

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Permasalahan

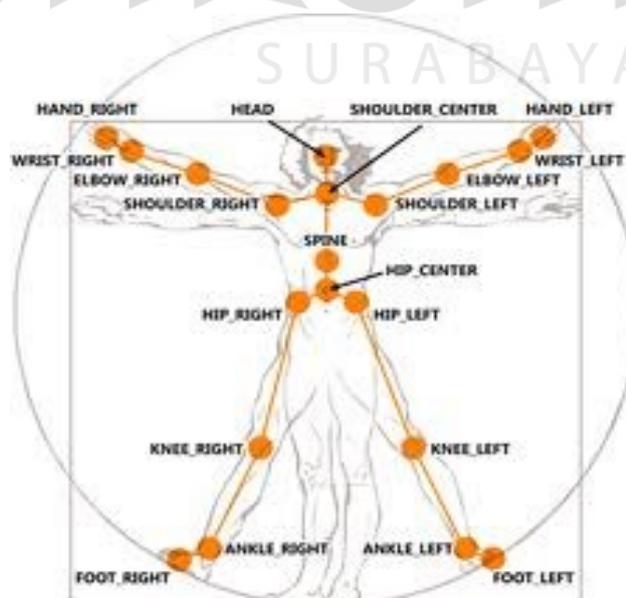
Proses penilaian posisi *Karate* selama ini dilakukan secara langsung oleh guru *Karate*. Langsung disini memiliki artian yaitu guru *Karate* menilai posisi *Karate* yang dilakukan oleh murid dengan bertatap muka secara langsung. Hal ini dilakukan secara rutin setiap dilakukan latihan *Karate* bersama dengan murid–murid lainnya. Di dalam proses ini guru *Karate* mendistribusikan pengetahuan dan pengalamannya selama mempelajari *Karate* dengan memberikan saran untuk para muridnya. Secara sederhana proses penilaian tersebut adalah sebagai berikut. Guru *Karate* mengarahkan muridnya untuk melakukan sebuah gerakan. Di akhir gerakan dimana murid berusaha untuk tetap dalam keadaan diam, guru *Karate* akan menghampiri murid dan akan melihat posisi dari murid. Jika posisi murid kurang tepat, guru akan mengarahkan langsung untuk mengoreksi posisi yang salah.



Gambar 3.1 Latihan Karate (Abdiwan, 2011)

Proses ini dapat dipersingkat dengan menambah jumlah pengajar sehingga waktu yang diperlukan dalam mengkoreksi posisi murid akan menjadi lebih singkat. Dengan adanya tambahan bantuan, maka proses latihan akan berlangsung lebih efisien. Namun tak semua kelas *Karate* dapat menambah tenaga pengajar untuk membantu proses latihan mereka. Maka untuk mengatasi permasalahan tersebut maka dapat dibantu oleh sebuah aplikasi penilaian posisi *Karate* yang dapat dilakukan mandiri oleh murid tanpa memerlukan bantuan guru namun tidak bermaksud untuk menggantikan posisi guru *Karate* sebagai sumber ilmu utama.

Sensor *Kinect* dapat membaca pergerakan manusia dengan mendeteksi persendian manusia yang tampak seperti pada Gambar 3.2. 12 persendian utama akan digunakan untuk perbandingan data yang mencakup pergelangan tangan, siku tangan, bahu, pinggul, lutut dan pergelangan kaki sebelah kiri dan kanan. Sedangkan untuk 3 titik yang dapat dibaca sensor *Kinect* selain persendian di atas (titik bahu tengah, titik tulang belakang, dan titik panggul tengah) tidak digunakan dalam proses penilaian.



Gambar 3.2 *Skeleton Tracking* pada Tubuh Manusia (Microsoft Corporation)

Tidak semua teknik *Kihon* dapat dinilai menggunakan sensor *Kinect*. Hal ini disebabkan oleh beberapa hal antara lain adalah keterbatasan perangkat keras dan kompleksitas gerakan. Sensor *Kinect* generasi pertama memiliki waktu beberapa detik untuk melacak posisi tubuh manusia dimana pengguna harus bergerak untuk membedakan antara manusia dan latar belakangnya. Masalah selanjutnya terdapat pada penggunaan dua sensor *Kinect* untuk mengoptimalkan *skeleton tracking* yang masih diteliti untuk meningkatkan keakurasiannya. Dan masalah yang terakhir yaitu posisi tubuh pada akhir gerakan *Kihon* tidak dapat dipertahankan kestabilannya. Hal ini merupakan masalah utama dalam penilaian tendangan dimana posisi akhir tendangan kembali ke kuda-kuda. Beberapa *kihon* yang diperkirakan dapat dinilai terlampir di Lampiran 2, 3, dan 4.

Kebutuhan aplikasi yang perlu disiapkan untuk menggunakan aplikasi ini adalah perangkat keras untuk output, pemrosesan data dan sensor penangkap posisi manusia. Untuk perangkat keras output aplikasi dapat menggunakan monitor maupun proyektor. Sedangkan untuk pemrosesan data dibutuhkan sebuah komputer yang akan memproses data menjadi informasi yang diharapkan. Dan yang terakhir dibutuhkan perangkat sensor *Kinect* untuk menangkap posisi manusia dan mengubahnya menjadi data digital.

Aplikasi ini akan mengeluarkan informasi berupa nilai ketepatan posisi tubuh pada posisi karate yang ingin dinilai. Nilai ini didapat dari perhitungan rata-rata selisih tiap persendian. Proses penilaian ini dimulai dengan aplikasi akan menampilkan pilihan posisi yang akan dicoba. Lalu, pengguna aplikasi memilih salah satu dari posisi yang ingin dinilai. Selanjutnya sistem akan mengeluarkan informasi dasar mengenai gerakan *Karate* yang dipilih. Setelah informasi tersebut

disampaikan, sistem akan mengkonfirmasi kesiapan pengguna untuk mencoba gerakan tersebut. Ketika pengguna siap, pengguna akan melakukan gerakan yang telah dipilih dan pada akhir gerakan pengguna akan memberikan perintah suara yang menandakan pengguna sudah ada di posisi akhir gerakan yang selanjutnya akan dinilai. Penilaian ini akan dilakukan dengan melihat posisi akhir dari pengguna dan akan dibandingkan dengan data yang sudah disimpan di basis data. Lalu selanjutnya sistem akan mengeluarkan hasil penilaian dari posisi pengguna.

Dari hasil analisis diatas maka sistem aplikasi yang akan dibangun memiliki spesifikasi kebutuhan sebagai berikut :

1. Aplikasi harus dapat membaca hasil bacaan sensor *Kinect*.
2. Aplikasi harus dapat membandingkan hasil bacaan sensor *Kinect* dengan data model yang sudah disimpan.
3. Aplikasi harus dapat menyajikan informasi sebagai umpan balik ke pengguna aplikasi.
4. Aplikasi harus dapat menerima perintah suara.

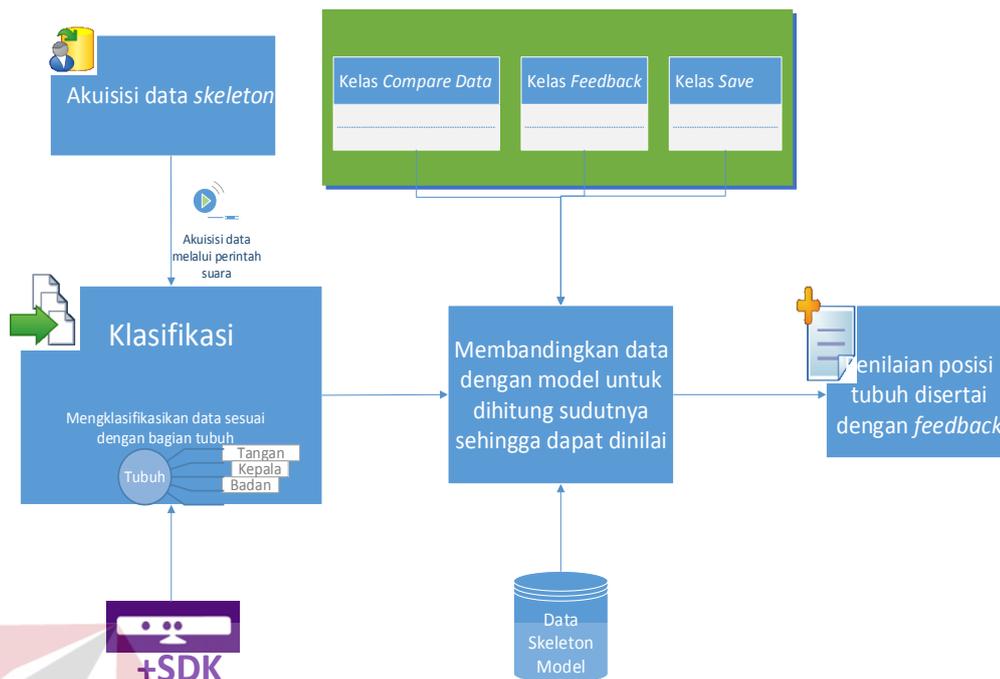
3.2 Perancangan Sistem

Untuk mencapai tujuan yang sudah disebutkan di bagian sebelumnya, aplikasi ini melalui beberapa tahap proses yang dapat dilihat pada Gambar 3.3.

Secara sederhana proses-proses yang ada di aplikasi ini adalah :

1. Menampilkan pilihan gerakan *Karate*
2. Pengguna memilih gerakan yang diinginkan menggunakan *gesture* dari tangan yang berfungsi sebagai tetikus virtual.

- Untuk menggerakkan *pointer* dapat dilakukan dengan salah satu tangan. Sistem ini tidak dapat menerima masukan ketika pengguna menggunakan dua tangannya.
 - Untuk memilih menu, pengguna dapat mendorong tangannya ke arah sensor.
3. *Kinect* memproses pilihan tersebut dan menyampaikannya ke program berupa pilihan input.
 4. Sistem akan mengeluarkan informasi mengenai gerakan *Karate* yang telah dipilih.
 5. Sistem mengkonfirmasi kesiapan *user* untuk mencoba gerakan dengan menampilkan pesan notifikasi berisi informasi mengenai proses pengambilan data.
 6. *User* mengkonfirmasi posisi tubuh dengan menggunakan perintah suara.
 7. Sistem akan memproses masukan posisi tubuh pengguna setelah pengguna memberikan perintah suara yang menandakan gerakan telah berakhir dan pengguna dalam posisi terakhir dari gerakan tersebut.
 8. Sistem akan memberikan hasil penilaian dari posisi pengguna.



Gambar 3.3 Blok Diagram Aplikasi

Dalam proses pengambilan data posisi tubuh, terdapat beberapa kendala antara lain keterbatasan sensor *Kinect* dalam mendeteksi beberapa posisi tubuh. Hal ini menyebabkan perlu adanya beberapa penyesuaian dalam proses penggunaan aplikasi. Beberapa kendala tersebut berhubungan dengan proses dan kemampuan deteksi tubuh manusia.

Kendala yang pertama adalah sensor *Kinect* hanya akan melakukan proses deteksi tubuh manusia jika pengguna menghadap sensor. Hal ini menyebabkan pengguna harus menghadap sensor ketika akan dilakukan pendeteksian. Masalah ini diperparah dengan adanya jeda waktu yang diperlukan sensor yang lamanya tergantung dari pendeteksian yang dipicu oleh pergerakan tubuh manusia. Kendala ini dapat dipecahkan dengan penggunaan dua modul sensor dari dua sisi. Namun penggunaan dua sensor ini pun memiliki kelebihan dan kekurangan.

Kendala lainnya timbul dari kemampuan interpretasi sensor *Kinect*. Hasil deteksi *skeleton tracking* berupa sebuah gambaran rangka tubuh manusia dalam garis yang terdiri dari 20 titik. Masing-masing titik akan dihubungkan dan membentuk rangka tubuh manusia. Dalam proses pembentukan rangka ini, sensor *Kinect* akan mengalami kesulitan pada bagian tubuh yang saling silang sehingga hasil interpretasi tidak akurat. Beberapa peneliti telah mengembangkan sebuah solusi untuk masalah ini dengan menggunakan dua sensor. Solusi ini dipaparkan dalam sebuah jurnal berjudul “*Improved Skeleton Tracking by Duplex Kinects: A Practical Approach for Real-Time Applications*” oleh Yeung, Kwok, dkk. Hasil penggunaan dua sensor dalam pendeteksian mampu membantu interpretasi posisi tubuh manusia menjadi lebih baik, namun untuk bagian tubuh yang saling silang masih belum dapat diinterpretasikan dengan benar.

Maka dari beberapa masalah tersebut, beberapa gerakan yang perlu untuk dilakukan penilaian dari dua sisi akan dilakukan secara bergantian menggunakan satu buah sensor. Penilaian yang dilakukan dua kali ini masing-masing akan fokus pada bagian tubuh bagian atas dan bagian bawah.

Dengan menggunakan bantuan SDK *Kinect*, proses akuisisi data skeleton *Kinect* akan menghasilkan data persendian tubuh manusia yang terdiri dari jarak dan posisi persendian. Sistem akan mengambil data dari 12 persendian pengguna aplikasi yang terdiri dari bahu, lengan, pergelangan tangan, pinggul, lutut dan pergelangan kaki dari tubuh pengguna. 3 *joint* hasil *skeletal tracking* yang tidak digunakan dalam aplikasi ini adalah *center shoulder*, *center hip*, dan *spine* mengingat 3 *joint* tidak memberikan kontribusi yang signifikan dalam rancangan proses penilaian posisi tubuh.

Selanjutnya proses perbandingan data sendi dengan data akuisisi akan dibandingkan. Proses perbandingan ini berjalan dengan bantuan basis data dimana data model dan data akuisisi berada di basis data. Proses perbandingan ini dilakukan dengan cara mencari rata-rata persentase selisih sudut elevasi dari 12 persendian.

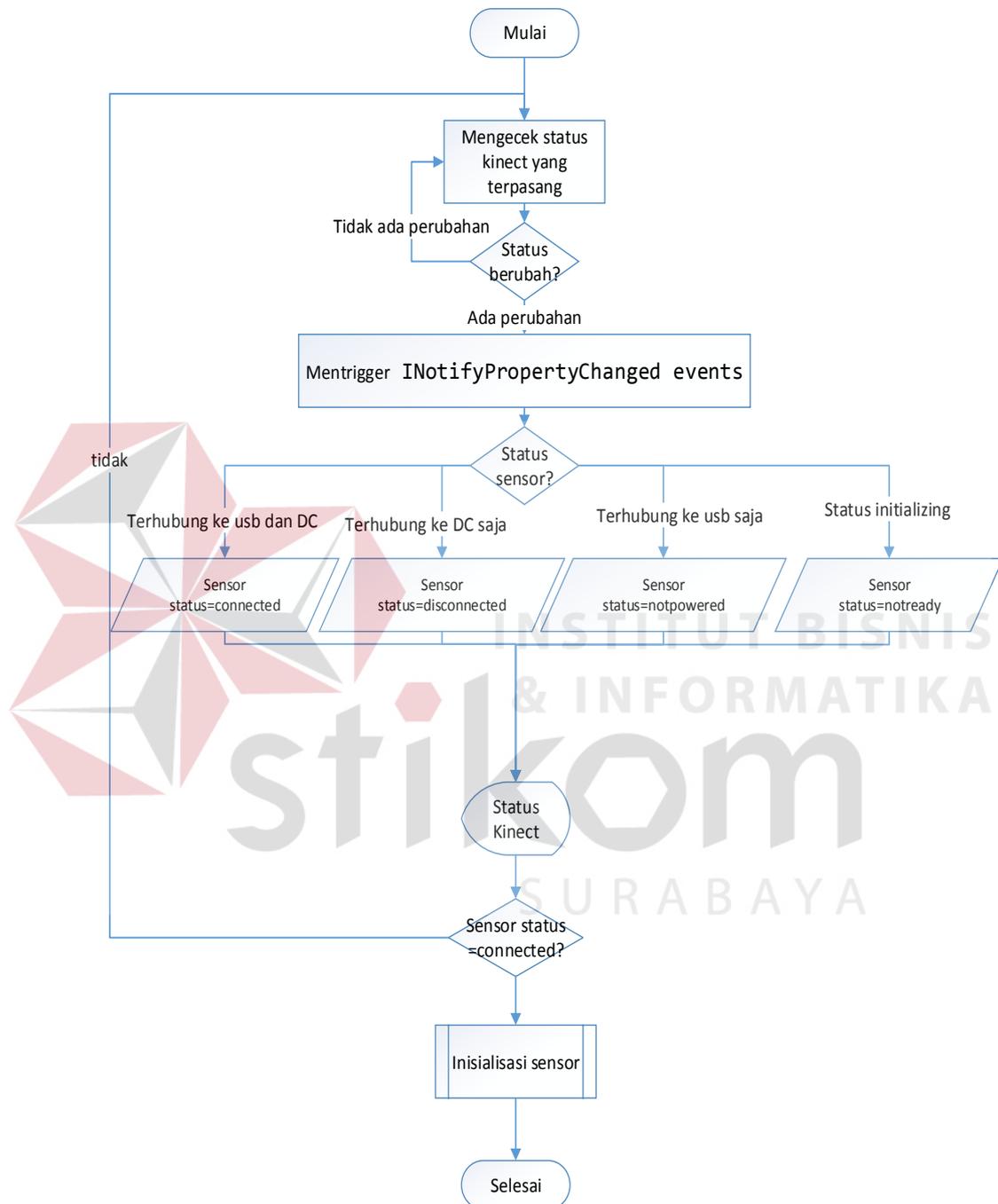
$$\bar{x} = \frac{1}{12} \sum_{i=1}^{12} (100\% - \left(\frac{\Delta_j}{j_1} \times 100\%\right)) \dots \dots \dots 3.1$$

Persamaan 3.1 adalah proses perhitungan untuk mencari nilai dari tingkat keakurasian data sendi dari model dengan data model yang didapat. J adalah derajat persendian dimana j_1 adalah derajat persendian model dan j_2 adalah derajat persendian pengguna aplikasi. Hasil penilaian ini tidak menunjukkan tingkat keakurasian pendeteksian *skeleton* oleh sensor *Kinect* melainkan tingkat keakurasian data model dengan data yang didapat.

