

BAB III

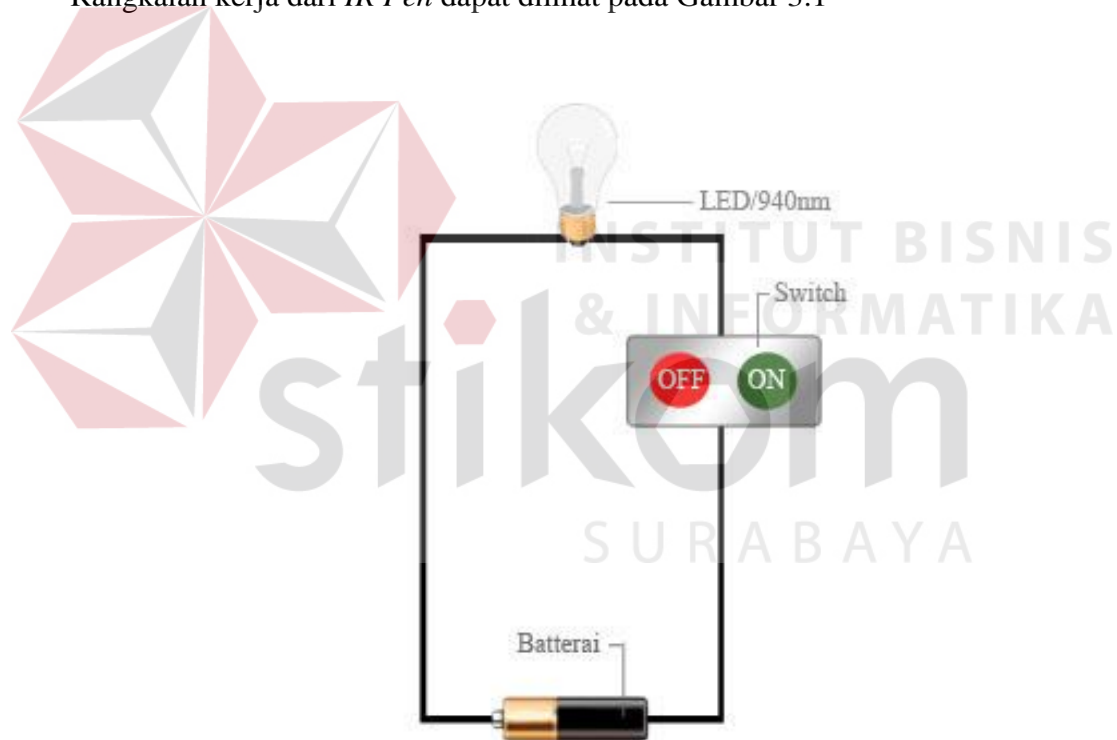
ANALISA DAN PERANCANGAN SISTEM

3.1 Analisa Permasalahan Sistem

Seperti yang sudah kita ketahui tentang *interactive whiteboard* yaitu sebuah panel layar sentuh yang berfungsi sebagai papan tulis atau sebagai proyektor komputer yang dapat mengendalikan obyek dalam komputer dengan menyentuh permukaan panel tanpa menggunakan mouse atau keyboard ini sebenarnya bukan merupakan teknologi baru, teknologi ini sebenarnya sudah lama ada dan berada di sekitar kita, seperti *smartboard* dan *tablet*. Namun pada kenyataannya media tersebut memiliki beberapa keterbatasan yaitu harga yang tinggi untuk *smartboard* sedangkan untuk *tablet* hanya memiliki ruang atau ukuran yang kecil untuk dijadikan sebuah *whiteboard*.

Karena keterbatasan media tersebut maka diperlukan suatu perangkat yang kiranya bisa sebagai alternatif untuk mengatasi masalah tersebut. Adapun perangkat yang bisa dimanfaatkan untuk mengatasi masalah tersebut adalah perangkat kontrol yang dikeluarkan oleh perusahaan Nintendo yang biasa dipakai sebagai alat kontrol dalam permainan game Nintendo Wii. Perangkat tersebut bernama *wiimote*, *wiimote* adalah suatu teknologi yang memanfaatkan sensor gerak yang memungkinkan pengguna dapat berinteraksi dengan obyek - obyek yang ada pada layar melalui pendeteksi gerakan dengan menggunakan teknologi accelerometer. Adapun cara kerjanya adalah *wiimote* harus ditempatkan di depan layar proyektor dengan jarak dan sudut tertentu untuk bisa menjangkau semua daerah layar, kemudian mengkoneksikan *wiimote* ke komputer dengan media *bluetooth* sebagai sarana untuk mengirimkan data dari *wiimote* ke komputer dan

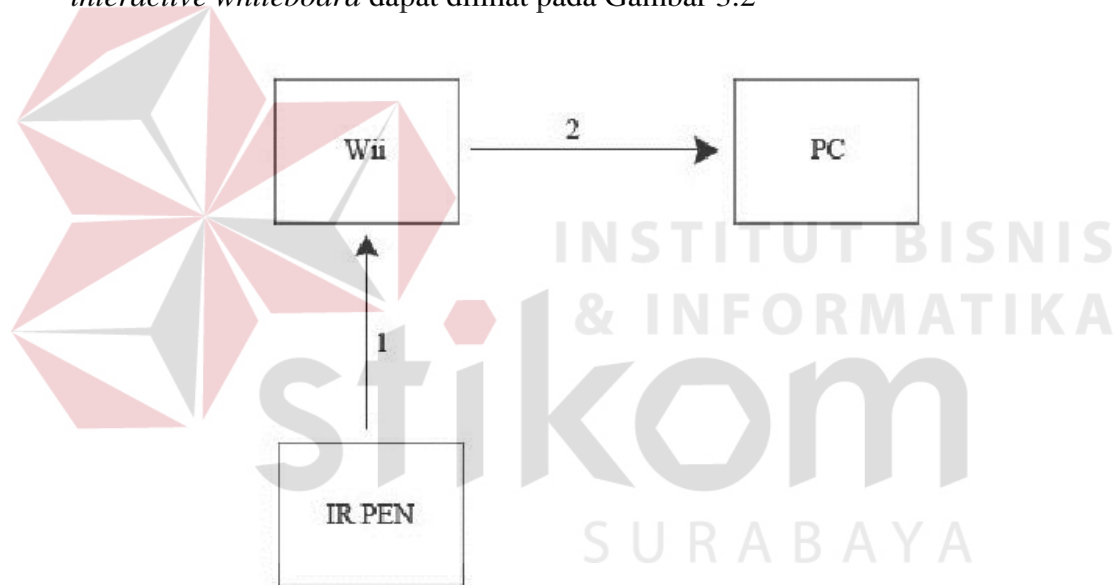
menggunakan *IR Pen* sebagai sensor pada *wiimote*. Adapun cara untuk membuat *IR Pen* diperlukan Lampu *IR*, Baterai AA, Kabel, *Switch*, Spidol dan untuk merangkainya digunakan rangkaian listrik seri. Jika *switch IR Pen* ditekan maka sensor *infrared* akan menyala dan apabila sensor tersebut ada pada jangkauan kamera *wiimote*, maka *wiimote* akan mendeteksi keberadaan dari sensor tersebut, apabila *wiimote* mendeteksi satu *sensor infrared* maka sistem akan memicu kejadian menekan tombol kiri *mouse* kalau *wiimote* mendeteksi dua sensor *infrared* maka sistem akan memicu kejadian menekan tombol kanan *mouse*. Rangkaian kerja dari *IR Pen* dapat dilihat pada Gambar 3.1



Gambar 3.1 Rangkaian Kerja *IR Pen*

Perangkat lunak yang akan dibuat nantinya akan berfungsi sebagai pengganti papan tulis yang bisa digunakan untuk menulis, menghapus serta juga bisa menyimpan dan menampilkan gambar dan mengganti slide sesuai keinginan. Perangkat lunak ini dapat dijalankan pada lingkungan sistem operasi Microsoft

Windows XP/Vista/7 dan dibangun dengan menggunakan bahasa pemrograman Microsoft Visual C#. Pengguna berinteraksi dengan perangkat lunak dengan menggunakan *IR Pen* sebagai media untuk menulis di *interactive whiteboard*. Pengguna aplikasi ini adalah guru dan siswa yang sedang melangsungkan proses belajar mengajar di dalam kelas. Kebutuhan perangkat keras yang dibutuhkan oleh perangkat lunak ini adalah : PC, *wiimote*, *IR Pen*, *Bluetooth*. Sedangkan library yang dibutuhkan adalah : *WiimoteLib* v1.7 untuk melakukan koneksi antara PC dengan *wiimote* melalui *script* Microsoft Visual C#. *Block diagram* untuk aplikasi *interactive whiteboard* dapat dilihat pada Gambar 3.2



