

# Prototype Penggunaan Kinect Untuk Aplikasi Pembelajaran Nada

Yosafat Yuwono

Jurusan Teknik Elektro, Universitas Kristen Petra

Jl. Siwalankerto 121-131, Surabaya

E-mail: m23408027@john.petra.ac.id

**Abstrak**— Sensor Kinect adalah salah satu sensor yang memiliki kemampuan untuk mendeteksi pemain serta gerakan-gerakan pemain. Sensor Kinect cocok sebagai *controller* untuk game-game simulator. Salah satu game simulator yang bisa digunakan untuk pembelajaran adalah program simulator pembelajaran nada menggunakan piano *virtual*. Oleh karena itu dalam tugas akhir ini akan dibuat aplikasi pembelajaran nada menggunakan piano *virtual* berbasis Kinect. Aplikasi ini akan terhubung dengan server, di mana server ini akan menyimpan data MIDI, pemain dan skor. Server ini diperlukan untuk menyimpan catatan skor pemain dan perkembangan pemain. Sebagai hasilnya, aplikasi telah dapat memainkan file MIDI, melakukan koneksi ke server, dan menggunakan Kinect sebagai *controller*. Aplikasi juga sudah berhasil mencatat skor hasil permainan dan menjalankan musik MIDI dan animasi dengan sinkron.

**Kata Kunci**—Kinect, Database Server, HTTP Server, MIDI, Pembelajaran Nada

## I. PENDAHULUAN

SEIRINGNYA dengan kemajuan teknologi, maka banyak hal yang menjadi lebih praktis. Contohnya adalah inovasi aplikasi simulator yang dapat membuat *user* dapat menjalankan aplikasi simulator dengan lebih nyata. Sayangnya masih belum didukung dengan alat yang memadai sehingga masih kurang terasa nyata, karena masih menggunakan *joystick* maupun menggunakan *keyboard*. Dengan penambahan *device* Kinect, yang merupakan sensor yang bisa mendeteksi warna, jarak, gerakan, wajah, dan juga suara, kita bisa mengendalikan aplikasi simulator tanpa menggunakan *joystick* dan *keyboard*, tapi dengan gerakan anggota badan, sehingga aplikasi simulator menjadi lebih terasa nyata.

Salah satu aplikasi simulator yang dapat digunakan sebagai pembelajaran adalah aplikasi pembelajaran nada dengan menggunakan piano *virtual*. Dengan adanya aplikasi ini, *user* dapat belajar mengetahui nada dengan menggunakan piano *virtual*. Dalam penggunaan aplikasi ini masih tidak terlihat nyata karena dalam memainkan aplikasi ini masih menggunakan *joystick* atau *keyboard*.

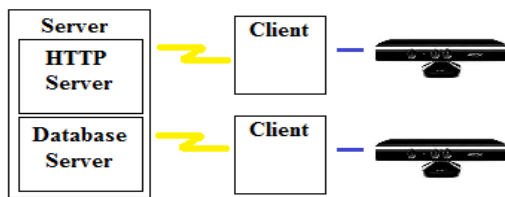
Oleh karena itu dibutuhkan suatu aplikasi simulator pembelajaran nada dengan menggunakan piano *virtual* yang dapat dimainkan secara nyata, yaitu dengan menggunakan *device* kinect sebagai *controller*.

## II. METODE PENELITIAN

- Studi literatur  
Studi ini mempelajari tentang format data, protokol, maupun teknik inisialisasi antar modul atau perangkat yang digunakan serta bagaimana membaca gerakan menjadi sebuah *input* yang dapat digunakan untuk bermain *game*.
- *Browsing* internet  
*Browsing* internet dilakukan untuk mencari tambahan informasi yang dibutuhkan untuk menunjang pembuatan sistem secara keseluruhan.
- Pembuatan aplikasi  
Pada *project* tugas akhir ini sistem yang dibuat diharapkan mampu mengakses dan men-*download file* MIDI yang ada di server, serta sistem piano *virtual* bisa berjalan dengan baik. Oleh karena itu sisi aplikasi menjadi harus diperhatikan.
- Analisa dan pengujian aplikasi
  - Pengujian *update* lagu dari *server*.
  - Pengujian animasi *MIDI*
  - Pengujian musik *MIDI*.
- Kesimpulan  
Kesimpulan diambil dari evaluasi tahap akhir terhadap pengerjaan tugas akhir.
- Pembuatan laporan  
Laporan dibuat berdasarkan dari seluruh kegiatan yang meliputi tahap perancangan, pembuatan, dan evaluasi tahap akhir pada pembuatan sistem pembelajaran nada berbasis *C#* dengan memanfaatkan Kinect sebagai *input sensor*.