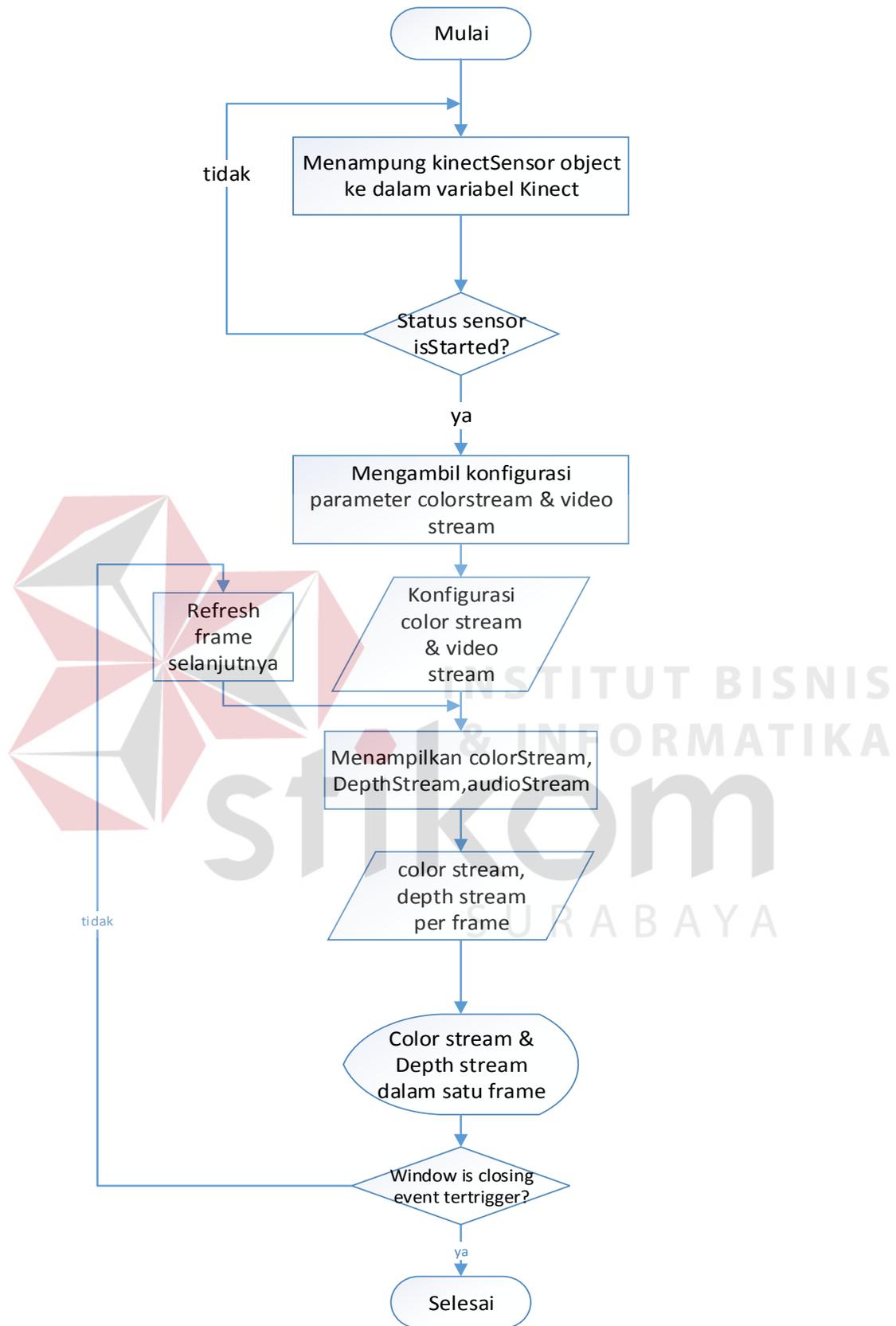
Dari analisis permasalahan diatas, terdapat beberapa modul yang perlu dibuat demi memenuhi kebutuhan perangkat lunak yang dibangun. Beberapa modul yang diperlukan dalam pengembangan perangkat lunak ini yaitu modul untuk menyalakan sensor *Kinect*, modul untuk menghitung derajat, modul untuk membandingkan data referensi dengan data pengguna, dan modul untuk perintah suara. Masing-masing modul tersebut saling berhubungan satu sama lain untuk menunjang kerja perangkat lunak.

Modul untuk menyalakan sensor *Kinect* dibagi menjadi dua, yaitu proses pengecekan status sensor *Kinect* dan proses menginisialisasi sensor *Kinect*. Proses pengecekan status sensor bertujuan untuk mencari sensor *kinect* yang terhubung dengan komputer dan siap untuk digunakan. Sedangkan proses inisialisasi sensor bertujuan untuk menyalakan sensor agar siap digunakan dalam proses berikutnya.

Proses pengecekan status sensor tampak pada *flowchart* di Gambar 3.4. Sedangkan proses inialisasi sensor dijelaskan melalui *flowchart* pada Gambar 3.5.

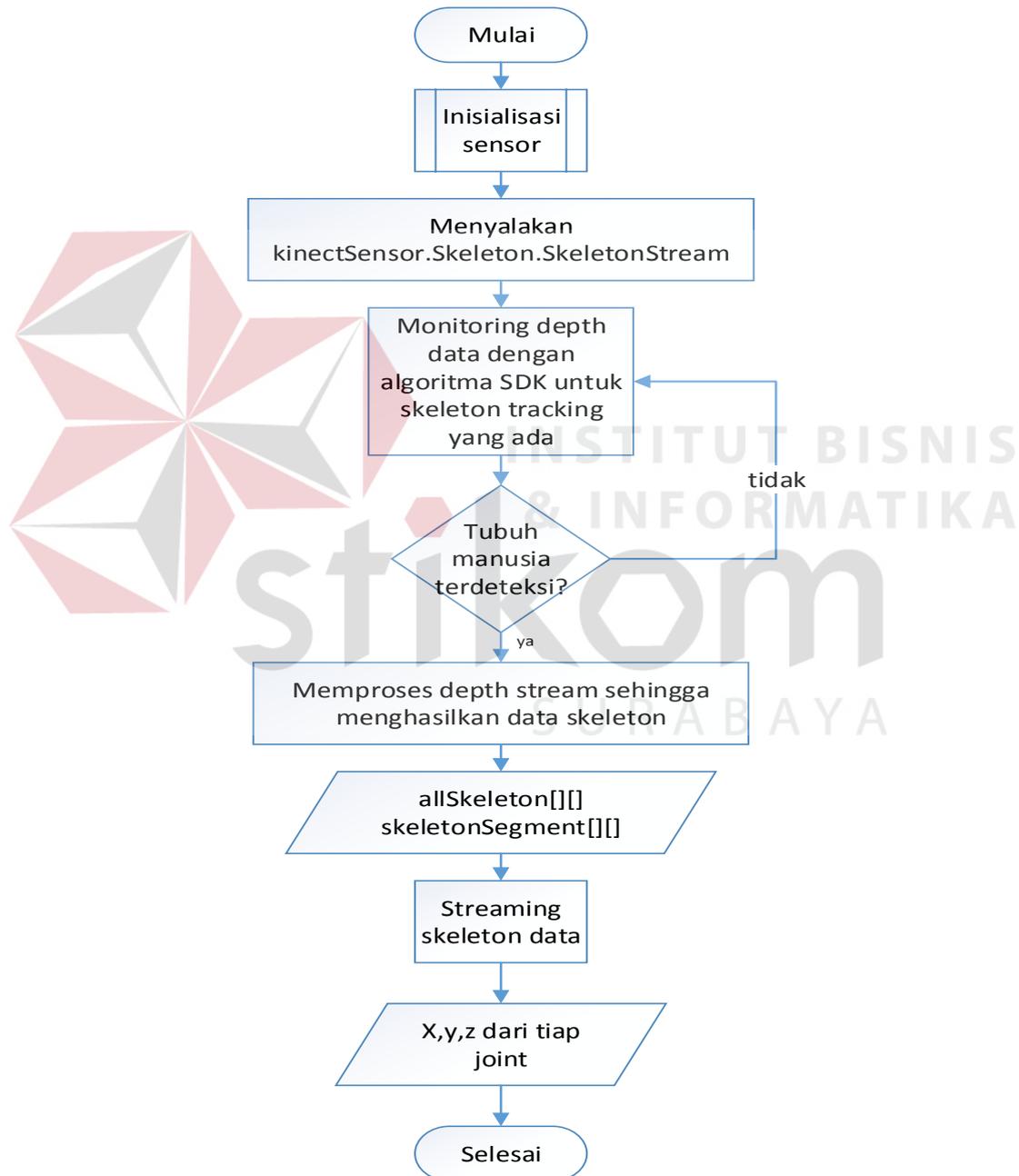


Gambar 3.4 *Flowchart* Proses Pengecekan Status Sensor



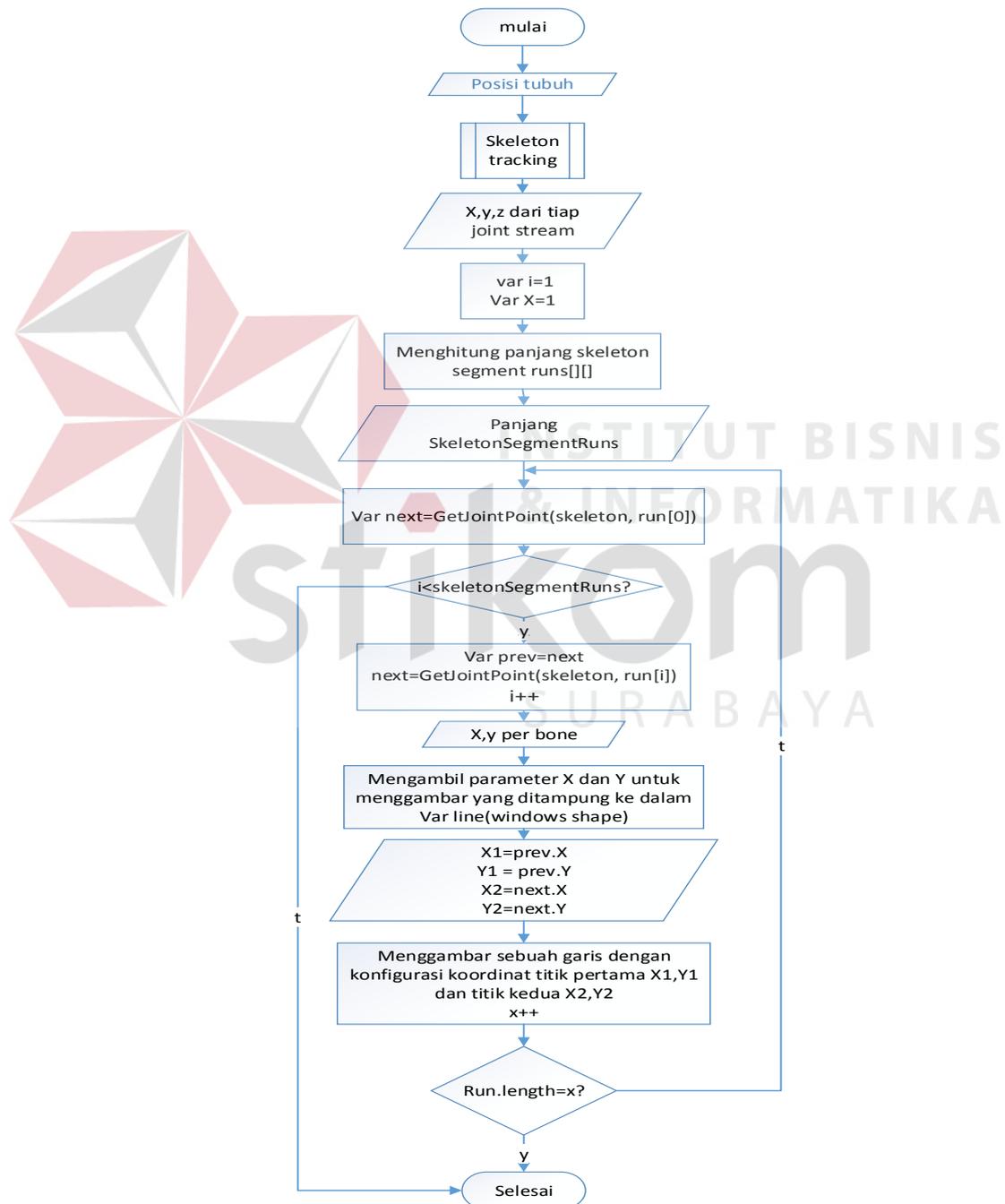
Gambar 3.5 *Flowchart* Proses Inisialisasi Sensor

Sensor Kinect memiliki kemampuan untuk melacak posisi tubuh manusia yang disebut dengan *skeleton tracking*. Proses *skeleton tracking* ini akan melalui tahap proses yang dijelaskan sebelumnya, yaitu proses pengecekan sensor dan inialisasi sensor. Secara garis besar proses *skeleton tracking* tampak pada *flowchart* di Gambar 3.6.



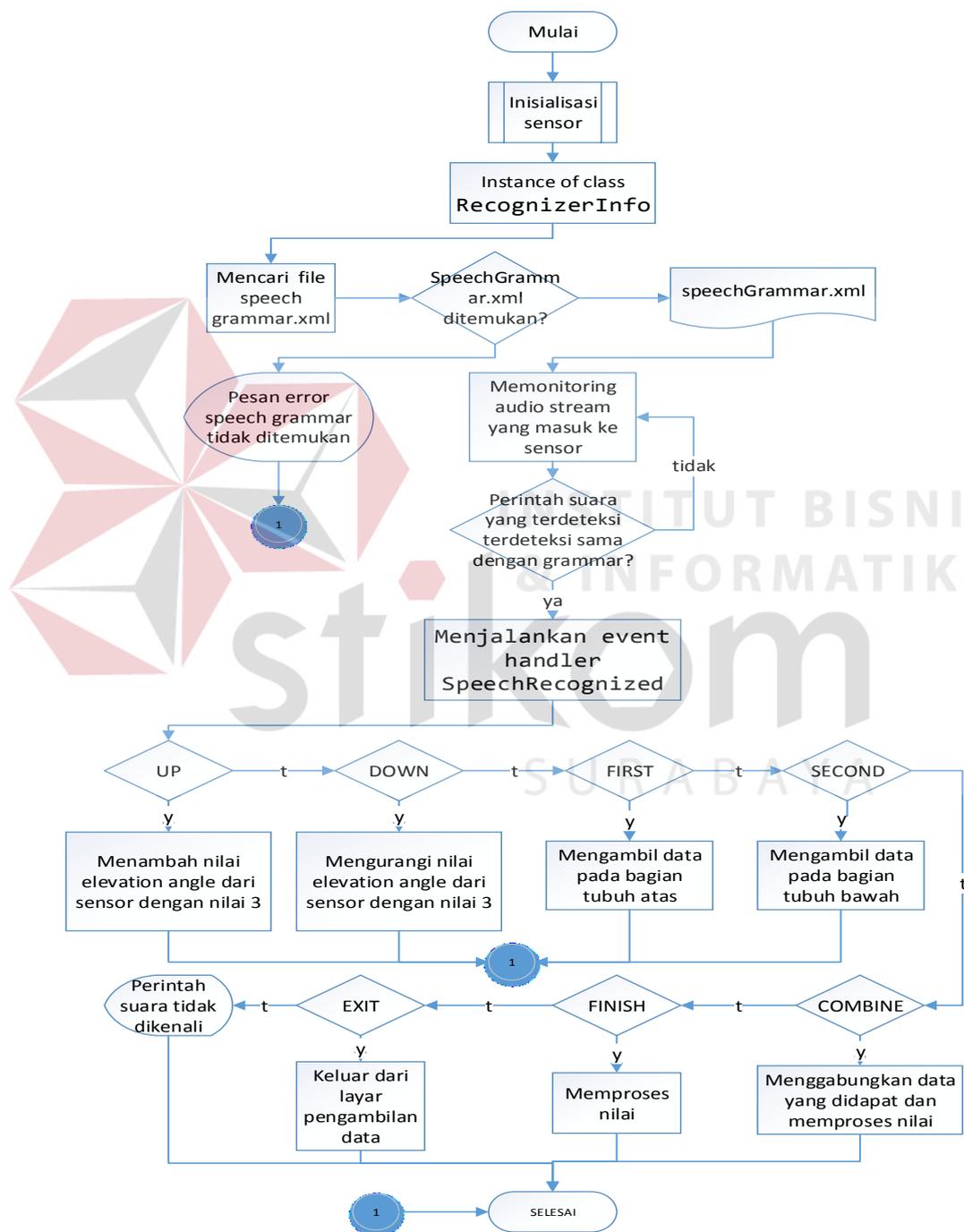
Gambar 3.6 *Flowchart* Proses *Skeleton Tracking*

Proses selanjutnya setelah *skeleton tracking* adalah memvisualisasikan data yang didapat menjadi bentuk rangka tubuh manusia. Proses ini disebut proses *drawing skeleton* yang akan menghasilkan sebuah visualisasi kerangka manusia. Proses *drawing* ini melewati beberapa proses seperti yang ada pada *flowchart* proses *drawing skeleton* di Gambar 3.7.



