

# Penerapan Ekstraksi *Keypoint* Menggunakan Algoritma *Scale-Invariant Feature Transform* (SIFT) sebagai Dasar Melakukan Registrasi *Point Cloud*

Riska Khairunnisah<sup>1</sup> dan Lussiana ETP<sup>2</sup>

<sup>1</sup>Program Magister Manajemen Sistem Informasi, Program Pascasarjana  
Universitas Gunadarma

<sup>2</sup>Program Magister Teknologi Informasi, Program Pascasarjana  
STMIK Jakarta STI&K

E-mail : riskakhairunnisah@gmail.com, lussiana.etp@gmail.com

## Abstrak

Registrasi kumpulan titik dari citra (*point cloud*) merupakan tahap penting pada proses rekonstruksi dalam rangka membangun objek 3 dimensi. Salah satu algoritma yang sering digunakan untuk meregistrasi *point cloud* adalah *Iterative Closest Point* (ICP). Keterbatasan Algoritma ICP terletak pada proses pencarian korespondensi dari semua titik pada *point cloud* untuk menentukan transformasi, yang membutuhkan waktu yang panjang. Tujuan penelitian ini adalah mempersingkat waktu proses dengan melakukan ekstraksi *keypoint* menggunakan algoritma *Scale Invariant Feature Transform* (SIFT) sebelum proses registrasi *point cloud* dengan menggunakan algoritma *Iterative Closest Point* (ICP). Berdasarkan hasil uji coba menunjukkan bahwa tahap ekstraksi *keypoint* berhasil mendapatkan jumlah titik yang lebih sedikit sehingga dapat menurunkan waktu proses registrasi sebelumnya. Dengan demikian, dapat disimpulkan bahwa kombinasi algoritma SIFT dan ICP dapat diterapkan untuk registrasi *point cloud* tanpa mengubah objek asli dari *point cloud*.

**Kata Kunci:** registrasi, point cloud, SIFT, ICP

## Pendahuluan

Objek 3D kerap digunakan untuk merepresentasikan objek di dunia nyata dalam bentuk digital seperti pada animasi, *game* dan juga iklan. Pembuatan objek 3D dapat dilakukan menggunakan aplikasi pemodelan 3 dimensi (3D), namun membutuhkan keahlian pengguna untuk dapat menghasilkan objek yang menyerupai aslinya, serta membutuhkan waktu yang cukup lama [1]. Cara lain menghasilkan objek 3D adalah dengan melakukan rekonstruksi objek 3D dari citra. Proses rekonstruksi dimulai dengan merepresentasikan citra ke dalam bentuk set titik, sebagai nilai terkecil dari suatu citra. Tahap rekonstruksi selanjutnya adalah meregistrasi kumpulan titik dari citra atau yang biasa disebut dengan *point cloud*, dan kemudian membangun permukaan objek 3D dari point

cloud yang telah diregistrasi.

Registrasi adalah proses mensejajarkan beberapa kumpulan titik dari citra (*point cloud*) sehingga menjadi satu kesatuan. Sebagai salah satu tahapan dalam rekonstruksi, registrasi memiliki peran penting untuk menghasilkan *point cloud* dari citra yang dapat mewakili suatu objek secara utuh. Dibutuhkan dua *point cloud* yang berbeda, yang masing-masing merepresentasikan objek dari sudut pandang berbeda. Registrasi dilakukan untuk menemukan posisi dan arah yang tepat antar pandangan di dalam koordinat global, sehingga area yang tumpang tindih dapat dicocokkan sebaik mungkin. Untuk setiap point cloud, diperlukan cara yang dapat mensejajarkan mereka menjadi kumpulan titik dari citra tunggal [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 akan mencari korespondensi dari setiap titik *point cloud* target pada *point cloud* model agar dapat menentukan matriks transformasi. *Point cloud* model kemudian ditransformasi agar sejajar dengan *point cloud* target.

Keterbatasan utama algoritma ICP adalah algoritma ini membutuhkan waktu pemrosesan yang tinggi, terutama ketika menggunakan data dengan resolusi tinggi. Beberapa algoritma telah dikembangkan oleh para peneliti untuk menanggulangi permasalahan registrasi *point cloud*, diantaranya Nejad dan Nasri (2017) melakukan ekstraksi *keypoint* menggunakan metode *Scale Invariant Feature Transform* (SIFT) untuk proses registrasi [4]. Algoritma SIFT diperkenalkan oleh Lowe pada tahun 2004 untuk ekstraksi *keypoint* dan fitur dari suatu citra yang dapat digunakan untuk mencocokkan citra yang berbeda dari suatu objek atau scene [5]. Bay, Tuytelaars, dan Gool (2006) mengembangkan ekstraksi *keypoint* yang dikenal dengan algoritma *Speeded Up Robust Features* (SURF) [6]. Algoritma SURF digunakan untuk menghitung lokal invarian fitur dari suatu *point cloud input*.

Filho, Moreira, Xavier, Gomes, Santos dan Freitas (2015) membandingkan metode SIFT dan *Speeded Up Robust Feature* (SURF), hasilnya menunjukkan bahwa algoritma SIFT lebih stabil dan memiliki performa yang lebih baik dibandingkan dengan SURF [7]. SIFT juga tidak berpengaruh terhadap perubahan skala dan rotasi, serta stabil terhadap perubahan pencahayaan, distorsi maupun derau (*noise*) [5]. Dengan keunggulan tersebut, SIFT merupakan algoritma yang paling baik dalam registrasi citra [4].

Berdasarkan kondisi tersebut, penelitian ini bertujuan mengekstraksi *keypoint* dengan menerapkan algoritma *Scale Invariant Feature Transform* (SIFT) untuk proses registrasi *point cloud* dengan menggunakan algoritma *Iterative Closest Point* (ICP).