## III. PERANCANGAN

Sistem yang terdiri dari *server* dan *client*. *Server* akan digunakan sebagai media penyimpanan data skor dan daftar lagu *MIDI* yang ada. Sedangkan *client* berisikan program piano *virtual* yang akan menggunakan Kinect sebagai *controller*-nya.. *Client* akan terhubung dengan *server* menggunakan jaringan internet. *Client* haruslah terhubung dengan Kinect, karena Kinect digunakan untuk mendeteksi pemain. Blok diagram sistem ditunjukkan pada gambar 1



Gbr 1 Blok diagram sistem

A. Desain Server

Dalam sistem yang dibuat server hanya berfungsi sebagai database server dan sebagai file server (HTTP server). Database yang dibuat berfungsi untuk menyimpan 3 hal utama, yaitu data mengenai player, file MIDI, dan skor. Berikut adalah struktur dari masing-masing tabel beserta relationship dari tabel-tabel tersebut.

Tabel 1 Struktur tabel player

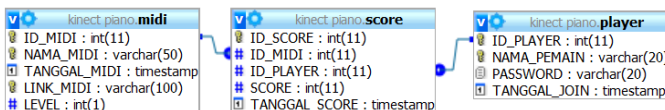
Field	Type	Deskripsi
ID_Player	Int	(Primary Key) (Auto Increment)
Nama_Pemain	VarChar	(Unique) Nick Name pemain
Password	VarChar	Password untuk login
Tanggal_Join	TimeStamp	Mencatat tanggal registrasi pemain

Tabel 2 Struktur tabel MIDI

Field	Type	Deskripsi
ID_MIDI	Int	(Primary Key) (Auto Increment)
Nama_MIDI	VarChar	(Unique) Judul lagu MIDI
Tanggal_MIDI	TimeStamp	Mencatat tanggal registrasi MIDI
Link_MIDI	VarChar	(Unique) Mencatat link MIDI
Level	Int	Mencatat level MIDI

Tabel 3 Struktur tabel score

Field	Type	Deskripsi
ID_Score	Int	(Primary Key) (Auto Increment)
ID_Player	Int	(Foreign key) Menunjukkan ID_Player pemilik score
ID_MIDI	Int	(Foreign key) Menunjukkan ID_MIDI yang dimainkan
Score	Int	Nilai score
Tanggal_Score	TimeStamp	Mencatat waktu Score di-submit ke server

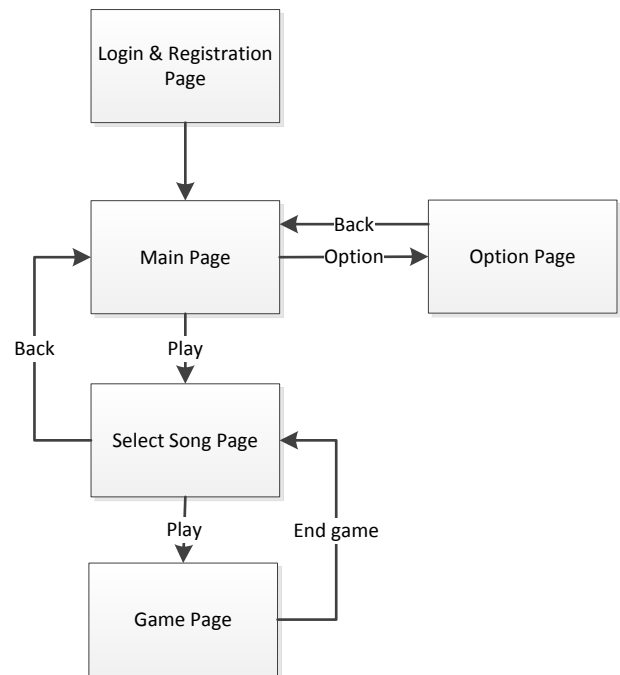


Gbr.2 Entity Relationship diagram dari database

Sedangkan untuk HTTP server, digunakan untuk media transfer data file MIDI dan melayani download request dari client. Link-link MIDI yang tersimpan pada tabel MIDI, semuanya akan merujuk pada alamat HTTP server ini.