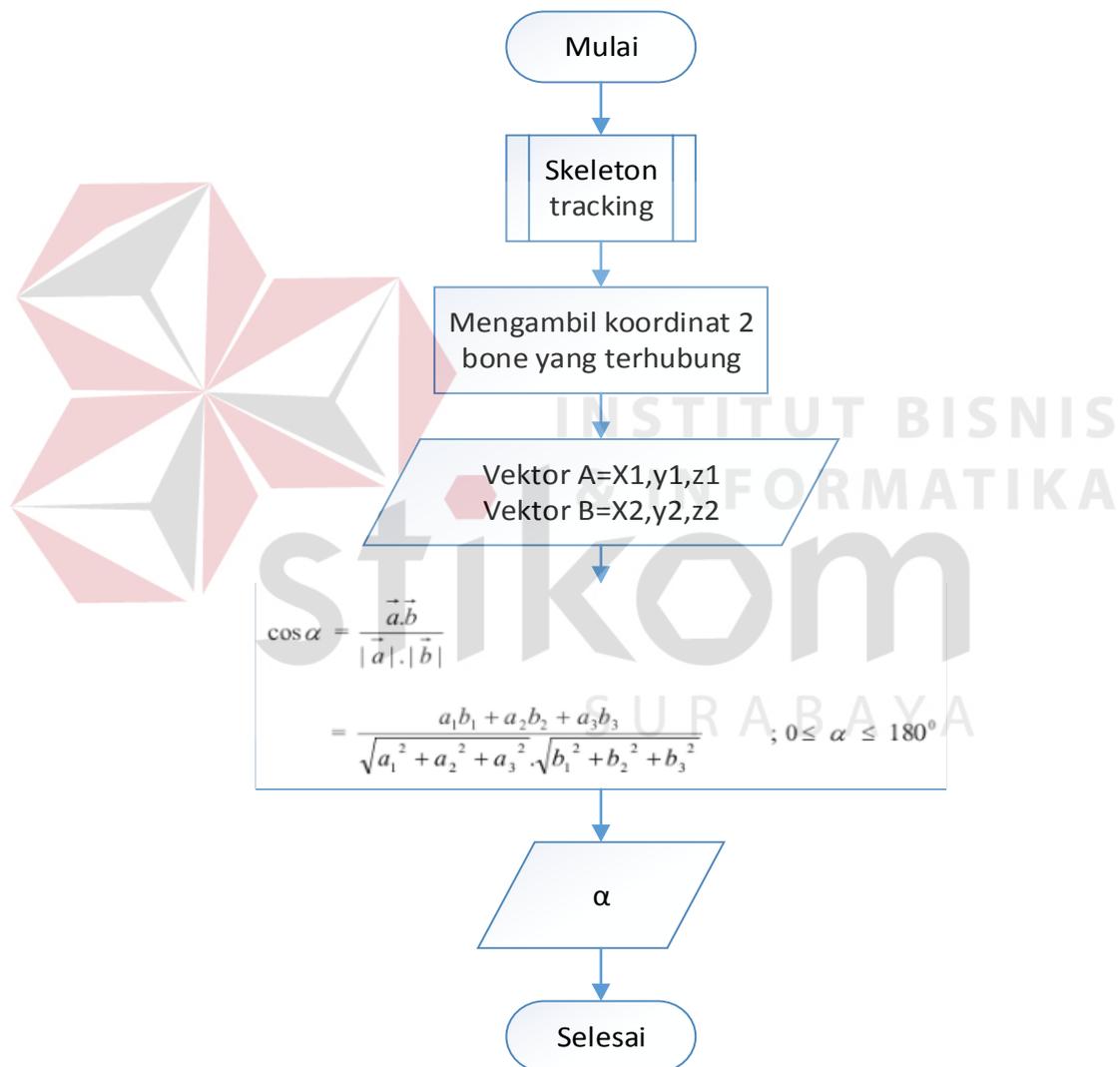
Gambar 3.7 *Flowchart* Proses *Drawing Skeleton*

Modul selanjutnya merupakan sebuah fungsionalitas yang memungkinkan pengguna untuk memerintah perangkat lunak untuk melakukan beberapa perintah yang sudah disimpan yaitu perintah suara. Proses untuk penangkapan perintah suara tampak pada *flowchart* pada Gambar 3.8.

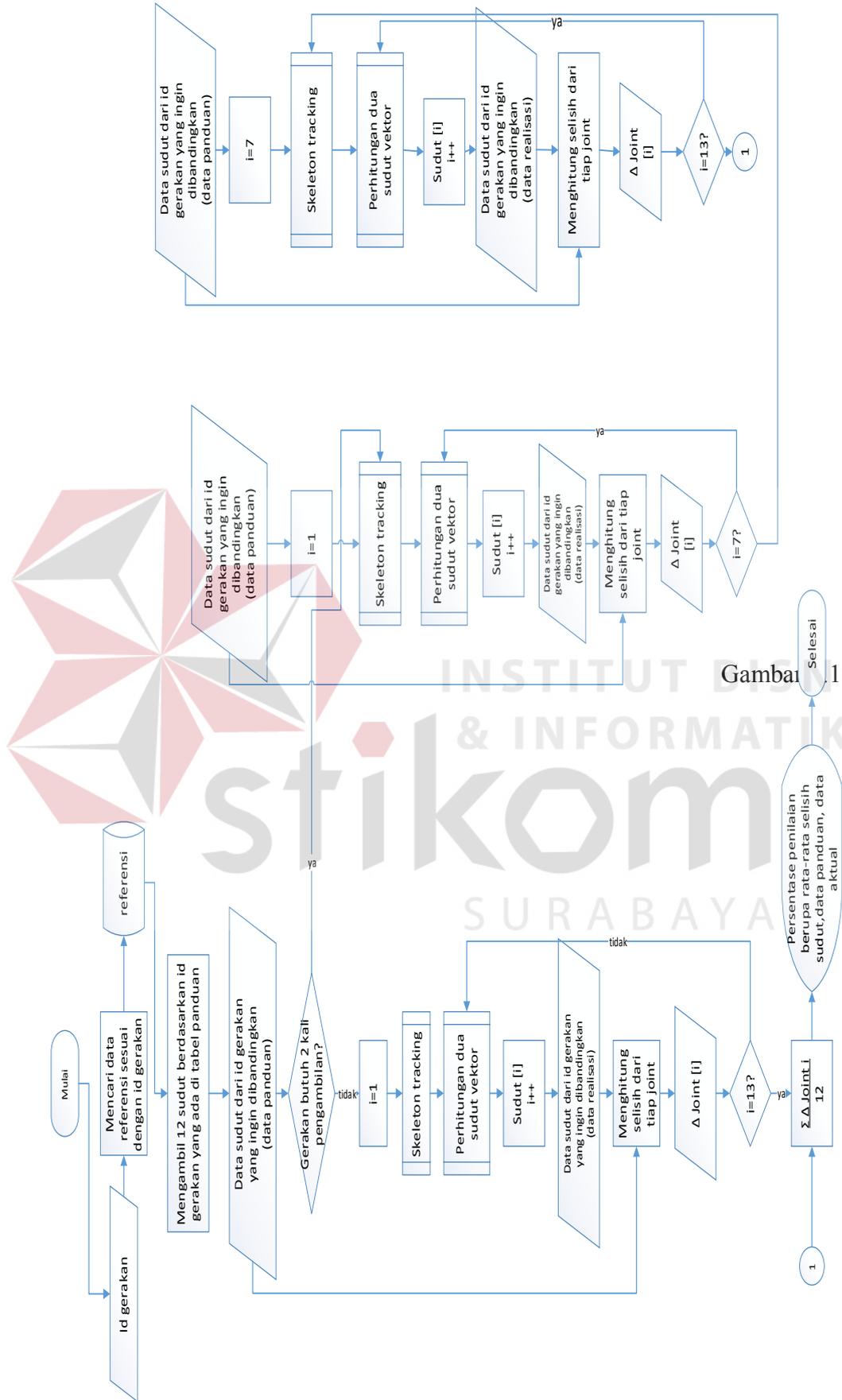


Gambar 3.8 *Flowchart* Proses Penangkapan Perintah Suara

Dua modul lainnya yang perlu dibuat adalah modul penting yang berfungsi dalam proses penilaian, yaitu proses perhitungan sudut dan proses perbandingan data. Proses perhitungan sudut melalui beberapa proses yang dijelaskan melalui *flowchart* pada Gambar 3.9. Sedangkan alur proses yang dilalui perangkat lunak untuk mendapatkan penilaian tampak pada Gambar 3.10. Proses perhitungan sudut menggunakan formula 2.1 yaitu rumus perkalian *dot product*.

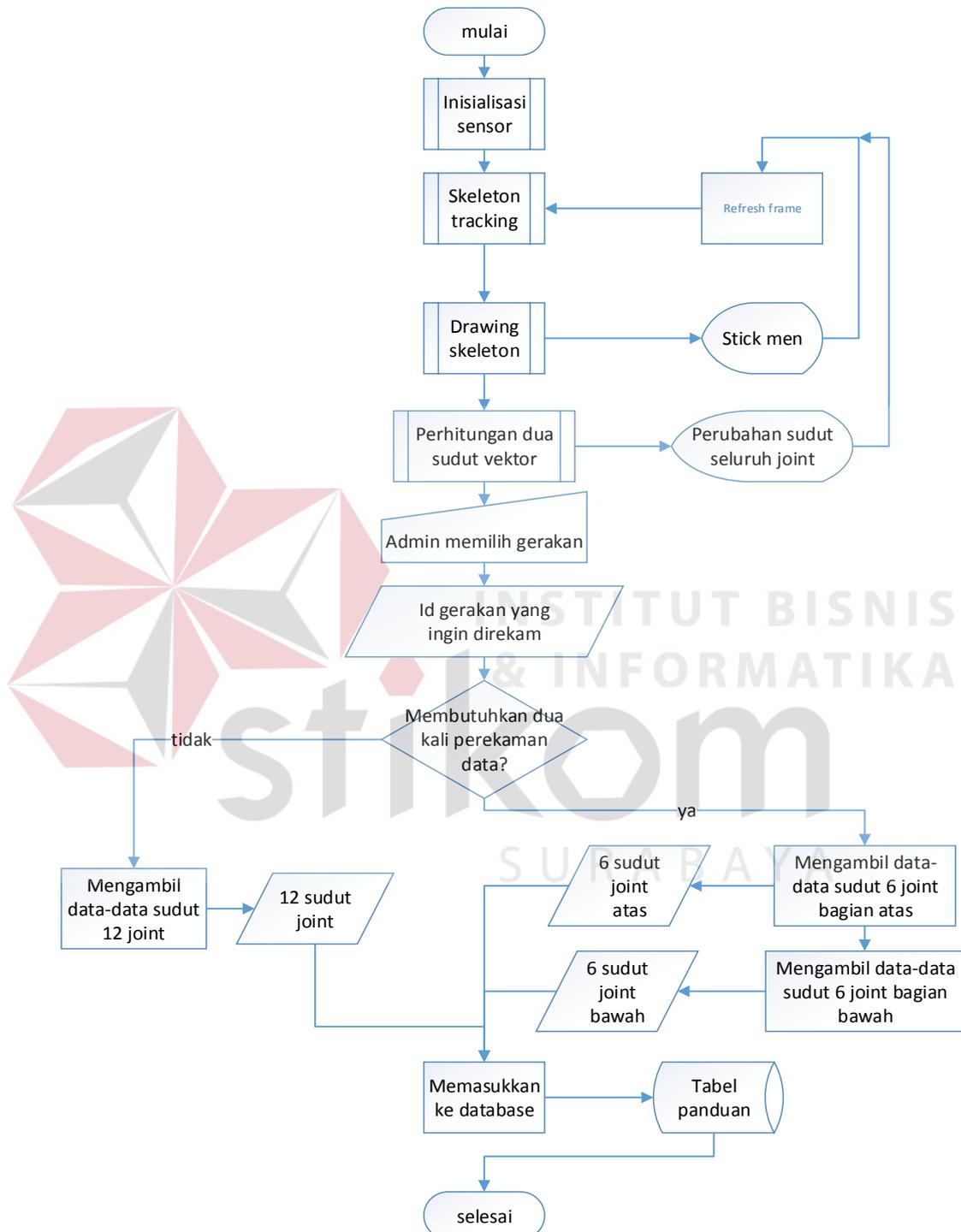


Gambar 3.9 *Flowchart* Proses Perhitungan Sudut



Gambar 10 Flowchart Proses

Proses untuk melakukan perekaman data sebagai referensi tampak pada Gambar 3.11.



Gambar 3.11 Flowchart Proses Perekaman Data Referensi

3.2.1 Agile Planning

Dari aktifitas-aktifitas yang sudah diidentifikasi diatas, maka dibuatlah *user story* yang masing-masing memiliki tujuan. Aktifitas-aktifitas ini akan dikembangkan satu persatu seiring dengan berjalannya waktu. Aktifitas tersebut tampak pada *user story* di Tabel 3.1.

Masing-masing dari *user story* tersebut akan mengalami fase *inception*, *elaboration*, *construction*, dan *transition* secara *iterative* jika dibutuhkan. *Setiap user story* yang berhasil dibuat akan disimpan dan dilanjutkan ke *user story* berikutnya secara *incremental*. Jika di tengah jalan akan diadakan penyesuaian, maka proses yang sedang dijalani akan hentikan sementara dan kembali ke proses sebelumnya yang perlu dilakukan perbaikan maupun perubahan.

Tabel 3.1 *User Story*

Nama Aktifitas	Sebagai	Kegiatan	Tujuan
Menampilkan pilihan gerakan karate	Pengguna	Melihat gerakan yang tersedia	Memilih gerakan yang diinginkan
Menampilkan informasi gerakan	Pengguna	Mencari tahu informasi mengenai gerakan yang dipilih	Mengetahui informasi dari gerakan
Memilih gerakan karate	Pengguna	Memilih gerakan yang diinginkan	Membuka window penilaian sesuai dengan gerakan yang dipilih
Melakukan proses penilaian	Pengguna	Melakukan penilaian posisi	Memproses posisi user menjadi nilai berdasarkan data referensi
Update data referensi	Admin	Melakukan update data referensi	Mengubah data referensi sesuai dengan data yang dimasukkan

3.2.2 Unified Process

Pengembangan perangkat lunak ini menggunakan *framework Unified Process* dimana terdapat beberapa fase yang harus dilalui. Fase-fase tersebut adalah

inception, elaboration, construction, dan transition. Pada tahap *inception* dilakukan proses penentuan *scope* produk dan batasan-batasannya, menguraikan permasalahan menjadi sub bagian yang lebih kecil, menentukan perencanaan jadwal dan biaya yang terkait.

Dalam tahap *inception* ini telah ditentukan scope program sebagai berikut :

1. Program yang dikembangkan akan menghasilkan penilaian posisi
2. Program yang dikembangkan menggunakan sensor Kinect
3. Program yang dikembangkan akan dapat menerima inputan data referensi dari *role model*
4. Program yang dikembangkan mampu menerima perintah melalui suara

Sedangkan batasan-batasan produk yang teridentifikasi antara lain :

1. Sensor Kinect yang digunakan adalah satu buah
2. Proses pengembangan menggunakan SDK Kinect for Windows versi 1.7

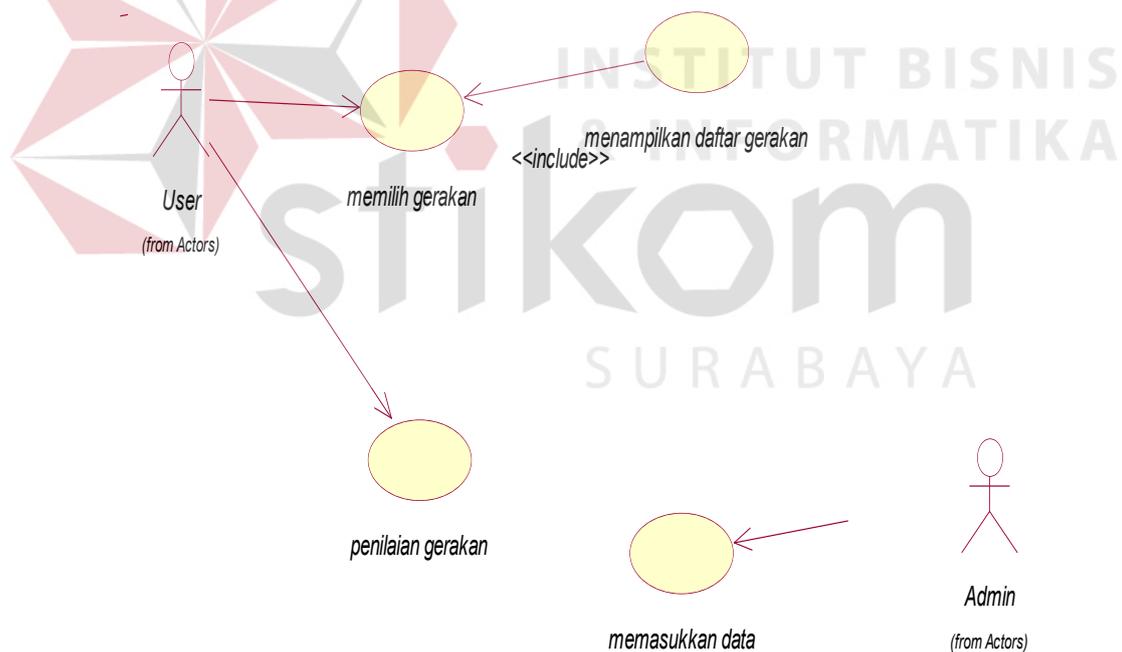
Uraian permasalahan dapat dilihat pada user story dalam bagian *Agile planning*.

Sedangkan pada tahap *elaboration* dihasilkan rancangan-rancangan mendetail dari perangkat lunak seperti *use case diagram, activity diagram* dan *class diagram*. Rancangan-rancangan ini menjadi panduan dalam tahap berikutnya yaitu *construction*. Rancangan-rancangan ini dibagi menurut fungsionalitasnya. Berbagai fungsi yang diperlukan dalam perangkat lunak ini menggunakan beberapa kelas dalam SDK Kinect. Beberapa rancangan dalam proses pengembangan perangkat lunak ini terdiri dari proses pengecekan status sensor, proses inialisasi sensor, proses *skeleton tracking*, proses *drawing skeleton*, proses perhitungan sudut vektor, proses perbandingan data dan proses perintah suara yang merupakan hasil analisis kebutuhan perangkat lunak.

Pada tahap ini perangkat lunak mulai dibangun untuk memenuhi kebutuhan pengguna yang sudah dijabarkan melalui *user story*. Ketika terdapat masalah maupun penyesuaian dalam rancangan spesifikasi perangkat lunak, maka akan dilakukan proses perubahan yang diperlukan sampai kebutuhan terpenuhi secara *iterative*. Pada tahap transition, dilakukan uji coba pada perangkat lunak agar dapat di *deploy*.

