

## ESTIMASI POSISI DENGAN MENGGUNAKAN KAMERA MONOKULAR

Hadha Afrisal<sup>1,\*</sup>, Indah Soesanti<sup>2</sup>, dan Adha Imam Cahyadi<sup>2</sup>

<sup>1</sup>)Departemen Teknik Elektro, Universitas Diponegoro Semarang  
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

<sup>2</sup>)Departemen Teknik Elektro dan Teknologi Informasi, Universitas Gadjah Mada  
Jl. Grafika no. 2, Kampus UGM Bulaksumur, Yogyakarta

\*E-mail: hadha.afrisal@elektro.undip.ac.id

### Abstrak

Estimasi posisi dengan pendekatan ciri visual pada umumnya dapat dilakukan dengan menggunakan 2 kamera (stereoskopik) atau kamera 3D yang sudah dilengkapi dengan sensor kedalaman (*depth*). Teknik-teknik tersebut membutuhkan sensor lebih banyak sehingga harga perangkat keras sensor untuk aplikasi estimasi posisi menjadi lebih mahal. Sebagai alternatifnya, pada penelitian ini estimasi posisi akan dilakukan dengan menggunakan teknik monokular (kamera tunggal) untuk simplifikasi perangkat keras, yakni dengan pendekatan metode *sequential multiple view geometry*. Pada aplikasi robot bergerak (*mobile robot*), teknik estimasi posisi diperlukan untuk bernavigasi sebagai fondasi pengembangan kapabilitas *localization* dan *mapping* secara bersamaan. Penelitian ini bertujuan untuk merancang dan mensimulasikan algoritma estimasi posisi yang merupakan langkah awal pada *monocular SLAM* (*Simultaneous Localization and Mapping*) 2-dimensi. Penelitian ini membahas tentang fungsi kamera sebagai sensor posisi untuk *mobile robot* dengan menitikberatkan diskusi pada transformasi citra dengan menggunakan model kamera *pinhole* beserta parameter intrinsik dan ekstrinsiknya. Berdasarkan hasil pengujian, estimasi pose dan posisi terhadap citra papan catur (*checkerboard*) berhasil memetakan jarak dan sudut dengan cukup akurat serta menghasilkan galat estimasi yang cukup kecil, yakni 2,9 mm untuk estimasi jarak dan 0,3° untuk estimasi sudut. Simulasi algoritma estimasi posisi juga telah dilakukan secara *offline* dengan menggunakan 1200 buah *image sequences* yang diekstraksi dari dataset citra video.

*Kata kunci: estimasi posisi, fitur titik, kamera monokular*

### Abstract

Position estimation of using visual features is generally performed by utilizing stereo camera or 3D camera which is equipped with a depth sensor. Those techniques require more sensors hence can increase the cost of hardware for this application. As an alternative, this experiment introduces position estimation of using monocular technique (single camera) to simplify the hardware requirement, which is done by using the method of sequential multiple view geometry. This technique is mainly important for robot to build navigation algorithm and performing simultaneous localization and mapping. This research aims to design and to simulate a position estimation algorithm as the fundamental aspect to equip the robot with a capability of doing 2-D monocular SLAM (*Simultaneous Localization and Mapping*). This research will discuss about the function of monocular camera as a position sensor for mobile robot, especially the use of pinhole camera model with its intrinsic and extrinsic parameter. The experiment shows that the pose and position estimation using checkerboard is quite accurate with a relatively small estimation error, which is 2,9 mm for distance estimation and 0,3° for angle estimation. An offline simulation has also been demonstrated using 1200 image sequences from a video images dataset.

*Keywords: position estimation, point features, monocular camera*

### 1. Pendahuluan

Dibandingkan dengan kompas digital dan IMU (*Inertial Measurement Unit*), estimasi posisi berbasis visual dinilai lebih unggul karena lebih kebal terhadap derau akibat medan magnet dan isolasi bangunan, serta memiliki kepresisian yang relatif lebih tinggi [1]. Kompas digital

mengukur orientasi robot relatif terhadap kutub bumi, sedangkan IMU mengestimasi perpindahan secara relatif terhadap posisi awal. Dengan menggunakan pendekatan visual berbasis kamera, posisi dan orientasi sensor terhadap lingkungan dapat diestimasi secara bersamaan. Estimasi posisi dan orientasi melalui pendekatan visual pada umumnya dapat dilakukan dengan menggunakan

kamera stereo atau kamera 3 dimensi yang sudah dilengkapi dengan sensor kedalaman (*depth*). Kamera stereo banyak digunakan untuk aplikasi estimasi posisi di luar ruangan (*outdoor*) karena lebih kebal terhadap variasi pencahayaan. Sedangkan kamera 3 dimensi lebih banyak digunakan untuk aplikasi estimasi posisi di dalam ruangan dimana derau akaibat cahaya *infra-red* lebih minim [2].

Teknik kamera jamak (*multi-camera*) cukup efisien dan presisi dalam memetakan ciri visual untuk aplikasi estimasi posisi dan orientasi, seperti yang telah dilakukan dalam beberapa penelitian terdahulu [1-3]. Namun pada prakteknya, teknik ini membutuhkan gabungan beberapa sensor sehingga lebih mahal untuk diimplementasikan secara luas pada produk robot bergerak untuk keperluan komersial, seperti misalnya robot untuk inspeksi di lingkungan industri [3]. Selain itu, penggabungan beberapa jenis sensor, seperti IMU, kompas digital, dan sonar untuk estimasi posisi dan orientasi juga memiliki tantangan dalam hal *data fusion* yang cukup rumit [4].

