateState - 1] = x;
dstY[calibrateState - 1] = y;
break;
case 5:
//compute warp

w.SetDestination(dstX[0], dstY[0], dstX[1], dstY[1],
dstX[2], dstY[2], dstX[3], dstY[3]);
w.SetSource(srcX[0], srcY[0], srcX[1], srcY[1],
srcX[2], srcY[2], srcX[3], srcY[3]);
w.ComputeWarp();
calibrateState = 0;

```

```

        cursorCtrl = true;

curFrm.BeginInvoke((System.Windows.Forms.MethodInvoker)delegate() {
    frm.Close(); });

        SaveCalibrationData();
        UpdateTrackingUtilization();
        break;
    default:
        break;
    }
}

```

#### 4.1.2 Instalasi Dan Kebutuhan Perangkat

##### A. Kebutuhan Perangkat Keras

Kebutuhan minimum perangkat keras yang diperlukan untuk menjalankan aplikasi ini adalah :

1. *Processor* Intel Core I5.
2. *Memory* 4 GB.
3. *Harddisk* 150 GB.
4. Monitor dengan resolusi minimal 1366 x 768.
5. *VGA Card* 1GB, *Mouse*, dan *keyboard*.
6. *Wiimote*.
7. *IR Pen* (Lampu *IR*, Baterai AA, Kabel, *Switch*, Spidol).
8. *Bluetooth*.

##### B. Kebutuhan Perangkat Lunak

Adapun kebutuhan perangkat lunak yang digunakan dalam pembuatan aplikasi *interactive whiteboard* adalah sebagai berikut:

1. Sistem operasi menggunakan Microsoft Windows 7 Ultimate.

2. Microsoft Visual Studio 2010 Ultimate.
3. Untuk perancangan sistem menggunakan Rational Rose.
4. Untuk perancangan desain input/output menggunakan GUI Design Studio.
5. Untuk dokumentasi menggunakan Microsoft Office Word 2010.
6. Untuk mengakses *wiimote* dibutuhkan *SDK WiimoteLib*.

### C. Instalasi Program dan Pengaturan Sistem

Pengembangan aplikasi *interactive whitebord* membutuhkan perangkat lunak yang telah terinstalasi, adapun tahapan-tahapan instalasi dan pengaturan (*setting*) sistem adalah sebagai berikut:

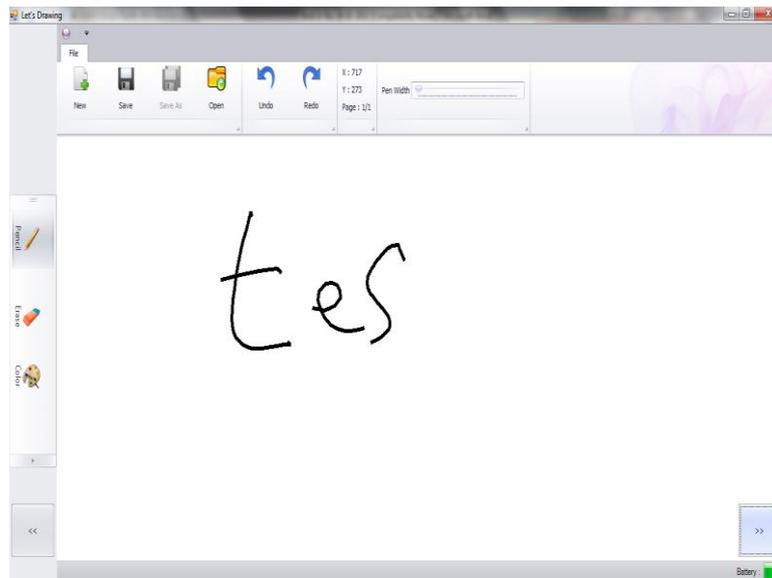
1. Install sistem operasi Microsoft Windows 7 Ultimate.
2. Install Microsoft Visual Studio 2010 Ultimate.
3. Install *wimoteLib* v1.7.

#### 4.2 Evaluasi

Untuk mengetahui apakah aplikasi *interactive whiteboard* berjalan sesuai dengan yang diinginkan maka dilakukan evaluasi untuk setiap proses yang terjadi. Evaluasi yang dilakukan antara lain :

##### 4.2.1 Menulis

Evaluasi yang dilakukan pada menulis ialah mencoba untuk menulis kata “tes”. Hasil dari evaluasi ini bisa dilihat pada Gambar 4.2.