### Iterative Closest Point (ICP)

*Iterative Closest Point* (ICP) adalah metode registrasi *point cloud* yang digunakan untuk menghitung titik terdekat dari suatu objek geometri terhadap titik yang dijadikan sebagai referensi. Penerapan ICP membutuhkan dua

objek geometri yaitu “data” dan “model” yang telah diuraikan menjadi bentuk *point cloud* [3]. Algoritma ini secara berulang memperbaiki transformasi (translasi dan rotasi) yang bertujuan untuk meminimalkan jarak antara titik dari dua data *point cloud*. Selain dua *point cloud*, perkiraan awal dan kriteria batas perulangan juga dibutuhkan sebagai masukan dari algoritma ICP, sedangkan keluaran dari ICP adalah transformasi [8].

Diberikan *point cloud*  $P$  (model) dengan jumlah titik  $N_p$  dan *point cloud*  $X$  (target) dengan jumlah titik  $N_x$ , dimana  $X = \{x_i\}$  dengan  $i = 1 \dots N_x$  dan  $P = \{p_i\}$  dengan  $i = 1 \dots N_p$ . *Point cloud* model akan ditransformasikan menuju *point cloud* target sebagai koordinat referensi dimensi tiga. Tetapkan nilai awal *point cloud* data dengan kondisi  $P_0 = P$ ,  $R_0 = [I]$ ,  $t_0 = [3 \times 0]$  dan iterasi ( $k$ ) = 0. Proses registrasi *point cloud* selanjutnya akan dilakukan seperti pada langkah 1, 2, 3 dan 4 [9].

1. Menghitung korespondensi titik dengan mencari tetangga terdekatnya

$$d(p, x) = \|\vec{p} - \vec{x}\|^2 \dots (1)$$

2. Memperkirakan parameter transformasi menggunakan fungsi *mean square*

$$e(R_k, t_k) = \frac{1}{N_p} \sum_{i=1}^{N_p} \| (R_k \vec{p}_{i,0} + t_k) - \vec{y}_{i,k} \|^2 \dots (2)$$

3. Mentransformasikan titik-titik pada *point cloud* menggunakan parameter yang diperoleh pada langkah 2

$$\vec{p}_{i,k+1} = R_k \vec{p}_{i,0} + t_k \dots (3)$$

4. Terminasi iterasi

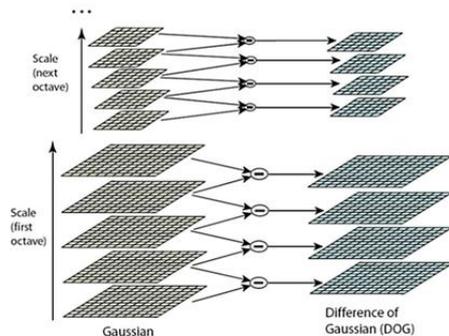
Iterasi berakhir jika jumlah iterasi yang dilakukan telah mencapai batas iterasi yang ditetapkan.

### Scale-Invariant Feature Transform (SIFT)

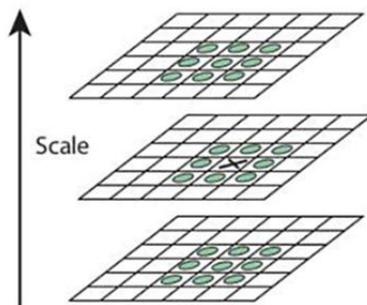
Algoritma SIFT digunakan untuk ekstraksi titik kunci (*keypoint*) dari masing-masing citra. *Keypoint* adalah sebuah titik yang menarik, yang mempunyai properti khusus di dalam *scene*, misalnya seperti sudut pada sebuah buku. Pencarian *keypoint* dari suatu *point cloud* menggunakan algoritma SIFT dilakukan dengan langkah berikut:

1. Mendeteksi nilai ekstrem pada ruang skala (*Scale – space extrema detection*)

Pencarian nilai ekstrem pada ruang skala merupakan tahap awal dalam ekstraksi keypoint. Tahap ini dilakukan dengan mencari fitur yang stabil di semua skala, menggunakan fungsi skala yang kontinu (*continuous function of scale*) yang dikenal dengan ruang skala (*scale space*).



Gambar 1: Deteksi Nilai Ekstrem pada Ruang Skala [7]



Gambar 2: Maksima dan minima dari citra difference-of-Gaussian [7]

Untuk setiap oktaf pada ruang skala, citra awal dikonvolusi secara berulang dengan Gaussian untuk menghasilkan himpunan citra ruang skala, yang ditunjukkan pada Gambar 1 sebelah kiri. Citra Gaussian yang berdekatan dicari selisihnya untuk menghasilkan citra *difference – of – Gaussian* seperti terlihat pada Gambar 1 sebelah kanan. Setelah setiap oktaf, citra Gaussian di *downsampling* dengan nilai faktor 2 dan proses ini dilakukan secara berulang, selanjutnya dicari ekstrem lokal (maksima dan minima) pada citra terhadap skala dan ruang. Jika suatu piksel merupakan ekstrem lokal (*local extrema*), maka piksel tersebut berpotensi sebagai *keypoint*.

Gambar 2 menunjukkan proses perolehan maksima dan minima dari citra *difference – of – Gaussian*.

Maksima dan minima dari citra *difference – of – Gaussian* diperoleh dengan membandingkan piksel yang ditandai dengan X, dengan 26 tetangganya di dalam daerah 3 x 3 pada skala tersebut dan skala yang berdekatan (ditandai dengan lingkaran).