B. Desain Client

Pada client, terdapat beberapa halaman tampilan, antara lain adalah login page, main page, option page, song select page, dan game page. Alur halaman tampilan client bisa dilihat pada gambar 3.



Gbr 3. Alur Halaman Tampilan

Pemain harus terlebih dahulu login atau melakukan registrasi, baru pemain akan diarahkan ke main page. Di dalam main page terdapat 2 pilihan, yaitu untuk menuju ke Option Page, atau masuk ke Select Song Page. Pada Option Page terdapat pilihan seperti volume permainan, dan update file MIDI. Sedangkan pada Select Song Page terdapat list file MIDI yang ada di Client. Pemain akan memilih salah satu file dari list tersebut untuk memainkannya, kemudian pemain akan diarahkan ke Game Page. Pada Game Page, pemain akan bermain dengan menggunakan file MIDI yang telah dipilih.

Untuk komunikasi client dengan Database Server menggunakan API ODBC (Open Database Connectivity). ODBC berfungsi untuk menghubungkan client dengan database yang ada di server. Pada sistem yang dibuat, client menggunakan ODBC 3.51. Komunikasi antara server dan client digunakan untuk proses login dan registration, mengambil data skor yang ada, mengambil list file MIDI yang ada di server, dan untuk menambahkan skor hasil permainan ke server.

Fitur yang akan diterapkan pada *client* antara lain adalah:

- Memainkan *file* MIDI
- Fitur *update file* MIDI
- Menggunakan Kinect sebagai *controller*
- Memainkan animasi MIDI.

C. Implementasi sistem

Implementasi sistem dapat dirangkum dalam 2 proses sebagai berikut:

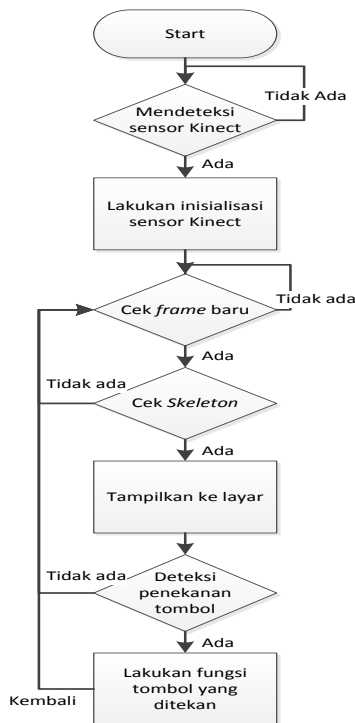
- Implementasi Kinect.
- Implementasi MIDI.

1) Implementasi Kinect.

Proses ini dapat dibagi menjadi beberapa bagian utama, yaitu:

- Inisialisasi Kinect.  
Mengaktifkan sensor-sensor yang akan digunakan pada sensor Kinect yang akan digunakan.
- Menampilkan *skeleton* pemain.  
Setiap ada frame baru, *client* akan memunculkan *skeleton* pemain ke layar. *Skeleton* pemain berupa 2 lingkaran yang menandakan tangan kanan dan tangan kiri pemain.
- Mendeteksi penekanan tombol.  
Setelah menampilkan *skeleton* pemain, *client* akan mendeteksi penekanan tombol. Pemain dianggap menekan tombol apabila jarak tangan pemain terhadap bahu pemain mencapai jarak yang sudah ditentukan.