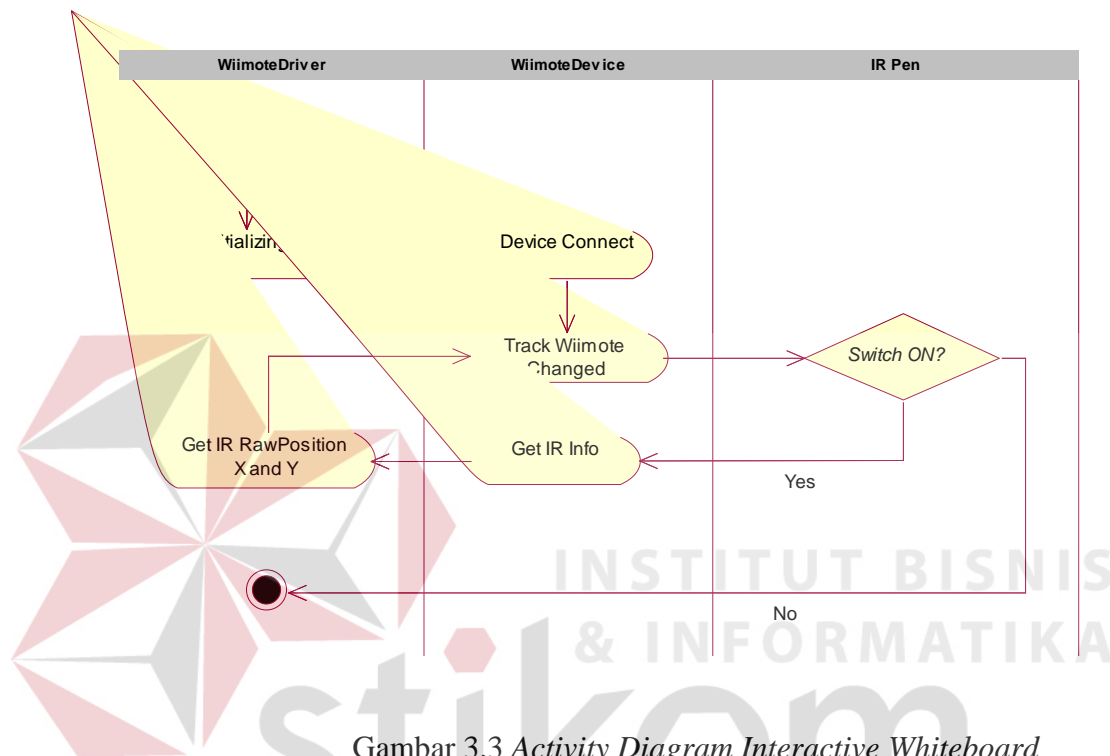
Gambar 3.2 *Block Diagram Interactive Whiteboard*

Berikut adalah cara kerja yang diterapkan pada aplikasi *Interactive Whiteboard* :

1. *Wiimote* diletakkan pada jarak tertentu dari layar PC agar kamera pada *wiimote* dapat menjangkau semua daerah layar kemudian dihubungkan ke PC melalui media *bluetooth*.

2. *IR Pen* berfungsi sebagai sensor. Jika sensor tersebut dinyalakan pada jangkauan kamera *wiimote*, maka kamera akan menangkap titik koordinat dari sensor tersebut.

Berikut adalah gambar *activity diagram* dari *Interactive Whiteboard*.



Gambar 3.3 Activity Diagram Interactive Whiteboard

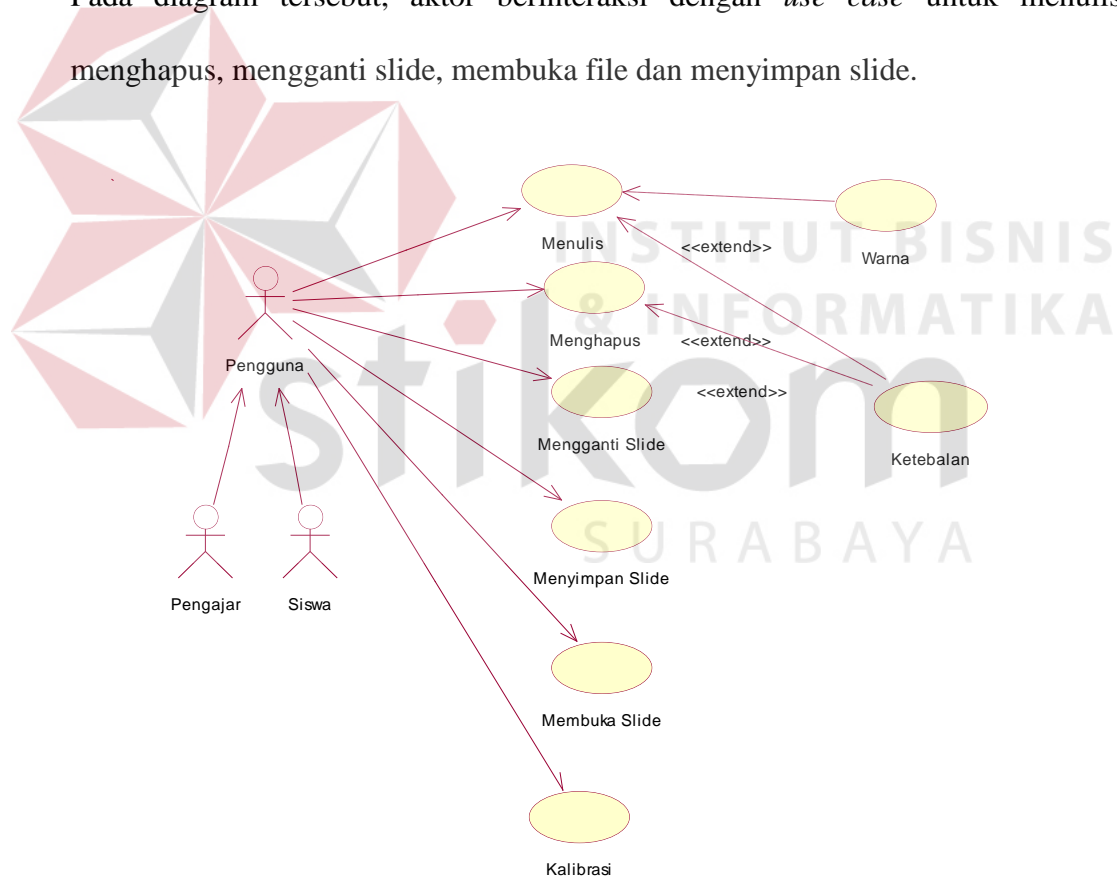
Dapat dilihat dari gambar diatas bahwa pertama – tama *wiimote driver* melakukan koneksi ke *wiimote* setelah itu *wiimote device* melakukan pelacakan terhadap perubahan yang terjadi pada *wiimote device* dengan menggunakan event *wiimotechanged* yang bisa diakses dalam *library*, jika *IR Pen* dinyalakan maka event tersebut akan mendeteksi informasi dari perubahan yang terjadi pada *IR Pen*, setelah itu *wiimote device* mengirimkan informasi berupa titik koordinat X dan Y tersebut ke *wiimote driver* dan dapat diakses melalui *function* yang bernama `IRState.IRSensors[Index].RawPosition.X` dan `IRState.IRSensors[Index].RawPosition.Y`.

Untuk mendapatkan hasil yang maksimal maka posisi kamera *wiimote* harus menghadap di depan layar dengan jarak minimal 0.5 meter dan maksimal 1.5 meter dari layar dengan posisi horizontal dan posisi *IR Pen* harus mendekati layar dengan jarak diantara 0.5 – 2cm dengan sudut horisontal 20 – 45 derajat dan sudut vertical 0 – 45 derajat.

3.2 Use Case Diagram Aplikasi *Interactive Whiteboard*

Pada Gambar 3.4 menunjukkan interaksi antara aktor dengan sistem.

Pada diagram tersebut, aktor berinteraksi dengan *use case* untuk menulis, menghapus, mengganti slide, membuka file dan menyimpan slide.



Gambar 3.4 Use Case Diagram Aplikasi *Interactive Whiteboard*

3.3 Flow Of Event

Untuk memahami lebih detail dari setiap fitur yang telah digambarkan pada use case diagram maka dijelaskan dengan *flow of event* (aliran kejadian). *Flow of event* yang disebut juga skenario, adalah langkah detail yang terjadi dalam sebuah use case dan dinyatakan dengan kalimat.