## 2. Menentukan titik kunci (*Keypoint Localization*)

Setelah kandidat keypoint diperoleh melalui tahapan pada poin 1, langkah selanjutnya adalah menentukan keypoint dengan cara mengambil detail dari kandidat keypoint yang ada. Detail yang diambil berupa lokasi, skala dan rasio kelengkungan dari masing-masing kandidat keypoint. Pada tahap ini juga terjadi pengurangan jumlah kandidat keypoint dengan mengeliminasi kandidat keypoint yang dianggap rentan terhadap gangguan (*noise*). Eliminasi keypoint mengacu pada nilai ambang (*threshold*) yang ditetapkan. Jika nilai kontrasnya kurang dari nilai ambang, maka keypoint tersebut akan dieliminasi. Sehingga yang tersisa hanyalah keypoint yang kuat [7].

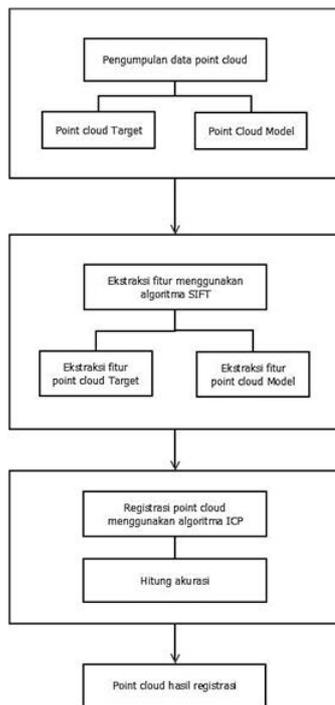
## Point Cloud Library (PCL)

*Point Cloud Library* (PCL) merupakan projek berskala besar yang dibuat untuk citra 2D / 3D serta pemrosesan *point cloud* data. Dari sudut pandang algoritmik, PCL dimaksudkan untuk menggabungkan banyak algoritma pemrosesan 3D yang beroperasi pada *point cloud* data, termasuk penyaringan (*filtering*), pencarian *keypoint*, estimasi fitur, rekonstruksi permukaan, segmentasi dan registrasi. Setiap rangkaian algoritma didefinisikan sebagai suatu kelas dasar. PCL dibangun sebagai sebuah framework untuk pemrosesan *point cloud* data yang dapat digunakan dengan bahasa pemrograman C++ [2].

## Metode Penelitian

Gambar 3 merupakan skema metode penelitian yang dilakukan. Tahap awal pada penelitian

ini adalah mendapatkan dataset dalam bentuk kumpulan titik dari citra sebagai *point cloud* Target dan *point Cloud* Model, kemudian mengekstraksi titik kunci (*keypoint*) dari masing-masing *point cloud* dan tahap akhir adalah tahap registrasi. Subjek penelitiannya adalah citra yang direpresentasikan dalam bentuk kumpulan titik (*point cloud*) yang diperoleh dari halaman web *Repository Point Cloud Library*.

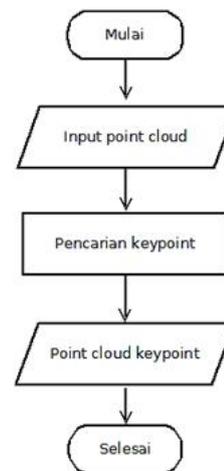


Gambar 3: Skema Metode Penelitian

Terdapat lima dataset yang digunakan yaitu dataset Tupperware, Chair, Desktop, Laptop, dan Bed. Dataset Tupperware merekam objek meja makan dengan peralatan makan, dataset Chair merekam dua buah kursi di ruang kantor, kemudian dataset Desktop merekam monitor dan keyboard di atas meja sedangkan dataset Laptop merekam laptop dan buku di atas meja, dan dataset Bed merekam objek tempat tidur termasuk lantai. Semua dataset direkam menggunakan kamera Asus Xtion Pro dengan jarak sekitar 3-4 meter dari objek utama, kemudian perlahan semakin dekat dan berputar di sekitar objek. Setiap dataset berisi citra 3 Dimensi yang masing-masing direpresentasikan dalam bentuk *point cloud* dan disimpan dalam dokumen dengan ekstensi pcd. Setiap titik pada *point cloud* mengandung informasi posisi pada koordinat x, y, z serta intensitas warna *Red*, *Green*, *Blue* (RGB).

## Ekstraksi Keypoint

*Keypoint* adalah sebuah titik yang menarik, yang mempunyai properti khusus di dalam scene, misalnya seperti sudut pada sebuah buku. Tahap ekstraksi *keypoint* diperlukan untuk menghasilkan kumpulan *keypoint* dari masing-masing *point cloud* yang digunakan, sebagai acuan dalam menentukan titik yang saling berkorespondensi. Gambar 4 merupakan ilustrasi tahap ekstraksi *keypoint* menggunakan algoritma *Scale – Invariant Feature Transform* (SIFT).



Gambar 4: Tahap Ekstraksi keypoint dengan algoritma SIFT

Penentuan *keypoint* diawali dengan memperkecil skala citra berulang kali dengan angka tertentu yang disebut dengan oktaf, sehingga menghasilkan kumpulan ruang skala (*scale space*). Setiap ruang skala dilakukan operasi konvolusi dengan filter Gaussian untuk membuat citra blur sehingga selisih dari ruang skala yang disebut dengan *difference-of-Gaussian* dapat ditemukan. Selanjutnya, dicari lokal ekstrema untuk menentukan titik yang menjadi kandidat *keypoint*. *Keypoint* dipilih berdasarkan ukuran stabilitasnya, *keypoint* dengan kualitas kontras rendah yang rentan terhadap noise dieliminasi dengan langkah-langkah sebagai berikut:

1. Meningkatkan lokalisasi titik ekstrim dengan akurasi subpiksel menggunakan ekspansi Taylor dari fungsi skala-ruang,  $D(x, y, \sigma)$ , digeser sehingga titik asal berada pada titik sampel, sehingga diperoleh [5]:

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \dots \quad (4)$$

Dimana  $D$  dan turunannya ditaksir dari titik sampel, dan  $(x, y, \sigma)^T$  adalah *offset* dari titik ini.