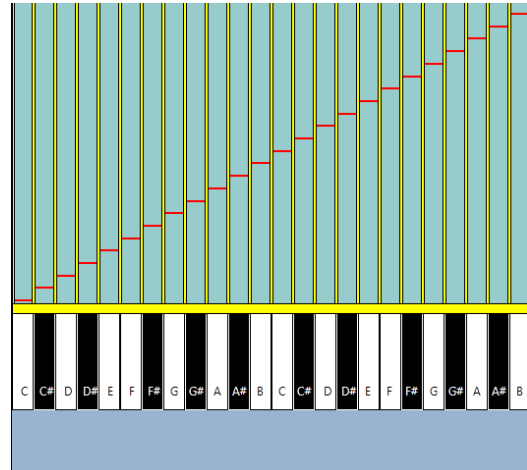
*Flowchart* dari keseluruhan program Kinect yang diterapkan dalam *client* bisa dilihat pada gambar 4.



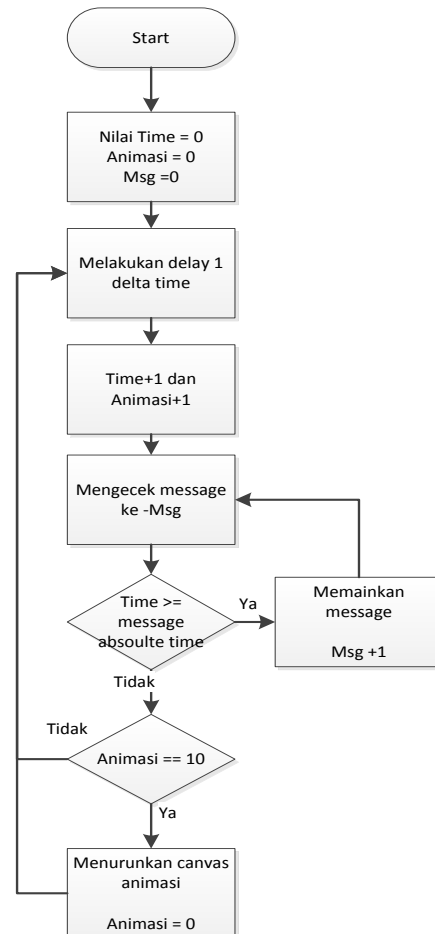
Gbr 4. Flowchart Program Kinect

2) Implementasi MIDI.

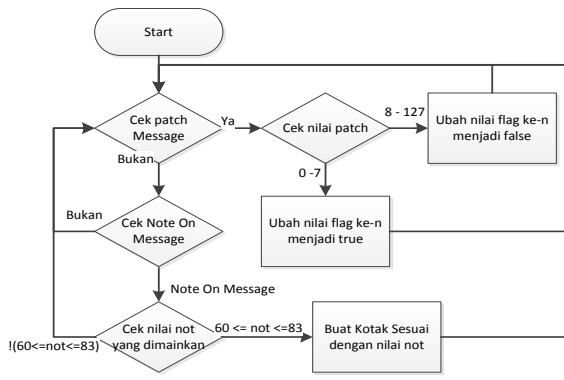
Dalam implementasi MIDI dibagi menjadi 2 bagian, memainkan *background* musik, dan menjalankan animasi permainan. Proses musik *background* dan animasi berjalan pada *thread* yang sama supaya animasi dan musik *background* bisa berjalan dengan sinkron. Berikut adalah gambar contoh hasil animasi dan *flowchart* program *thread* yang digunakan.



Gbr 5. Animasi yang digunakan *client*



Gbr 6. Flowchart Program Thread Memainkan MIDI



Gbr 7. Flowchart Mekanisme Pemilihan Not yang dimainkan

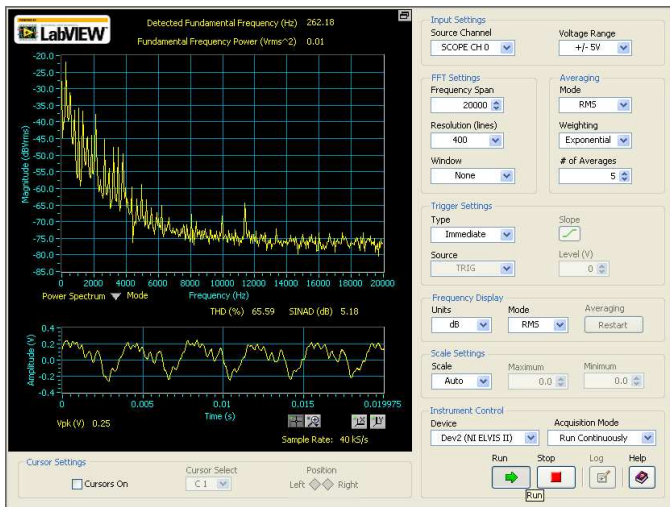
Piano *virtual* yang digunakan memiliki 2 oktaf not lagu dan juga menggunakan not-not kromatis. Nada yang dimainkan juga masih dibatasi dari not C4 - B5 saja.