Untuk itu, sebagai alternatif pengembangan teknik estimasi posisi dan orientasi berbasis visual, pada penelitian ini akan digunakan teknik monokular (kamera tunggal) dengan pendekatan *sequential multiple view geometry*. Teknik ini lebih sederhana dalam segi perangkat keras, serta dapat memiliki prospek yang bagus untuk diaplikasikan secara luas pada *consumer electronics* dan robot bergerak dengan harga yang lebih terjangkau karena hanya menggunakan 1 buah kamera saja.

Metode estimasi pose dan posisi dengan menggunakan kamera monokular pada robot bergerak merupakan salah satu dasar untuk mengembangkan teknik *Simultaneous Localization and Mapping* (SLAM). *Localization* merupakan proses yang dilakukan robot untuk memposisikan diri (lokalisir) pada suatu lingkungan baru, sedangkan *mapping* merupakan proses untuk membangun peta memori terhadap lingkungan baru tersebut. Pada navigasi robot, proses *Localization* dan *Mapping* yang dilakukan bersamaan akan membentuk kemampuan *Simultaneous Localization and Mapping* (SLAM). Konsep SLAM tersebut pertama kali diperkenalkan oleh Hugh Durrant-Whyte dan John J. Leonard berdasarkan penelitian Smith, Self, dan Cheesman [5].

Pada robot yang menggunakan kamera sebagai sensor utama, teknik SLAM dapat diterapkan dengan mengambil ciri visual untuk dijadikan pedoman navigasi. Ciri visual tersebut dapat berupa titik, *corner* (sudut), *edge* (tepi), garis, maupun ciri-ciri buatan lain dari metode seperti SIFT, SURF, HOG, dan sebagainya [6]. Ciri-ciri tersebutlah yang kemudian dijadikan sebagai *landmark* (penanda) visual untuk mengestimasi posisi dan orientasi kamera relatif terhadap kelompok ciri visual. Dalam hal ini, fungsi kamera adalah sebagai sensor posisi yang secara terus menerus memprediksi dan mengestimasi jarak dan posisi kamera relatif terhadap titik acuan visual

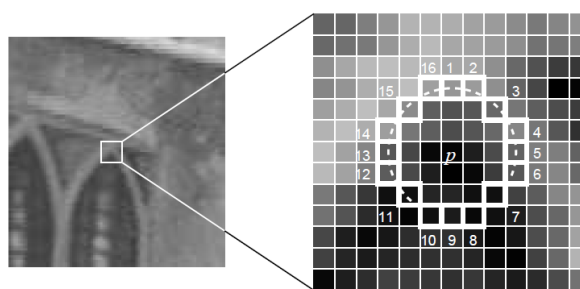
pada lingkungan tersebut. Ada banyak jenis metode visual SLAM, salah satunya adalah *monocular SLAM*, yaitu teknik *visual SLAM* yang dilakukan hanya dengan satu buah kamera saja. Penelitian ini berfokus pada pembahasan mengenai penggunaan kamera dan proses transformasi citra dalam proses SLAM monokular tersebut, serta mendemonstrasikan fisibilitas penggunaan teknik monokular untuk estimasi posisi dan orientasi di dalam ruangan dengan objek dan setting tertentu.

## 2. Metode

### 2.1. Ekstraksi Ciri Visual

Ciri atau fitur yang dapat digunakan sebagai penanda visual sebagai acuan posisi dan orientasi dapat berbentuk titik, *corner* (sudut), *edge* (tepi), garis, maupun ciri-ciri buatan lain seperti SIFT, SURF, HOG, dan sebagainya [6]. Pada eksperimen yang dilakukan, ciri yang digunakan adalah berbentuk *corner* (sudut). Pendeteksi sudut merupakan salah satu algoritma penting pada proses *monocular SLAM* guna memetakan *interest point* berupa ciri sudut. Banyak teknik yang digunakan sebagai *corner detector* antara lain Harris [7], SUSAN (*Smallest Univalued Segment Assimilating Nucleus*) [8], FAST (*Features from Accelerated Segment Test*) [9], dan teknik-teknik turunannya yang lain.

Masing-masing teknik tersebut memiliki keunggulan, namun dalam kaitannya dengan proses *monocular SLAM*, salah satu pertimbangan pemilihan algoritma *corner detector* adalah memiliki kecepatan pemrosesan yang cepat serta memiliki ripitabilitas yang baik. Metode FAST *corner detector* merupakan salah satu yang cukup cepat dan efektif dibandingkan dengan *corner detector* yang lain [9], sehingga pada penelitian ini FAST *corner detector* dipilih sebagai algoritma untuk mengekstraksi ciri berupa *corner*.



Gambar 1. *Corner detector* dengan menggunakan metode FAST [9]

FAST (*Features from Accelerated Segment Test*) *corner detector* bekerja dengan cara menguji kandidat sudut sebagai titik yang dikelilingi oleh 16 buah piksel yang membentuk lingkaran sebagai piksel pengujian kebenaran ciri sudut, seperti diilustrasikan pada Gambar 1. Caranya adalah dengan membandingkan tingkat kecerahan 16 buah piksel pengujian tersebut. Nilai ke-16 buah piksel

penguji tersebut dibandingkan kecerahannya dengan piksel  $p$  sebagai titik uji, kemudian dikelompokkan menjadi pola terang dan pola gelap untuk menentukan ciri *corner*. Proses penentuan apakah sebuah piksel dapat dianggap sebagai *corner* atau tidak ditentukan oleh proses komparasi deret cincin. Ada beberapa parameter utama pada FAST yaitu, (1) Titik uji *corner feature*,  $p$ , (2) Radius piksel uji cincin,  $r$ , (3) *Threshold* intensitas,  $t$ , dan (4) Jumlah *contiguous pixels* (piksel-piksel yang bersebelahan dan memiliki ciri yang serupa, terang atau gelap),  $N$ . Nilai  $N$  dapat diatur sesuai dengan kebutuhan pengguna. Menurut eksperimen Rosten dan Drummond [9], nilai  $N$  yang paling efektif adalah 9.