2. Lokasi titik ekstrim,  $\hat{x}$ , diperoleh dengan menurunkan persamaan 4 terhadap  $x$ , sehingga [5]:

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \dots \quad (5)$$

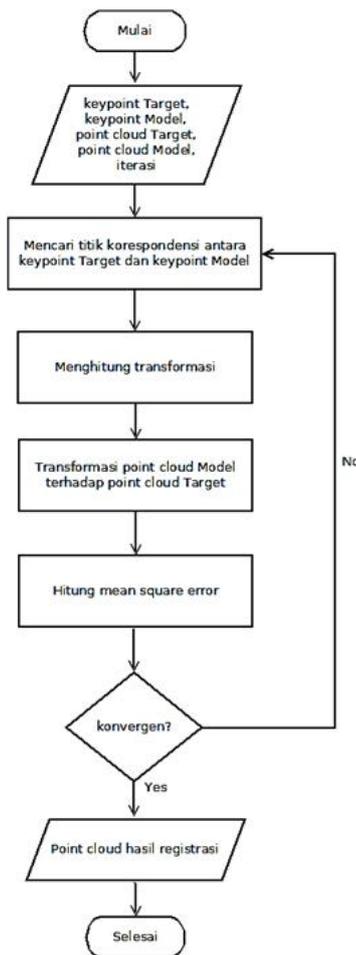
3. Nilai fungsi pada titik ekstrem  $D(\hat{x})$ , diperoleh dengan mensubstitusi persamaan 5 pada persamaan 4 sehingga menghasilkan [5]:

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} \dots \quad (6)$$

Pada penelitian ini, semua titik ekstrem dengan nilai  $|D(\hat{x})|$  kurang dari 0.001 akan dieliminasi.

### Registrasi Point Cloud

Gambar 5 merupakan ilustrasi dari proses registrasi *point cloud* setelah mengekstraksi *keypoint* dari *point cloud* Target dan *point cloud* Model.



Gambar 5: Tahap Registrasi point cloud dengan algoritma ICP

Dibutuhkan lima nilai input untuk proses registrasi *point cloud* yang dibangun pada penelitian ini yaitu *point cloud* Target, *point cloud* Model, *keypoint* Target, *keypoint* Model dan parameter iterasi sebagai kriteria konvergensi untuk proses registrasi. Tahap pertama yang dilakukan adalah mencari titik yang saling berkorespondensi berdasarkan jarak antara titik di *keypoint* Target dengan pasangannya di *keypoint* Model. Korespondensi dari setiap titik di *keypoint* Target adalah titik pada *keypoint* Model yang memiliki jarak *euclidean* paling dekat 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 7.

$$d(\vec{r}_1, \vec{r}_2) = \|\vec{r}_1 - \vec{r}_2\|$$

$$= \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \dots \quad (7)$$

Berdasarkan titik yang saling berkorespondensi, tahap selanjutnya adalah menghitung transformasi yang harus dilakukan *point cloud* Model agar sejajar dengan *point cloud* Target. Transformasi terdiri dari rotasi dan translasi yang menentukan perputaran serta pergeseran *point cloud* Model, sehingga nilai *error* yang merupakan jarak antara titik-titik yang saling berkorespondensi bisa diminimalkan.

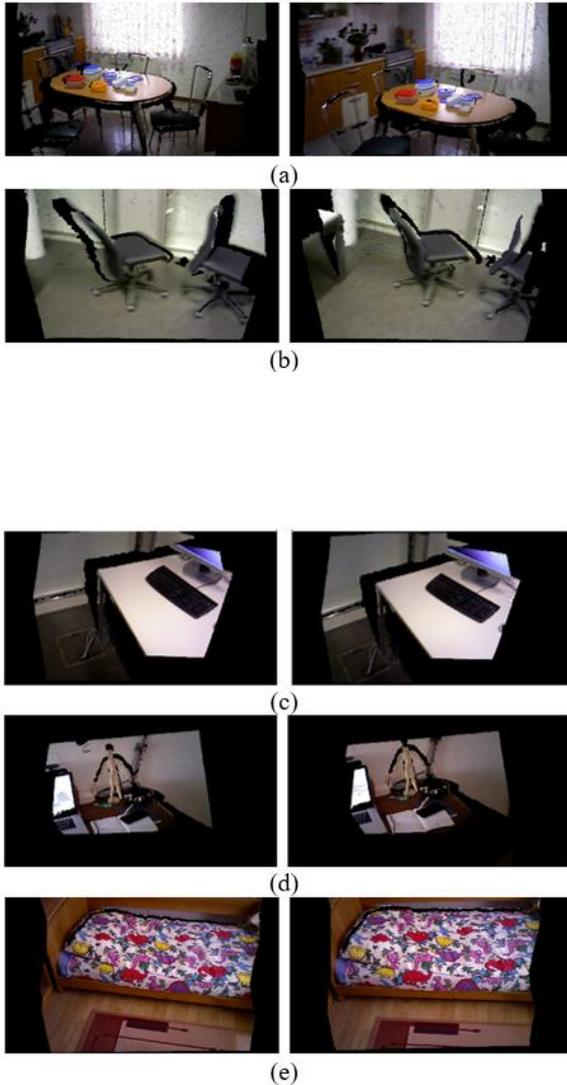
Pada tahap ketiga, *point cloud* Model di-transformasi dengan matriks transformasi yang diperoleh di tahap sebelumnya. Selanjutnya nilai *mean square error* dihitung sebagai nilai *error* untuk iterasi pertama. Langkah ini dilakukan terus menerus sampai kriteria konvergensi terpenuhi, yaitu ketika jumlah iterasi yang dilakukan telah mencapai batas iterasi yang ditetapkan. Terdapat lima batas iterasi yang digunakan yaitu 1, 5, 10, 15 dan 20 iterasi.

