

Pelacakan Jari Tangan Menggunakan Data Kedalaman Berbasis Tracking

Afdhol Dzikri¹, Rumen Filkov²

¹ Jurusan Teknik Informatika, Program Studi Teknik Multimedia dan Jaringan, Politeknik Negeri Batam

² Teknologi dan Sistem Komputer, Universitas Teknikal Varna, Bulgaria

Email : afdhol@polibatam.ac.id¹, rumen.filkov@gmail.com²

Abstrak— Pelacakan jari tangan merupakan salah satu bagian dari pengembangan Interaksi Manusia dan Komputer (IMK). Penelitian terdahulu tentang pendeteksian tangan masih banyak menggunakan kamera *stereo* dan *Kinect*. Dengan teknologi terbaru, pengembangannya dilakukan secara *real-time* menggunakan *Leap Motion*. Pelacakan jari tangan terdeteksi berdasarkan data kedalaman yang dimiliki oleh *Leap Motion* yaitu tiga komponen vektor sebagai posisi dalam ruang tiga dimensi (sumbu koordinat XYZ). Sedangkan jarak maksimum jari yang terdeteksi dengan asumsi lima jari tangan terdeteksi oleh *Leap Motion* yaitu berkisar antara -0.3 sampai 0.6 meter terhadap sumbu koordinat XYZ.

Kata kunci— Pelacakan jari tangan, IMK, *Kinect*, *Leap Motion*, sumbu koordinat XYZ.

I. PENDAHULUAN

Pengenalan gerak tangan manusia telah banyak diteliti dan terus dikembangkan sampai sekarang. Salah satu pengenalan gerak tangan tersebut meliputi pendeteksian jari tangan, jumlah jari yang terdeteksi, dan posisi tangan terbuka atau tertutup.

Penelitian pengenalan gerak tangan tersebut telah diteliti oleh para peneliti antara lain oleh X. Liu and K. Fujimura [4], Yan Wen [5], Liu Yun [6], semua peneliti menggunakan teknologi kedalaman data yaitu *Kinect* dan *webcam*.

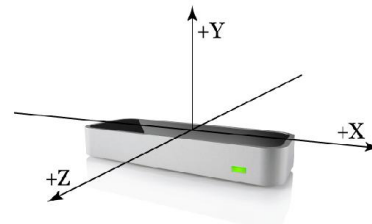
Teknologi kedalaman data yang dilakukan oleh *Kinect* atau *webcam* hanya menangkap gambar tangan dan menampilkan dalam bentuk gambar diam. Tetapi sekarang ada penemuan teknologi terbaru dalam *hand tracking* yaitu *Leap Motion Controller* yang ditunjukkan pada gambar 1.

Jari tangan yang ditangkap oleh *Leap Motion* divisualkan dalam bentuk dimensi 3 (3D) dan gambar bergerak secara *real-time*. Jari tangan yang ditangkap oleh sensor *Leap Motion* merupakan data masukan yang akan dikomputasi oleh algoritma dari *Leap Motion*. Penangkapan pergerakan jari tangan yang bergerak dengan cepat membutuhkan suatu komputasi yang sangat kompleks agar dapat diinterpretasikan dengan komputer yang dilakukan oleh *Leap Motion*.

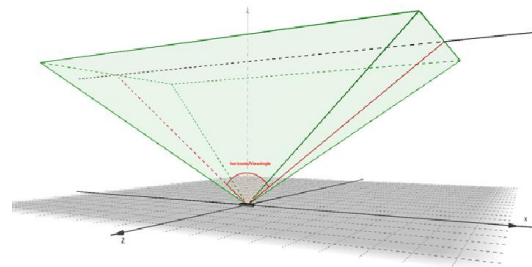
Leap Motion menggunakan sistem koordinat *Cartesian* tangan kanan. Cara kerjanya adalah dengan menciptakan ruang empat kaki kubik interaktif yang ditunjukkan pada gambar 2. Pusat penangkapan gerakan jari tangan berada pada

sensor tengah *Leap Motion Controller*. Dan *Leap Motion Controller* memiliki tiga komponen vektor sebagai tempat seperti arah atau posisi dalam ruang 3D yaitu sumbu koordinat XYZ [1]. Ketiga komponen vektor ini berfungsi untuk memperoleh nilai data kedalaman dari pelacakan atau penangkapan gerakan tangan. *Leap Motion* dapat menangkap pergerakan tangan, jumlah jari maupun posisi jari yang terdeteksi dalam area sumbu koordinat XYZ. Tujuan pengembang menciptakan *Leap Motion* adalah untuk menggantikan fungsi *keyboard* dan *mouse* hanya dengan menggerakkan jari di atas *Leap Motion* tanpa menyentuh layar komputer.