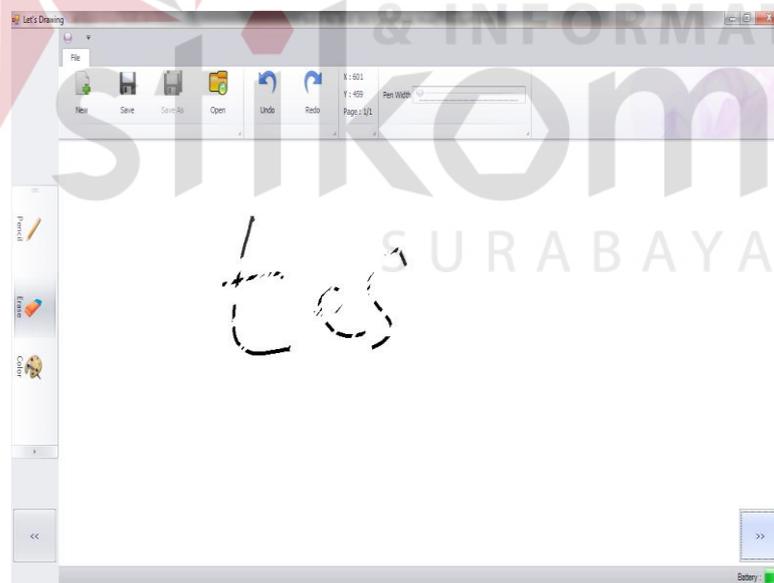


Gambar 4.2 Hasil Dari Evaluasi Menulis

#### 4.2.2 Menghapus

Evaluasi yang dilakukan pada menghapus ialah menghapus kata “tes” yang telah ditulis sebelumnya. Hasil dari evaluasi ini bisa dilihat pada Gambar

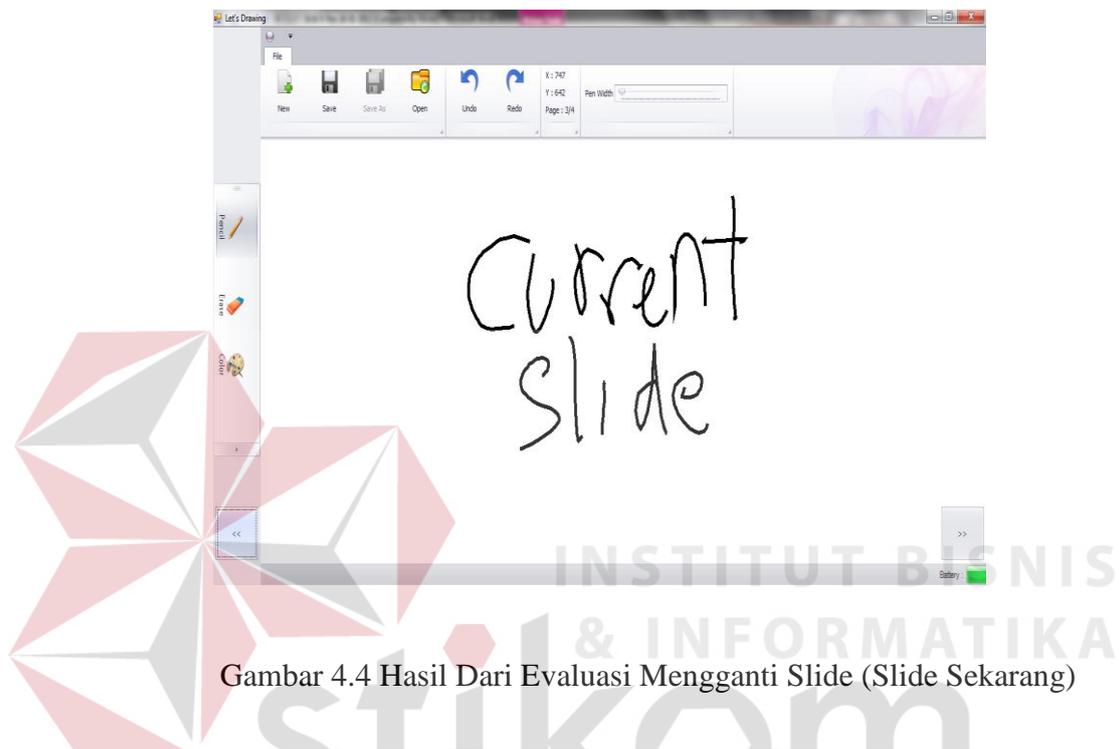
4.3.