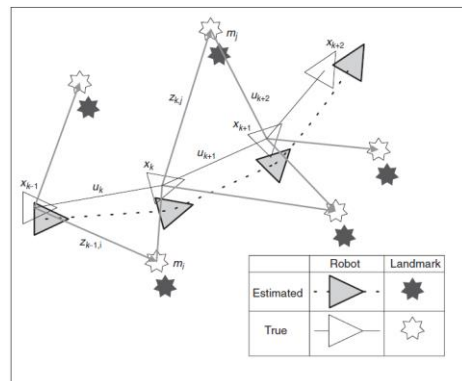
Secara garis besar, ada 3 proses yang dilakukan algoritma FAST untuk mendeteksi sebuah ciri *corner*, yaitu,

1. Proses inisialisasi parameter  
Proses ini berperan untuk mengatur nilai-nilai parameter radius ( $r$ ), *threshold* ( $t$ ), dan jumlah *contiguous pixels* ( $N$ ).
2. Proses deteksi  
Jika ada lebih dari atau sama dengan  $N$  buah piksel pada rentet cincin uji memiliki ciri lebih terang atau lebih gelap dibanding nilai intensitas  $p$ , maka diasumsikan titik  $p$  adalah *corner feature*. Kemudian selanjutnya dilakukan *rapid rejection* pada empat buah piksel uji pada lokasi 1, 5, 9, dan 13. Jika salah setidaknya ada 3 buah piksel dari ke-4 buah piksel uji *rapid rejection* memenuhi kriteria *threshold* (lebih terang atau lebih gelap dibanding intensitas titik  $p$ ), maka benar titik  $p$  merupakan *corner feature*.
3. *Non-maximal suppression*  
Proses ini berguna untuk mengeliminasi beberapa *corner feature* yang bersebelahan sehingga dapat dianggap sebagai sebuah *corner feature* yang sama. Tahapan awal pada proses ini adalah menghitung *score function*,  $V$ . *Score function* merupakan jumlah absolut dari selisih nilai piksel  $p$  dengan nilai piksel pada cincin. Kemudian jika dideteksi ada *corner feature* yang bersebelahan, maka *corner feature* dengan nilai  $V$  yang lebih kecil akan dieliminasi. Rumusan matematisnya dijelaskan pada persamaan

$$V = \max \left\{ \begin{array}{l} \sum (pix\ val - p) \text{ jika } (value - p) > t \\ \sum (p - pix\ val) \text{ jika } (p - value) > t \end{array} \right\} \quad (1)$$

## 2.2. Simultaneous Localization and Mapping (SLAM)

*Simultaneous Localization and Mapping* (SLAM) merupakan suatu proses dimana robot bergerak memetakan wilayah yang dilaluinya ke dalam memori dan sekaligus pada waktu yang bersamaan menggunakan informasi tersebut untuk memperkirakan posisinya saat itu [10]. Pada perkembangannya, proses tersebut menjadi suatu teknik yang umum digunakan pada robot bergerak sebagai metode untuk bernavigasi. Proses SLAM diilustrasikan pada Gambar 2.



Gambar 2. Proses SLAM dan Visualisasi Korelasi Antar Landmark

Pada penelitian ini, dengan menggunakan penanda berupa ciri *corner*, korelasi antar *landmark* kemudian dapat diestimasi dan dipetakan menggunakan pendekatan metode bayesian [11]. Pada konstanta waktu  $k$ , beberapa parameter dapat didefinisikan sebagai berikut,

1.  $x_k$  : yaitu *state vector* yang merepresentasikan lokasi dan arah dari sebuah *mobile robot*.
2.  $u_k$  : yaitu *control vector* yang digunakan untuk menggerakkan *mobile robot* menuju *state* selanjutnya.
3.  $m_i$  : merupakan vektor yang merepresentasikan lokasi *landmark* ke- $i$  yang diasumsikan *time invariant*.
4.  $z_{ik}$  : sebuah parameter yang digunakan untuk menampung data observasi *mobile robot* terhadap *landmark* ke- $i$  pada waktu  $k$ .

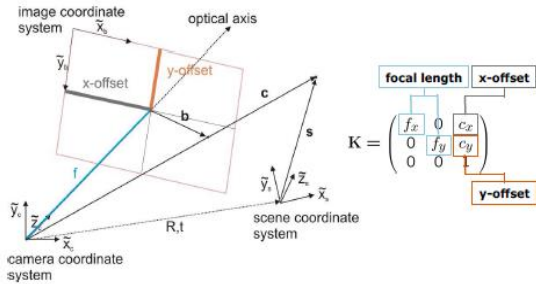
Pada langkah selanjutnya didefinisikan *data set* yang merepresentasikan *history* dari parameter-parameter yang sudah dijabarkan tersebut, antara lain:

1.  $X_{0:k} = \{x_0, x_1, \dots, x_k\} = \{X_{0:k-1}, x_k\}$  : yaitu *history* dari lokasi *mobile robot*.
2.  $U_{0:k} = \{u_0, u_1, \dots, u_k\} = \{U_{0:k-1}, u_k\}$  : yaitu *history* dari *control input*.
3.  $m = \{m_1, m_2, \dots, m_n\}$  : yaitu data set dari semua *landmark* yang terekam.
4.  $Z_{0:k} = \{z_1, z_2, \dots, z_k\} = \{Z_{0:k-1}, z_k\}$  : data set dari parameter *landmark observation*.