Penelitian ini bertujuan untuk melacak jari tangan yang meliputi penangkapan jumlah jari tangan yang terdeteksi secara *real-time* serta mengukur kedalaman data pergerakan tangan dari sensor *Leap Motion* berdasarkan area sumbu koordinat XYZ.



Gbr. 1 *Leap Motion* dan posisi koordinat XYZ [1]



Gbr. 2 Ruang empat kaki kubik *Leap Motion* [1]

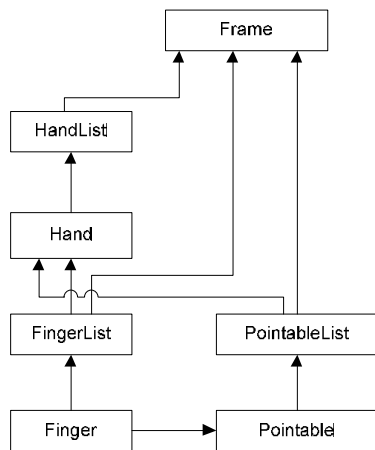
II. METODOLOGI

Leap Motion mendukung beberapa bahasa pemrograman dan dalam penelitian ini menggunakan bahasa pemrograman C#. *Leap*

Motion juga mempunyai *Software Development Kit (SDK)* yang dapat terintegrasi dengan perangkat lunak *Unity 3D*. Pada *Unity 3D* perlu menyertakan *library C# Leap Motion* ke dalam *UnityAssets* yang terdapat dalam *SDK Leap Motion*.

A. Tracking Data

Pada gambar 3 menunjukkan bahwa *Leap Motion* yang telah terhubung dengan komputer akan mengirimkan data *tracking* ke objek *Frame* pada aplikasi.



Gbr. 3 Tracking data

Finger (jari) merupakan objek *Pointable*, karena jari adalah sesuatu yang dapat menunjuk atau *pointed*. Objek *Finger* yang *valid* akan dimasukkan ke objek *Hand*. Sebuah jari memiliki tipe, arah dan satu set posisi sendi. Seluruh lima jari terdaftar dalam *FingerList* pada sebuah tangan (*Hand*). Semua kelas *PointableList*, dan *FingerList* memiliki struktur yang sama dan tersedia dalam objek *Hand* dan *Frame*.

Hand (tangan) adalah entitas utama yang dilacak oleh *Leap Motion Controller*. *Controller* mempertahankan model dari tangan manusia dan memvalidasi data dari sensor terhadap model. Hal ini memungkinkan *Controller* dapat melacak posisi jari bahkan ketika jari tidak sepenuhnya terlihat. Kelas *Hand* merupakan fisik tangan yang terdeteksi oleh *Leap Motion*. Sebuah objek *Hand* menyediakan akses ke daftar *pointables* (*PointableList*) seperti atribut yang menggambarkan posisi, orientasi, dan gerakan tangan. Sedangkan *HandList* bertindak seperti *array* vektor dan mendukung iterator.

