

## EVALUASI PENERAPAN ALGORITMA ITERATIVE CLOSEST POINT (ICP) PADA PROSES REGISTRASI KUMPULAN TITIK DARI CITRA (POINT CLOUD)

Riska Khairunnisah dan Lussiana ETP

STMIK Jakarta STI&K

Jl. BRI No.17, Radio Dalam, Kebayoran Baru, Jakarta Selatan 12140

{riskakhairunnisah, lussiana.etp}@gmail.com

### ABSTRAK

*Iterative Closest Point (ICP) adalah salah satu algoritma yang sering digunakan untuk registrasi point cloud. Dalam meregistrasi point cloud, algoritma ICP mencari korespondensi dari semua titik pada point cloud untuk menentukan transformasi yang harus dilakukan point cloud Model agar sejajar dengan point cloud Target. Tujuan penelitian ini adalah mengevaluasi kinerja algoritma Iterative Closest Point (ICP) pada proses registrasi point cloud, sehingga dapat menentukan pengembangan yang bisa diterapkan pada algoritma ICP untuk meregistrasi point cloud 3 dimensi. Berdasarkan hasil pengujian, menunjukkan bahwa banyaknya titik yang dimiliki suatu point cloud mempengaruhi waktu yang diperlukan untuk proses registrasi. Kondisi kesejajaran yang tinggi antar titik yang berkorespondensi dapat diperoleh dengan memeriksa nilai MSE, semakin kecil nilai MSE maka semakin didapatkan kondisi sejajar. Nilai MSE yang semakin rendah diperoleh dengan menambah iterasi proses registrasi. Berdasarkan kompleksitas waktu algoritma ICP, yaitu  $O(NpNx)$ , dapat dinyatakan bahwa waktu komputasi algoritma ICP linier terhadap jumlah titik pada point cloud. Dengan demikian dapat disimpulkan bahwa semakin tinggi resolusi suatu citra, menunjukkan jumlah titik yang semakin banyak, sehingga membutuhkan waktu yang tinggi untuk proses registrasi. Hal ini merupakan peluang dari pengembangan metode registrasi point cloud yang dapat mengurangi waktu yang diperlukan untuk proses registrasi point cloud tersebut.*

**Kata Kunci :** *registrasi, point cloud, ICP, jumlah titik*

### PENDAHULUAN

Registrasi merupakan tahap penting pada pengolahan *point cloud* 3 dimensi [1], salah satunya dalam proses rekonstruksi objek 3D dari citra. *Point cloud* adalah set titik yang merupakan nilai terkecil dari suatu citra 3 dimensi.

Tahap rekonstruksi objek 3 dimensi citra dimulai dengan merepresentasikan citra ke dalam bentuk *point cloud*, kemudian meregistrasi *point cloud* dengan cara mensejajarkan beberapa *point cloud* sehingga menjadi satu kesatuan. Setelah proses registrasi selesai dilakukan, tahap rekonstruksi berikutnya adalah membangun permukaan objek 3D dari *point cloud* hasil registrasi.

Dalam melakukan registrasi, dibutuhkan dua *point cloud* yang masing-masing merepresentasikan objek dari sudut pandang berbeda. Untuk setiap *point cloud*, diperlukan cara yang dapat mensejarkannya menjadi kumpulan titik dari citra tunggal. Tujuan dari registrasi *point cloud* adalah menemukan posisi dan arah yang

tepat antar pandangan, sehingga area yang tumpang tindih dapat dicocokkan sebaik mungkin [2].

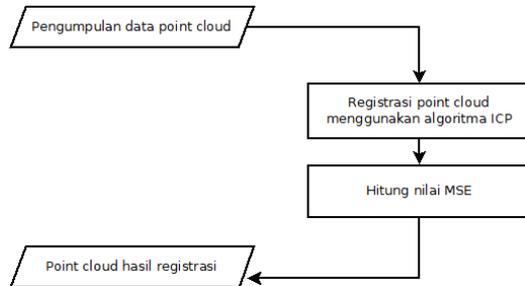
Salah satu algoritma registrasi *point cloud* adalah *Iterative Closest Point (ICP)* [3] yang diperkenalkan oleh Besl *et al* pada tahun 1992. Dalam setiap proses registrasi, algoritma ICP mencari korespondensi dari setiap titik *point cloud* Target pada *point cloud* Model. Berdasarkan korespondensi titik yang diperoleh, matriks transformasi dapat ditentukan. *Point cloud* Model kemudian ditransformasi agar sejajar dengan *point cloud* Target. Tahapan tersebut dilakukan secara berulang sampai diperoleh kondisi konvergen, sehingga diperoleh *point cloud* yang saling sejajar.