## 2.3. Kamera dan Pemodelannya

Pemodelan kamera memuat 2 jenis parameter utama yaitu parameter internal dan eksternal. Parameter internal merupakan parameter yang berhubungan langsung dengan karakter diri kamera, sedangkan parameter eksternal merupakan deskripsi hubungan antara kamera terhadap dunia luar atau *world reference frame*. Parameter internal kamera terdiri dari 2 buah komponen utama, yaitu *focal length* dan *pixel offset* yang masing-masing terdiri dari sumbu  $x$  dan  $y$ , seperti ditunjukkan pada Gambar 3. Parameter eksternal tersebut merepresentasikan kamera

terhadap dunia luar atau *world reference frame*. Komponen utamanya terdiri dari matriks rotasi dan translasi, yang dapat didefinisikan sebagai matriks transformasi [12].



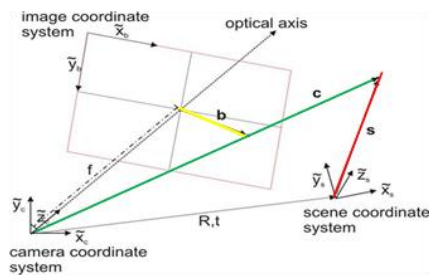
Gambar 3. Interpretasi Parameter Internal Kamera

Secara lengkap parameter internal pada kamera dapat didefinisikan dalam 4 buah komponen yaitu,

1. *Principal point* atau titik tengah kamera
2. *Focal length* atau panjang titik fokus kamera
3. *Skew Coefficient* atau koefisien kecondongan
4. Parameter distorsi pada kamera

Dengan menggunakan *perspective transformation* berdasarkan model kamera *pinhole*, maka dapat dirumuskan matriks pemodelan kamera yang memuat parameter internal dan eksternal kamera sebagai persamaan matriks pada persamaan (2), sebagaimana dapat divisualisasikan pada Gambar 4.

$$b = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (2)$$



Gambar 4. Pemodelan Kamera *Pinhole* Persamaan (2)

## 2.4. Kalibrasi Kamera Teknik 2-Dimensi

Ide penyelesaian kalibrasi kamera dengan menggunakan media 2 dimensi adalah dengan mengatur titik-titik ujung papan catur sebagai *world coordinate system*. Sehingga pada proses kalibrasi, patokan ukuran jarak dari ujung ke ujung yang lain papan catur sudah diketahui secara pasti. Yang membedakan antara kalibrasi 2-dimensi dengan kalibrasi 3-dimensi adalah terletak pada asumsi bahwa semua ciri titik yang berupa *corner feature* dan tersebar

pada bidang papan catur adalah berada pada sebuah bidang sumbu yang sama, sehingga bisa dianggap sumbu z tidak terpakai. Dengan mengembangkan persamaan (2) dan mengeliminasi z serta kolom ketiga matriks parameter eksternal (persamaan (3)), maka matriks persamaan kamera dapat disederhanakan menjadi seperti pada persamaan (4).

$$b = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} \begin{pmatrix} x \\ y \\ 0 \\ 1 \end{pmatrix} \quad (3)$$

$$b = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (4)$$

Tujuan dari kalibrasi kamera adalah mencari nilai matriks

$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$  dari beberapa citra kalibrator. Dengan

asumsi bahwa ukuran dan posisi pada *world coordinate system* sudah terukur, maka *homography* pada masing-masing citra kalibrator juga akan bernilai sama, sehingga nilai K dapat dihitung dengan persamaan (5) [13].

$$H = (h_1, h_2, h_3) = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{pmatrix} \quad (5)$$

$$(h_1, h_2, h_3) = K(r_1, r_2, t) \\ r_1 = K^{-1}h_1, \quad r_2 = K^{-1}h_2$$

Dengan mempertimbangkan bahwa  $(r_1, r_2, r_3)$  adalah saling orthonormal, maka perkalian vektornya selalu menghasilkan nilai 0. Sehingga dapat diperoleh persamaan,

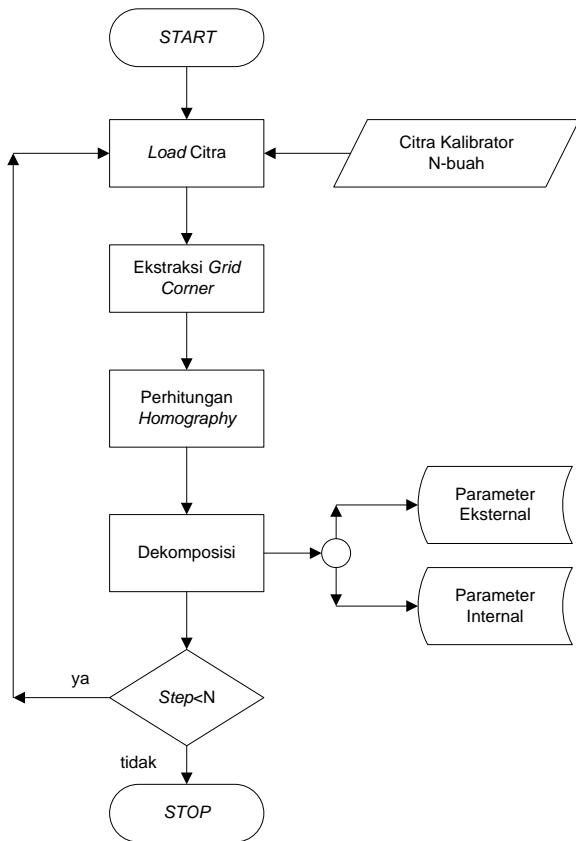
$$H_1^T K^{-T} K^{-1} h_2 = 0 \quad (6)$$

$$H_1^T K^{-T} K^{-1} h_1 - H_2^T K^{-T} K^{-1} h_2 = 0 \quad (7)$$

$$B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} \quad (8)$$

Dapat kita perhatikan bahwa  $K^{-T}K^{-1}$  bersifat simetris dan positif definit. Sehingga dapat kita rumuskan matriks B yang merupakan perkalian dari  $K^{-T}K^{-1}$ , seperti pada persamaan (8). Dari rumusan tersebut, maka nilai K dapat dihitung dari nilai B menggunakan faktorisasi [14].