IV. PENGUJIAN SISTEM

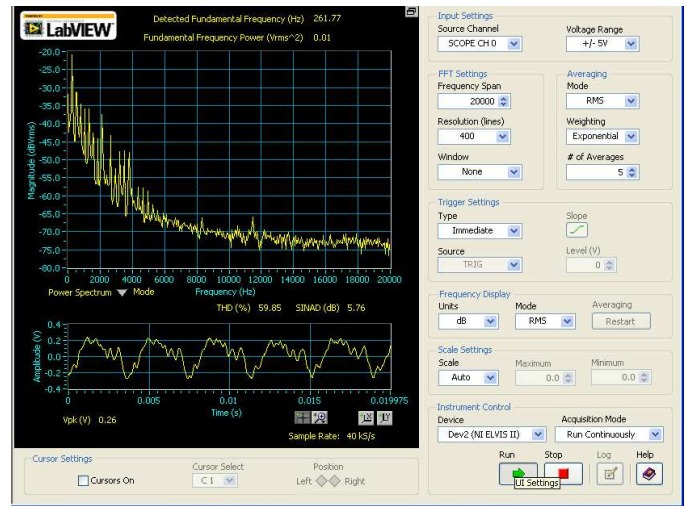
Berikut merupakan hasil pengujian dari aplikasi yang dibuat:

A. Pengujian proses memainkan MIDI.

Pengujian ini dilakukan dengan membandingkan hasil suara yang dihasilkan Windows Media Player, sebagai acuan, dengan *client*. Nada yang akan dimainkan adalah nada C4 yang akan diulang-ulang. Nada C4 memiliki frekuensi 261,63 hz. Hasil dari pengujian ini bisa dilihat pada gambar 8 dan 9.



Gbr 8. Pengukuran hasil suara dari Windows Media Player

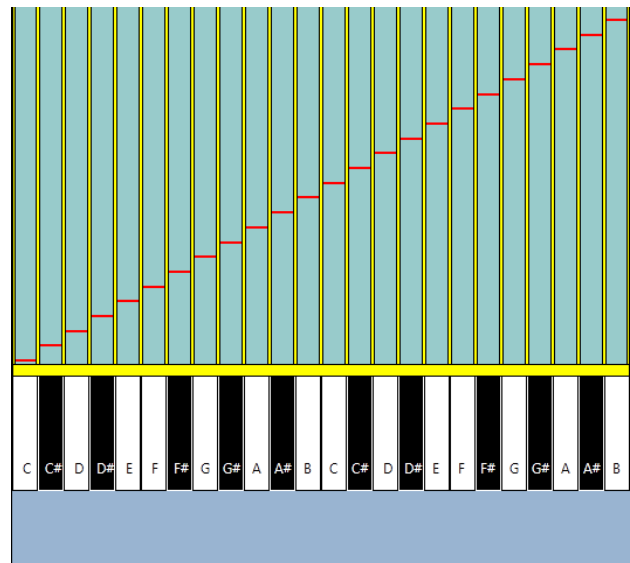


Gbr 9. Pengukuran hasil suara dari *client*

Dari gambar 8, bisa dilihat frekuensi hasil suara dari Windows Media Player 262.18 hz. Sedangkan dari gambar 8 bisa dilihat frekuensi hasil suara dari *Client* 261.77 hz. Dilihat dari hasil frekuensi suara yang dihasilkan antara Windows Media Player dengan *client*, frekuensi hasil suara dari kedua aplikasi menunjukkan nilai yang mendekati dengan nilai frekuensi nada C4 yang memiliki frekuensi 261,63, sehingga bisa dikatakan *client* sudah bisa memainkan *file* MIDI dengan baik.