ICP merupakan algoritma yang paling sering digunakan dalam meregistrasi *point cloud* [4], namun demikian masih memiliki keterbatasan dalam hal waktu pemrosesan yang tinggi dalam melakukan registrasi *point cloud*.

Penelitian ini bertujuan mengevaluasi kinerja algoritma *Iterative*

*Closest Point* (ICP) pada proses registrasi *point cloud*, sehingga dapat menentukan pengembangan yang bisa diterapkan pada algoritma ICP untuk meregistrasi *point cloud* 3 dimensi.

## METODE PENELITIAN



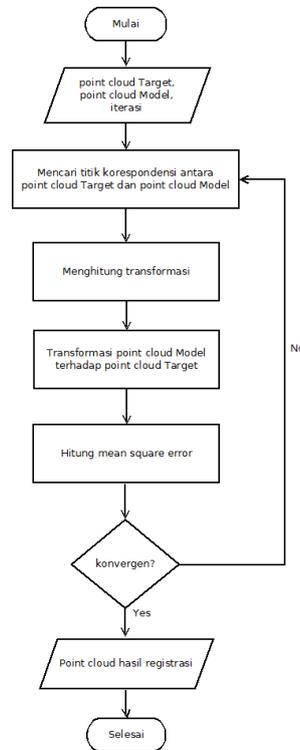
**Gambar 1.** Skema Metode Penelitian

Gambar 1 merupakan skema metode penelitian yang dilakukan. Pertama mengumpulkan *dataset* berupa citra yang direpresentasikan dalam bentuk kumpulan titik (*point cloud*). *Dataset* ini diperoleh dari halaman web *Repository Point Cloud Library*. Dari setiap *dataset*, dibutuhkan masing-masing dua *point cloud* yaitu *point cloud Target* dan *point cloud Model* untuk proses registrasi. Tahap selanjutnya adalah meregistrasi *point cloud* pada masing-masing *dataset* untuk memperoleh satu *point cloud* tunggal.

Pada penelitian ini terdapat tiga *dataset* yang digunakan untuk pengujian yaitu *dataset Windows*, *Door*, dan *Boxes*. *Dataset Windows* merekam objek jendela, *dataset Door* merekam sebuah pintu, dan *dataset Boxes* merekam sebuah kotak dengan barang di dalamnya. *Dataset* tersebut direkam menggunakan kamera Asus Xtion Pro dengan jarak 3 sampai dengan 4 meter dari objek utama, kemudian bergerak semakin mendekat dan berputar di sekitar objek. Setiap *dataset* terdiri dari citra 3 Dimensi yang direpresentasikan dalam bentuk *point cloud* dan disimpan dalam dokumen dengan ekstensi *pcd*. Titik pada *point cloud* mengandung informasi posisi pada koordinat  $x, y, z$  serta intensitas warna *Red, Green, Blue* (RGB).

## Registrasi Point Cloud

Gambar 2 adalah skema tahap registrasi *point cloud* setelah mendapatkan *point cloud Target* dan *point cloud Model* dari masing-masing *dataset*.



**Gambar 2.** Tahap Registrasi *point cloud* dengan algoritma ICP

Pada tahap registrasi, dibutuhkan tiga nilai input yaitu *point cloud Target*, *point cloud Model*, dan parameter iterasi sebagai kriteria konvergensi. Registrasi *point cloud* diawali dengan mencari titik yang saling berkorespondensi antara titik di *point cloud Target* dengan titik di *point cloud Model*, dengan cara memeriksa jarak antara keduanya. Pasangan titik di *point cloud Target* adalah titik pada *point cloud Model* yang memiliki jarak *euclidean* terdekat dengannya.