### Hasil dan Pembahasan

Registrasi *point cloud* pada penelitian ini diaplikasikan kepada lima 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

Input data *point cloud* dilakukan untuk menentukan *point cloud* yang ingin diregistrasi. Data *point cloud* dari dataset yang digunakan pada penelitian ini dapat dilihat pada Gambar 6.



Gambar 6: Data point cloud dari dua sudut pandang pada dataset Tupperware (a), dataset Chair (b), dataset Desktop (c), dataset Laptop (d), dan dataset Bed (e).

Gambar 6 menunjukkan dua *point cloud* dari masing-masing dataset yang digunakan pada proses registrasi. *Point cloud* di sisi kiri pada Gambar 6 merupakan *point cloud* Target sedangkan sisi kanan merupakan *point cloud* Model dari masing-masing dataset seperti yang dijelaskan pada Tabel 1.

Tabel 1: Data point cloud dari dataset Tupperware, dataset Chair, dataset Desktop, dataset Laptop dan dataset Bed

Dataset	Point Cloud Target		Point Cloud Model	
	Nama File (.pcd)	Gambar	Nama File (.pcd)	Gambar
Tupperware	frame_00021	(a) kiri	frame_00024	(a) kanan
Chair	frame_0	(b) kiri	frame_2	(b) kanan
Desktop	frame_00010	(c) kiri	frame_00015	(c) kanan
Laptop	frame_00005	(d) kiri	frame_00020	(d) kanan
Bed	frame_15	(e) kiri	frame_30	(e) kanan

Berdasarkan Tabel 1, *point cloud* yang digunakan pada uji coba penelitian ini adalah frame\_00021.pcd dan frame\_00024.pcd dari dataset Tupperware, frame\_0.pcd dan frame\_2.pcd dari dataset Chair, frame\_00010.pcd dan frame\_00015.pcd dari dataset Desktop, frame\_00005 dan frame\_00020 dari dataset Laptop, serta frame\_15.pcd dan frame\_30.pcd dari dataset Bed.

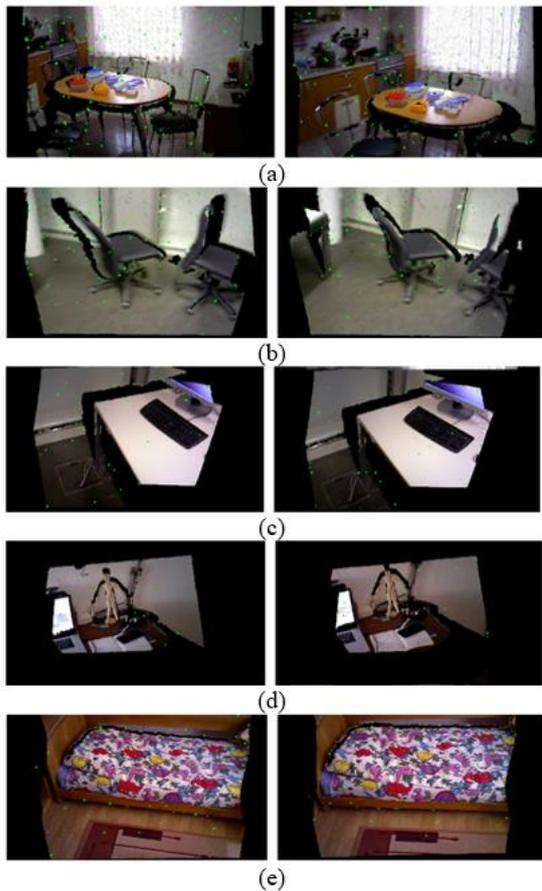
## B. Hasil Ekstraksi Keypoint

Gambar 7 menunjukkan visualisasi *keypoint* yang diperoleh dari suatu *point cloud*. *Keypoint* yang dihasilkan merupakan titik-titik yang dapat mewakili *point cloud* secara keseluruhan.



Gambar 7: Hasil tahap ekstraksi Keypoint

Keypoint yang diperoleh direpresentasikan dengan titik yang berwarna hijau seperti yang terlihat pada Gambar 7. Hal ini dilakukan untuk membedakan *keypoint* dengan titik lainnya dalam suatu *point cloud*. Gambar 8 merupakan visualisasi *keypoint* yang diperoleh pada *point cloud* Target dan *point cloud* Model untuk masing-masing dataset yang digunakan.



Gambar 8: Hasil ekstraksi keypoint pada point cloud Target dan point cloud Model pada dataset Tupperware (a), dataset Chair (b), dataset Desktop (c), dataset Laptop (d), dan dataset Bed (e).

Pada setiap kali proses registrasi menghasilkan dua kumpulan keypoint yaitu keypoint untuk point cloud Target dan keypoint untuk point cloud Model. Tabel 2 menunjukkan jumlah titik yang diperoleh dari tahap ekstraksi keypoint dan jumlah titik asli sebelum tahap ekstraksi keypoint.

Tabel 2: Perbandingan jumlah titik point cloud

Dataset	Point Cloud Target		Point Cloud Model	
	Nama File (.pcd)	Gambar	Nama File (.pcd)	Gambar
Tupperware	frame_00021	(a) kiri	frame_00024	(a) kanan
Chair	frame_0	(b) kiri	frame_2	(b) kanan
Desktop	frame_00010	(c) kiri	frame_00015	(c) kanan
Laptop	frame_00005	(d) kiri	frame_000020	(d) kanan
Bed	frame_15	(e) kiri	frame_30	(e) kanan

Berdasarkan Tabel 2, dapat dinyatakan

bahwa jumlah titik keypoint yang dihasilkan lebih sedikit dibandingkan point cloud aslinya. Kedua keypoint inilah yang menjadi acuan untuk menentukan titik yang saling berkorespondensi dalam registrasi antara point cloud Target dengan point cloud Model.