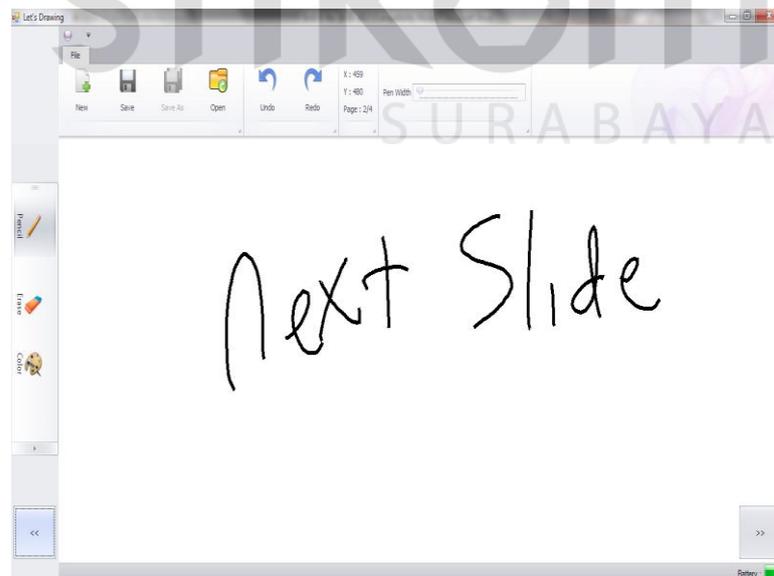
Gambar 4.3 Hasil Dari Evaluasi Menghapus

### 4.2.3 Mengganti Slide

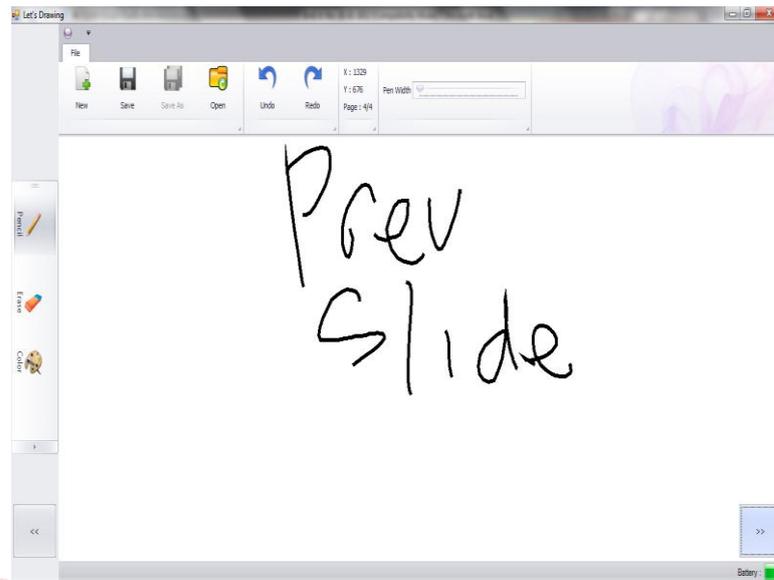
Evaluasi yang dilakukan pada mengganti slide ialah mengganti slide sekarang ke slide selanjutnya dan ke slide sebelumnya. Hasil dari evaluasi ini bisa dilihat pada Gambar 4.4, Gambar 4.5 dan Gambar 4.6.