B. Pengujian animasi.

Pada pengujian ini *client* memainkan *file* MIDI yang sudah disiapkan. Data pada *file* MIDI ini berisikan data *note on* dari not ke 60 – 83 dengan jeda waktu yang sama antar *note on*. Berikut adalah gambar dari hasil pengujian yang dilakukan.



Gbr 10. Hasil Pengujian Pengambilan Nada

Dari gambar 10, bisa dilihat *client* sudah berhasil mengambil semua data *note on* dan menampilkannya. Beda ketinggian antara tiap *note on* juga sama. Animasi yang

dihasilkan juga berjalan sinkron dengan suara nada yang dihasilkan, dimana apabila kotak not mencapai garis kuning, maka suara nada juga dihasilkan.

C. Pengujian update.

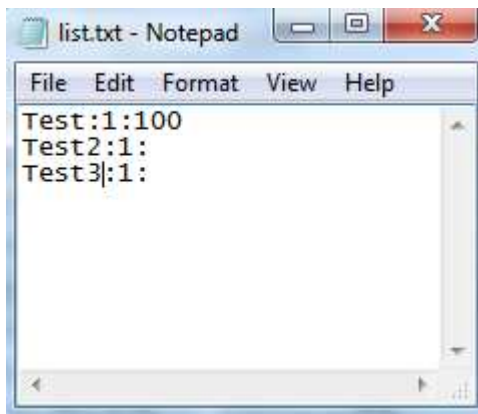
Pada pengujian ini *client* akan melakukan proses *update*. Isi dari tabel MIDI, folder MIDI dan isi dari file list.txt sebelum proses update dilakukan bisa dilihat pada gambar-gambar berikut.

ID_MIDI	NAMA_MIDI	TANGGAL_MIDI	LINK_MIDI	LEVEL
1	Test	2012-11-22 23:43:05	<a href="http://localhost/Midi/Test.mid">http://localhost/Midi/Test.mid</a>	1
2	Test2	2012-11-22 23:43:05	<a href="http://localhost/Midi/Test2.mid">http://localhost/Midi/Test2.mid</a>	1
3	Test3	2012-11-22 23:43:21	<a href="http://localhost/Midi/Test3.mid">http://localhost/Midi/Test3.mid</a>	1
4	Test4	2012-11-28 14:24:51	<a href="http://localhost/Midi/Test4.mid">http://localhost/Midi/Test4.mid</a>	1

Gbr 11. Isi Dari Tabel MIDI

Test	28/11/2012 22:16	File	64 KB
Test2	28/11/2012 22:16	File	66 KB
Test3	28/11/2012 23:08	File	80 KB

Gbr 12. Isi file Di Folder MIDI sebelum update

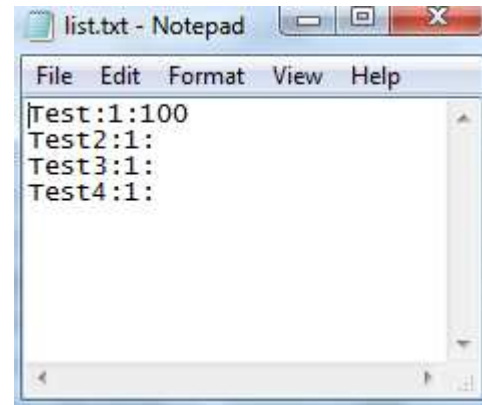


Gbr 13. Isi dari list.txt sebelum update

Dari gambar 11, isi dari tabel MIDI ada 4 *file*, sedangkan pada gambar 12, *folder* MIDI hanya memiliki 3 buah *file*, di mana kurang *file* Test4. Pada gambar 13, *file* list.txt juga tidak menyimpan data mengenai Test4. Kemudian proses *update* dilakukan. Hasilnya bisa dilihat dari gambar-gambar berikut.