### C. Hasil Registrasi Point Cloud

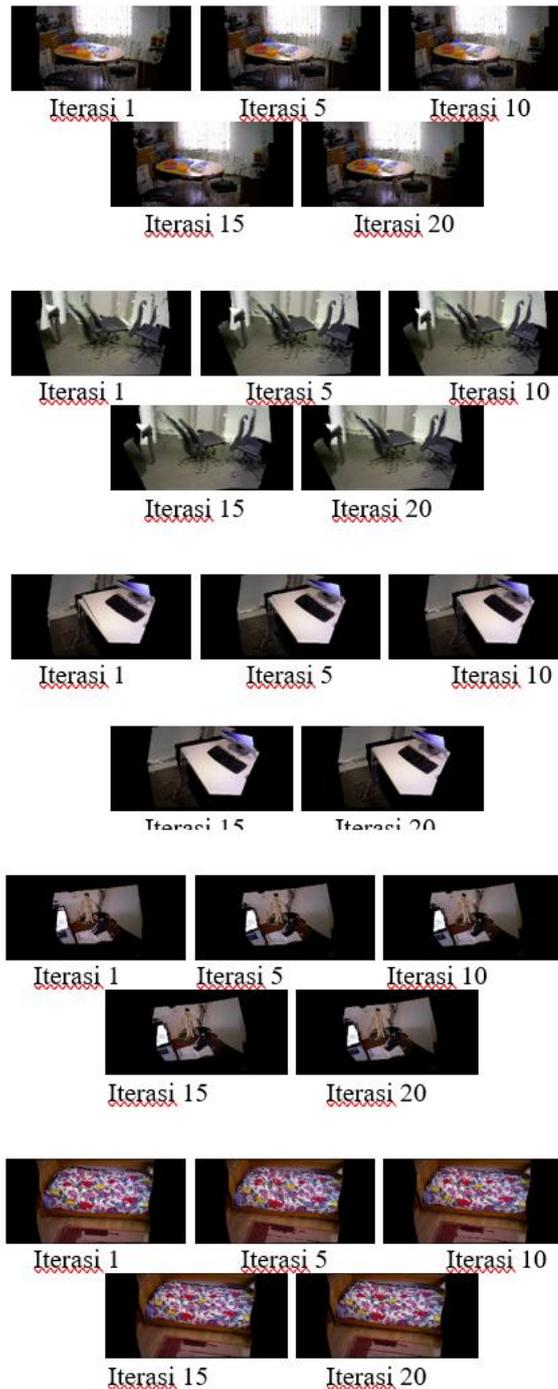
Tahap registrasi *point cloud* dilakukan dengan mengaplikasikan algoritma ICP menggunakan *keypoint* yang dihasilkan dari algoritma SIFT sebagai acuan dalam menentukan titik yang saling berkorespondensi. Pada masing-masing dataset, implementasi dilakukan sebanyak lima kali pengujian dengan parameter iterasi yang berbeda, yaitu 1, 5, 10, 15, dan 20, yang berarti bahwa dalam satu kali registrasi, pencarian titik yang saling berkorespondensi, penentuan transformasi, serta proses transformasi point cloud Model diulangi sebanyak parameter yang digunakan agar *point cloud* Model dapat sejajar dengan *point cloud* Target.

Gambar 9 menunjukkan potongan bagian dari hasil registrasi *point cloud* pada dataset Desktop dengan iterasi 1, 5, 10, 15 dan 20, dilihat dari kiri ke kanan. Berdasarkan hasil yang disajikan pada Gambar 9, sudut meja dari kedua point cloud semakin sejajar seiring dengan bertambahnya jumlah iterasi. Hasil registrasi *point cloud* pada masing-masing dataset disajikan pada Gambar 10.



Gambar 9: Hasil registrasi pada dataset Desktop dengan iterasi 1, 5, 10, 15, 20 dari kiri ke kanan

Hasil registrasi *point cloud* pada Gambar 10 menunjukkan bahwa jumlah iterasi mempengaruhi *point cloud* hasil registrasi. Berdasarkan Gambar 10, registrasi *point cloud* dengan jumlah iterasi sebanyak 20 kali dapat menghasilkan selisih jarak terkecil antara *point cloud* Model dengan *point cloud* Target untuk masing-masing dataset. Oleh karena itu, dapat dinyatakan bahwa semakin banyak iterasi yang dilakukan, maka selisih jarak antara *point cloud* Target dan *point cloud* Model semakin kecil.



Gambar 10: Hasil registrasi point cloud pada dataset Tupperware, dataset Chair, dataset Desktop, dataset Laptop dan dataset Bed

### 1. Evaluasi Nilai Mean Square Error (MSE)

Untuk membuktikan validitas metode yang diusulkan, proses registrasi *point cloud* pada masing-masing dataset juga dilakukan menggunakan algoritma tradisional *Iterative Closest*

*Point* (ICP). Evaluasi algoritma S-ICP dengan tradisional ICP dilakukan dengan menghitung nilai *Mean Square Error* (MSE) yang 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 perbandingan nilai MSE dari algoritma S-ICP dan ICP pada setiap iterasi untuk masing-masing dataset.