Gambar 4.4 Hasil Dari Evaluasi Mengganti Slide (Slide Sekarang)



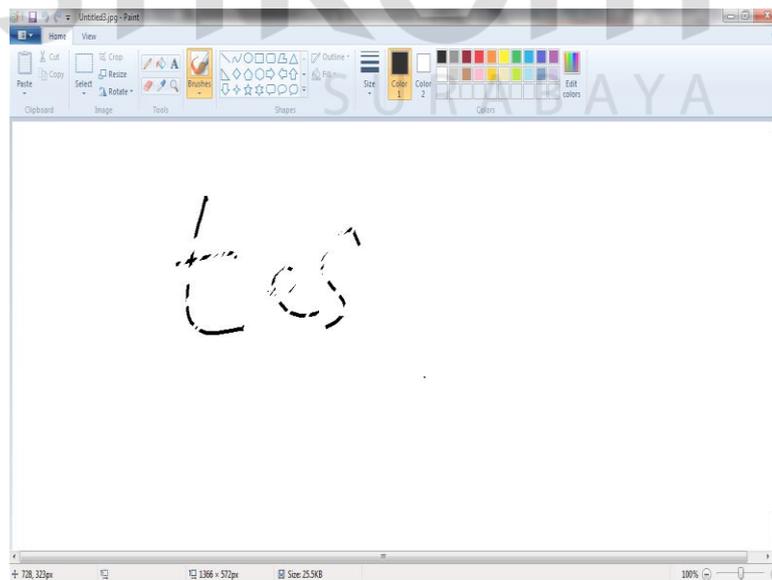
Gambar 4.5 Hasil Dari Evaluasi Mengganti Slide (Slide Berikutnya)



Gambar 4.6 Hasil Dari Evaluasi Mengganti Slide (Slide Sebelumnya)

#### 4.2.4 Menyimpan

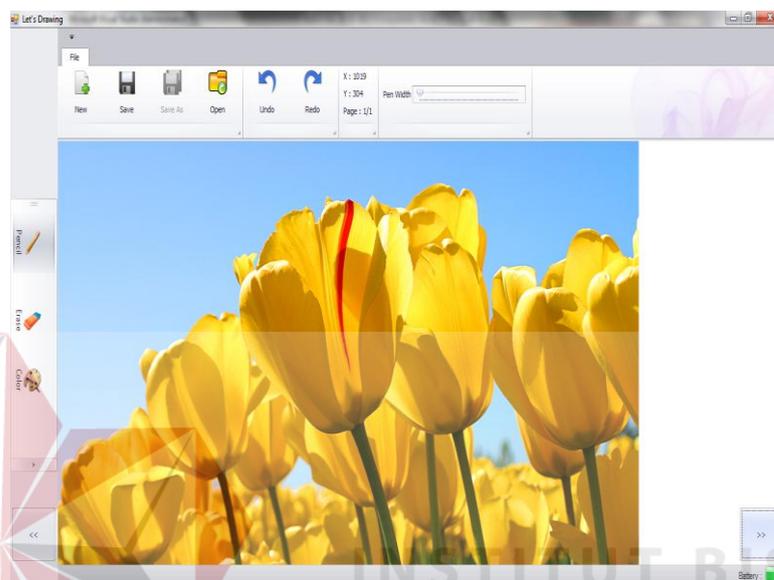
Evaluasi yang dilakukan pada menyimpan ialah menyimpan slide ke dalam bentuk *file* gambar dan menampilkan *file* tersebut di *Paint* (*Image Editor* bawaan windows). Hasil dari evaluasi ini bisa dilihat pada Gambar 4.7



Gambar 4.7 Hasil Dari Evaluasi Menyimpan

#### 4.2.5 Membuka *File*

Evaluasi yang dilakukan pada membuka *file* ialah mencoba untuk membuka *file* gambar dan menampilkan gambar tersebut di kanvas. Hasil dari evaluasi ini bisa dilihat pada Gambar 4.8



Gambar 4.8 Hasil Dari Evaluasi Membuka *File*

Setelah melakukan evaluasi yang telah dilakukan di atas, maka berikut hasil dari evaluasi tersebut yang dapat dilihat pada Table 4.1

**Tabel 4.1** Hasil Evaluasi Aplikasi Interactive Whiteboard.

Proses	Sesuai Dengan Yang Diharapkan
Menulis	Ya
Menghapus	Ya
Mengganti Slide	Ya
Menyimpan	Ya
Membuka File	Ya

### 4.3 Penjelasan Hasil Implementasi

Setelah kebutuhan sistem terpenuhi, langkah selanjutnya adalah mengimplementasikan rancangan sistem ke dalam sebuah aplikasi *interactive whiteboard*.