3.3.1 Flow Of Event dari use case Menulis

A. Precondition

IR Pen dalam kondisi *ON*

B. Main Flow

1. Sistem menampilkan toolbox dan canvas.
2. Wiimote menangkap pergerakan *IR Pen* yang berupa titik koordinat *x* dan *y* lalu mengirimkannya ke komputer melalui media *bluetooth* kemudian menggerakkan kursor sesuai titik koordinat tersebut.
3. Pengguna memilih tool pensil pada toolbox
4. Pengguna menggerakkan *IR Pen* pada canvas dan menekan *switch* untuk menggambar obyek pada canvas dengan cara memberi warna *default* hitam pada setiap titik koordinat *x* dan *y* yang berasal dari pergerakan *IR Pen*.

C. Sub Flow

Tidak ada

D. Alternatif Flow

Pergerakan *IR Pen* untuk menggambar bisa dideteksi jika masih dalam batas garis yang sudah ditentukan.

3.3.2 Flow Of Event dari use case Menghapus

A. Precondition

IR Pen dalam kondisi *ON*

B. Main Flow

1. Sistem menampilkan toolbox dan canvas.
2. Wiimote menangkap pergerakan *IR Pen* yang berupa titik koordinat *x* dan *y* lalu mengirimkannya ke komputer melalui media *bluetooth* kemudian menggerakkan kursor sesuai titik koordinat tersebut.
3. Pengguna memilih tool penghapus pada toolbox
4. Pengguna menggerakkan *IR Pen* pada kanvas dan menekan *switch* untuk menghapus obyek pada kanvas dengan cara memberi warna *default* putih atau sesuai dengan warna background kanvas pada setiap titik koordinat *x* dan *y* yang berasal dari pergerakan *IR Pen*.

C. Sub Flow

Tidak ada

D. Alternatif Flow

Pergerakan *IR Pen* untuk menghapus bisa dideteksi jika masih dalam batas garis yang sudah ditentukan.

3.3.3 Flow Of Event dari use case Mengganti Slide

A. Precondition

IR Pen dalam kondisi *ON*

B. Main Flow

1. Sistem menampilkan tombol sebelumnya dan sesudahnya.
2. Wiimote menangkap pergerakan IR Pen yang berupa titik koordinat x dan y lalu mengirimkannya ke komputer melalui media *bluetooth* kemudian menggerakkan kursor sesuai titik koordinat tersebut.
3. Pengguna menekan tombol sebelumnya untuk berpindah ke slide sebelumnya atau menekan tombol sesudahnya untuk berpindah ke slide sesudahnya. Pengguna dalam berpindah slide dengan cara mengarahkan posisi IR Pen ke tombol tersebut dengan menekan *switch* lalu melepas *switch* tersebut sehingga akan memicu kejadian klik.

C. Sub Flow

Tidak ada

D. Alternatif Flow

Tidak ada

3.3.4 Flow Of Event dari use case Menyimpan Slide

A. Precondition

IR Pen dalam kondisi *ON*

B. Main Flow

1. Sistem menampilkan menu file.
2. Wiimote menangkap pergerakan IR Pen yang berupa titik koordinat x dan y lalu mengirimkannya ke komputer melalui

media *bluetooth* kemudian menggerakkan kursor sesuai titik koordinat tersebut.

3. Pengguna menekan tombol save dan sistem menampilkan jendela dialog untuk meminta konfirmasi kepada pengguna sebelum menyimpan slide ke dalam bentuk file gambar.

C. Sub Flow

Tidak ada

D. Alternatif Flow

Jika memilih tombol cancel maka slide tidak akan tersimpan dan akan kembali ke kanvas

3.3.5 Flow Of Event dari use case Membuka Slide

A. Precondition

IR Pen dalam kondisi *ON*

B. Main Flow

1. Sistem menampilkan menu file.
2. Wiimote menangkap pergerakan *IR Pen* yang berupa titik koordinat x dan y lalu mengirimkannya ke komputer melalui media *bluetooth* kemudian menggerakkan kursor sesuai titik koordinat tersebut.
3. Pengguna menekan tombol open dan sistem menampilkan jendela dialog untuk memilih file gambar yang ingin ditampilkan di kanvas.

C. Sub Flow

Tidak ada

D. Alternatif Flow

Jika memilih tombol cancel maka gambar tidak akan ditampilkan dan akan kembali ke canvas.

3.3.6 Flow Of Event dari use case Kalibrasi

A. Precondition

IR Pen dalam kondisi *ON*

B. Main Flow

1. Sistem menampilkan target pertama pada pojok kiri atas.
2. Wiimote menangkap pergerakan *IR Pen* yang berupa titik koordinat *x* dan *y* lalu mengirimkannya ke komputer melalui media *bluetooth* kemudian menggerakkan kursor sesuai titik koordinat tersebut.
3. Pengguna menekan *switch* pada posisi target yang ditampilkan secara otomatis oleh sistem. Hal ini dilakukan sampai 4 titik target.

C. Sub Flow

Tidak ada

D. Alternatif Flow

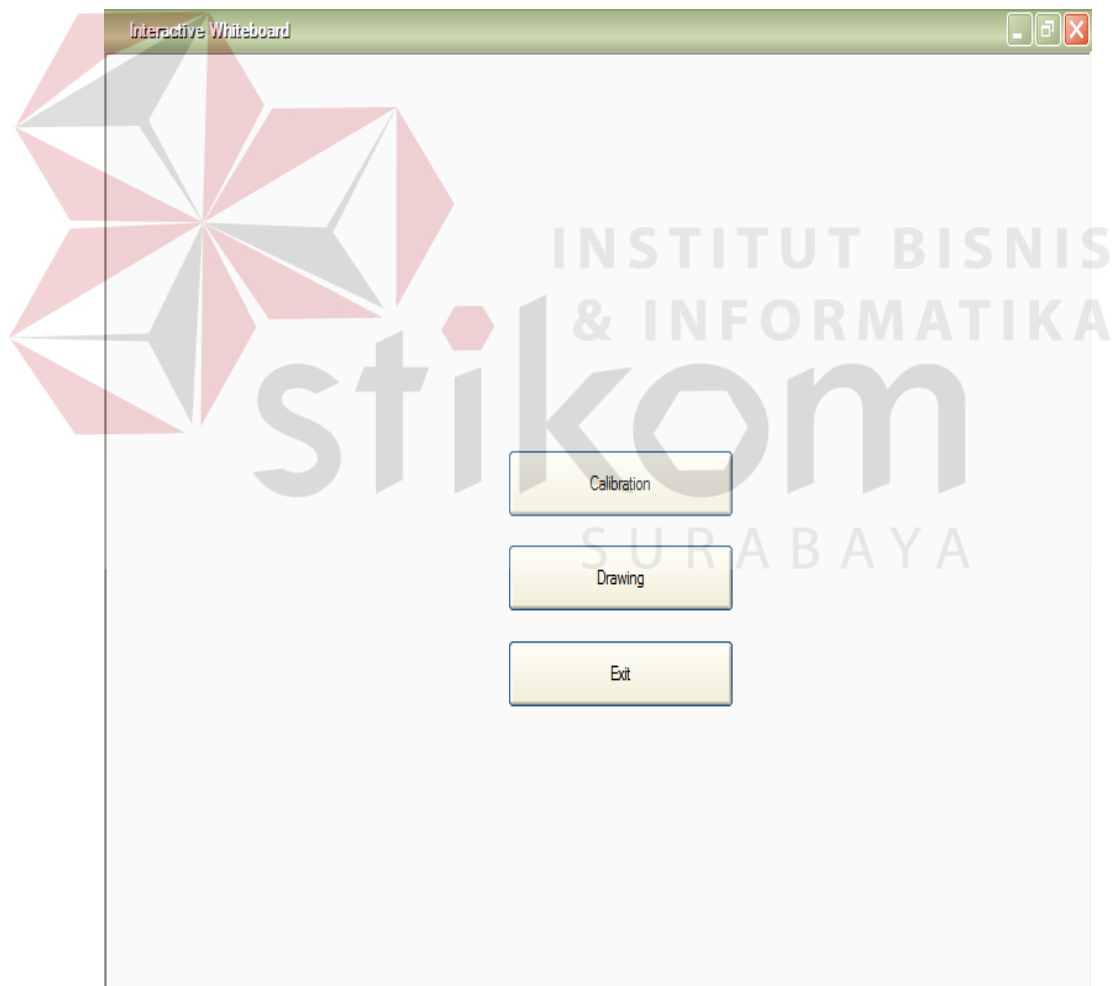
Tidak ada

3.4 User Interface

Pembuatan tampilan sangat diperlukan agar pengguna dapat berinteraksi dengan sistem sehingga dibutuhkan perancangan secara detil mengenai tampilan aplikasi berdasarkan informasi yang ditampilkan pada monitor atau proyektor.