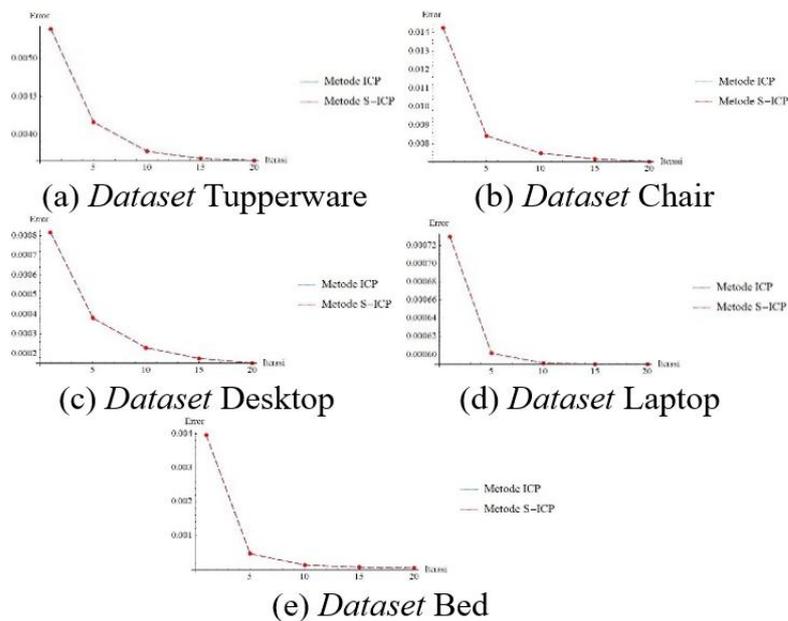
Tabel 3: Perbandingan jumlah titik point cloud

Iterasi	Tupperware		Chair		Desktop		Laptop		Bed	
	ICP	SIFT – ICP (S – ICP)	ICP	SIFT – ICP (S – ICP)	ICP	SIFT – ICP (S – ICP)	ICP	SIFT – ICP (S – ICP)	ICP	SIFT – ICP (S – ICP)
1	0.00539137	0.00539137	0.01425500	0.01425500	0.000817112	0.000817112	0.000729853	0.000729853	0.0039687600	0.0039687600
5	0.00416105	0.00416105	0.00844323	0.00844323	0.000380852	0.000380852	0.000601701	0.000601701	0.0004690490	0.0004690490
10	0.00377612	0.00377612	0.00750350	0.00750350	0.000230135	0.000230135	0.000590646	0.000590646	0.0001340380	0.0001340380
15	0.00368228	0.00368228	0.00718791	0.00718791	0.000175255	0.000175255	0.000589675	0.000589675	0.0000671798	0.0000671798
20	0.00365674	0.00365674	0.00705454	0.00705454	0.000152243	0.000152243	0.000589590	0.000589590	0.0000480026	0.0000480026

Berdasarkan data yang disajikan, diketahui bahwa proses registrasi *point cloud* untuk dataset Tupperware menggunakan algoritma ICP dan SIFT-ICP menghasilkan nilai MSE yang sama pada setiap iterasi. Jumlah iterasi yang semakin banyak dalam melakukan registrasi *point cloud* menghasilkan nilai MSE yang semakin kecil. Nilai MSE terkecil dari dataset Tupperware diperoleh pada jumlah iterasi ke 20 untuk kedua algoritma, yaitu 0.00365674. Hal serupa juga terjadi pada dataset Chair, Desktop, Laptop dan Bed yang memiliki nilai

MSE yang sama pada setiap iterasi, dalam proses registrasi menggunakan algoritma ICP dan SIFT-ICP. Nilai MSE terkecil diperoleh masing-masing dataset pada jumlah iterasi ke 20 untuk kedua algoritma, yaitu 0.00705454 untuk dataset Chair, 0.000152243 untuk dataset Desktop, 0.000589590 untuk dataset Laptop, dan 0.00705454 untuk dataset Bed.

Data perbandingan dari nilai MSE proses registrasi menggunakan algoritma ICP dan SIFT-ICP juga disajikan dalam bentuk grafik seperti yang dapat dilihat pada Gambar 11.



Gambar 11: Grafik perbandingan nilai MSE proses registrasi pada masing-masing dataset

Grafik pada Gambar 11 menunjukkan semakin banyak iterasi, semakin kecil nilai MSE. Nilai MSE yang semakin kecil menyatakan kedua point cloud yang digunakan pada proses

registrasi semakin sejajar, sehingga registrasi point cloud memiliki tingkat kepercayaan yang tinggi. Berdasarkan pada nilai MSE antara algoritma tradisional ICP dan SIFT-ICP, diper-

oleh nilai yang sama, sebagai contoh registrasi untuk dataset Tupperware pada iterasi 1 memiliki nilai MSE yang sama untuk kedua algoritma, yaitu 0.00539137. Oleh karena itu, dapat dinyatakan bahwa ekstraksi *keypoint* menggunakan algoritma SIFT sebagai dasar dalam proses registrasi tidak merubah objek dari *point cloud*.

## 2. Evaluasi Waktu Komputasi

Evaluasi waktu komputasi dilakukan setelah menyelesaikan proses registrasi *point cloud* menggunakan algoritma SIFT-ICP dan tradisional ICP pada perangkat dengan processor AMD A9-9425 RADEON R5. Waktu komputasi merupakan durasi proses registrasi dari kedua algoritma dengan menggunakan dataset dan jumlah iterasi yang sama untuk setiap registrasi.

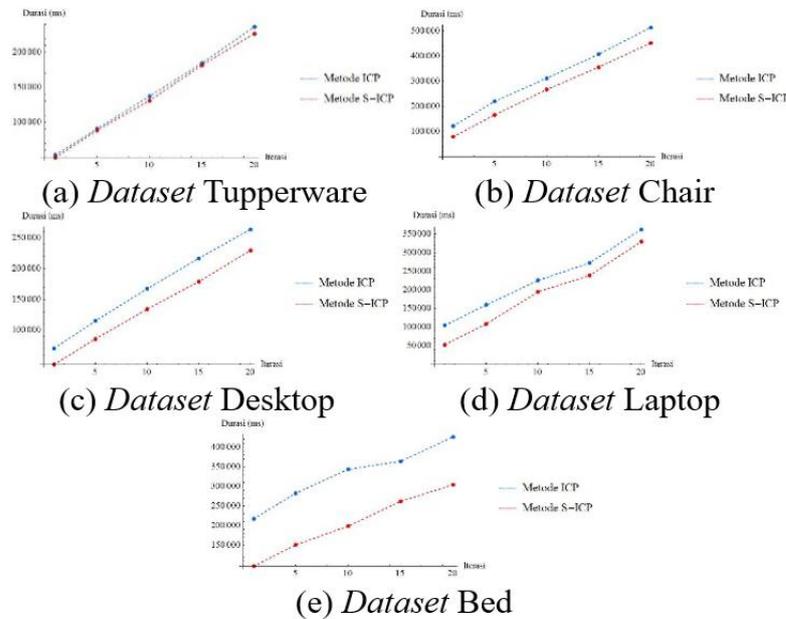
Tabel 4: Perbandingan durasi proses registrasi dalam satuan millisecond (ms)