#### 4.3.1 Form Menu

Pada form menu terdapat pilihan-pilihan menu yang dapat dipilih oleh pengguna aplikasi. Tampilan *form* menu utama dapat dilihat pada Gambar 4.9.



Gambar 4.9 Form Menu Utama aplikasi *Interactive Whiteboard*.

Berikut ini adalah penjelasan dari masing-masing menu yang digunakan pada aplikasi *Interactive Whiteboard* dapat dilihat pada Tabel 4.1.

**Tabel 4.1** Penjelasan Menu Aplikasi Interactive Whiteboard.

Menu	Deskripsi
Calibrate	Berfungsi untuk melakukan pengaturan batas paling atas, kanan, bawah dan kiri untuk membatasi pergerakan <i>IR Pen</i> .
Drawing	Berfungsi untuk menulis, menghapus, menyimpan slide, membuka slide.
Exit	Berfungsi untuk keluar dari aplikasi <i>interactive whiteboard</i>

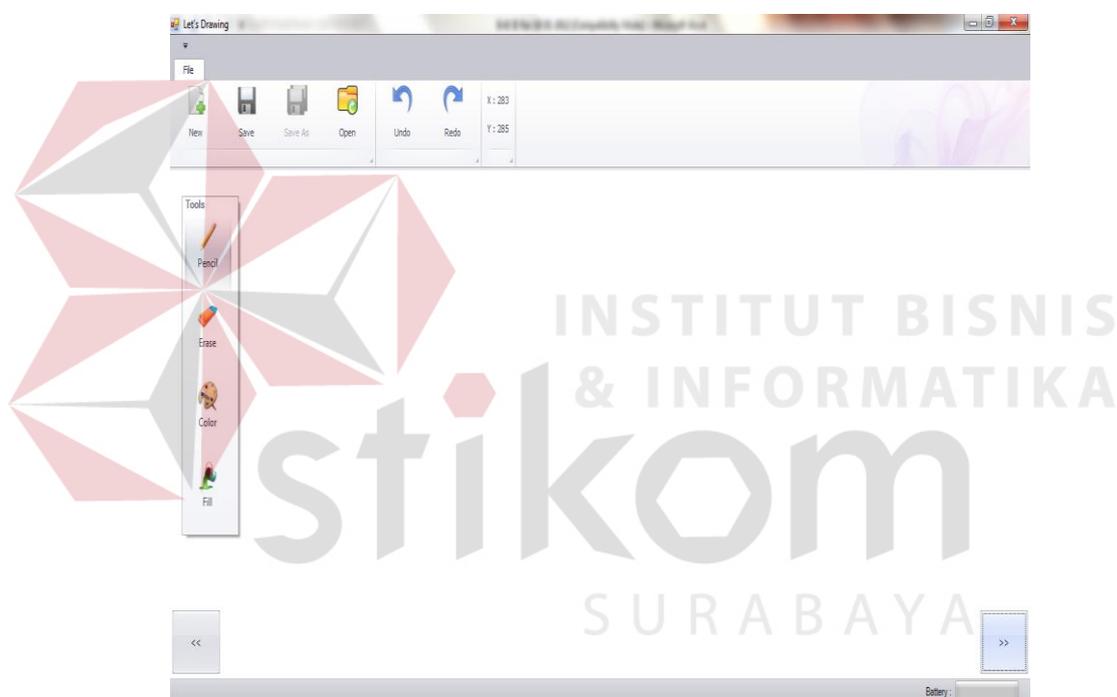
### 4.3.2 Form Kalibrasi

Pada form ini sistem sudah menentukan titik koordinat untuk batas atas, kanan, bawah dan kiri sesuai dengan besar layar monitor. Pengguna hanya mengarahkan *IR Pen* ke simbol pada target yang sudah disediakan oleh sistem dan menekan tombol *switch* menjadi *on* untuk melanjutkan ke posisi target berikutnya sampai 4 titik target. Calibration ini digunakan untuk membatasi pergerakan *IR Pen* pada canvas. Gambar 4.10.