3.4.1 User Interface Form Menu Utama

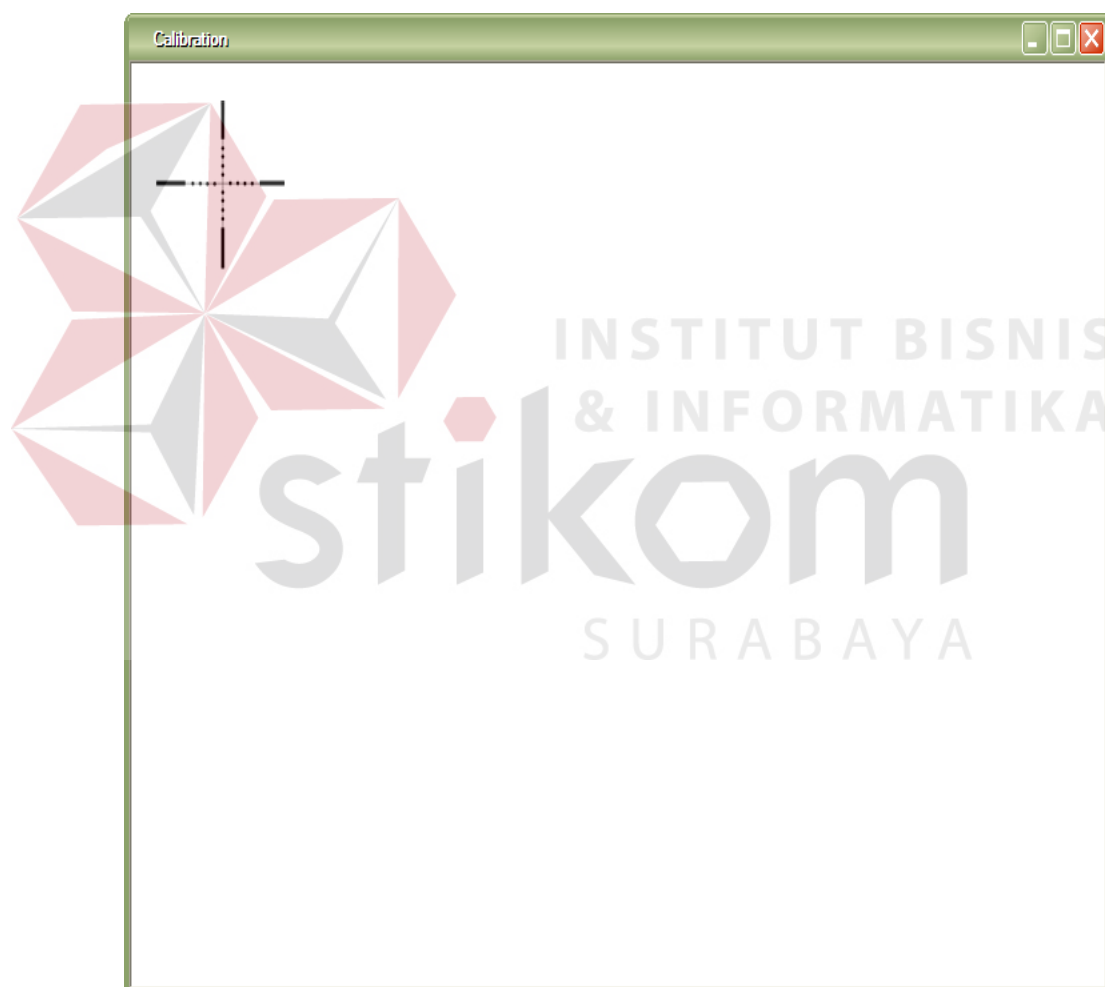
Fitur yang terdapat pada menu utama terdiri dari Calibration, Drawing dan Exit. Menu Calibration digunakan untuk mengatur batas pergerakan yang akan ditangkap oleh wiimote untuk sisi atas, kanan, bawah dan kiri. Menu Drawing berfungsi sebagai canvas yang digunakan untuk menulis, menghapus, mengganti slide, menyimpan slide dan membuka slide. Menu Exit digunakan untuk keluar dari aplikasi. Perancangan tampilan untuk menu utama dapat dilihat pada gambar 3.5.



Gambar 3.5 User Interface Menu Utama

3.4.2 User Interface Form *Calibration*

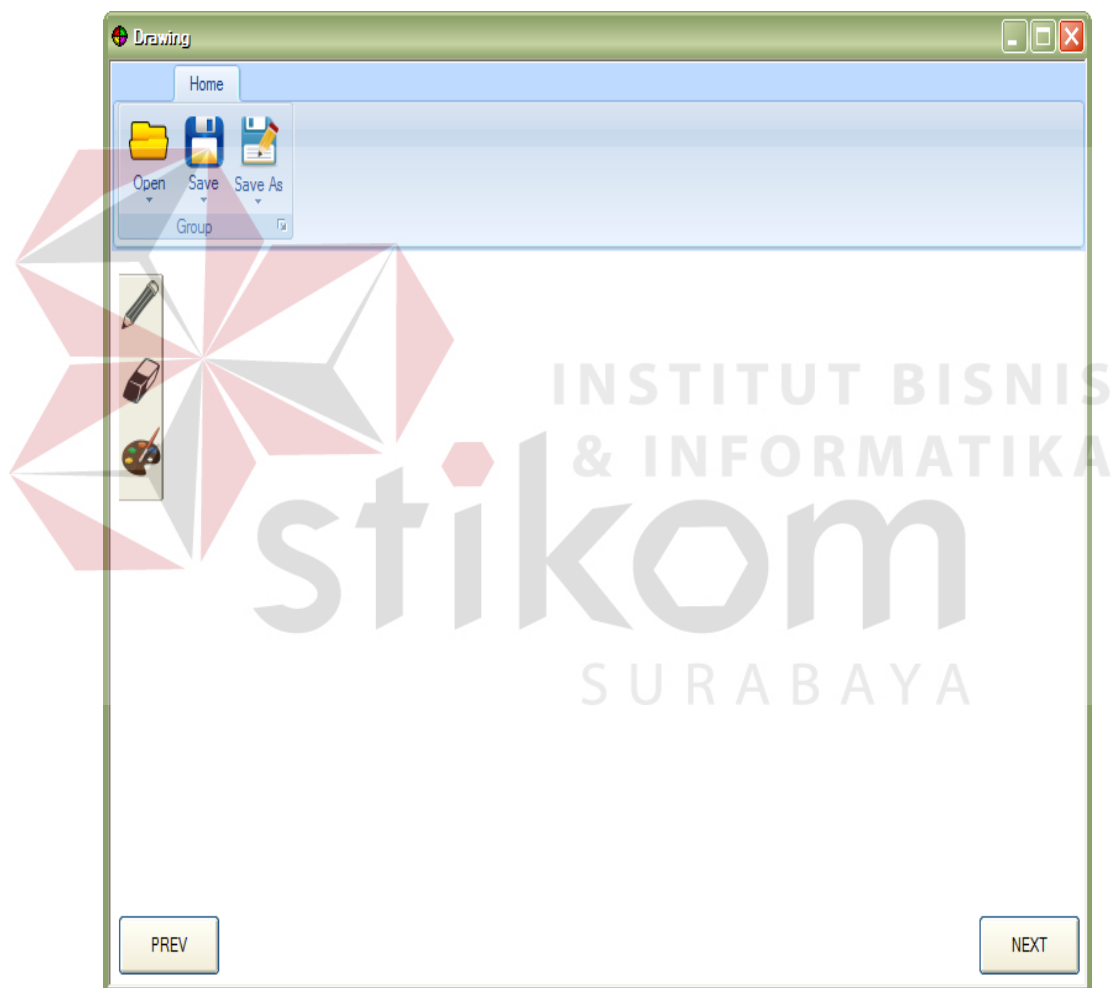
Pada form ini sistem sudah menentukan titik koordinat untuk batas atas, kanan, bawah dan kiri sesuai dengan besar layar monitor. Pengguna hanya mengarahkan *IR Pen* ke simbol pada target yang sudah disediakan oleh sistem dan menekan tombol *switch* menjadi *on* untuk melanjutkan ke posisi target berikutnya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