3.2.3 Use Case Diagram Aplikasi Penilaian Posisi Karate

Use case memberikan gambaran terhadap aktivitas utama pelaku dalam sistem atau *actor* dan kegiatan yang dapat dilakukannya dalam aplikasi. Sistem akan berinteraksi dengan pengguna sesuai diagram *use case* pada Gambar 3.12.



Gambar 3.12 Use Case Diagram Utama

Dari diagram *use case* tersebut dapat dilihat bahwa terdapat dua aktor, yaitu *user* dan *admin*. Aktor *user* adalah pengguna aplikasi yang ingin menilai posisi tubuh dengan menggunakan aplikasi. Sedangkan aktor *admin* adalah aktor yang

bertanggung jawab dalam memasukkan data-data yang diperlukan agar aplikasi dapat berjalan. Terdapat empat *use case* yang masing-masing merupakan kegiatan yang memiliki tujuan masing-masing. Penjelasan mengenai masing-masing *use case* tampak pada Tabel 3.2.

Tabel 3.2 Detail *Use Case*

No.	Nama <i>Use Case</i>	Aktor	Penjelasan
1.	Memilih gerakan	<i>User</i>	Kegiatan <i>user</i> dalam menentukan gerakan yang ingin dicoba
2.	Penilaian gerakan	<i>User</i>	Kegiatan <i>user</i> dalam melakukan gerakan yang ingin dicoba
3.	Menampilkan daftar gerakan	Sistem	Bagian dari <i>use case</i> memilih gerakan yang dijalankan oleh sistem sehingga menampilkan pilihan gerakan
4.	Memasukkan data	<i>Admin</i>	Kegiatan <i>admin</i> dalam memasukkan data referensi ke dalam basis data agar dapat dipakai dalam aplikasi

3.2.4 *Flow of Events*

Flow of events merupakan gambaran dari aliran kejadian yang tersusun secara runtut. Disini akan tampak aliran sebuah aktifitas yang di lengkapi dengan bagaimana aktifitas tersebut dimulai dan bagaimana aktifitas itu berakhir. *Flow of events* merupakan penjabaran awal *user story* yang selanjutnya akan didetailkan dalam *activity diagram*. *Flow of events* dari beberapa aktifitas yang diidentifikasi pada *use case diagram* akan dibahas pada bagian selanjutnya.

a. *Flow of Event* dari *Use Case* Menampilkan Pilihan Gerakan

Use case menampilkan pilihan gerakan diawali ketika pengguna aplikasi memilih masuk pada *splash screen*. Aliran aktifitas dalam *use case* ini akan berakhir dengan kondisi akhir yaitu sistem menampilkan pilihan gerakan yang ada dalam basis data. Untuk lebih detailnya dapat dilihat pada *flow of event* di Tabel 3.3.

Tabel 3.3 *Flow of Event* dari *Use Case* menampilkan Pilihan Gerakan

Use Case Detail	Menampilkan Pilihan Gerakan		
Kebutuhan terkait	A1. Pengguna aplikasi perlu melihat pilihan-pilhan gerakan yang ditampilkan dalam program. Infotmasi yang perlu didapat dari pilihan-pilhan gerakan ini adalah nama gerakan, keterangan singkat mengenai gerakan, dan gambar posisi akhir gerakan yang disimpan dalam basis data		
Tujuan	Memberikan daftar pilihan gerakan ke pengguna		
Prasyarat	Sensor Kinect terhubung		
Kondisi akhir sukses	Pilihan gerakan ditampilkan beserta detailnya.		
Kondisi akhir Gagal	- Jika Pilihan gerakan tidak memiliki gambar dan informasi maka akan ditampilkan <i>selection display</i> yang kosong		
Aktor utama	Pengguna aplikasi		
Aktor sekunder	Sistem		
Pemicu	Pengguna aplikasi memilih untuk masuk ke aplikasi melalui <i>splash screen</i>		
Alur utama	Langkah	Aktor	Aksi
	1.	Sistem	Menampilkan form selamat datang
	2.	User	Memilih tombol mulai
	3.	Sistem	Mendeteksi dan menyalakan sensor
	4.	Sistem	Menampilkan keterangan sensor siap
	5.	Sistem	Menampilkan menu pilihan gerakan
	6.	Sistem	Mendeteksi pergerakan tangan user
	7.	Sistem	Menampilkan <i>kinect cursor</i>
Alur perluasan	4.1	Sistem	Menampilkan keterangan sensor tidak ditemukan
	6.1	Sistem	<i>Active movement detection</i> dilakukan untuk memonitoring pergerakan tangan selama form dibuka

b. *Flow of Event* dari *Use Case* Memilih Gerakan

Use case memilih gerakan diawali ketika pengguna aplikasi memilih gerakan baik menggunakan *kinect cursor* ketika ditemukan sensor Kinect maupun dengan tetikus. Aliran aktifitas dalam *use case* ini akan berakhir dengan kondisi akhir yaitu sistem menerima pilihan gerakan dari *user* dengan nilai id gerakan sesuai dengan gerakan yang dipilih. Untuk lebih detailnya dapat dilihat pada *flow of event* di Tabel 3.4.

Tabel 3.4 *Flow of Event* dari *Use Case* memilih Gerakan

Use Case	Memilih gerakan		
Detail			
Kebutuhan terkait	A1. Pengguna aplikasi mampu memilih gerakan yang diinginkan untuk dinilai menggunakan <i>hand tracking</i>		
Tujuan	Memberikan pilihan gerakan ke sistem		
Prasyarat	Sensor Kinect terhubung		
Kondisi akhir sukses	Pilihan gerakan disampaikan ke sistem berupa id gerakan		
Kondisi akhir Gagal			
Aktor utama	Pengguna aplikasi		
Aktor sekunder	Sistem		
Pemicu	Sistem menampilkan menu pilihan gerakan untuk dipilih		
Alur utama	Langkah	Aktor	Aksi
	1.	User	Mengarahkan <i>kinect cursor</i> ke info pilihan gerakan yang dipilih
	2.	Sistem	Menampilkan <i>selection display</i> dari pilihan gerakan yang dipilih selama 10 detik
	3.	User	Mendorong tangannya ke arah sensor Kinect diatas pilihan gerakan yang diinginkan
	4.	Sistem	Mendeteksi id gerakan dari pilihan gerakan yang dipilih
	5.	Sistem	Menampilkan <i>kinectWindow</i> dengan parameter id gerakan yang sudah disampaikan dari form Menu pilihan gerakan
Alur perluasan	4.1	User	Mengarahkan <i>pointer mouse</i> ke info pilihan gerakan yang dipilih
	3.1	Sistem	Memilih pilihan gerakan yang diinginkan dengan menggunakan <i>mouse</i>

c. *Flow of Event* dari *Use Case* Penilaian Gerakan

Use case penilaian gerakan diawali ketika pengguna aplikasi telah memilih gerakan yang selanjutnya akan membuka *kinect window* yang menerima id gerakan yang disampaikan oleh form sebelumnya. Aliran aktifitas dalam *use case* ini akan berakhir dengan kondisi akhir yaitu sistem menampilkan hasil penilaian yang merupakan hasil pemrosesan data. Untuk lebih detailnya dapat dilihat pada *flow of event* di Tabel 3.5.

Tabel 3.5 *Flow of Event* dari *Use Case* memilih Gerakan

Use Case	Penilaian Gerakan		
Detail			
Kebutuhan terkait	A1. Aplikasi menampilkan notifikasi mengenai proses pengambilan data yang perlu dilakukan beserta hitungan mundur. Pengambilan data dibagi menjadi tiga tipe yaitu pengambilan data penuh, pengambilan data bagian atas, dan pengambilan data bagian bawah.		
Tujuan	Memberikan Penilaian gerakan ke user		
Prasyarat	Sensor Kinect terhubung		
Kondisi akhir sukses	Nilai disampaikan ke <i>user</i>		
Kondisi akhir Gagal			
Aktor utama	Sistem		
Aktor sekunder	Pengguna aplikasi		
Pemicu	Pengguna aplikasi memilih salah satu dari pilihan gerakan yang ada		
Alur utama	Langkah	Aktor	Aksi
	1.	Sistem	Menampilkan notifikasi informasi pengambilan data dan akan hilang setelah hitungan mundur selesai
	2.	User	Melakukan gerakan yang dipilih
	3.	User	Mengakhiri gerakan dengan mengucapkan <i>finish</i>
	4.	Sistem	Mengambil nilai derajat per join
	5.	Sistem	Membandingkan nilai user dengan data referensi
	6.	Sistem	Menampilkan hasil penilaian
Alur perluasan	3.1	User	Mengakhiri gerakan pada pengambilan pertama dengan mengucapkan <i>save first</i>
	3.2	User	Mengakhiri gerakan pada pengambilan pertama dengan mengucapkan <i>save second</i>
	3.3	User	Mengakhiri proses pengambilan data dengan mengucapkan <i>combine</i>

d. *Flow of Event* dari *Use Case* Memasukkan Data

Use case memasukkan data diawali ketika admin masuk ke *window update* data referensi. Aliran aktifitas dalam *use case* ini akan berakhir dengan kondisi akhir yaitu basis data dengan id gerakan yang ingin diubah memiliki nilai yang

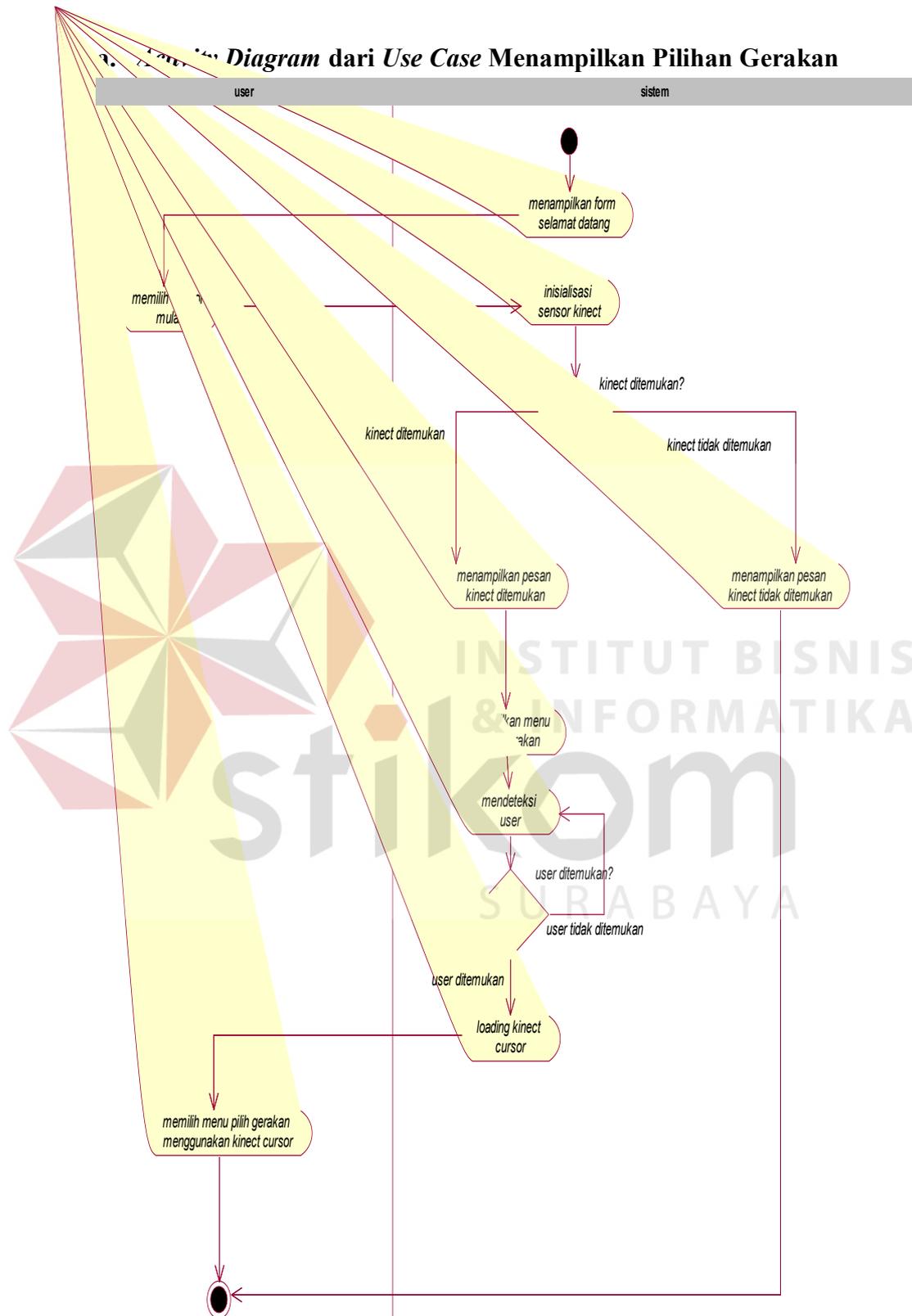
sudah berubah sesuai dengan perekaman data sudut sendi yang dilakukan. Untuk lebih detailnya dapat dilihat pada *flow of event* di Tabel 3.6.

Tabel 3.6 *Flow of Event* dari *Use Case* Memasukkan Data

Use Case	Memasukkan data		
Detail			
Kebutuhan terkait	A1. Admin dapat memasukkan data referensi posisi melalui sensor Kinect dengan Kinect skeleton tracking dan pengukuran derajat sendi		
Tujuan	Menyimpan data referensi persendian sebagai referensi		
Prasyarat	Sensor Kinect terhubung dan diinisialisasi		
Kondisi akhir sukses	Data referensi gerakan tersimpan		
Kondisi akhir Gagal	Data referensi tidak tersimpan		
Aktor utama	Admin		
Aktor sekunder	Sistem		
Pemicu	Admin masuk ke window update untuk mengubah data referensi yang sudah ada		
Alur utama	Langkah	Aktor	Aksi
	1.	Sistem	Menampilkan pilihan gerakan yang dapat diupdate datanya
	2.	Admin	Memilih gerakan yang ingin direkam datanya
	3.	Sistem	Menginisialisasi sensor Kinect
	4.	Sistem	Menggambar <i>stickmen</i> beserta sudutnya
	5.	Admin	Menekan tombol simpan
	6.	Sistem	Memasukkan data ke dalam basis data
Alur perluasan	-	-	-

3.2.5 Activity Diagram

Activity diagram merupakan representasi grafis dari alur kegiatan (*workflows*) yang menjelaskan secara mendetail dari berbagai kegiatan. Setiap *use case* memiliki *activity diagram* masing-masing. Pada bagian selanjutnya akan dijelaskan *activity diagram* per *use case* yang ada.



Gambar 3.13 Activity Diagram dari Use Case Menampilkan Pilihan Gerakan

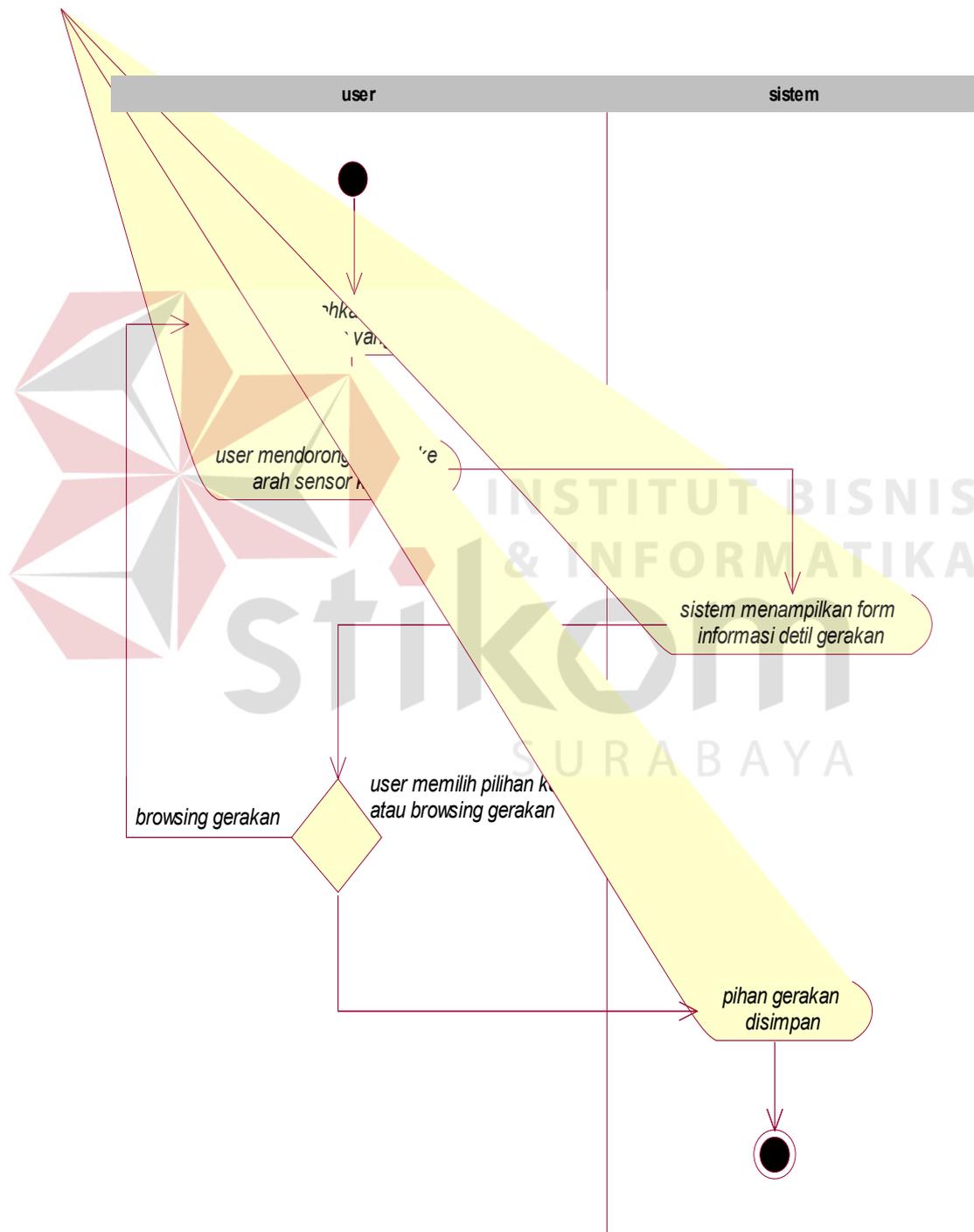
Activity diagram dari *use case* menampilkan pilihan gerakan tampak pada Gambar 3.13. Aktifitas pertama dalam diagram ini adalah menampilkan form selamat datang dan aktifitas terakhir adalah memilih menu pilihan gerakan menggunakan *kinect cursor*. Penjelasan dari *activity diagram* ini adalah sebagai berikut :

1. Aktifitas pertama diawali oleh sistem dengan menampilkan form selamat datang yang memberikan penjelasan singkat mengenai maksud dan tujuan aplikasi beserta keterangan bahwa aplikasi ini membutuhkan perangkat sensor *Kinect*.
2. *User* akan merespon dengan memilih tombol mulai yang selanjutnya akan memicu aplikasi untuk menginisialisasi sensor *Kinect*.
3. Jika sensor *Kinect* tidak ditemukan akan muncul pesan bahwa aplikasi ini membutuhkan sensor *Kinect*. Sedangkan jika ditemukan sensor *Kinect* maka akan muncul pesan yang menunjukkan proses sensor *Kinect* sudah tersambung.
4. Setelah sensor *Kinect* berhasil diinisialisasi, akan muncul menu pilihan gerakan.
5. Aplikasi akan segera mendeteksi *user* yang ada di depan sensor *Kinect*.
6. Jika ditemukan *user*, selanjutnya aplikasi akan mencari pergerakan telapak tangan *user* yang akan menjadi penggerak *kinect cursor*.
7. *User* dapat memilih gerakan yang tersedia menggunakan *kinect cursor*.

b. Activity Diagram dari Use Case Memilih Gerakan

Activity diagram dari use case memilih gerakan tampak pada Gambar 3.14.