Pada pengujian, metode kalibrasi kamera yang digunakan adalah teknik 2 dimensi yang dikembangkan oleh Bouquet [14] menggunakan citra kalibrator papan catur berukuran 4x7 dengan masing-masing kotak hitam dan putih memiliki ukuran 30x30 mm. Output kalibrasi adalah parameter internal kamera. Tahapan pada proses kalibrasi kamera adalah kebalikan dari proses simulasi 3 dimensi ke 2 dimensi, karena data-data *sample* kalibrator yang diperoleh berupa *point feature* 2 dimensi pada *camera reference frame* untuk diekstraksi informasinya agar sistem mengetahui nilai matriks *homography*-nya sehingga diperoleh nilai parameter internal. Alur proses kalibrasi kamera ditunjukkan pada Gambar 5.



Gambar 5. Diagram Alir Proses Kalibrasi Kamera

2.5. Teknik Monocular SLAM

Untuk dapat melakukan proses *monocular SLAM*, sistem harus dirumuskan dengan suatu model yang representatif. Secara umum representasi dan pemodelan parameter-parameter yang digunakan pada *monocular SLAM* berdasarkan Davison [15] [16] adalah sebagai berikut,

a. Representasi State Kamera

Informasi yang memuat tentang *state* kamera disandikan dalam vektor  $x_v$  seperti ditunjukkan pada persamaan (9).

$$x_v = \begin{bmatrix} r^{WC} \\ q^{WC} \\ v^W \\ \omega^C \end{bmatrix} \tag{9}$$

Masing-masing komponen parameter terdiri dari,

- $r^{WC} = (x_c \ y_c \ z_c)^T$
- $q^{WC} =$  orientasi arah  $(x,y,z)$
- $v^W = (v_x \ v_y \ v_z)$
- $\omega^C = (\omega_x \ \omega_y \ \omega_z)$

dimana masing-masing superskrip merupakan koordinat kerangka acuan: *WC* sebagai *World-Camera frame*, *W* sebagai *World frame*, dan *C* sebagai *Camera frame*.

b. Representasi Fitur XYZ

Parameterisasi fitur 3 dimensi  $x_i$  dapat dilakukan secara langsung dengan menggunakan notasi,

$$x_i = (x_i \ y_i \ z_i)^T \tag{10}$$

c. Representasi Inverse Depth Feature

*Inverse depth* direpresentasikan sebagai,

$$y_i = (x_{c,i} \ y_{c,i} \ z_{c,i} \ \theta_i \ \phi_i \ \rho_i)^T \tag{11}$$

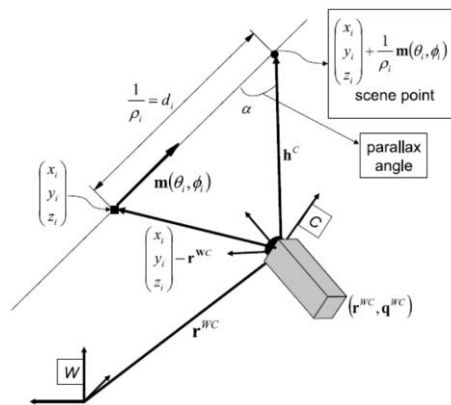
Dimana  $x_{c,i}$ ,  $y_{c,i}$ ,  $z_{c,i}$  merupakan posisi 3 dimensi dari titik yang diamati,  $y_i$  saat inialisasi,  $\theta_i$  sebagai sudut azimuth dan  $\phi_i$  sebagai sudut elevasi, serta  $\rho_i$  sebagai *inverse depth* dari  $y_i$ . Selanjutnya titik yang diamati dapat dimodelkan dengan sebagai titik 3 dimensi dengan referensi pada *world reference frame* sebagai  $x_i$  dengan persamaan 12 yang divisualisasikan pada Gambar 6.

$$x_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} x_{c,i} \\ y_{c,i} \\ z_{c,i} \end{bmatrix} + \frac{1}{\rho_i} m(\theta_i, \phi_i) \tag{12}$$

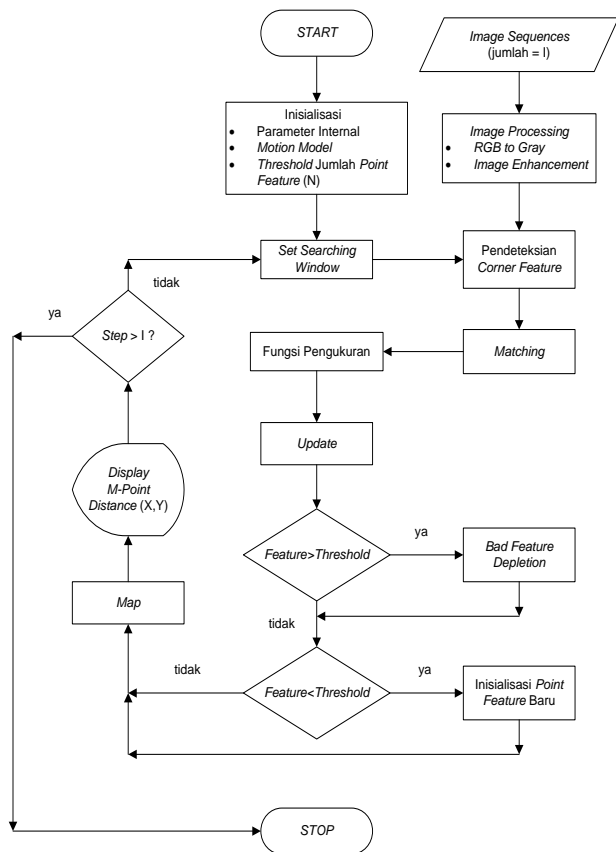
$$m(\theta_i, \phi_i) = (\sin\theta_i \cos\phi_i \ -\sin\phi_i \ \cos\theta_i \ \sin\phi_i)^T \tag{13}$$