Gambar 4.10 *Form* Kalibrasi.

### 4.3.3 Form *Drawing*

Pada *form* ini sistem menyediakan beberapa fitur yaitu save untuk menyimpan slide, open untuk membuka slide. Terdapat toolbox yang digunakan untuk menulis, menghapus dan mengganti warna sesuai keinginan. Pengguna juga bisa mengganti slide dengan cara menekan tombol prev untuk berpindah ke slide sebelumnya atau menekan tombol next untuk berpindah ke slide berikutnya. Gambar 4.11.



Gambar 4.11 *Form Drawing*.

## 4.4 Uji Coba dan Evaluasi

Uji coba dan evaluasi bertujuan untuk memastikan bahwa aplikasi telah dibuat dengan benar sesuai dengan kebutuhan atau tujuan yang diharapkan.

#### 4.4.1 Uji Coba Fungsionalitas Aplikasi

Uji coba fungsionalitas aplikasi dilakukan untuk mengetahui apakah fungsi-fungsi yang ada pada aplikasi berjalan dengan baik atau tidak. Adapun fungsi-fungsi yang akan diujikan adalah sebagai berikut :

##### A. Proses Menulis

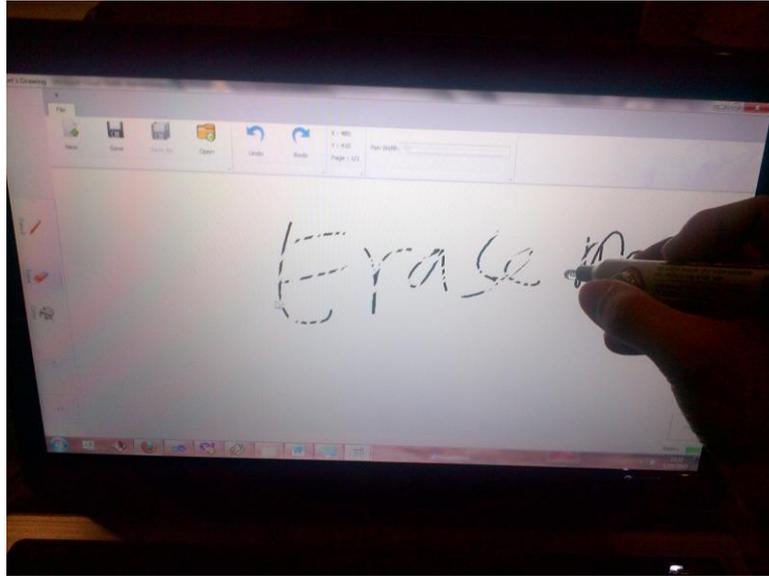
Proses ini dilakukan untuk menguji apakah program bisa menampilkan tulisan yang akan ditulis oleh pengguna melalui *IR Pen* yang digerakkan di atas layar yang ditangkap oleh *wiimote* yang kemudian dikirim ke sistem. Gambar 4.12.



Gambar 4.12 Menulis dengan *IR Pen*.