Gambar 3.6 User Interface Form Calibration

3.4.3 User Interface 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 3.7 User Interface Form Drawing

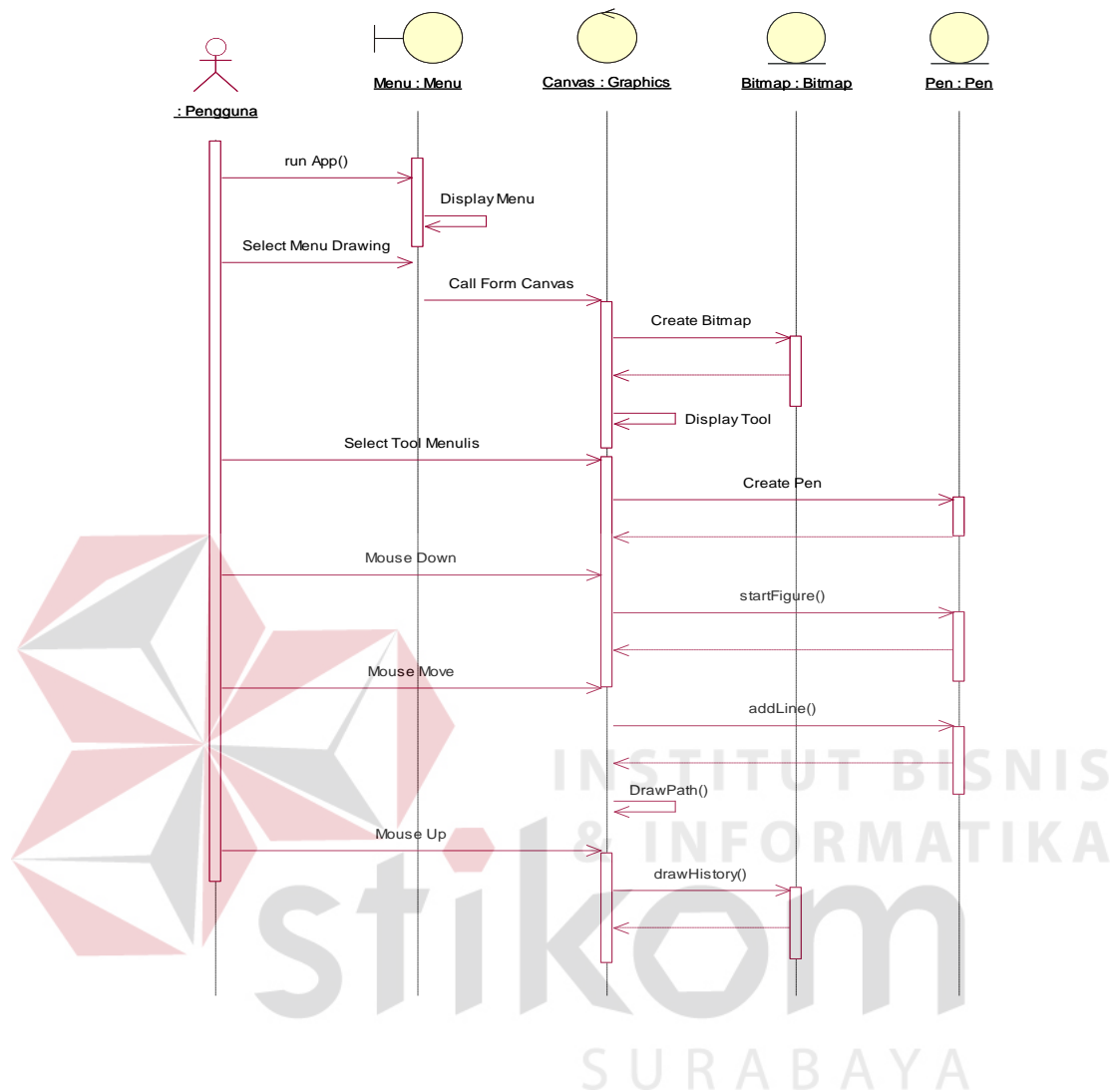
3.5 Sequence Diagram

Sequence diagram digunakan untuk menggambarkan jalannya suatu proses yang melibatkan *object* atau *instance* dari suatu class dalam aplikasi *interactive whiteboard* pada masing – masing *use case* yang terdapat pada use case diagram. *Object* yang dipakai dalam membuat *Use Case Diagram Aplikasi Interactive Whiteboard* adalah :

1. *Bitmap*, digunakan untuk tempat menulis yang memiliki atribut *width* untuk mengatur lebar *bitmap* dan *height* untuk mengatur panjang *bitmap*.
2. *Pen*, memiliki atribut *color* dan *width* yang digunakan untuk mengatur warna dan ketebalan pensil.
3. *GraphicsPath*, memiliki method *startFigure* yang digunakan untuk memulai figur baru dan method *addLine* yang digunakan untuk memberi tanda pada setiap titik koordinat yang dilewati oleh kursor.
4. *Graphics*, memiliki method *FromImage* yang digunakan untuk mengatur sumber gambar sebagai acuan dalam menulis dan method *DrawPath* yang digunakan untuk memberi warna pada setiap titik koordinat dari *Pen* yang telah dibuat sebelumnya.
5. *IO*, yang memiliki atribut *filename* untuk mendapatkan nama file yang dipilih oleh pengguna, *pathname* untuk mendapatkan nama *path* yang dipilih oleh pengguna, *openfiledialog* untuk menampilkan jendela untuk membuka slide dan *savefiledialog* untuk menampilkan jendela untuk menyimpan slide.

3.5.1 Sequence Diagram Menulis

Sequence diagram dari *use case* Menulis dapat digambarkan seperti pada gambar 3.8



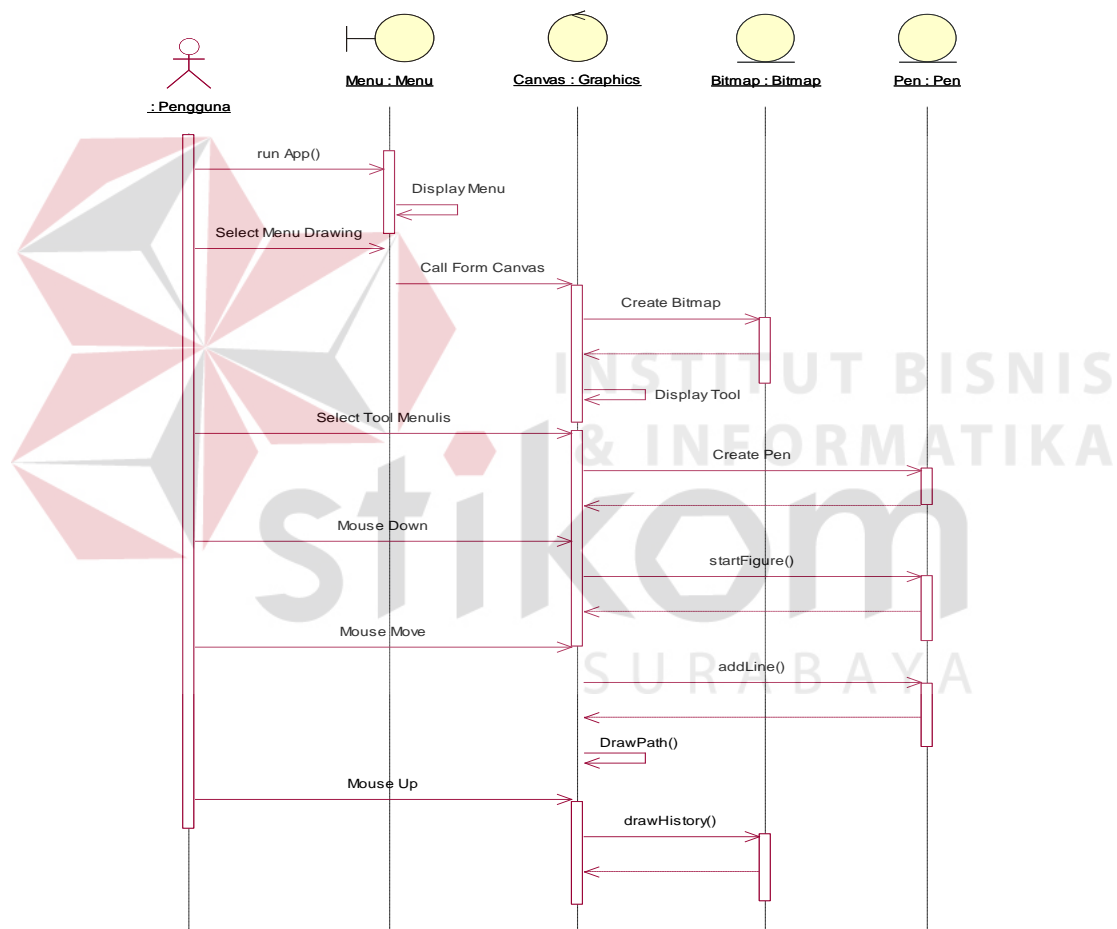
Gambar 3.8 *Sequence Diagram* Menulis

Pengguna menjalankan aplikasi lalu tampil form menu utama kemudian pengguna memilih menu drawing yang akan memanggil form canvas yang dibuat dari bitmap. Di dalam form canvas terdapat tool yang memiliki beberapa alat yang digunakan untuk menulis, menghapus dan memilih warna, disini pengguna memilih tool pensil untuk bisa menulis di canvas. Cara kerja tool pensil ialah pengguna menahan tombol kiri mouse pada canvas untuk memulai figur baru untuk menggambar *line* setelah itu pengguna menggerakkan mouse di atas canvas

untuk melakukan `addLine` untuk setiap pergerakan mouse dan diakhiri dengan melepas tombol kiri mouse dan membuat riwayat menulis.