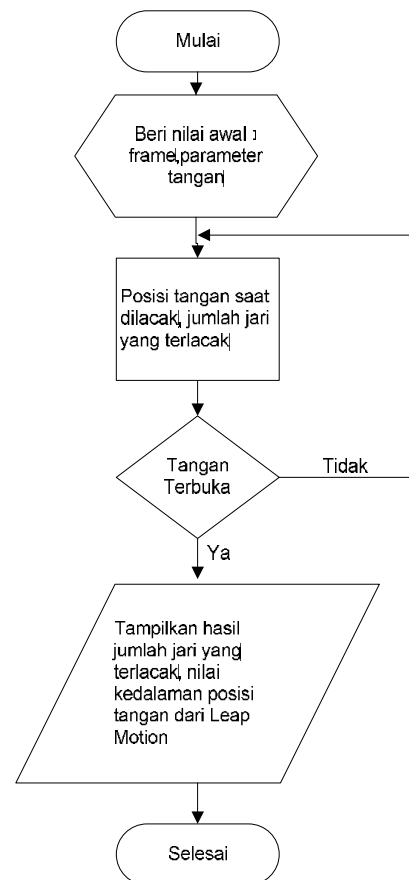
Objek *Frame* merupakan akar dari model data dan menyediakan akses ke semua entitas yang dilacak. Sebuah *Frame* baru dibuat pada setiap interval terbaru. *Frame* berisi daftar tangan, jari, *pointable* yang terlacak dalam satu *Frame*. Akses objek *Frame* juga melalui sebuah *instance* kelas *Controller*. Objek *Controller* yang telah

dihubungkan menghasilkan data *tracking* pada objek *Frame*. Sebuah *frame* didapatkan ketika aplikasi siap untuk memproses *frame* tersebut menggunakan *method frame()* dari kelas *Controller*. Kelas *Frame* juga mendefinisikan beberapa *function* yang menyediakan akses untuk data pada *frame*, seperti *HandList*, *PointableList*, dan *FingerList*.

B. Rancangan Aplikasi

Perancangan yang pertama kali dilakukan sesuai dengan gambar 4 yaitu memberikan nilai awal untuk *frame*, menentukan tangan kiri atau kanan yang akan digunakan.

Pada proses menghitung data dilakukan untuk melacak posisi tangan dan menghitung jumlah jari yang terlacak. Pada saat terlacak, posisi tangan atau *palm position* keadaan terbuka, sedangkan pada saat tangan tertutup atau mengepal gerakan tangan tidak dapat terlacak. Posisi tangan terbuka bisa menampilkan data gerakan jari tangan. *Leap Motion* bisa melacak jari tangan dengan cepat dan akurat serta menampilkan nilai terhadap sumbu X, Y, dan Z. Nilai tersebut diperoleh dengan kondisi lima jari tangan terdeteksi. Dan pergerakan tangan dilakukan dalam lingkup pandang maksimal dari *Leap Motion*.



Gbr. 4 Flowchart Rancangan Aplikasi

III. HASIL DAN DISKUSI

A. Pengujian Fungsional

Pengujian yang pertama dilakukan adalah pengujian terhadap semua fungsi kelas *Controller* yang digunakan untuk mengaktifkan *gesture* dari *Leap Motion*. Implementasi aplikasi dibangun dengan menggunakan perangkat keras *notebook* dengan spesifikasi processor intel core i7 CPU 2 GHz RAM 4 GB. Sedangkan sistem operasi dan *tools* yang digunakan untuk membangun aplikasi ini adalah sistem operasi Microsoft Windows 7 SP 1, *Unity 3D 4.3.1*, dan *Leap Motion*.

B. Implementasi Pelacakan Jari Tangan secara Real-Time

Berdasarkan hasil percobaan, semakin tinggi jarak tangan dari *Leap Motion* (Sumbu Y) maka hasilnya semakin besar. Pergerakan tangan secara horizontal ke sebelah kiri alat (sumbu $-X$) hasilnya semakin kecil (negatif), begitu sebaliknya semakin ke arah kanan (sumbu X) hasilnya semakin positif. Untuk hasil sumbu Z, semakin ke arah layar hasilnya semakin positif dan besar, hal ini dikarenakan pada aplikasi *unity* untuk sumbu Z ke arah layar adalah bernilai positif. Dan ini bertentangan dengan aturan tangan kanan yang diadopsi oleh sumbu koordinat *Leap Motion*. Nilai pada koordinat sumbu XYZ pada saat tangan berada dalam area pandang maksimal *Leap Motion* yaitu sumbu X = 0.3, sumbu $-X$ = -0.3, sumbu Y = 0.6, sumbu Z = 0.3, sumbu $-Z$ = -0.3. Semua nilai tersebut dalam satuan meter. Hasil percobaan pergerakan tangan berdasarkan koordinat sumbu X, Y, Z tersebut dengan asumsi lima jari tangan terdeteksi semua. Adapun hasilnya ditunjukkan pada gambar 5.