Jarak *euclidean*  $d(\vec{r}_1, \vec{r}_2)$  antara dua titik  $\vec{r}_1 = (x_1, y_1, z_1)$  dan  $\vec{r}_2 = (x_2, y_2, z_2)$  diperoleh menggunakan persamaan 1.

$$d(\vec{r}_1, \vec{r}_2) = \|\vec{r}_1 - \vec{r}_2\| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (1)$$

Setelah memperoleh titik yang saling berkorespondensi, tahap selanjutnya

adalah menghitung transformasi untuk *point cloud* Model agar sejajar dengan *point cloud* Target. Transformasi terdiri dari matriks rotasi dan vektor translasi yang menentukan pergeseran serta perputaran *point cloud* Model agar dapat meminimalkan nilai *error* yang merupakan jarak antara titik-titik di *point cloud* Target yang saling berkorespondensi dengan titik di *point cloud* Model. Pada tahap ketiga, *point cloud* Model ditransformasi menggunakan matriks transformasi yang telah diperoleh di tahap sebelumnya. Tahap berikutnya adalah menghitung nilai MSE (*mean square error*) sebagai nilai *error* untuk iterasi pertama.

Pada penelitian ini, nilai MSE dari proses registrasi di suatu iterasi  $k$  ( $e_k$ ) merupakan rata-rata selisih jarak antara titik di *point cloud* Target  $P$  ( $p_{ik}$ ) dengan titik pasangannya di *point cloud* Model ( $y_{ik}$ ) pada iterasi ke  $k$ , yang dihitung menggunakan persamaan 2 [3].

$$e_k = \frac{1}{N} \sum_{i=1}^N \|\vec{y}_{ik} - \vec{p}_{ik}\|^2 \quad (2)$$

Dimana  $\|\cdot\|^2$  menunjukkan *Norm Euclidean* antara dua titik, dan  $N$  menyatakan jumlah titik pada *point cloud* Model.

Langkah-langkah tersebut dilakukan terus menerus sampai kriteria konvergensi terpenuhi, yaitu ketika jumlah iterasi yang dilakukan telah mencapai batas iterasi yang ditetapkan. Pada penelitian ini, digunakan lima batas iterasi yang berbeda yaitu 1, 5, 10, 15 dan 20 iterasi.

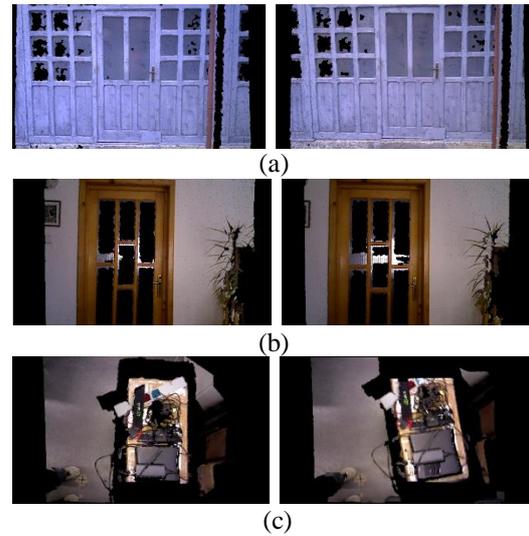
## HASIL DAN PEMBAHASAN

Registrasi *point cloud* pada penelitian ini diaplikasikan kepada tiga *dataset* yang direkam menggunakan kamera Asus Xtion Pro. Semua proses dilakukan pada perangkat dengan spesifikasi: *Processor* AMD A9-9425 RADEON R5, 5 COMPUTE CORES 2C+3G 3.10 Ghz, Sistem Operasi Windows 10, dan memori 4,00 GB RAM.

### A. Input data *point cloud*

Pada masing-masing *dataset*, ditentukan *point cloud* yang ingin diregistrasi. Data *point cloud* dari *dataset* yang digunakan pada penelitian ini dapat

dilihat pada Gambar 4.



**Gambar 4.** Data *point cloud* dari dua sudut pandang pada *dataset* Windows (a), *dataset* Door (b), dan *dataset* Boxes (c).

Gambar 4 menunjukkan dua *point cloud* dari masing-masing *dataset* yang digunakan pada proses registrasi. *Point cloud* di sisi kiri pada Gambar 4 merupakan *point cloud* Target sedangkan sisi kanan merupakan *point cloud* Model dari *dataset* Windows, *dataset* Door dan *dataset* Boxes, seperti yang dijelaskan pada Tabel 1.