3.5.2 Sequence Diagram Menghapus

Sequence diagram dari *use case* Menghapus dapat digambarkan seperti pada gambar 3.9



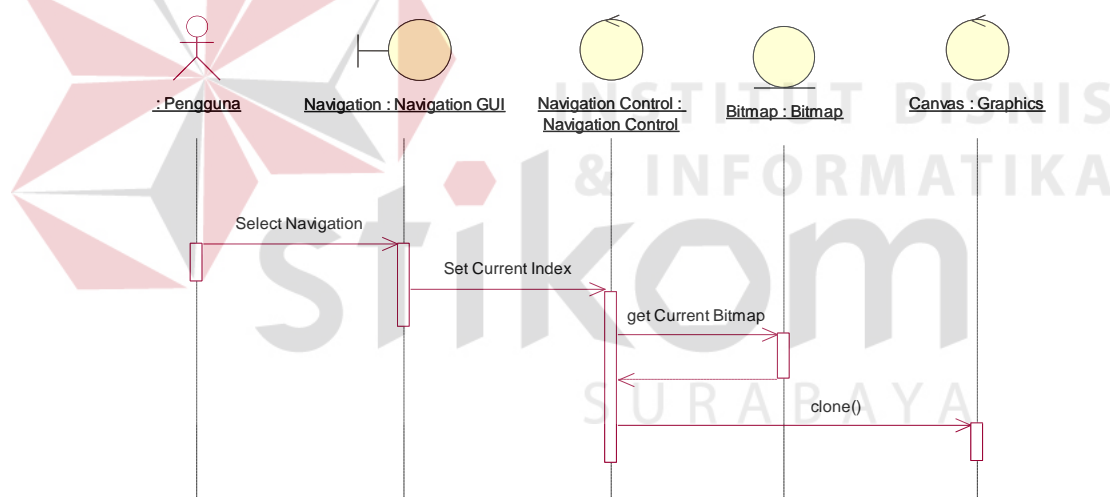
Gambar 3.9 *Sequence Diagram* Menghapus

Pengguna menjalankan aplikasi lalu tampil form menu utama kemudian pengguna memilih menu drawing yang akan memanggil form canvas yang dibuat dari bitmap. Di dalam form canvas terdapat tool yang memiliki beberapa alat yang

digunakan untuk menulis, menghapus dan memilih warna, disini pengguna memilih tool pensil untuk bisa menulis di canvas. Cara kerja tool pensil ialah pengguna menahan tombol kiri mouse pada canvas untuk memulai figur baru untuk menggambar *line* setelah itu pengguna menggerakkan mouse di atas canvas untuk melakukan *addLine* untuk setiap pergerakan mouse dan diakhiri dengan melepas tombol kiri mouse dan membuat riwayat menulis.

3.5.3 Sequence Diagram Mengganti Slide

Sequence diagram dari *use case* Mengganti Slide dapat digambarkan seperti pada gambar 3.10

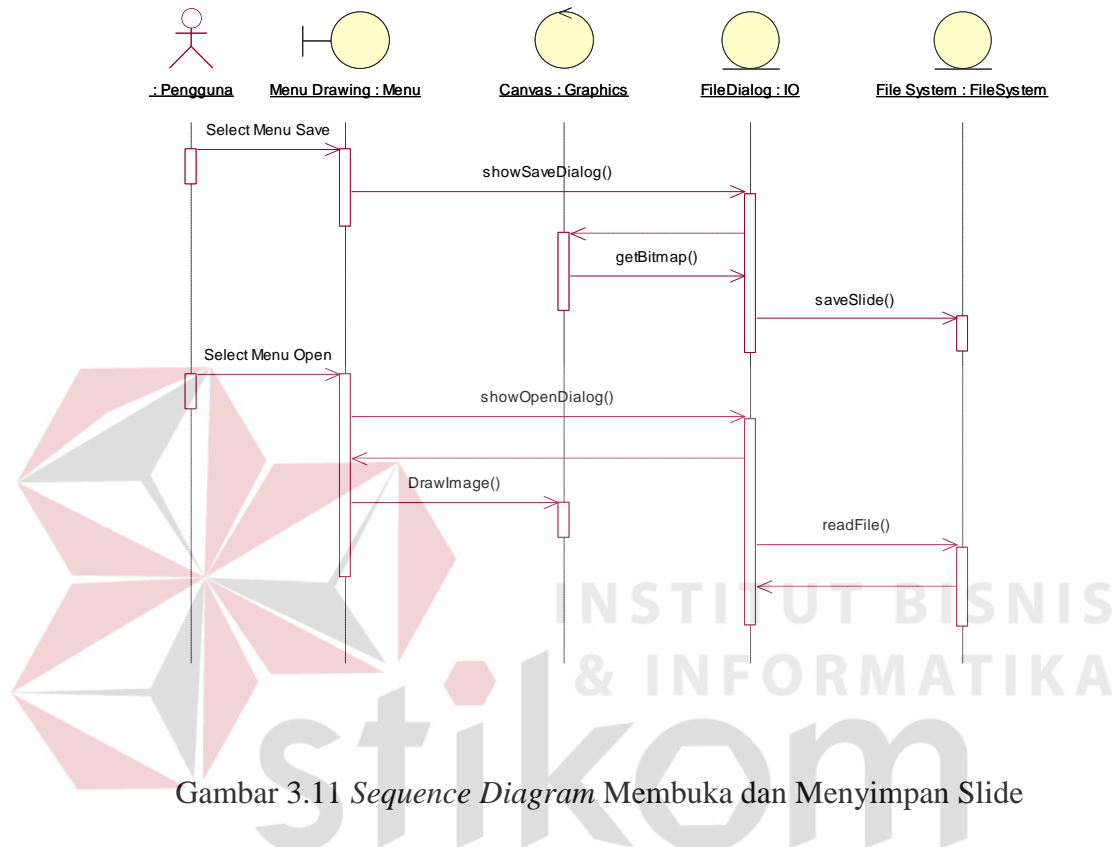


Gambar 3.10 *Sequence Diagram* Mengganti Slide

Jika pengguna menekan tombol *navigation* maka akan system akan mengatur nilai dari variable index untuk mendapatkan bitmap yang telah disimpan di dalam variable *array* pada posisi *index* sekarang dan menampilkan bitmap tersebut di *canvas*.

3.5.4 Sequence Diagram Membuka dan Menyimpan Slide

Sequence diagram dari *use case* membuka dan menyimpan Slide dapat digambarkan seperti pada gambar 3.11

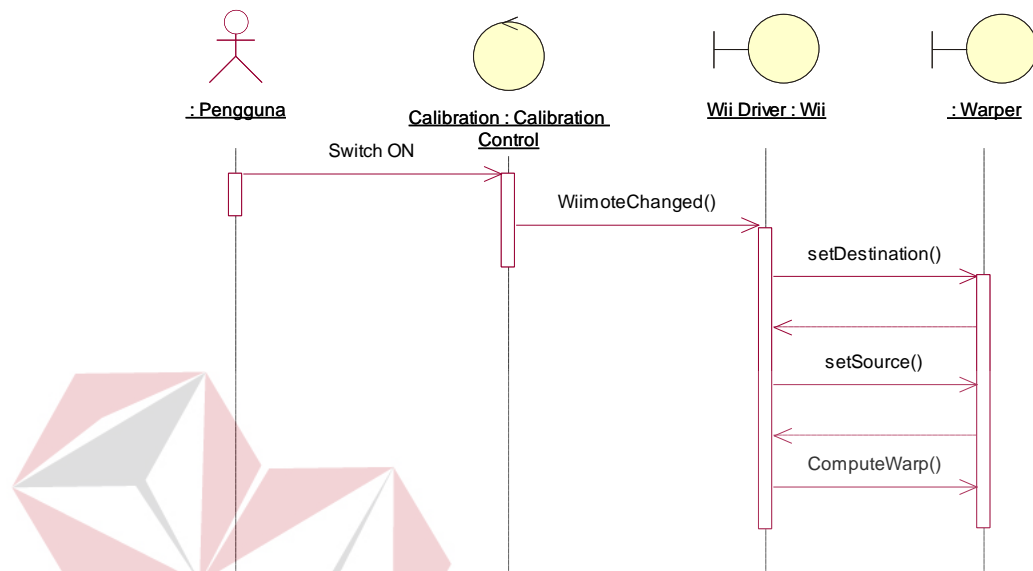


Gambar 3.11 *Sequence Diagram* Membuka dan Menyimpan Slide

Jika pengguna menekan tombol *open* maka *system* akan memunculkan jendela *open dialog* lalu mendapatkan *file* yang telah dipilih oleh pengguna kemudian menampilkannya di *canvas*, apabila pengguna menekan tombol *save* maka *system* akan menampilkan jendela *save dialog* untuk mendapatkan *path* yang telah dipilih oleh pengguna untuk menyimpan *slide* dengan mengatur *file format* dalam bentuk *jpeg* dan memberikan nilai untuk kualitas gambar yang akan disimpan.