Aktifitas pertama dalam diagram ini adalah *user* mengarahkan *kinect cursor* gerakan yang dipilih dan berakhir ketika pilihan gerakan disimpan.



Gambar 3.14 Activity Diagram dari Use Case Memilih Gerakan

Penjelasan *activity diagram* ini adalah sebagai berikut :

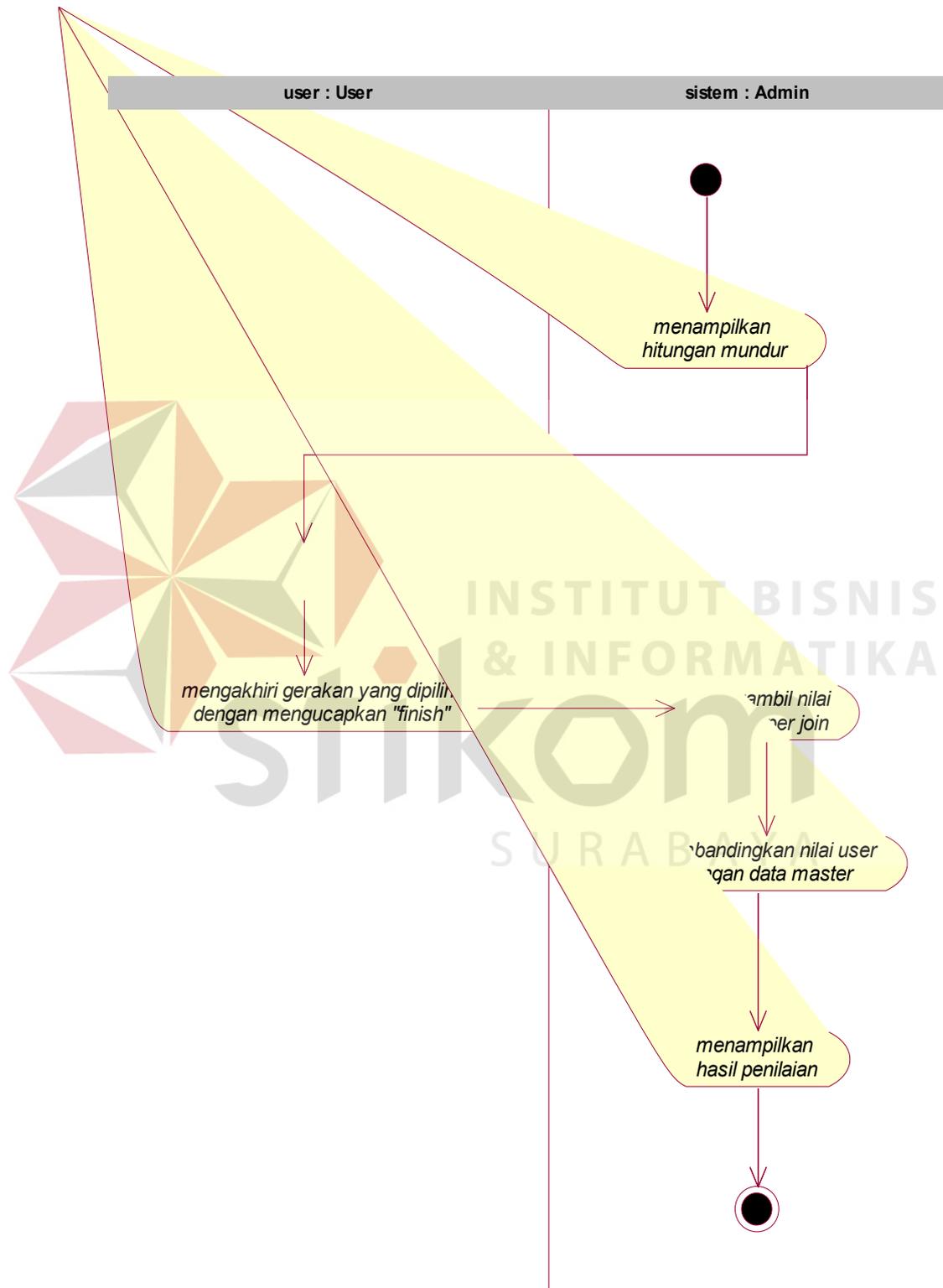
1. *User* mengarahkan kinect cursor diatas gerakan yang ingin diketahui penjelasannya.
2. Sistem akan menampilkan informasi detil gerakan.
3. Untuk memilih gerakan yang ingin dicoba, *user* mengarahkan *kinect cursor* di pilihan yang dikehendaki.
4. *User* memilih dengan mendorong telapak tangan ke arah sensor *kinect*.
5. Pilihan gerakan akan disimpan oleh aplikasi untuk mengambil parameter yang dibutuhkan

c. *Activity Diagram* dari *Use Case* Penilaian Gerakan

Activity diagram dari *use case* penilaian gerakan tampak pada Gambar 3.15. Aktifitas pertama dalam diagram ini adalah menampilkan hitungan mundur dan diakhiri dengan menampilkan hasil penilaian. Penjelasan dari *activity diagram* ini adalah sebagai berikut ;

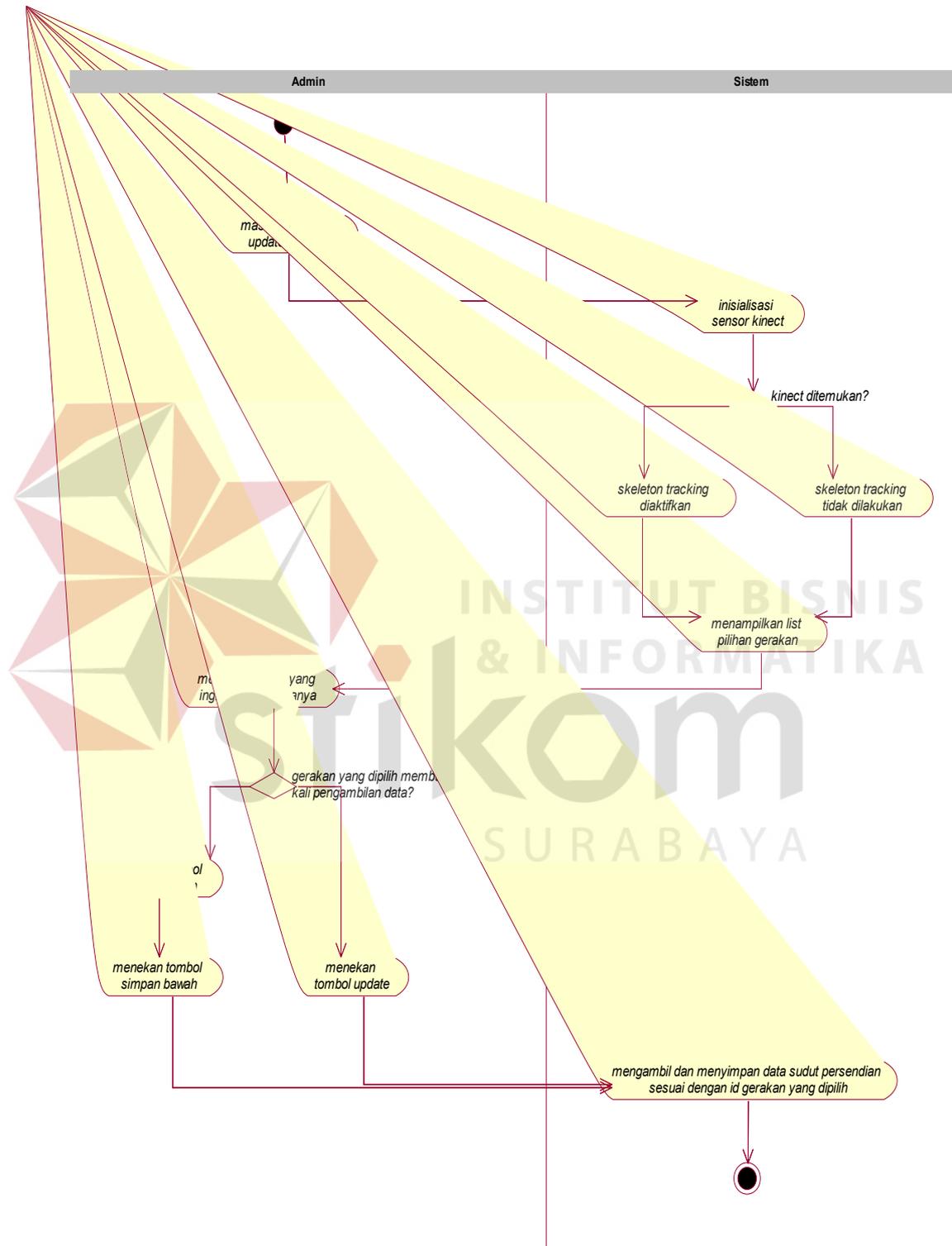
1. Setelah *user* memilih pilihan gerakan yang ingin dinilai, sistem akan memunculkan hitungan mundur sebagai waktu agar *user* siap.
2. Setelah hitungan mundur habis, *user* dapat memeragakan gerakan yang dipilih dan mempertahankan posisi akhirnya.
3. Untuk memberitahu aplikasi bahwa *user* sudah dalam posisi akhir, *user* perlu memberikan perintah suara ke sistem dengan mengucapkan “*finish*”
4. Aplikasi akan mengambil data derajat per join dari hasil *skeleton tracking*.
5. Aplikasi akan mengolah data tersebut dengan membandingkannya dengan data model yang sudah disimpan.

6. Aplikasi menampilkan hasil perbandingan data yang didapat dengan data yang sudah disimpan beserta tafsirannya.



Gambar 3.15 Activity Diagram dari Use Case Penilaian Gerakan

d. *Activity Diagram* dari *Use Case* Memasukkan Data



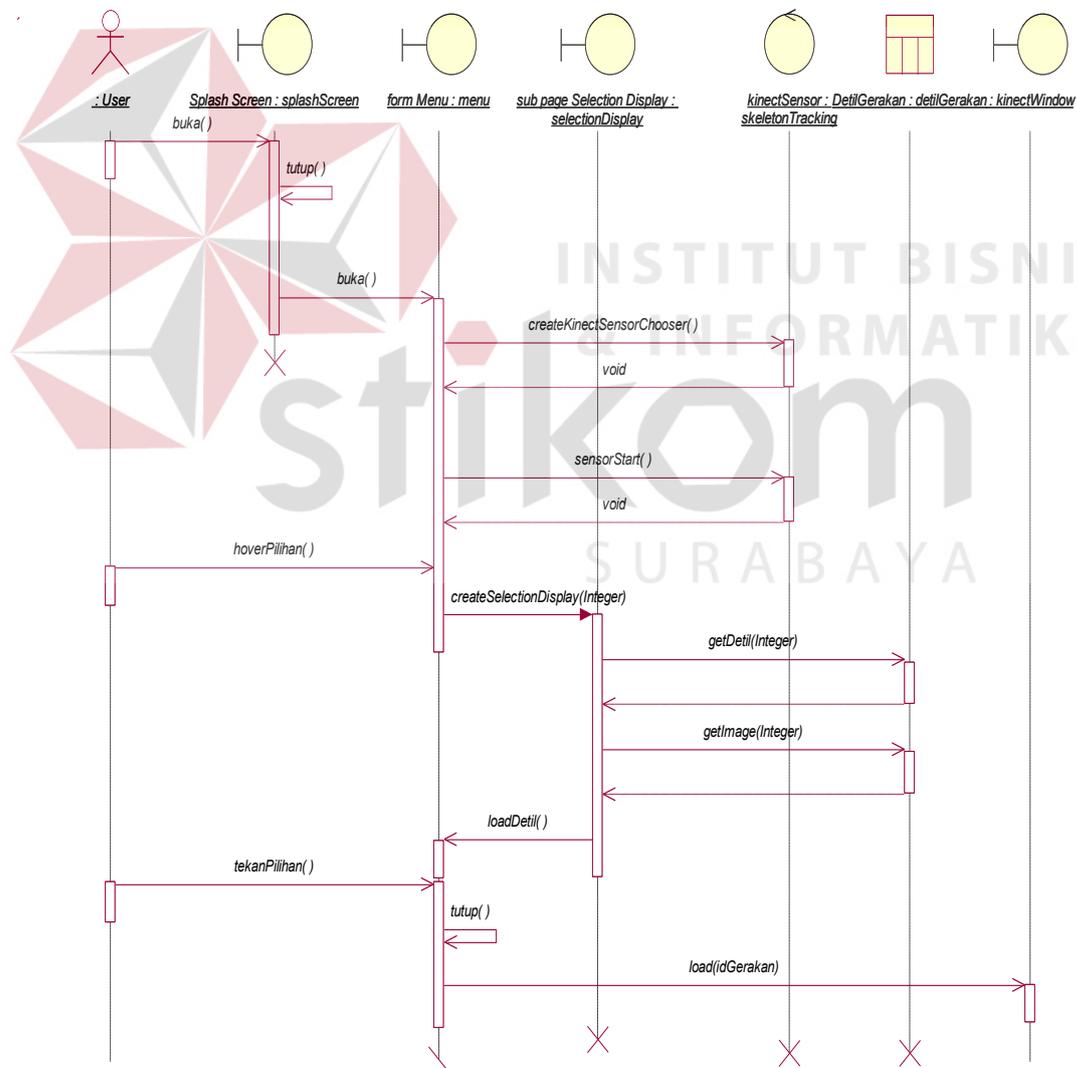
Gambar 3.16 *Activity Diagram* dari *Use Case* Memasukkan Data

3.2.6 Sequence Diagram

Sequence diagram menggambarkan jalannya suatu proses yang melibatkan *object* atau *instance* dari suatu *class*. Berikut adalah *sequence diagram* dari aplikasi ini.

a. Sequence Diagram dari Menampilkan pilihan Gerakan

Sequence diagram dari *use case* menampilkan pilihan gerakan dapat dilihat pada Gambar 3.17.



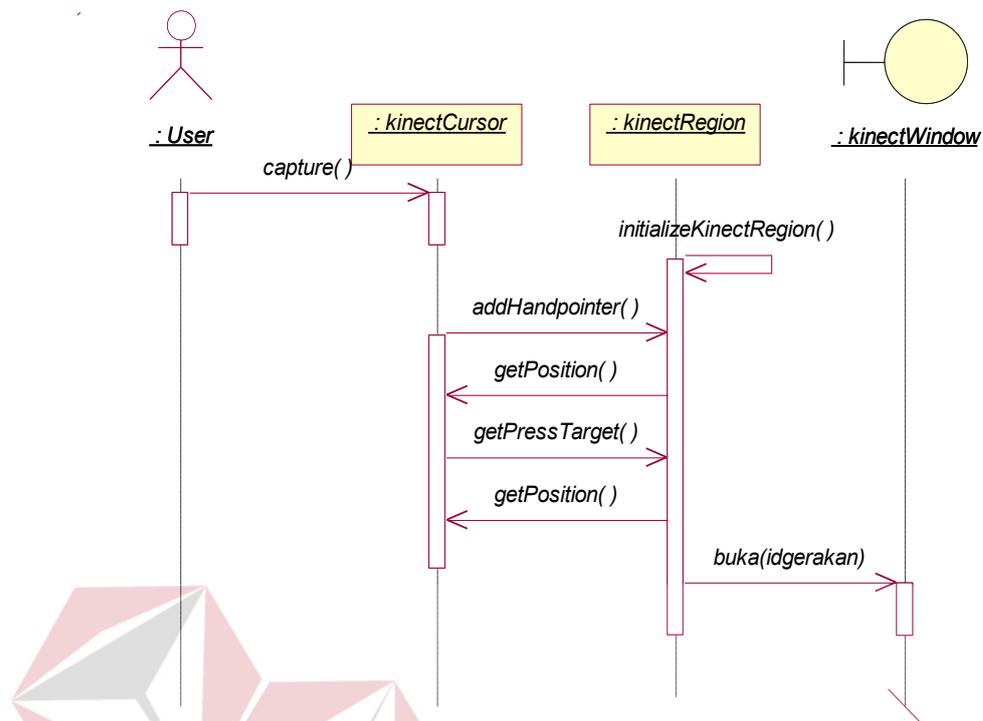
Gambar 3.17 *Sequence Diagram* dari Menampilkan Pilihan Gerakan

Penjelasan dari *sequence diagram* tersebut adalah sebagai berikut :

1. Kegiatan memilih gerakan di *trigger* ketika *user* menjalankan file *executable* dari aplikasi.
2. Aplikasi akan menampilkan *splash screen*.
3. *User* menekan tombol masuk.
4. *Splash screen* akan menutup dirinya dan memanggil form menu.
5. Form menu akan menggunakan *class* dari SDK Kinect untuk melakukan inisialisasi dalam menggunakan sensor *Kinect*.
6. Jika ditemukan sensor *Kinect* maka sensor digunakan dalam form Menu.
7. Ketika *user* meletakkan *pointer* di atas pilihan gerakan, form menu akan memanggil *Selection Display*.
8. Ketika *Selection Display* menginisialisasi untuk tampil, pada saat itu juga method *getDetil* dan *getImage* akan dipanggil yang akan memberikan data-data yang akan dimuat di halaman *Selection Display*.
9. Selanjutnya *user* dapat memilih sebuah gerakan dengan dengan cara menekan pilihan tersebut lalu form Menu akan menutup dan form *Kinect Window* akan dimuat. Form *Kinect Window* akan terbuka dengan variabel *idgerakan* yang dipilih menjadi parameternya.

b. *Sequence Diagram* dari Memilih Gerakan

Dalam aktifitas memilih gerakan akan digunakan kelas dari SDK yaitu kelas *kinectCursor* dan *KinectRegion* dalam hal mendeteksi pergerakan tangan pengguna beserta aksi yang akan dilakukan oleh pengguna, yaitu mendorong, menggenggam, dan menggerakkan kursor. *Sequence diagram* dari penilaian gerakan dapat dilihat pada Gambar 3.18.