**Tabel 1.** Data *point cloud* dari *dataset* Windows *dataset* Door dan *dataset* Boxes

Dataset	Point Cloud Target		Point Cloud Model	
	Nama File (.pcd)	Gambar	Nama File (.pcd)	Gambar
Windows	frame_00005	(a) kiri	frame_00015	(a) kanan
Door	frame_00007	(b) kiri	frame_00029	(b) kanan
Boxes	frame_00000	(c) kiri	frame_00015	(c) kanan

Berdasarkan Tabel 1, *point cloud* yang digunakan pada uji coba penelitian ini adalah frame\_00005.pcd dan frame\_00015.pcd dari *dataset* Windows, frame\_00007.pcd dan frame\_00029.pcd dari *dataset* Door, serta frame\_00000.pcd dan frame\_00015.pcd dari *dataset* Boxes. Jumlah titik *point cloud* pada masing-masing *dataset* yang menunjukkan banyaknya titik yang membangun *point cloud* disajikan pada Tabel 2.

**Tabel 2.** Jumlah titik point cloud

Dataset	Jumlah Titik Point Cloud Target	Jumlah Titik Point Cloud Model
Windows	248396	254736
Door	222446	222634
Boxes	249043	256897

### B. Hasil Registrasi Point Cloud

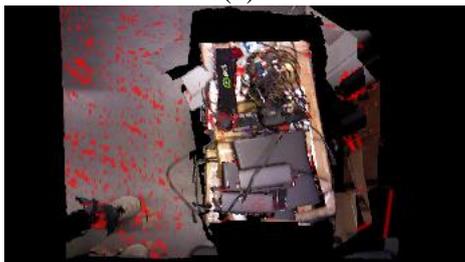
Gambar 5 merupakan visualisasi titik yang saling berkorespondensi antara point cloud Target dan point cloud Model pada iterasi pertama proses registrasi. Korespondensi titik divisualisasikan dengan garis berwarna merah.



(a)



(b)



(c)

**Gambar 5.** Korespondensi titik antara point cloud Target dan point cloud Model pada dataset Windows (a), dataset Door (b), dan dataset Boxes (c).

Setelah memperoleh titik yang saling berkorespondensi, proses registrasi dilanjutkan dengan menghitung transformasi yang perlu dilakukan point cloud Model. Contoh transformasi yang dihasilkan dapat dilihat pada Gambar 6.

0.999979	0.0029285	-0.00587958	-0.000530061
-0.00294936	0.99999	-0.00354761	0.0114023
0.00586921	0.00356489	0.999977	0.0070219
0	0	0	1

**Gambar 6.** Matriks Transformasi

Pada Point Cloud Library (PCL), matriks transformasi direpresentasikan dengan matriks berukuran 4 x 4. Matriks 3 x 3 dari kolom pertama menyatakan matriks rotasi, sedangkan kolom terakhir menyatakan vektor translasi. Pada matriks transformasi tersebut, baris terakhir tidak digunakan dan harus selalu bernilai [0, 0, 0, 1].

Pada masing-masing dataset, implementasi dilakukan sebanyak lima kali pengujian dengan parameter iterasi yang berbeda, yaitu 1, 5, 10, 15, dan 20. Dalam satu kali proses registrasi, pencarian titik yang saling berkorespondensi, penentuan transformasi, serta proses transformasi point cloud Model dan penghitungan nilai MSE diulangi sebanyak parameter yang digunakan agar point cloud Model dapat sejajar dengan point cloud Target.

Berdasarkan hasil proses registrasi, evaluasi algoritma ICP dilakukan pada dua variabel utama yaitu durasi waktu registrasi dan nilai MSE. Penghitungan nilai MSE diperoleh dari setiap hasil registrasi pada masing-masing dataset. Nilai MSE merupakan rata-rata jarak antara titik pada point cloud Target dengan titik pasangannya di point cloud Model setelah proses registrasi selesai dilakukan [3].

Tabel 3 menunjukkan nilai MSE dari algoritma ICP pada setiap iterasi untuk masing-masing dataset.

**Tabel 3.** Nilai MSE proses registrasi

Iterasi	Windows	Door	Boxes
1	0.000845723	0.00142084	0.00209345
5	0.000663266	0.000146822	0.00162482
10	0.000573271	0.000114843	0.00153838
15	0.000515977	0.000103152	0.00150691
20	0.000479651	0.0000979574	0.00149395

Berdasarkan data pada Tabel 3, tampak bahwa registrasi point cloud dengan jumlah iterasi yang banyak akan memperkecil nilai MSE proses registrasi. Nilai MSE terkecil untuk dataset Windows

adalah 0.000479651, yang diperoleh pada iterasi ke 20, demikian pula untuk dataset Door dan Boxes diperoleh MSE terkecil pada iterasi ke 20. Nilai MSE yang semakin kecil menunjukkan kedua *point cloud* yang digunakan pada proses registrasi dapat dicocokkan dengan baik dan semakin sejajar, sehingga registrasi *point cloud* memiliki tingkat kepercayaan yang tinggi.