(a)



(b)



(c)



(d)



(e)

Gbr. 5 Jumlah jari tangan yang terdeteksi dan nilai sumbu koordinat XYZ dari gerakan tangan pengguna : (a). Maks. Sumbu Y, (b). Maks. Sumbu X, (c). Maks. Sumbu $-X$, (d). Maks. Sumbu Z, (e). Maks. Sumbu $-Z$.

IV. KESIMPULAN

Berdasarkan hasil pengamatan dan metodologi maupun hasil percobaan, telah dilakukan percobaan pelacakan jari tangan dalam posisi telapak tangan terbuka ke bawah dan bergerak dalam area jangkauan *Leap Motion*. Metode yang digunakan dalam penelitian ini adalah metode dari *Leap Motion* sendiri yaitu data kedalaman berdasarkan tiga komponen vektor (area sumbu koordinat XYZ). Telapak tangan pengguna bergerak secara horizontal maupun vertikal, ke atas dan kebawah untuk mendapatkan nilai terhadap sumbu XYZ. Nilai yang diperoleh berkisar antara -0.3 sampai 0.6 meter.

REFERENSI

- [1] Documentation Leap Motion. (2014) C# and Unity. <http://developer.leapmotion.com/documentation/csharp/>.
- [2] Lee Garber, "Gestural Technology: Moving Interfaces in a New Direction", Technology News, *IEEE Journals & Magazine*, 2013.

- [3] Michal Nowicki, Olgierd Pilarczyk, Jakub Wasikowski, Katarzyna Zjawin, "Gesture Recognition Library for Leap Motion Controller", Bachelor's thesis, Poznan University of Technology, 2014.
- [4] X. Liu, K. Fujimura (2004), "Hand gesture recognition using depth data", in Automatic Face and Gesture Recognition, Proceeding, *Sixth IEEE International Conference on*, pp. 529 – 534, 2004.
- [5] Yan Wen, Chuanyan Hu, Guanghui Yu, Chanbo Wang, "A Robust Method of Detecting Hand Gesture Using Depth Sensors", *IEEE*, 978-1-4673-1567-8, 2012.
- [6] Liu Yun, Zhang Lifeng, Zhang Shujun, "A Hand Gesture Recognition Method Based on Multi-Feature Fusion and Template Matching", *Procedia Engineering 29 1678-1684, IWIEE, SciVerse ScienceDirect*, 2012.
- [7] Y. Fang, K. Wang, J. Cheng, H. Lu, "A Real-Time Hand Gesture Recognition Method", *Proc. IEEE Int. Conf. on Multimedia and Expo*, pp. 995-998, 2007.
- [8] Pedro Trindade, Jorge Lobo, Jao P. Barreto (2012), "Hand gesture recognition using color and depth images enhanced with hand angular pose data", *IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems, Germany*, September 13-15, 2012.
- [9] Zhou Ren, Junsong Yuan, Jingjing Meng, Zhengyou Zhang (2013), "Robust Part-Based Hand Gesture Recognition Using Kinect Sensor", *IEEE Transaction On Multimedia*, Vol. 15, No. 5, August, 2013.
- [10] Unity3D.(2014), <http://docs.Unity3d.com/Documentation/Manual/>.
- [11] Ren, Z., Yuan, J., Meng, J., Zhang, Z. "Robust Part-based Hand Gesture Recognition based on Finger-Earth Mover's Distance", *IEEE*, 2012.
- [12] Li, Y, "Hand Gesture Recognition using Kinect", *IEEE*, 2012.
- [13] Panagiotis Apostolellis, Brennon Bortz, Mi Peng, Nicholas Polys, Andy Hoegh, "Poster: Exploring the Integrality and Separability of the Leap Motion Controller for Direct Manipulation 3D Interaction", *Symposium on 3D User Interfaces, IEEE*, 2014.