Gambar 3.18 Sequence Diagram dari Memilih Gerakan

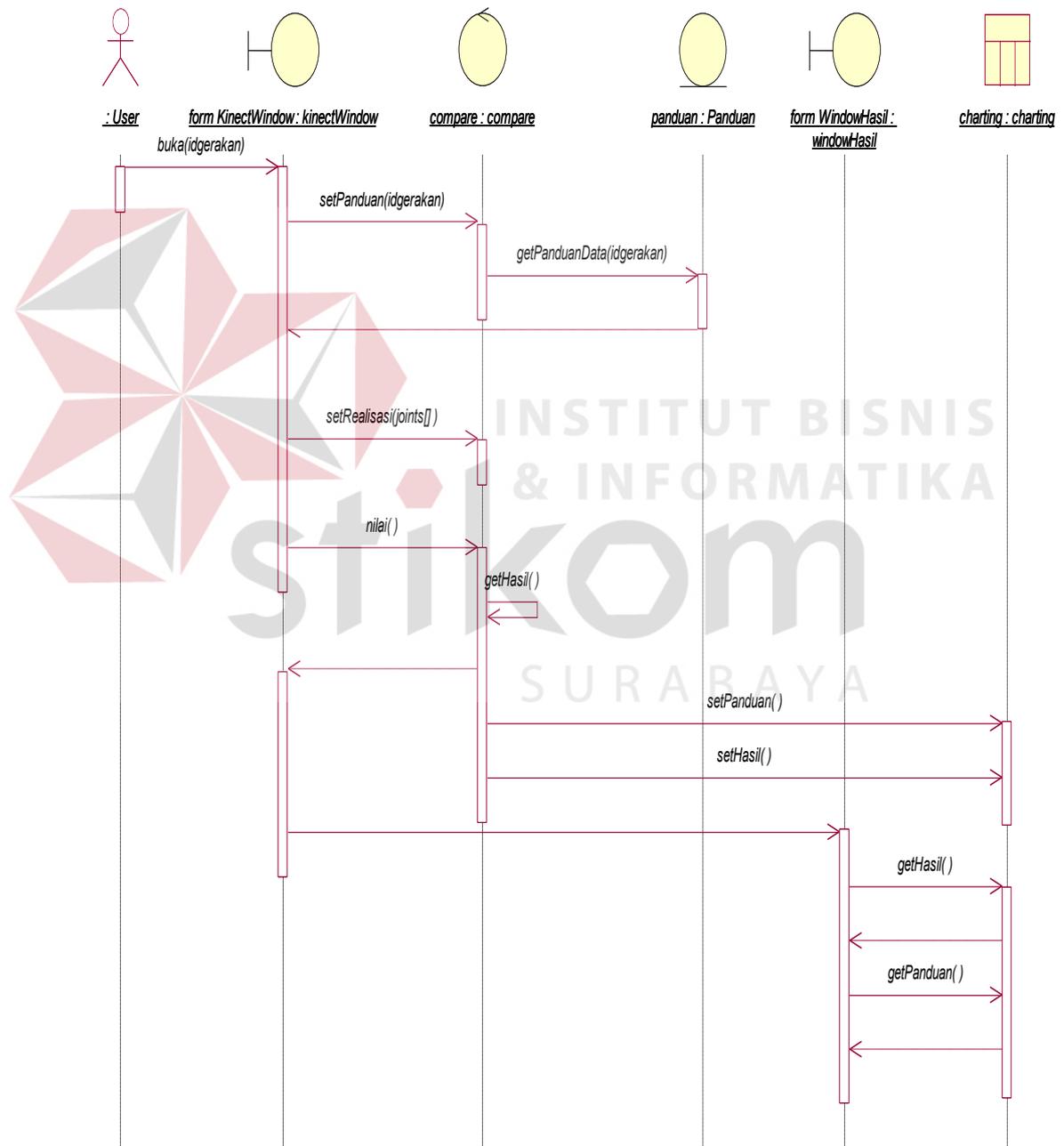
Penjelasan dari *sequence diagram* tersebut adalah sebagai berikut :

1. User menggerakkan tangannya di depan sensor sehingga aplikasi akan menangkap pergerakan tangan.
2. Pergerakan tangan dideteksi oleh program melalui kelas *KinectCursor*
3. Setelah *user* terdeteksi, aplikasi akan membuat *object hand pointer*.
4. Kelas *kinectRegion* yang berfungsi sebagai *content control* memiliki area dalam layar aplikasi dimana *hand pointer* yang sudah dibuat dapat digunakan.
5. Ketika tangan pengguna bergerak, aplikasi akan mengupdate letak *hand pointer* menggunakan metode *getPosition*.
6. Ketika tangan pengguna mengalami perubahan jarak pada sumbu z atau dapat dikatakan mendekati sensor, maka kelas *KinectCursor* akan memproses posisi

dari *handcursor* dan jika terdapat target pilihan maka target tersebut akan dieksekusi menurut *event handler*.

c. *Sequence Diagram* dari Penilaian Gerakan

Sequence diagram dari penilaian gerakan dapat dilihat pada Gambar 3.19.



Gambar 3.19 *Sequence Diagram* dari Penilaian Gerakan

Penjelasan dari *sequence diagram* tersebut adalah sebagai berikut :

1. Form *Kinect Window* dibuka atas dasar kegiatan aktifitas *user* sebelumnya dalam form Menu yang disertai dengan penyampaian parameter idgerakan.
2. Form *Kinect Window* akan mengambil nilai panduan dari gerakan yang dimaksud dan menampungnya dalam *array joint* sesuai dengan parameter idgerakan.
3. Ketika *user* dalam keadaan stabil dan mengucapkan *voice command* untuk memerintahkan aplikasi untuk masuk ke tahap penilaian maka akan dipanggil metode *setRealisasi* yang akan menjadi *array joint* yang akan dibandingkan.
4. Setelah itu akan dipanggil method nilai yang akan melakukan perhitungan untuk membandingkan dua data yang sudah didapat sebelumnya untuk mendapatkan hasil melalui metode *getHasil* dalam kelas *compare*.
5. Kelas *compare* juga akan memanggil metode *setHasil* dan *setPanduan* untuk menampung data ke tabel *charting* untuk membantu proses penyajian data.
6. Lalu form *Kinect Window* akan memanggil form *Window Hasil* dimana sebelum dimuat akan mencari data dalam tabel *charting* yang digunakan untuk penyajian data penilaian.

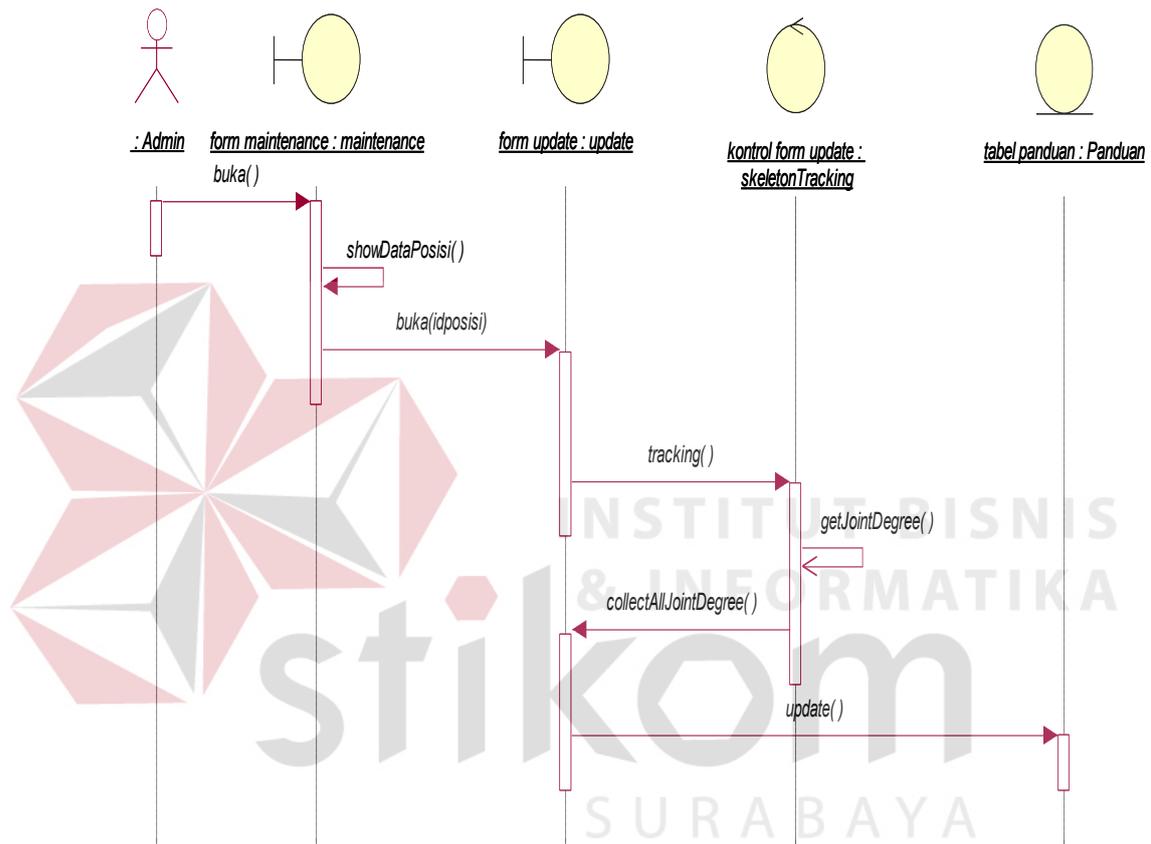
d. *Sequence Diagram* Memasukkan Data

Sequence diagram dari penilaian gerakan dapat dilihat pada Gambar 3.20.

Penjelasan dari *sequence diagram* ini adalah sebagai berikut :

1. *Admin* membuka form *maintenance*
2. Form *maintenance* akan memuat data panduan dari tabel panduan
3. *Admin* memasukkan idposisi yang ingin diubah lalu form *maintenance* akan memanggil form *update*

4. Form *update* akan menginisialisasi kelas *skeletonTracking* dari kelas yang ada di *Kinect SDK*
5. *Admin* dapat menyimpan data derajat per join dari pembacaan sensor *Kinect* ke tabel panduan setelah memilih tombol *update*.

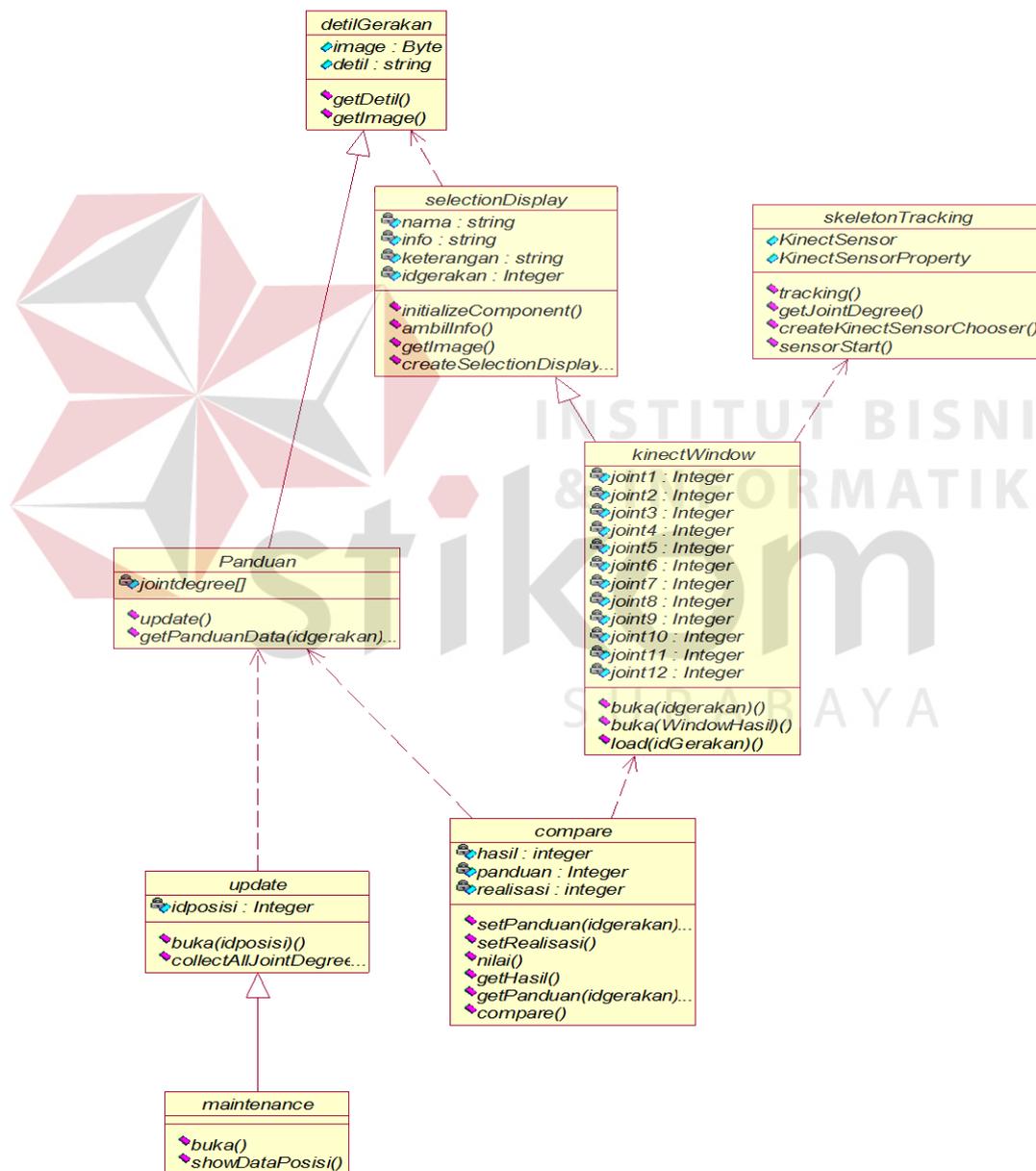


Gambar 3.20 *Sequence Diagram* dari Memasukkan Data

3.2.7 Class Diagram

Berdasarkan rancangan pemodelan diatas, maka dibutuhkan beberapa kelas untuk dapat menjalankan fungsi-fungsi yang ada dalam aplikasi ini. Kelas – kelas tersebut tampak pada *class diagram* pada Gambar 3.13. *Class diagram* ini berisi kelas-kelas utama yang dibutuhkan beserta hubungan antar kelas. Tiap kelas

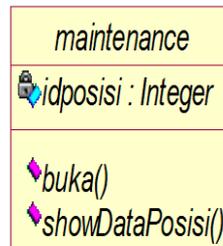
diwakili oleh sebuah kotak yang terbaki menjadi 3 kolom yaitu nama kelas, atribut kelas dan metode kelas. Masing-masing kelas tersebut memiliki relasi ke kelas yang lainnya. Selain kelas-kelas yang ada pada diagram kelas yang dapat dilihat pada Gambar 3.21, aplikasi ini juga akan menggunakan kelas-kelas yang disediakan oleh SDK *Kinect*.



Gambar 3.21 Class Diagram Aplikasi Penilaian Posisi Karate

a. Kelas *Maintenance*

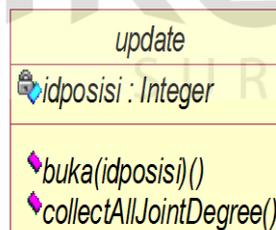
Kelas *Maintenance* digunakan dalam proses awal update data panduan pada form *maintenance*. Fungsi utamanya adalah menampilkan data posisi. Struktur kelas ini dapat dilihat pada Gambar 3.22.



Gambar 3.22 Kelas *Maintenance*

b. Kelas *Update*

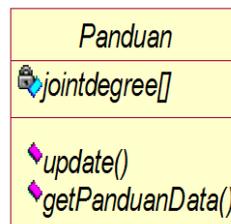
Kelas *Update* digunakan dalam proses update data panduan. Fungsi utamanya mengubah data pada database yang memiliki id posisi yang sama dengan atribut dari kelas ini dengan hasil tangkapan sensor *Kinect* yang didapat dari metode *collectAllJointDegree*. Struktur kelas ini dapat dilihat pada Gambar 3.23.