3.5.5 Sequence Diagram Kalibrasi

Sequence diagram dari *use case* kalibrasi dapat digambarkan seperti pada gambar 3.12

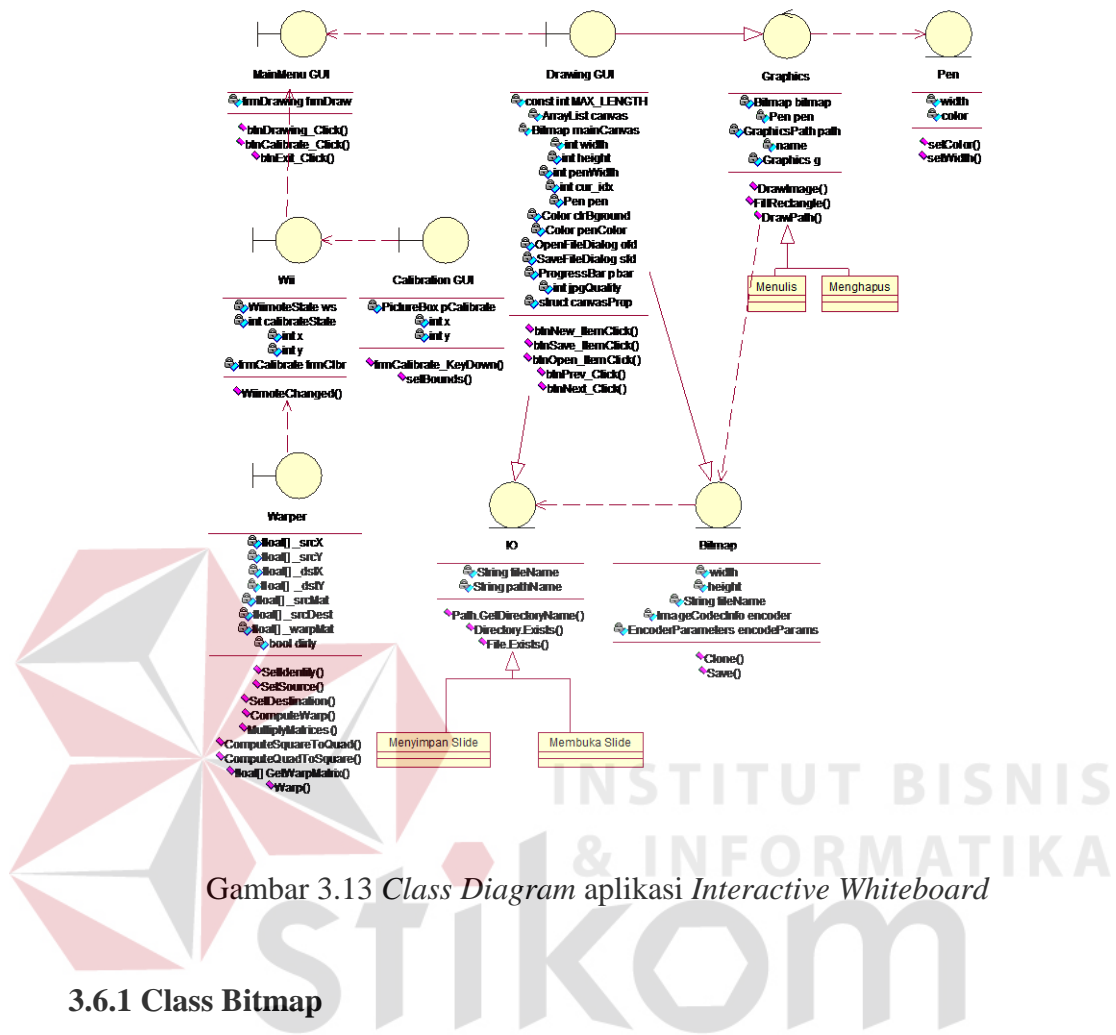


Gambar 3.12 *Sequence Diagram* Membuka dan Menyimpan Slide

Pengguna menyalakan tombol pada *IR Pen* yang kemudian ditangkap oleh wiimote kemudian dikirim ke *PC* dan diterima oleh sistem melalui *Library* dari *WiimoteLib* setelah itu menentukan *bounds* untuk batas kanan, atas, bawah dan kiri yang digunakan untuk membatasi pergerakan *IR Pen*.

3.6 Class Diagram

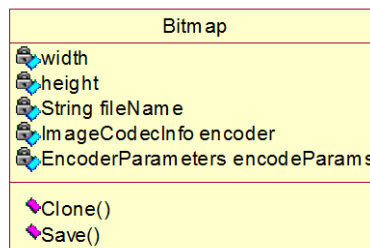
Berdasarkan perencanaan sistem *use case diagram*, dibutuhkan class-class untuk membangun dan mendukung jalannya aplikasi. *Class Diagram* atau Diagram Kelas digunakan untuk menampilkan kelas-kelas atau paket-paket di dalam sistem serta relasi antar kelas tersebut. Hubungan antar class-class tersebut dapat digambarkan sebagai berikut :



Gambar 3.13 Class Diagram aplikasi Interactive Whiteboard

3.6.1 Class Bitmap

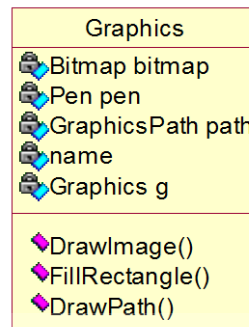
Kelas *Bitmap* merupakan kelas yang digunakan untuk mengatur panjang dan lebar sebuah *canvas* yang akan digunakan untuk menulis, serta digunakan untuk mendapatkan image yang terdapat pada *canvas*. Untuk lebih jelasnya dapat dilihat pada gambar 3.14.



Gambar 3.14 Class Bitmap

3.6.2 Class Graphics

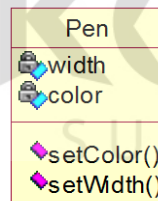
Kelas *Graphics* digunakan untuk melakukan penggambaran shape yang digunakan untuk menulis, memberi warna background dan menampilkan gambar dari *file*. Untuk lebih jelasnya dapat dilihat pada gambar 3.15.



Gambar 3.15 *Class Graphics*

3.6.3 Class Pen

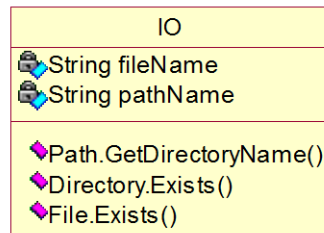
Kelas *Pen* digunakan untuk mengatur warna dan ketebalan *pen* yang dipakai waktu menulis di canvas pada kelas *graphics*. Untuk lebih jelasnya dapat dilihat pada gambar 3.16.