Fungsi  $m(\theta_i, \phi_i)$  pada persamaan (13) tersebut merupakan representasi koordinat dari XYZ ke koordinat bola. Dengan menambahkan *depth* ( $1/\rho_i$ ) kepada posisi awal, maka diperoleh titik 3 dimensi. Sehingga dari proses tersebut kemudian dapat disusun *state vector* dari sistem yang mampu menangkap sejumlah *n*-fitur seperti pada persamaan (14) sebagai berikut,

$$x_t = (x_v^T \ f_1^T \ f_2^T \ \dots \ f_n^T)^T \tag{14}$$



Gambar 6. Pemodelan Inverse Depth [16]



Gambar 7. Diagram Alir Algoritma Monocular SLAM

Dengan berpedoman pada rumusan-rumusan tersebut, maka selanjutnya perancangan program *monocular SLAM* mengikuti diagram alir yang ditunjukkan pada Gambar 7.

### 3. Hasil dan Pembahasan

Eksperimen yang telah dilakukan meliputi,

1. Kalibrasi kamera, dengan hasil yang dielaborasi pada subbab 3.1.
2. Estimasi posisi dan orientasi sederhana dengan menggunakan bantuan pola papan catur (*checkerboard*), dengan hasil yang dibahas pada subbab 3.2.
3. Simulasi estimasi pada ruang bebas di dalam ruangan dengan menggunakan citra video *offline* sebagaimana dibahas pada subbab 3.3.

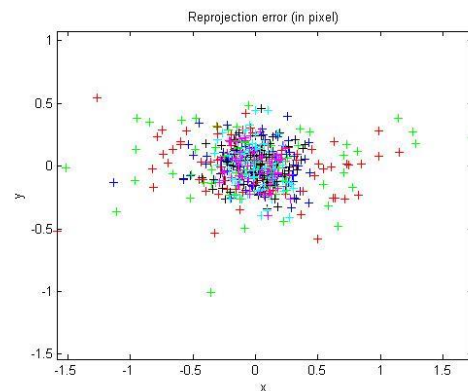
#### 3.1. Hasil Eksperimen Kalibrasi Kamera

Tahapan kalibrasi dilakukan pada 12 citra kalibrator dengan pola papan catur. Dari hasil kalibrasi tersebut diperoleh nilai parameter internal yang ditampilkan pada Tabel 1. Kemudian untuk menguji keberhasilan kalibrasi digunakan grafik *reprojection error* seperti yang ditunjukkan pada Gambar 8., yakni untuk menilai keberhasilan proses kalibrasi. Dari grafik *reprojection*

*error* diperoleh bahwa galat pengukuran kurang dari 2 piksel pada sumbu X maupun sumbu Y sehingga proses kalibrasi dinilai cukup berhasil.

Tabel 1. Nilai Parameter-Parameter Internal dan Eksternal Kamera yang Diperoleh dari Proses Kalibrasi Kamera

No.	Parameter	Nilai (mm)
1	Focal Length X	1,37
2	Focal Length Y	1,37
3	Principal Point X	984,68
4	Principal Point Y	570,27
5	Skew X	0
6	Skew Y	0
7	Distorsi	0
8	Pixel Error X	0,69
9	Pixel Error Y	1,08



Gambar 8. Visualisasi Reprojection Error

Galat kalibrasi merupakan parameter untuk mengukur kesuksesan proses kalibrasi kamera. Jika pada akhir tahap kalibrasi ternyata ditemui nilai galat yang besar (2% dari jumlah piksel citra kalibrator), maka proses kalibrasi harus diulang dengan penyeleksian citra kalibrator yang lebih cermat.

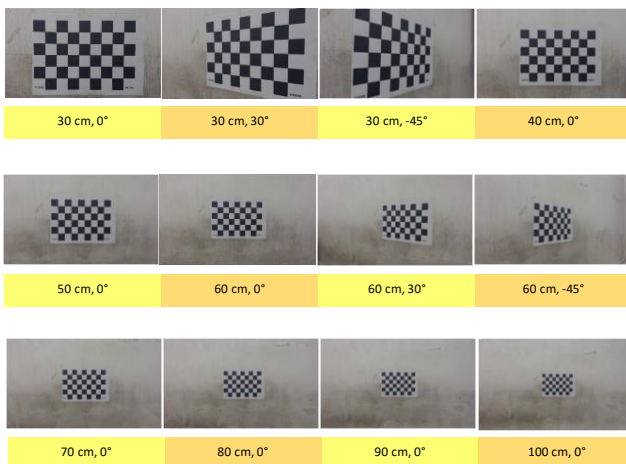


Tabel 2. Data Pengukuran dan Galat Estimasi pada Eksperimen Estimasi Posisi dan Orientasi