Iterasi	Tupperware		Chair		Desktop		Laptop		Bed	
	ICP	SIFT – ICP (S – ICP)	ICP	SIFT – ICP (S – ICP)	ICP	SIFT – ICP (S – ICP)	ICP	SIFT – ICP (S – ICP)	ICP	SIFT – ICP (S – ICP)
1	52833	49255	122412	79453	70674	44460	104741	53200	217939	96782
5	90453	88581	220687	166624	115369	86130	159740	108732	282747	151149
10	137299	131116	312669	267481	167376	134293	225134	194391	343864	198749
15	184435	181795	408053	356959	217052	179186	272127	238533	364017	262015
20	236772	226903	514641	452157	264047	229917	362196	329132	426417	304373

Tabel 4 menunjukkan perbandingan durasi waktu registrasi menggunakan algoritma SIFT-ICP dan ICP pada setiap iterasi untuk masing-masing dataset. Berdasarkan data pada Tabel 4, melakukan registrasi point cloud menggunakan algoritma SIFT-ICP membutuhkan durasi yang lebih singkat dibandingkan dengan algoritma tradisional ICP. Hal ini berlaku untuk semua point cloud pada dataset yang digunakan, yaitu Tupperware, Chair, Desktop, Laptop dan Bed. Selain itu, pada masing-masing dataset, durasi terpanjang diperoleh ketika melakukan registrasi point cloud dengan jumlah iterasi sebanyak 20 kali, baik dengan menggunakan algoritma SIFT-ICP maupun menggunakan algoritma ICP. Perbandingan durasi waktu registrasi antara algoritma tradisional ICP dan SIFT-ICP untuk masing-masing dataset juga disajikan pada grafik yang tertera di Gambar 12. Grafik tersebut sekaligus menunjukkan keterkaitan antara jumlah iterasi terhadap durasi registrasi.

Dengan mengacu pada grafik di Gambar 12, terdapat dua nilai yang mempengaruhi durasi waktu yang dibutuhkan untuk registrasi *point cloud* pada masing-masing dataset, yaitu jumlah iterasi dan jumlah titik pada *point cloud input* yang digunakan pada proses registrasi.

Gambar 12 menunjukkan semakin banyak jumlah iterasi yang dilakukan, maka semakin lama durasi yang dibutuhkan untuk menyelesaikan proses registrasi. Perbandingan durasi berdasarkan tabel dan grafik yang telah disajikan juga menunjukkan bahwa proses registrasi menggunakan algoritma SIFT-ICP pada semua dataset membutuhkan durasi yang lebih singkat dibandingkan dengan algoritma tradisional ICP. Hal ini disebabkan karena algoritma SIFT-ICP hanya mencari titik yang saling berkorespondensi antar *keypoint*, sedangkan algoritma ICP mencari titik-titik yang saling berkorespondensi pada semua titik di point cloud.



Gambar 12: Grafik perbandingan durasi proses registrasi pada masing-masing dataset

Berdasarkan Gambar 12, ditemukan keunikan tersendiri pada hasil registrasi untuk dataset Tupperware. Melakukan registrasi menggunakan algoritma SIFT-ICP dan ICP pada dataset Tupperware menghasilkan durasi waktu registrasi yang hampir sama. Hal ini terjadi karena dalam proses registrasi yang dilakukan, transformasi diaplikasikan untuk semua titik di *point cloud* Model, yang merupakan *point cloud* input asli.

## Penutup

Berdasarkan hasil uji coba yang dapat dilihat pada grafik perbandingan durasi dan nilai MSE antara kedua algoritma registrasi, dapat disimpulkan bahwa tahap ekstraksi *keypoint* menggunakan algoritma *Scale – Invariant Feature Transform* (SIFT) dapat mereduksi jumlah titik objek hingga 99%. Kombinasi algoritma SIFT sebelum meregistrasi *point cloud* menggunakan algoritma *Iterative Closest Point* (ICP) mampu mengurangi durasi proses registrasi tanpa merubah objek asli dari *point cloud*.

## Daftar Pustaka

- [1] R. Adipranata dan R. Lim, “Rekonstruksi Obyek 3D dari Gambar 2D dengan Metode
- [2] R. B. Rusu and S. Cousins, “3d is here: Point Cloud Library (PCL). IEEE International Conference on Robotics and Automation. Mei 2011. IEEE. 1-4, 2011.
- [3] P. J. Besl and 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] Z. H. Nejad and M. N, “An adaptive image registration method based on SIFT features and RANSAC transform”, Computers and Electrical Engineering 62(C): 524-537, 2017.
- [5] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60(2): 91-110, 2004.
- [6] Herbert Bay, T. Tuytelaars, and L. V. Gool, “SURF: Speeded Up Robust Features”, Proc. of the 9th European Conference on Computer Vision (ECCV’06). Mei 2006. Springer. 404-417, 2006.
- [7] P. P. R. Filho, F. D. L. Moreira, F. G. L. Xavier, S. L. Gomes, J. C. Santos, F. N.

Generalized Voxel Coloring”, Prosiding Seminar Nasional KOMMIT, 2006.