Gambar 3.23 Kelas *Update*

c. Kelas *Panduan*

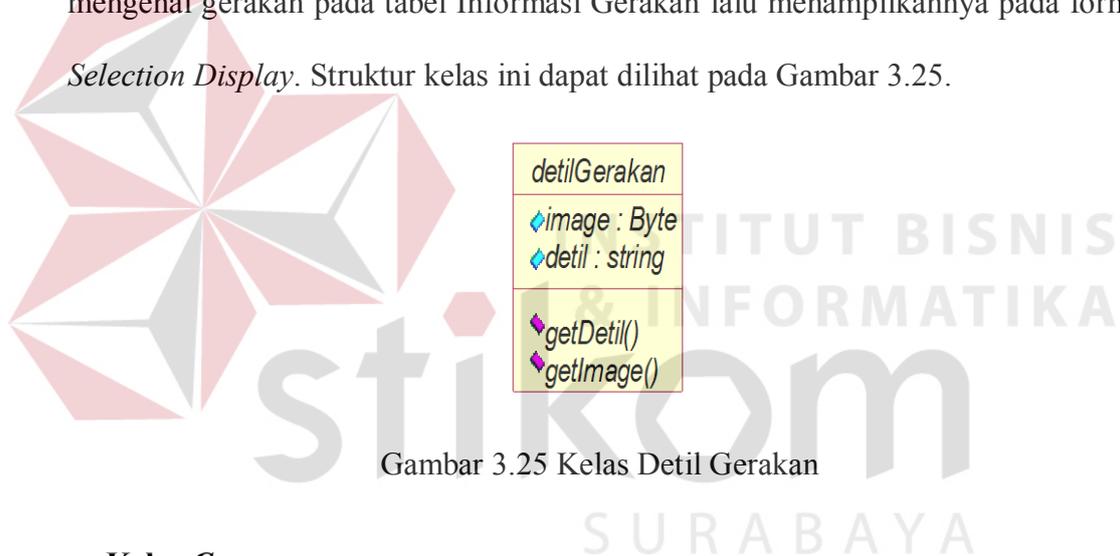
Kelas *Panduan* digunakan dalam proses mengambil data panduan. Fungsi utamanya mengambil data pada tabel panduan lalu menampungnya pada sebuah *array*. Struktur kelas ini dapat dilihat pada Gambar 3.24.



Gambar 3.24 Kelas Panduan

d. Kelas Detil Gerakan

Kelas Detil Gerakan digunakan dalam proses mengambil data informasi gerakan pada database. Fungsi utamanya mengambil data gambar dan keterangan mengenai gerakan pada tabel Informasi Gerakan lalu menampilkannya pada form *Selection Display*. Struktur kelas ini dapat dilihat pada Gambar 3.25.



Gambar 3.25 Kelas Detil Gerakan

e. Kelas Compare

Kelas *Compare* digunakan dalam proses perbandingan data dari derajat join dari hasil pembacaan sensor *Kinect* dengan derajat joint dari tabel Panduan. Fungsi utamanya mengambil data panduan posisi dari basis data, hasil bacaan sensor *Kinect* dan memprosesnya. Struktur kelas ini dapat dilihat pada Gambar 3.26.

<i>compare</i>
hasil : integer
panduan : Integer
realisasi : integer
setPanduan()
setRealisasi()
nilai()
getHasil()
getPanduan()
compare()

Gambar 3.26 Kelas *Compare*

f. Kelas Kinect Window

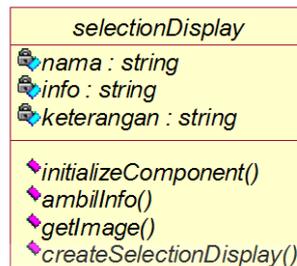
Kelas *kinectWindow* digunakan dalam proses inisialisasi form *Kinect Window* beserta kegiatan yang terjadi di form ini. Struktur kelas ini dapat dilihat pada Gambar 3.27.

<i>kinectWindow</i>
joint1 : Integer
joint2 : Integer
joint3 : Integer
joint4 : Integer
joint5 : Integer
joint6 : Integer
joint7 : Integer
joint8 : Integer
joint9 : Integer
joint10 : Integer
joint11 : Integer
joint12 : Integer
buka(idgerakan())
buka(WindowHasil())
load(idGerakan())

Gambar 3.27 Kelas *Kinect Window*

g. Kelas Selection Display

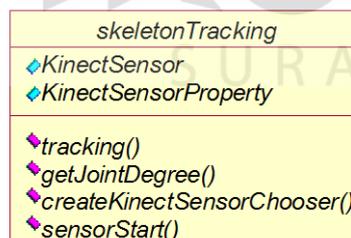
Kelas *selectionDisplay* digunakan dalam proses form Menu memuat detail gerakan ke halaman *Selection Display*. Fungsi utamanya adalah mengambil data dari tabel Informasi Gerakan. Struktur kelas ini dapat dilihat pada Gambar 3.28.



Gambar 3.28 Kelas *Selection Display*

h. Kelas Skeleton Tracking

Kelas *skeletonTracking* digunakan dalam proses inialisasi sensor *Kinect*. Fungsi utamanya menyalakan sensor *Kinect* untuk menjalankan skeleton tracking. Struktur kelas ini dapat dilihat pada Gambar 3.29.



Gambar 3.29 Kelas *Skeleton Tracking*

Kelas ini memiliki atribut yang berasal dari *enumeration* dengan struktur yang tampak pada Gambar 3.30.

<<enumeration>> KinectSensor	<<enumeration>> KinectSensorProperty
<ul style="list-style-type: none"> Activated ActivatedDetected NotActive 	<ul style="list-style-type: none"> AudioSource ColorStream CoordinateMapper DepthFilter DeviceConnectionID ElevationAngle ForcedInfraredEmitter IsRunning KinectSensors MaxElevationAngle minElevationangle SkeletonStream Status UniqueKinectId

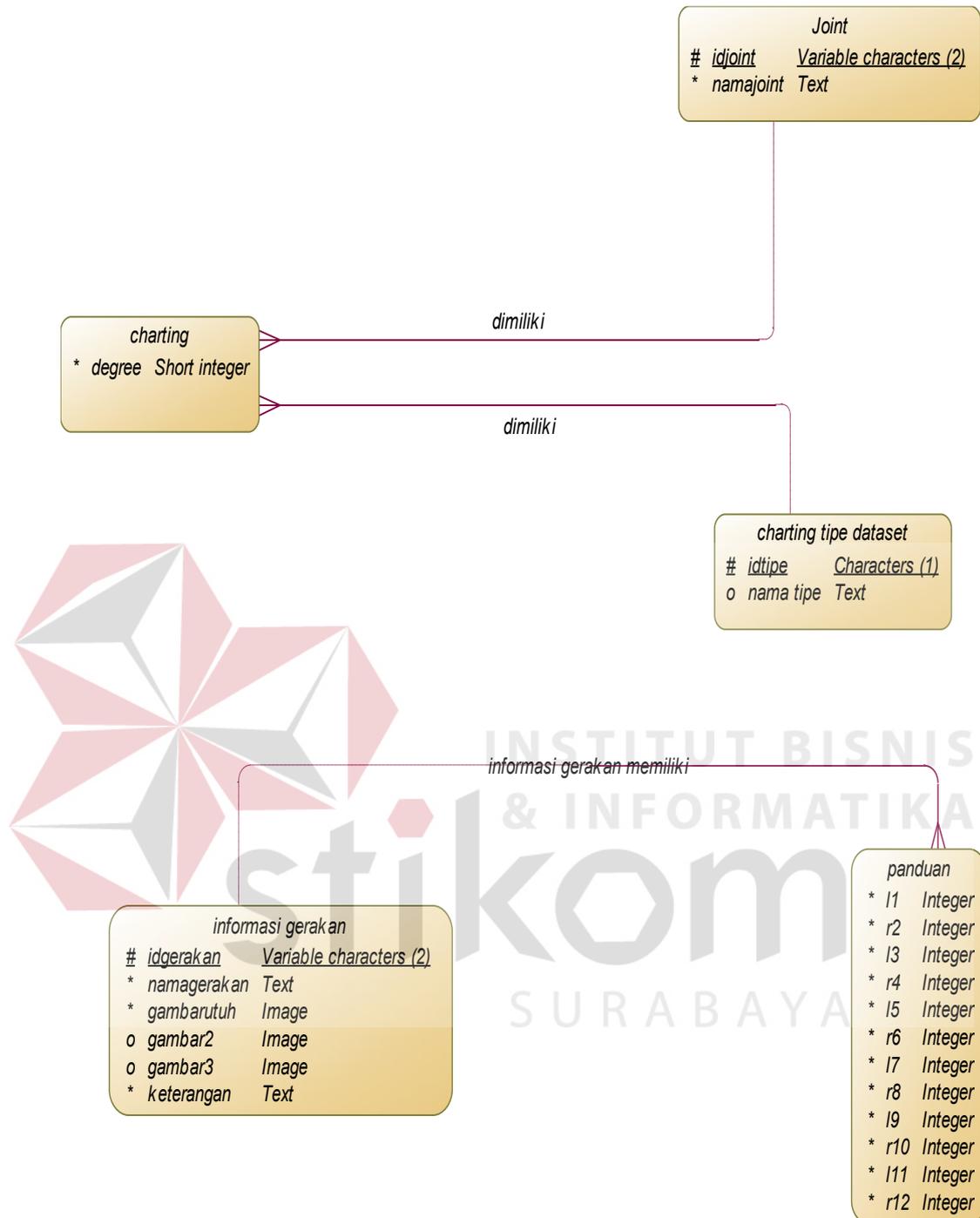
Gambar 3.30 Struktur Properti dari Kelas *Skeleton Tracking*

3.2.8 Struktur Tabel

Aplikasi ini memerlukan sebuah basis data yang mampu menyediakan berbagai data yang diperlukan dalam menjalankan berbagai fungsi. Basis data ini terdiri dari beberapa tabel yang sesuai dengan kebutuhan aplikasi ini. Perancangan basis data ini dilakukan dengan pemodelan data secara konseptual yang menghasilkan diagram *Contextual Data Diagram* (CDM) lalu diolah lagi menjadi bentuk fisik yang sesuai dengan target basis data (Microsoft SQL Server 2008) yang digunakan yang menghasilkan *Physical Data Model* (PDM). Pada bagian akhir akan dijelaskan mengenai kamus data dari struktur tabel yang digunakan.

a. CDM

Diagram CDM dari struktur tabel yang akan digunakan tampak pada Gambar 3.31.

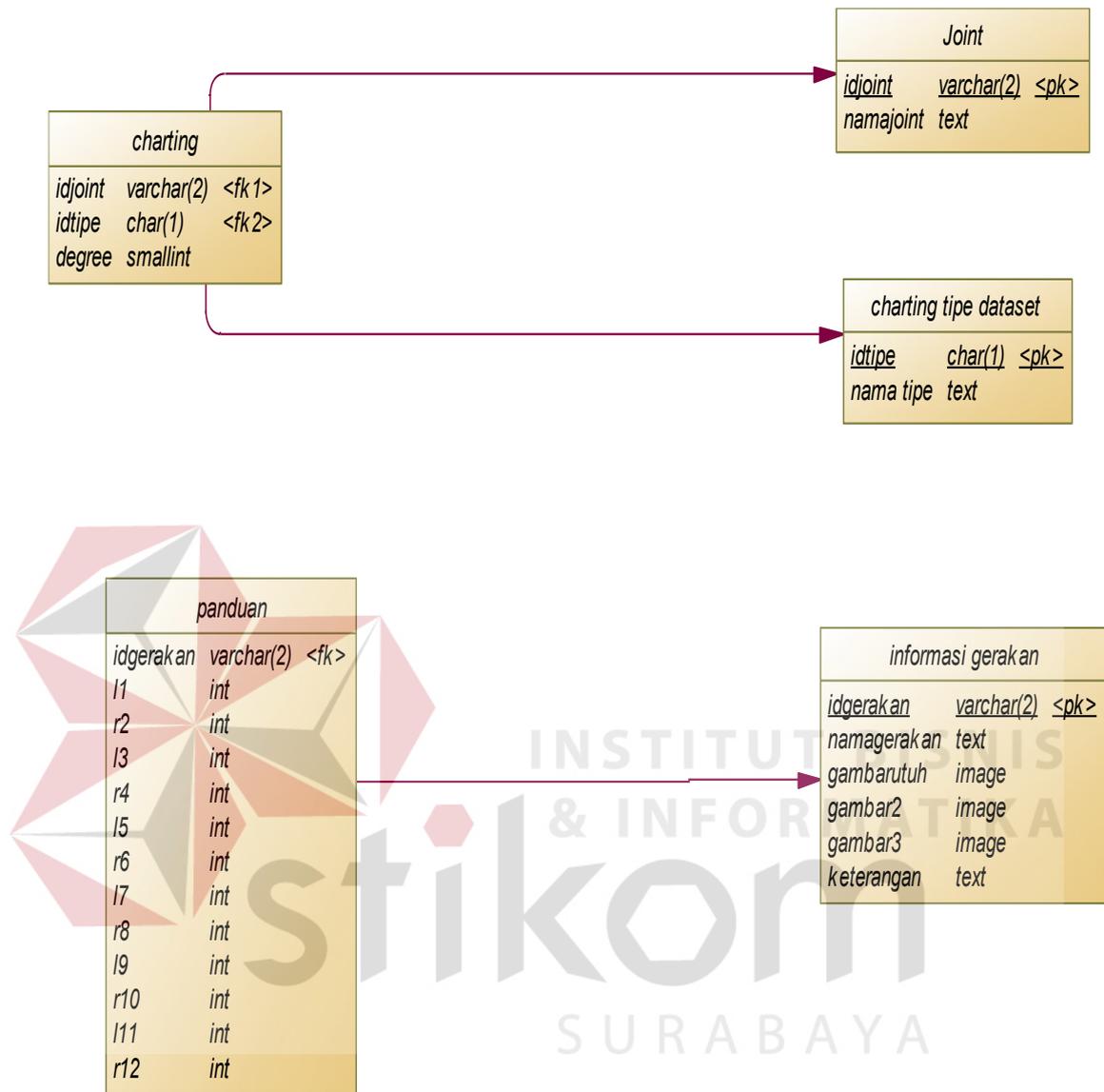


Gambar 3.31 *Contextual Data Model* dari Aplikasi Penilaian Posisi Karate

b. PDM

Diagram PDM dari struktur tabel yang akan digunakan tampak pada

Gambar 3.32.



Gambar 3.32 *Physical Data Model* dari Aplikasi Penilaian Posisi Karate

c. Kamus Data

Dalam pengembangan aplikasi ini, digunakan beberapa tabel untuk menyimpan berbagai data penting. Tabel tersebut adalah tabel *Joint*, tabel *Charting*, tabel *Tipe Dataset*, tabel *Panduan* dan tabel *Informasi Gerakan*. Berikut adalah penjelasan tentang tabel-tabel yang dipakai beserta kamus data dari tabel yang bersangkutan.

c.1 Tabel Joint

Primary key : idjoint

Foreign key : -

Fungsi :

menyimpan nama-nama joint yang ada

Struktur tabel :

Tabel 3.7 Kamus Data Tabel *Joint*

Nama	Tipe Data	Mandatory	Keterangan
<u>idjoint</u>	<i>varchar(2)</i>	Ya	-
namajoint	<i>text</i>	Ya	-

c.2 Tabel Charting

Primary key : -

Foreign key : idjoint, idtipe

Fungsi :

menampung panduan derajat dari posisi yang dipilih dan derajat dari hasil pembacaan sensor kinect yang dikelompokkan menurut idtipe

Struktur tabel :

Tabel 3.8 Kamus Data Tabel *Charting*

Nama	Tipe Data	Mandatory	Keterangan
idjoint	<i>varchar(2)</i>	Ya	-
idtipe	<i>char(1)</i>	Ya	-
degree	<i>integer</i>	Ya	Derajat dari perjoin

c.3 Tabel Tipe Dataset

Primary key : idtipe

Foreign key : -

Fungsi :

menampung tipe dari jenis data yang akan disimpan ke tabel *charting*, yaitu panduan atau hasil

Struktur tabel :

Tabel 3.9 Kamus Data Tabel Tipe Dataset

Nama	Tipe Data	<i>Mandatory</i>	Keterangan
idtype	<i>char(1)</i>	Ya	-
nama tipe	<i>text</i>	Ya	Tipe dibagi menjadi dua yaitu tipe hasil dan tipe panduan

c.4 Tabel Panduan

Primary key : -

Foreign key : idgerakan

Fungsi :

Menampung derajat dari posisi akhir gerakan yang disimpan berupa susunan 12 joint.

Struktur tabel :

Tabel 3.10 Kamus Data Tabel Panduan

Nama	Tipe Data	<i>Mandatory</i>	Keterangan
idgerakan	<i>varchar(2)</i>	Ya	
l1	<i>integer</i>	Ya	Bahu kiri
r2	<i>integer</i>	Ya	Bahu kanan
l3	<i>integer</i>	Ya	Siku kiri
r4	<i>integer</i>	Ya	Siku kanan
l5	<i>integer</i>	Ya	Pergelangan tangan kiri
r6	<i>integer</i>	Ya	Pergelangan tangan kanan
l7	<i>integer</i>	Ya	Panggul kiri
r8	<i>integer</i>	Ya	Panggul kanan
l9	<i>integer</i>	Ya	Lutut kiri
r10	<i>integer</i>	Ya	Lutut kanan
l11	<i>integer</i>	Ya	Pergelangan kaki kiri
r12	<i>integer</i>	Ya	Pergelangan kaki kanan

c.5 Tabel Informasi Gerakan

Primary key : idgerakan

Foreign key : -

Fungsi :

Menampung keterangan yang akan ditampilkan di kelas *selectionDisplay()* yang terdiri dari nama gerakan, gambar utama, gambar tambahan kedua, gambar tambahan ketiga dan keterangan mengenai gerakan tersebut.