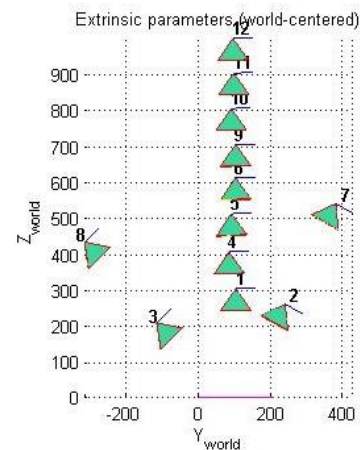
No	Pengukuran			Estimasi			Galat		
	Jarak (mm)	Sudut(°)	pp y (mm)	Jarak (mm)	Sudut(°)	pp y (mm)	Jarak (mm)	Sudut(°)	pp y (mm)
1	300,00	0,0	0,00	303,35	0,0	-5,46	3,35	0,0	5,46
2	300,00	30,0	0,00	304,78	30,5	4,74	4,78	0,5	4,74
3	300,00	-45,0	0,00	302,31	-46,4	7,32	2,31	1,4	7,32
4	400,00	0,0	0,00	403,56	0,0	21,12	3,56	0,0	21,12
5	500,00	0,0	0,00	501,08	0,0	11,13	1,08	0,0	11,13
6	600,00	0,0	0,00	602,10	0,0	8,00	2,1	0,0	8,00
7	600,00	30,0	0,00	602,78	30,6	16,20	2,78	0,6	16,2
8	600,00	-45,0	0,00	601,94	-45,8	13,78	1,94	0,8	13,78
9	700,00	0,0	0,00	700,92	0,0	17,35	0,92	0,0	17,35
10	800,00	0,0	0,00	800,45	-0,1	5,21	0,45	0,1	5,21
11	900,00	0,0	0,00	903,62	0,1	12,50	3,62	0,1	12,5
12	1000,00	0,0	0,00	1007,91	0,2	13,40	7,91	0,2	13,4
<b>Rerata Galat</b>							<b>2,9</b>	<b>0,3</b>	<b>11,35</b>

3.2. Hasil Eksperimen Estimasi Posisi dan Orientasi dengan Bantuan Pola Papan Catur (Checkerboard)

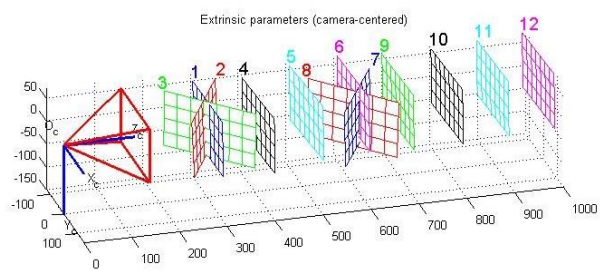
Estimasi posisi dan orientasi dilakukan dengan menggunakan 12 citra papan catur yang telah diatur dengan jarak dan sudut pengambilan yang beragam seperti ditunjukkan pada Gambar 9. Kemudian estimasi posisi dan orientasi dilakukan dengan menampilkan kembali posisi citra kalibrator papan catur dengan *world-centered view* seperti ditunjukkan pada Gambar 10 dan *camera-centered view* pada Gambar 11 dengan nilai jarak ukur, estimasi, dan galat yang ditunjukkan pada Tabel 2.



Gambar 9. Pola Papan Catur dengan Berbagai Pose dan Posisi Pengukuran yang Beragam



Gambar 10. Hasil Estimasi Pose dan Posisi terhadap Pola Papan Catur dengan Tampilan World-Centered View

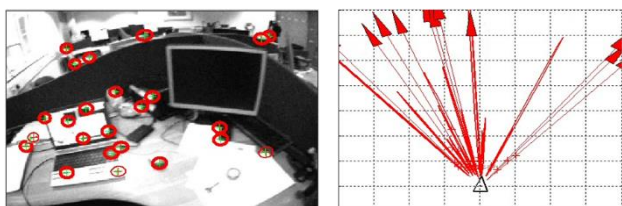


Gambar 11. Hasil Estimasi Pose dan Posisi terhadap Pola Papan Catur dengan Tampilan Camera-Centered View

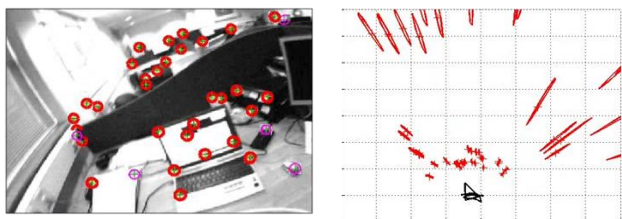
Hasil eksperimen yang dilakukan dengan software Matlab menunjukkan bahwa pengujian estimasi posisi dan orientasi terhadap citra kalibrator dengan ciri titik yang tersusun rapi seperti pada citra papan catur telah berhasil dilakukan dengan baik, yakni menghasilkan nilai galat yang kecil, yakni rata-rata hanya 0,29% dari rentang pengukuran, serta 0,67% dari rentang pengukuran sudut. Dengan persentase galat yang cukup kecil (kurang dari 2%) menunjukkan bahwa algoritma estimasi posisi dan orientasi berbasis monokular telah berhasil dilakukan dengan cukup akurat.

### 3.3. Hasil Simulasi Proses Monocular SLAM Sederhana dengan Estimasi Jarak terhadap Fitur Titik

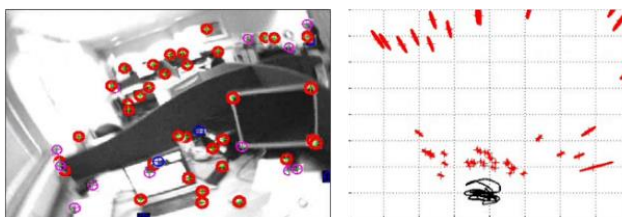
Untuk mendemonstrasikan algoritma monokular SLAM sederhana, eksperimen ini menggunakan citra video *offline* yang diambil dari dataset. Citra video direkam di dalam ruangan kantor menggunakan webcam dengan resolusi 640x480 yang telah diturunkan kualitasnya menjadi setengahnya saja guna mempercepat proses komputasi. Berbeda dengan citra papan catur yang memiliki ciri titik yang tersebar secara rapi dalam satu bidang yang sama, pada kondisi di dalam ruang kantor ciri titik tersebar secara tidak teratur dan dalam bidang 3 dimensi.