Gambar 3.16 *Class Pen*

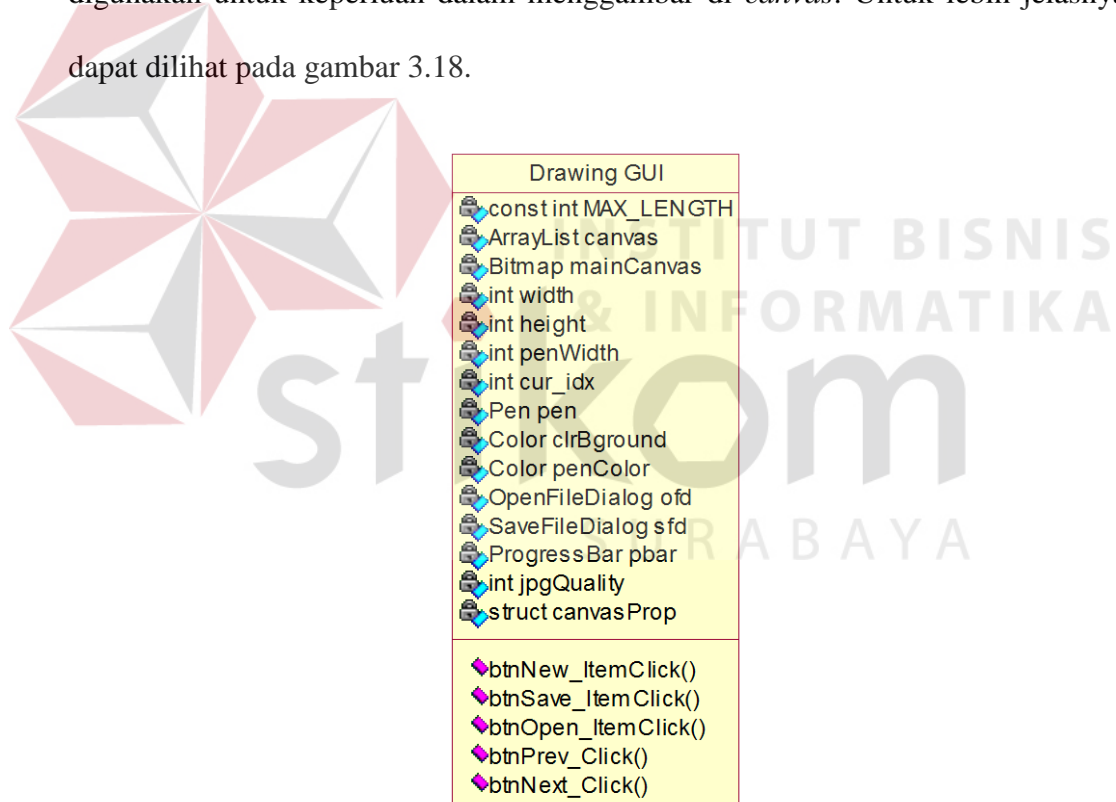
3.6.4 Class IO

Kelas *IO* merupakan kelas yang digunakan untuk membuka koneksi ke *file system* yang digunakan untuk menyimpan dan membuka file gambar yang telah dipilih melalui jendela dialog. Untuk lebih jelasnya dapat dilihat pada gambar 3.17.

Gambar 3.17 *Class IO*

3.6.5 Class Drawing

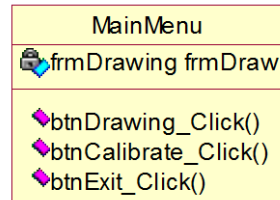
Class Drawing digunakan untuk menampilkan menu dan tool yang digunakan untuk keperluan dalam menggambar di *canvas*. Untuk lebih jelasnya dapat dilihat pada gambar 3.18.

Gambar 3.18 *Class Drawing*

3.6.6 Class MainMenu

Class MainMenu digunakan untuk menampilkan semua menu yang disediakan oleh aplikasi *interactive whiteboard*. *Class MainMenu* juga

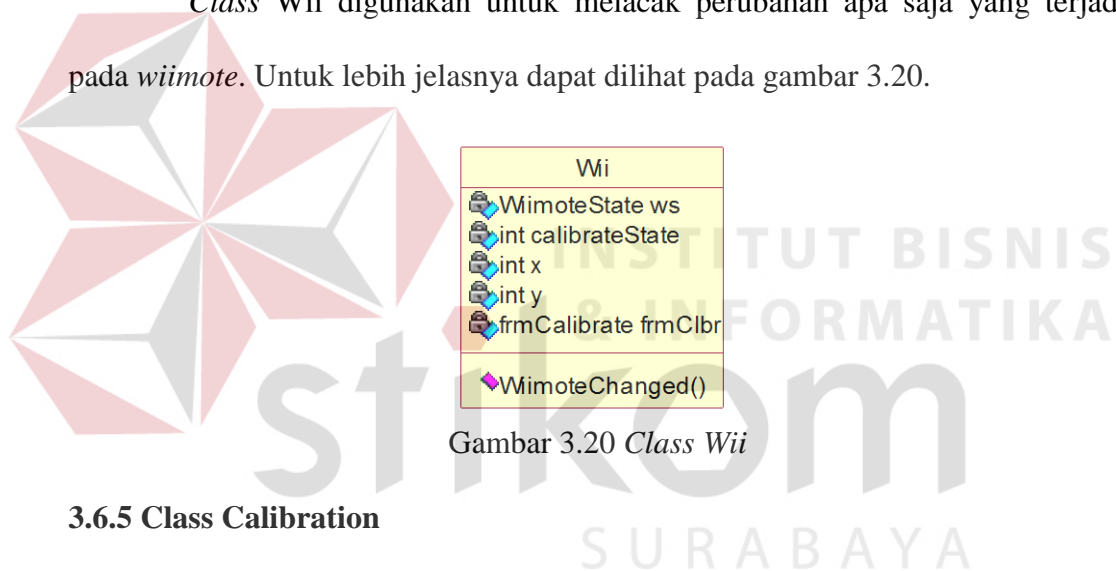
mengontrol *class-class* menu yang lain untuk berinteraksi dengan pengguna aplikasi. Untuk lebih jelasnya dapat dilihat pada gambar 3.19.



Gambar 3.19 *Class MainMenu*

3.6.5 Class Wii

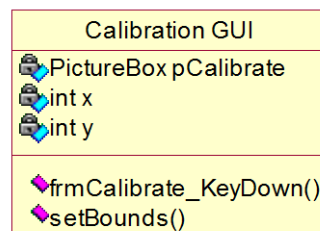
Class Wii digunakan untuk melacak perubahan apa saja yang terjadi pada *wiimote*. Untuk lebih jelasnya dapat dilihat pada gambar 3.20.



Gambar 3.20 *Class Wii*

3.6.5 Class Calibration

Class Calibration digunakan untuk menampilkan gambar target untuk mengirimkan koordinat x dan y dari posisi target tersebut ke *class Warper*. Untuk lebih jelasnya dapat dilihat pada gambar 3.21.

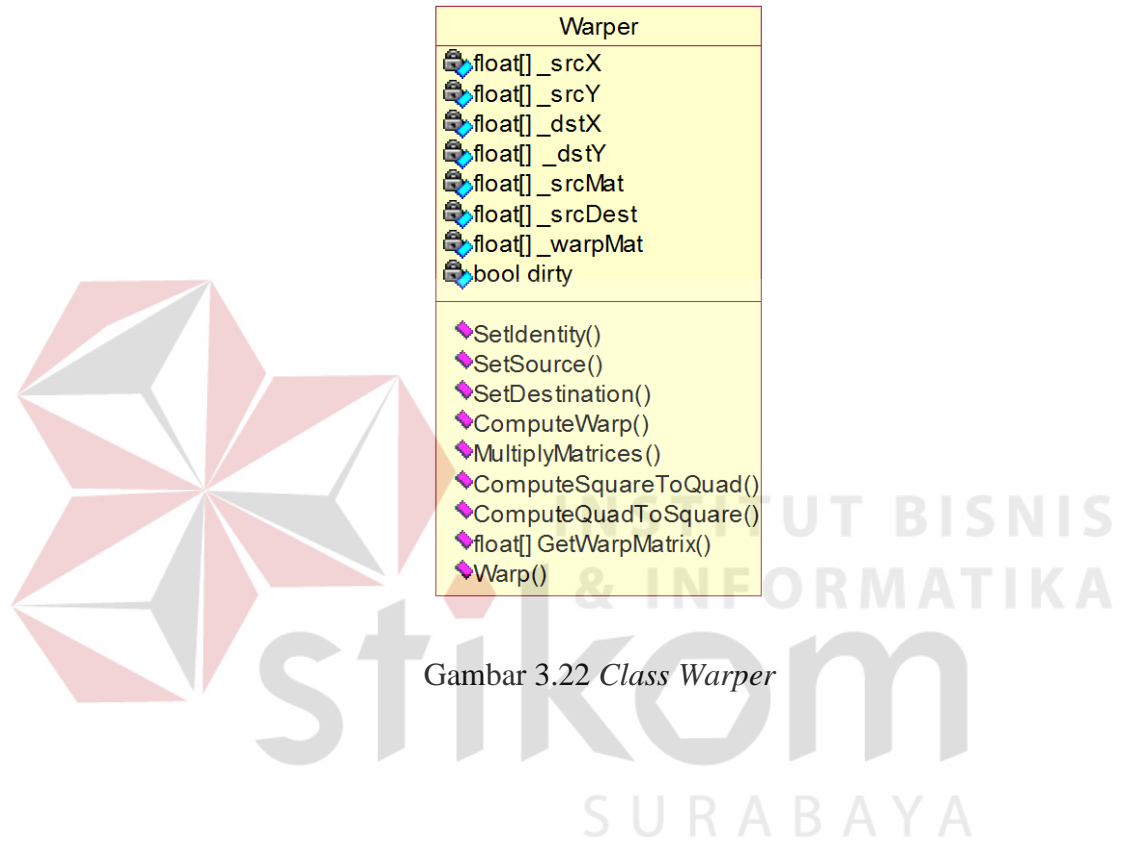


Gambar 3.21 *Class Calibration*

3.6.5 Class Warper

Class Warper digunakan untuk menginisialisasi area untuk batas atas, kanan, bawah dan kiri dan digunakan juga untuk membatasi pergerakan *IR Pen*.

Untuk lebih jelasnya dapat dilihat pada gambar 3.22.



Gambar 3.22 *Class Warper*