Struktur tabel :

Tabel 3.11 Kamus Data Tabel Informasi Gerakan

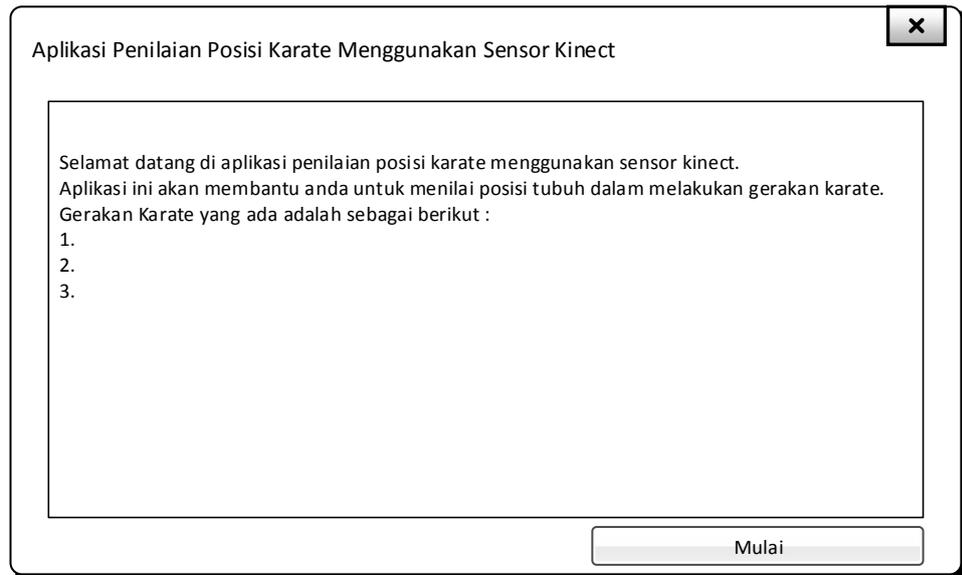
Nama	Tipe Data	Mandatory	Keterangan
<u>idgerakan</u>	<i>varchar(2)</i>	Ya	
namagerakan	<i>integer</i>	Ya	
gambarutuh	<i>image</i>	Ya	
gambar2	<i>image</i>	Ya	
gambar3	<i>image</i>	Ya	
keterangan	<i>text</i>	Ya	

3.2.9 Desain User Interface

Tampilan *user interface* yang didesain diarahkan ke arah minimalis dan sederhana. Desain ini sangat diperlukan untuk membantu dalam proses pengembangan aplikasi. Dalam aplikasi ini ada beberapa form utama yaitu :

a. Desain *Splash Screen*

Form *Splash Screen* adalah halaman awal mengenai aplikasi yang dibuat. Pengguna aplikasi dapat melihat keterangan mengenai aplikasi dan gerakan yang dapat dinilai. Perancangan tampilan untuk form *Splash Screen* ini dapat dilihat pada Gambar 3.33.



Gambar 3.33 Desain Form *Splash Screen*

Pada Form ini pengguna aplikasi dapat menutup program dengan menggunakan *close button* yang disediakan di kanan atas form. Pengguna aplikasi dapat menuju form selanjutnya dengan memilih tombol mulai.

b. Desain Form Menu

Form Menu adalah halaman aplikasi yang menampilkan seluruh daftar gerakan yang dapat dinilai oleh aplikasi ini. Mulai dari form ini, sensor Kinect dideteksi dan digunakan untuk menangkap pergerakan tangan pengguna aplikasi sebagai masukan dari penggerak *kinect cursor*. Terdapat *kinect tile button* yang dapat dipilih menggerakkan tetikus secara normal ataupun menggunakan pergerakan tangan yang sudah terdeteksi oleh sensor Kinect. Dalam menggunakan *kinect cursor*, pengguna dapat memilih sebuah gerakan dengan cara mendorong tangannya mendekati sensor Kinect. Perancangan tampilan untuk form Menu ini dapat dilihat pada Gambar 3.34.

Aplikasi Penilaian Posisi Karate Menggunakan Sensor Kinect

Silahkan memilih gerakan yang ingin dinilai :

1	2	3	4

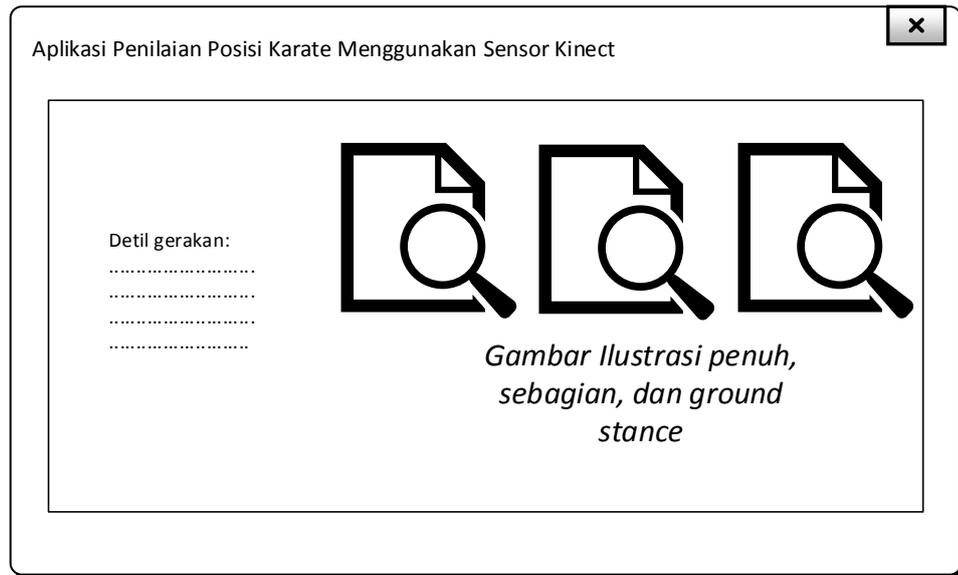
Navigation: Left arrow, Right arrow

Gambar 3.34 Desain Form Menu

Pada form ini terdapat *Kinect vertical scroll* yang berguna untuk menjelajah pilihan gerakan yang ada di halaman selanjutnya. *Kinect vertical scroll* ini dapat digunakan dengan cara meletakkan *Kinect cursor button* di atas *Kinect vertical scroll*, setelah itu secara otomatis daftar pilihan akan bergerak ke kiri atau kanan sesuai dengan *Kinect vertical scroll* yang dipilih.

c. Desain Form *Selection Display*

Form *Selection Display* adalah halaman aplikasi yang menampilkan detail gerakan yang telah dipilih pengguna aplikasi dari form sebelumnya, form Menu. Di dalam form ini akan ditampilkan detail singkat gerakan beserta gambar yang menjelaskan posisi tubuh yang benar. Perancangan tampilan untuk form *Selection Display* dapat dilihat di Gambar 3.35.

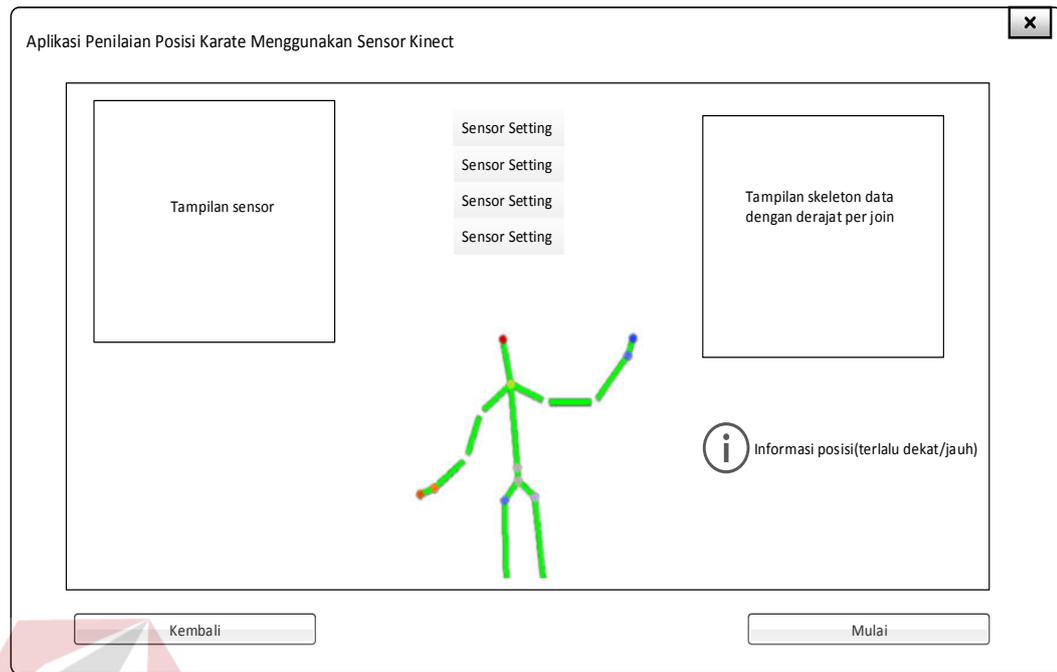


Gambar 3.35 Desain Form *Selection Display*

Di dalam form *Selection Display* ini terdapat dua tombol, yaitu tombol kembali dan tombol mulai. Form ini merupakan sub form dari Menu dimana akan tampil jika pengguna aplikasi meletakkan kursor tetikus maupun *hand cursor* diatas informasi gerakan. Tampilan form ini akan ditampilkan selama 10 detik dan akan hilang secara otomatis setelah 10 detik.

d. Desain Form *Kinect Window*

Form *Kinect Window* adalah form yang menampilkan proses dan data penting yang akan diproses oleh aplikasi dalam proses penilaian posisi tubuh. Di dalam form ini akan ditampilkan hasil penangkapan sensor dan *skeleton* yang sudah terdeteksi. Dari *skeleton* yang sudah terdeteksi akan ditampilkan data tambahan berupa derajat *skeleton line per joint*. Form ini mutlak membutuhkan sensor *Kinect*, jika sensor *Kinect* tidak ditemukan maka muncul pesan kesalahan bahwa sensor *Kinect* tidak ditemukan. Perancangan tampilan untuk form *Kinect Window* ini dapat dilihat pada Gambar 3.36.

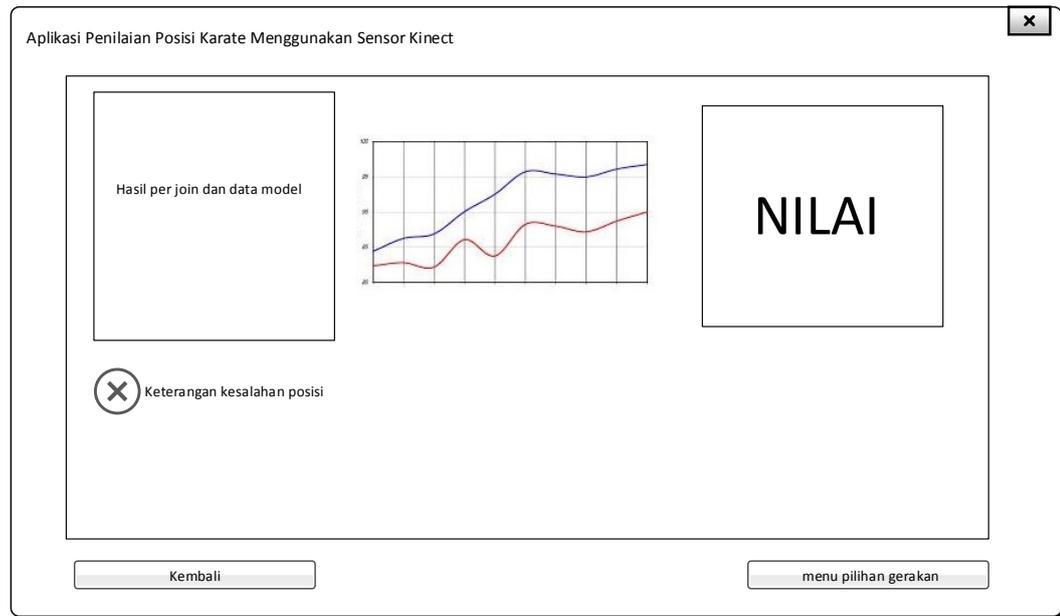


Gambar 3.36 Desain Form *Kinect Window*

Akan muncul peringatan jika pengguna aplikasi tidak berada dalam jarak ideal dari sensor Kinect berupa garis merah dalam tampilan sensor di kanan atas. Peringatan ini akan hilang ketika *skeleton* pengguna berhasil dilacak oleh sensor Kinect.

e. Desain Form Window Hasil

Form Window Hasil adalah form yang menampilkan informasi tentang proses penilaian yang sudah dilakukan aplikasi. Form ini akan menampilkan data perbandingan dari data pengguna aplikasi dan data dari posisi yang sudah tersimpan. Di dalam form ini akan disajikan data tabular dari data perbandingan dan data berupa grafik untuk melengkapi. Hasil penilaian posisi direpresntasikan melalui nilai persentase yang ada di form ini. Perancangan tampilan untuk form Hasil dapat dilihat pada Gambar 3.37.



Gambar 3.37 Desain Form Window Hasil

Form ini memiliki dua tombol yaitu kembali dan menu pilihan gerakan. Tombol kembali berfungsi untuk kembali melakukan penilaian pada posisi yang dipilih. Sedangkan tombol kembali ke menu pilihan gerakan berfungsi untuk menampilkan form Menu sehingga pengguna aplikasi bisa memilih gerakan lain.

3.3 Rancangan Pengujian Sistem

Pengujian sistem akan dilaksanakan dengan dua metode yaitu *Blackbox* dan yang fokus pada *correctness testing*. Uji coba dengan metode *Blackbox* akan memastikan fungsionalitas aplikasi berjalan sesuai dengan yang diharapkan tanpa melihat struktur internal (Khan, 2010). Pelaksanaan uji coba sistem ini akan mengacu ke rencana tes yang akan dijelaskan di bagian berikutnya.

3.3.1 Rancangan Pengujian dengan Metode Blackbox

a. Test Case Fungsi Memilih Gerakan

Tabel 3.12 Test Case Fungsi Memilih Gerakan

Keterangan Use Case	Flow	Hasil yang diharapkan
Nama Memilih gerakan Deskripsi Fitur ini adalah fitur yang berguna bagi pengguna dalam memilih gerakan yang ingin dinilai	Main Flow	
	Aktor	Sistem
	Memilih Gerakan dengan menggunakan Sensor Kinect <ol style="list-style-type: none"> 1. Menjalankan aplikasi 2. Menekan tombol masuk 3. Inisialisasi sensor Kinect 4. Menggerakkan telapak tangan di depan sensor 5. Menggengam dan menggeser tangan di pilihan "heishoku" 6. Meletakkan kinect hand cursor diatas informasi detail "heishoku" 7. Memilih gerakan "heishoku" dengan cara mendorong telapak tangan mendekati sensor 	H1. <i>Splash screen</i> ditampilkan H2. Menu Pilihan ditampilkan H3. Sensor Kinect menyala & tersambung H4. <i>Kinect hand cursor</i> ditampilkan H5. Pilihan gerakan akan bergeser ke arah tangan bergeser dengan patokan tile "heishoku" H6. Selection display dari "heishoku" ditampilkan H7. <i>Kinect Window</i> dengan parameter idgerakan "heishoku" ditampilkan
	Alternate Flow	
Memilih Gerakan tanpa menggunakan Sensor Kinect <ol style="list-style-type: none"> 1. Menjalankan aplikasi 2. Menekan tombol masuk 3. Inisialisasi sensor Kinect 4. Meletakkan cursor diatas informasi detail "heishoku" 	H1. <i>Splash screen</i> ditampilkan H2. Menu Pilihan ditampilkan H3. Tidak ditemukan sensor H4. Selection display dari "heishoku" ditampilkan	

Keterangan Use Case	Flow	Hasil yang diharapkan
	5. Meletakkan cursor diatas tombol panah kanan 6. Memilih gerakan dengan cara mengklik <i>tile</i> "heishoku"	H5. Pilihan gerakan akan bergeser ke kanan H6. <i>Kinect Window</i> dengan parameter "heishoku" ditampilkan

b. *Test Case* Fungsi Penilaian gerakan

Tabel 3.13 *Test Case* Fungsi Penilaian Gerakan

Keterangan Use Case	Flow	Hasil yang diharapkan
Nama Penilaian Gerakan Deskripsi Fitur ini adalah fitur yang berguna bagi pengguna dalam proses penilaian	Main Flow	
	Aktor	Sistem
	Menilai Gerakan	
	1. Memilih gerakan "heishoku"	H1. <i>Kinect Window</i> ditampilkan dengan nama gerakan "heishoku" ditampilkan
	2. Inisialisasi sensor Kinect	H2. Sensor Kinect menyala
3. Berada di posisi dimana sensor Kinect dapat membaca pergerakan tubuh	H3. Skeleton tracking ditampilkan	
4. Melakukan gerakan dan pada akhir gerakan mengucapkan "finish"	H4. Window Hasil <i>penilaian heishoku</i> ditampilkan	
Alternate Flow		
Tidak ada		

c. *Test Case* Fungsi Pembaruan DataTabel 3.14 *Test Case* Fungsi Pembaruan Data

Keterangan Use Case	Flow	Hasil yang diharapkan
Nama Memasukkan data Deskripsi Fitur ini adalah fitur yang berguna bagi Admin dalam melakukan perubahan data model dengan memasukkan data baru	Main Flow	
	Aktor	Sistem
	Update Data Gerakan	
	1. Menekan tombol Ctrl+H di saat yang bersamaan	H1. Form Data ditampilkan
	2. Memilih gerakan "heishoku" untuk diupdate	H2. Data <i>joint</i> ditampilkan di <i>data gridview</i>
3. Sensor Kinect dinisialisasi	H3. Sensor Kinect menyala & tersambung	
4. Hasil baca sensor disampaikan ke aplikasi	H4. Data 12 <i>joint</i> ditampilkan	
5. Admin memilih tombol simpan	H5. Tabel Panduan dengan nama "heishoku" ter-update	
Alternate Flow		
Tidak ada		