Gambar 12. Hasil Estimasi Frame ke-18 pada Image Sequence



Gambar 13. Hasil Estimasi Frame ke-302 pada Image Sequence



Gambar 14. Hasil Estimasi Frame ke-1004 pada Image Sequence

Pada simulasi proses SLAM monokular, sejumlah 1200 buah *image sequences* diekstrak dari video. Pada awal proses monocular SLAM, semua *point feature* diasumsikan sebagai *point at infinity* (berada di titik jarak tak terbatas), seiring dengan pergerakan kamera, *point feature* teramati dari posisi yang berbeda, sehingga kemudian dengan menggunakan *point feature* tersebut dapat dilakukan estimasi jarak dan posisinya relative terhadap kamera, nilai estimasi *depth* diperoleh secara terus menerus dan dilakukan *update confidence value* yang lebih baik, dengan metode yang dijelaskan pada Bab 2 subbab 2.5.

Seperti ditunjukkan pada Gambar 12 sampai Gambar 14, Pada *frame-frame* awal, sistem SLAM monokular belum dapat memetakan jarak dan orientasi tiap ciri titik, namun pada *frame* ke-302, ciri-ciri titik dengan posisi dekat sudah dapat dipetakan, dan pada *frame* ke-1004 sistem sudah dapat mengestimasi jarak sejumlah ciri titik dengan lebih baik.

## 4. Kesimpulan

Berdasarkan eksperimen yang telah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut:

1. Proses kalibrasi kamera sudah dilakukan dengan cukup baik dengan *reprojection error* yang relatif kecil (kurang dari 2%) serta dapat digunakan untuk menentukan nilai parameter internal sebuah kamera.
2. Estimasi posisi citra papan catur dapat dilakukan dengan benar serta dapat divisualisasikan kembali dalam bentuk *world-centered view* dan *camera-centered view* dengan galat pengukuran-estimasi yang relatif kecil.
3. Simulasi SLAM monokular sederhana secara *offline* telah berhasil dilakukan dan mampu mengestimasi jarak beberapa ciri titik dengan cukup baik pada lingkungan yang statis (tetap, tidak ada objek yang bergerak selain kamera sebagai sensor).

Estimasi posisi dan orientasi terhadap beberapa citra papan catur berhasil memetakan jarak dan sudut dengan cukup akurat, dengan rerata galat estimasi yang kecil (2,9 mm untuk estimasi jarak dan 0,3° untuk estimasi sudut). Kemudian pada SLAM monokular sederhana, sistem berhasil memetakan posisi beberapa ciri titik dengan baik dan konsisten dengan menggunakan 1200 buah *image sequences*. Penggunaan MATLAB untuk analisis dan simulasi sudah mencukupi untuk rancangan algoritma SLAM monokular. Namun ke depan pengembangan algoritma estimasi posisi dapat dilakukan lebih jauh untuk aplikasi *real-time*, sehingga dapat diaplikasikan untuk SLAM pada *mobile robot*, yakni dengan menggunakan bahasa C atau python untuk proses perancangan algoritma dengan input dari *live video sequences* dari kamera secara *real-time*.



## Referensi

- [1]. M. O. A. Aqel, M. H. Marhaban, M. I. Saripan and N. B. Ismail, "Review of visual odometry: types, approaches, challenges, and applications," SpringerPlus, vol. 5, p. 1897, 2016.
- [2]. A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox and N. Roy, "Visual odometry and mapping for autonomous flight using an RGB-D camera," in *Robotics Research*, Springer, 2017, pp. 235-252.
- [3]. L. Pérez, Í. Rodríguez, N. Rodríguez, R. Usamentiaga and D. García, "Robot guidance using machine vision techniques in industrial environments: A comparative review," *Sensors*, vol. 16, p. 335, 2016.
- [4]. X. Li, W. Chen, C. Chan, B. Li and X. Song, "Multi-sensor fusion methodology for enhanced land vehicle positioning," *Information Fusion*, vol. 46, pp. 51-62, 2019.
- [5]. H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE robotics & automation magazine*, vol. 13, pp. 99-110, 2006.
- [6]. C. C. Hau, *Handbook of pattern recognition and computer vision*, World Scientific, 2015.
- [7]. C. Harris and M. Stephens, "A combined corner and edge detector.," in *Alvey vision conference*, 1988.
- [8]. S. M. Smith and J. M. Brady, "SUSAN\_a new approach to low level image processing," *International journal of computer vision*, vol. 23, pp. 45-78, 1997.
- [9]. E. Rosten and T. Drummond, "Machine Learning for High Speed Corner Detector," *ninth conference of European Conference on Computer Vision*, pp. 430 - 443, 2006.
- [10]. T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robotics & Automation Magazine*, vol. 13, pp. 108-117, 2006.
- [11]. S. Thrun, W. Burgard and D. Fox, *Probabilistic robotics*, MIT press, 2005.
- [12]. S. Albrecht, "An analysis of visual mono-slam," *Diss. Master's Thesis. Universität Osnabrück*, vol. 2009, 2009.
- [13]. R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, Cambridge university press, 2003.
- [14]. J.-Y. Bouguet, "Camera calibration toolbox for matlab," <http://www.vision.caltech.edu/bouguetj/calibdoc/index.html>, 2004.
- [15]. A. J. Davison, I. D. Reid, N. D. Molton and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, pp. 1052-1067, 2007.
- [16]. J. M. M. Montiel, J. Civera and A. J. Davison, "Unified inverse depth parametrization for monocular SLAM," 2006.