Evaluasi penerapan ICP juga dilakukan terhadap waktu yang dibutuhkan selama proses registrasi. Algoritma ICP mencakup 4 tahapan, mulai dari mencari korespondensi titik, menentukan dan melakukan transformasi, serta menghitung nilai MSE.

Jika jumlah titik pada *point cloud* Target adalah  $N_p$ , dan jumlah titik pada *point cloud* Model adalah  $N_x$ , maka kompleksitas waktu dari setiap tahap pada proses registrasi adalah  $O(N_p N_x)$  untuk pencarian titik yang saling berkorespondensi,  $O(N_p)$  untuk penghitungan transformasi, dan  $O(N_x)$  untuk proses transformasi [3]. Berdasarkan kompleksitas dari masing-masing tahap yang dilakukan, kompleksitas waktu yang diperoleh adalah  $O(N_p N_x + N_p + N_x)$ , dengan mengabaikan orde yang lebih rendah, diperoleh kompleksitas waktu  $O(N_p N_x)$ , dimana  $N_p$  dan  $N_x$  merepresentasikan jumlah titik dari *point cloud* yang digunakan [5].

Tabel 4 menunjukkan waktu yang dibutuhkan selama proses registrasi pada masing-masing *dataset*.

**Tabel 4.** Waktu proses registrasi dalam satuan *millisecond (ms)*

Iterasi	Windows	Door	Boxes
1	43265	143879	105670
5	77248	173978	160016
10	121611	211627	233253
15	163150	253665	315517
20	218140	285583	402096

Dari Tabel 4, waktu yang diperlukan untuk proses registrasi *point cloud* dipengaruhi oleh jumlah titik dari masing-masing *dataset*. Semakin banyak jumlah titik, semakin panjang waktu yang dibutuhkan untuk registrasi. Banyaknya titik ini juga akan mempengaruhi waktu yang diperlukan untuk mendapatkan nilai MSE

yang diinginkan. Jika diinginkan nilai MSE semakin kecil, maka menambah iterasi proses registrasi, seperti yang ditunjukkan pada Tabel 3.

## PENUTUP

Berdasarkan hasil uji coba pada ketiga *dataset*, menunjukkan bahwa banyaknya titik yang dimiliki suatu *point cloud* sangat mempengaruhi proses registrasi algoritma *Iterative Closest Point* (ICP). Hasil registrasi akan semakin baik didapatkan dengan memeriksa nilai MSE yang kecil, seperti yang diinginkan. Hal ini dapat mempengaruhi waktu yang diperlukan dalam melakukan registrasi, disebabkan bertambahnya iterasi.

Berdasarkan kompleksitas waktu algoritma ICP, yaitu  $O(N_p N_x)$ , dapat dikatakan bahwa waktu komputasi algoritma ICP linier terhadap jumlah titik pada *point cloud*. Oleh karena itu, semakin tinggi resolusi *point cloud*, semakin lama durasi waktu yang dibutuhkan untuk menyelesaikan proses registrasi. Dengan demikian masih terbuka peluang pengembangan metode registrasi *point cloud* yang dapat mengurangi waktu komputasi.

## DAFTAR PUSTAKA

- [1] Holz, D., A. E. Ichim, F. Tombari, R. B. Rusu, dan S. Behnke, "Registration with the Point Cloud Library: A Modular Framework for Aligning in 3-D", *IEEE Robotics & Automation Magazine* 22(4) 110-124, 2015.
- [2] Rusu, R. B. dan S. Cousins, "3d is here: Point Cloud Library (PCL)", *IEEE International Conference on Robotics and Automation*, 1-4, IEEE, Mei, 2011.
- [3] Besl, P. J. dan N. D. McKay, "A Method for Registration of 3-D Shapes", *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14(2) 239-256, 1992.
- [4] Shi, Xiuying., J. Peng, J. Li, P. Yan, dan H. Gong, "The Iterative Closest Point Registration Algorithm Based on the Normal Distribution Transformation", *Procedia Computer Science* 147 181-190, 2019.

- [5] Jost, T. dan H. Hügly, “Fast ICP algorithms for shape registration”, *Pattern Recognition* 91-99, 2002.