##### B. Proses Menghapus

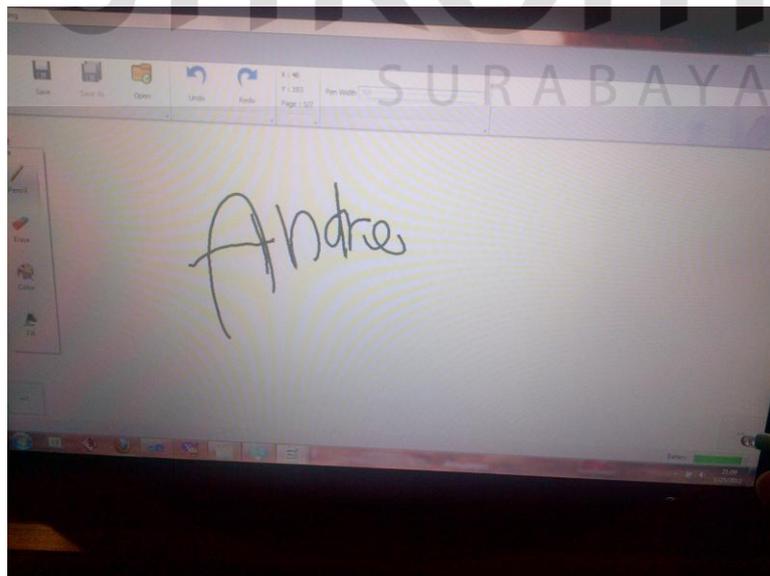
Proses ini dilakukan untuk menguji apakah program bisa menghapus tulisan yang telah ditulis oleh pengguna melalui *IR Pen* yang digerakkan di atas layar yang ditangkap oleh *wiimote* yang kemudian dikirim ke sistem. Gambar 4.13.



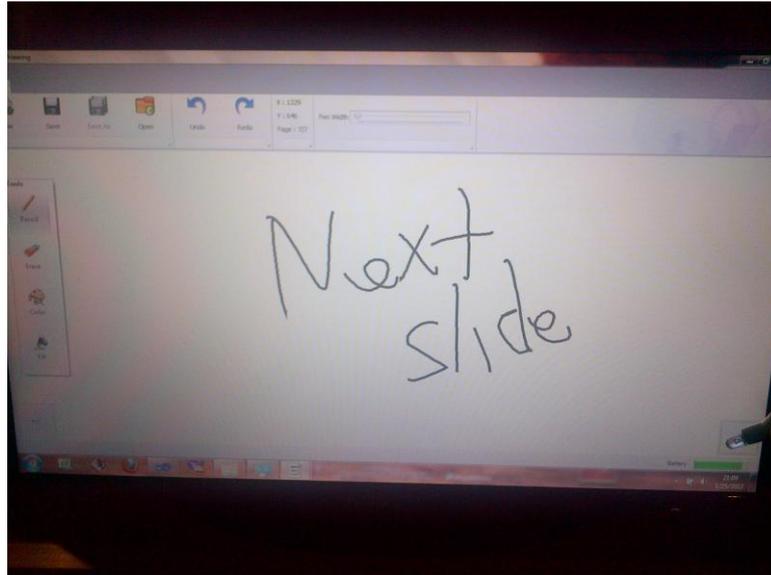
Gambar 4.13 Menghapus dengan *IR Pen*.

### C. Proses Mengganti *Slide*

Proses ini dilakukan untuk menguji apakah program bisa menampilkan slide yang dipilih oleh pengguna dengan menggunakan *IR Pen* dengan menekan tombol *next* atau *prev*. Gambar 4.14 dan 4.15



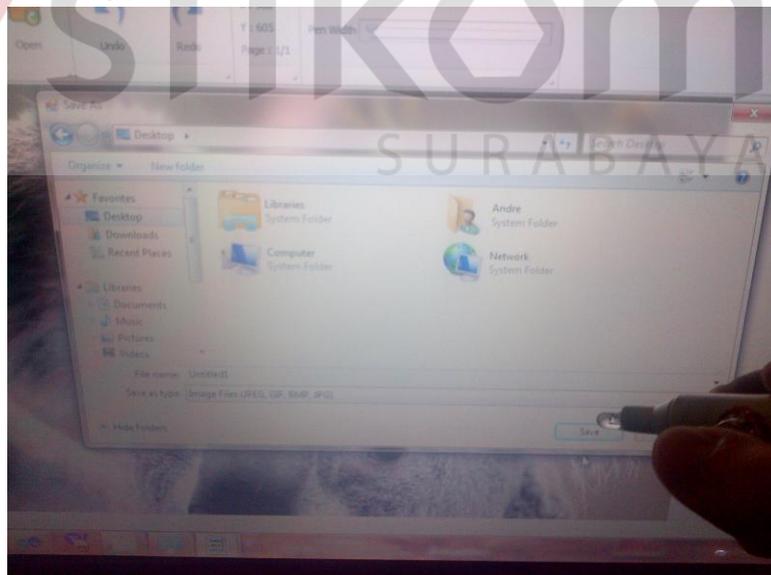
Gambar 4.14 Slide sekarang.