Test	29/11/2012 10:57	File	64 KB
Test2	29/11/2012 10:57	File	66 KB
Test3	29/11/2012 10:57	File	80 KB
Test4	29/11/2012 10:57	File	2 KB

Gbr 14. Isi dari folder MIDI setelah update

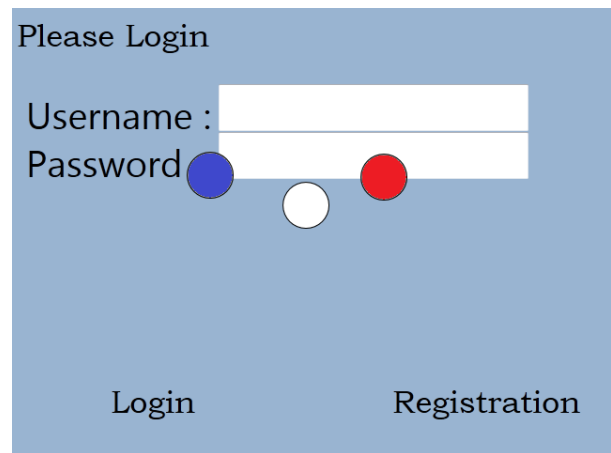


Gbr 15. Isi Dari list.txt Setelah Update

Bila gambar 12 dan 14 dibandingkan, maka bisa dilihat proses *update* yang dilakukan berhasil dengan baik. Pada gambar 15, data pada "list.txt" juga terdapat data *file* Test4.

D. Pengujian Kinect.

Pada pengujian ini *client* akan mendeteksi 1 orang pemain dengan jarak pemain dari Kinect sebesar ± 2M. Berikut adalah gambar hasil pendeteksian pemain.



Gbr 16. Hasil Pencitraan Skeleton Pemain

Pada gambar 16, lingkaran putih ialah badan pemain, lingkaran biru adalah tangan kiri, dan lingkaran merah adalah tangan kanan. Lingkaran-lingkaran tersebut akan mengikuti gerak gerak pemain, selama pemain masih terdeteksi oleh sensor Kinect.

V. KESIMPULAN/RINGKASAN

Berdasarkan pemaparan pada bagian-bagian sebelumnya maka dapat ditarik beberapa kesimpulan sebagai berikut:

- Kinect bisa digunakan untuk mendeteksi pemain dengan baik, walaupun masih belum bisa untuk mendeteksi jari pemain.
- Animasi dan musik MIDI harus berada dalam thread yang sama supaya animasi dan musik bisa berjalan dengan sinkron.
- Metode pemilihan not yang digunakan masih kurang baik, karena permainan masih monoton dan terbatas pada bagian nada tertentu saja.

- Untuk memudahkan memainkan file MIDI yang memiliki banyak *track*, maka *track-track* tersebut harus dijadikan 1 terlebih dahulu.
- Untuk meringankan beban kerja program, tampilan harus dibuat lebih ringan. Salah satu caranya ialah dengan tidak menampilkan gambar RGB.

#### UCAPAN TERIMA KASIH

“Penulis Y.Y mengucapkan terima kasih kepada Universitas Kristen Petra, Bapak Petrus Santoso, Ibu Triana Mugia Rahayu, Bapak Murtiyanto Santoso, Keluarga dan teman-teman yang tidak dapat disebutkan satu-persatu atas dukungan dan doanya dalam pengerjaan tugas akhir ini.”

#### DAFTAR PUSTAKA

- [1] Craig Sapp. “Outline of the Standard MIDI File Format.” *Center for Computer Assisted Research in Humanities*.2003. Stanford University. 28 Mei 2012. Available: < <http://253.ccarh.org/handout/smf/>>
- [2] Dan Fernandez. “Kinect For Windows Quickstart Series.” Channel9.msdn.com. 2012. Channel 9. 22 Maret 2012. Available: <<http://channel9.msdn.com/Series/KinectQuickstart>>
- [3] Edward Chow. “MIDI.” *CS525 Multimedia Computing And Communications*. 2006. University of Colorado. 14 Agustus 2012. Available: <<http://www.cs.uccs.edu/~cs525/midi/midi.html>>
- [4] Messick, Paul. *Maximum MIDI: Music Application in C++*. n.p : Manning Publication Co., 1998.