Gambar 4.15 Slide berikutnya.

#### D. Proses Menyimpan Slide

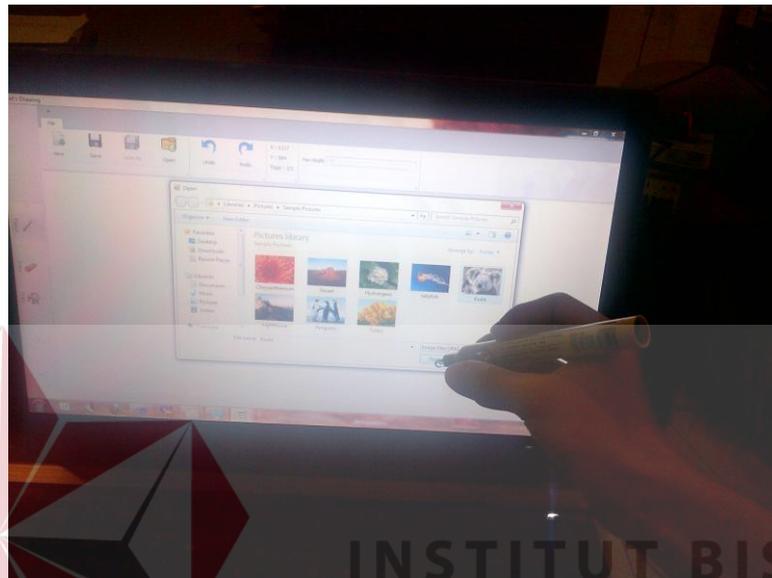
Proses ini dilakukan untuk menguji apakah program bisa menyimpan slide yang telah disimpan ke dalam bentuk *file* gambar ke media penyimpanan. Gambar 4.16



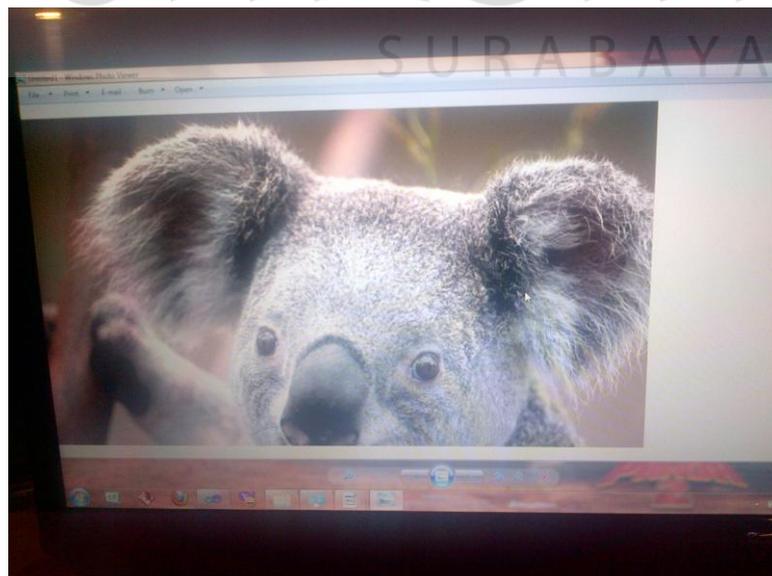
Gambar 4.16 Menyimpan Slide.

### E. Proses Membuka *Slide*

Proses ini dilakukan untuk menguji apakah program bisa menampilkan *file gambar* yang telah dipilih oleh pengguna dari media penyimpanan. Gambar 4.17 dan 4.18



Gambar 4.17 Membuka *Slide*.



Gambar 4.18 Hasil dari slide yang telah disimpan.

#### 4.4.2 Evaluasi

Analisa hasil uji coba dilakukan untuk mengetahui seluruh fungsi aplikasi agar bisa berjalan di pc yang digunakan oleh pengguna. Berdasarkan pada hasil uji coba yang telah dilakukan pada bagian 4.4.1 dapat disimpulkan bahwa fungsionalitas aplikasi dapat berjalan dengan baik pada PC yang digunakan.

