



TESIS - TE185401

***MARKERLESS MOTION CAPTURE* BERBASIS
OPENPOSE MODEL MENGGUNAKAN METODE
TRIANGULASI**

UTI SOLICHAH
07111750052002

DOSEN PEMBIMBING
Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng
Dr. Eko Mulyanto Yuniarno, ST., MT

PROGRAM MAGISTER
BIDANG KEAHLIAN JARINGAN CERDAS MULTIMEDIA
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2020



TESIS - TE185401

***MARKERLESS MOTION CAPTURE* BERBASIS
OPENPOSE MODEL MENGGUNAKAN METODE
TRIANGULASI**

UTI SOLICHAH
07111750052002

DOSEN PEMBIMBING
Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng.
Dr. Eko Mulyanto Yuniarno, ST., MT

PROGRAM MAGISTER
BIDANG KEAHLIAN JARINGAN CERDAS MULTIMEDIA
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2020

LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar

Magister Teknik (MT)

di

Institut Teknologi Sepuluh Nopember

Oleh:

UTI SOLICHAH

NRP: 07111750052002

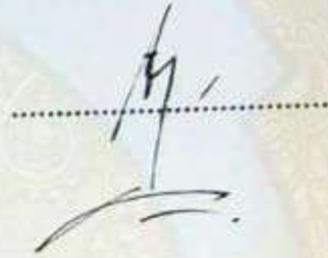
Tanggal Ujian: 14 Januari 2020

Periode Wisuda: Maret 2020

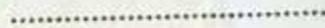
Disetujui oleh:

Pembimbing:

1. Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng.
NIP: 195809161986011001

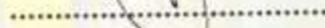


2. Dr. Eko Mulyanto Yuniarno, ST., MT.
NIP: 196806011995121009

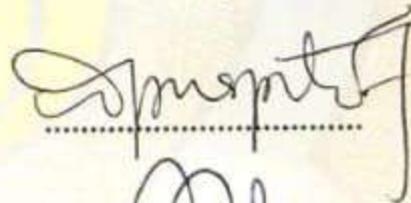


Penguji:

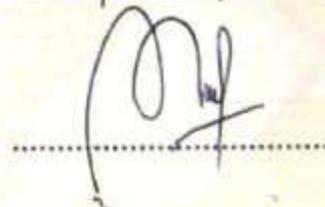
1. Dr. Supeno Mardi Susiki Nugroho, ST., MT
NIP: 197003131995121001



2. Dr. Diah Puspito Wulandari, ST., M.Sc
NIP: 198012192005012001



3. Reza Fuad Rachmadi, ST., MT., Ph.D
NIP: 198504032012121001



Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN TESIS

Dengan ini saya menyatakan bahwa isi keseluruhan Tesis saya dengan judul “*MARKERLESS MOTION CAPTURE* BERBASIS *OPENPOSE MODEL* MENGGUNAKAN METODE *TRIANGULASI*” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Maret 2020

Uti Solichah
07111750052002

Halaman ini sengaja dikosongkan

MARKERLESS MOTION CAPTURE ***BERBASIS OPENPOSE MODEL*** **MENGGUNAKAN METODE** **TRIANGULASI**

Nama Mahasiswa : Uti Solichah
NRP : 07111750052002
Pembimbing : 1. Prof. Dr. Ir. Mauridhi Hery Purnomo,
M.Eng.
2. Dr. Eko Mulyanto Yuniarno, ST., MT.

ABSTRAK

Human Pose Estimation merupakan salah satu topik menarik pada visi komputer. Dalam penelitian sebelumnya, *Openpose* telah berhasil menerapkan *human pose estimation* dalam bentuk 2D. 3D pose estimation didapatkan dari dua macam masukan, yaitu *single view* dan *multiview*, 3D *pose estimation* menggunakan *multiview* dapat mengestimasi lokasi kedalaman lebih *robust* dibandingkan dengan menggunakan masukan *single view*. 3D *human pose estimation* dapat dihasilkan dari dataset pose 3D atau dari lokasi *joint* 2D. Penggunaan dataset pose 3D memiliki keterbatasan dalam sumber maupun kegunaannya. Dari beberapa kasus di atas, Kami memilih untuk mengembangkan pendekatan *multiview* dan lokasi *joint* 2D sebagai masukan untuk mendapatkan 3D *human pose estimation*.

Input dari penelitian berupa dua *view image* yang berbeda, dimana masing-masing input akan diproses menggunakan *inference openpose model* untuk mendapatkan lokasi *joint* 2D. Proses kalibrasi kamera dibutuhkan untuk mendapatkan fitur instrinsik dan ekstrinsik kamera. Dengan fitur tersebut dan hasil dari 2D *pose estimation*, kita akan mendapatkan 3D *pose estimation* dengan pendekatan metode triangulasi. Sistem ini dapat digunakan dalam gender, baju dan pose yang berbeda. Desain jarak antar kamera yang terbaik pada jarak 66cm. Rentang error jarak dari kamera ke objek berkisar pada 13-21 cm. Error kedalaman objek ke kamera dapat diminimalkan dengan perbaikan hasil lokasi 2D *joint* dari *openpose*.

Kata Kunci : 2D *human pose estimation*, 3D *human pose estimation*, kalibrasi kamera, *multiview*, *openpose*

Halaman ini sengaja dikosongkan

MARKERLESS MOTION CAPTURE BASED ON OPENPOSE MODEL USING TRIANGULATION

By : Uti Solichah
Student Identity Number : 07111750052002
Supervisor(s) : 1. Prof. Dr. Ir. Mauridhi Hery Purnomo,
M.Eng.
2. Dr. Eko Mulyanto Yuniarno, ST., MT.

ABSTRACT

Human pose estimation is one of many topics in computer vision. Previous works in openpose are only applicable in 2D human pose estimation. 3D human pose estimation can be done using two types of input, single and multiview. 3D pose estimation using multiview are inherently more robust than single view, due to multiview allowing better depth estimation. 3D human pose estimation are obtained from 3D data set poses or 2D joint location poses. However, with the limitations inherent in the 3D data set, such as lack of sufficient data and usage difficulties, this research applies 2D joint location. From the cases outlined before, we choose to develop multiview camera and 2D joint location as input to obtain 3D motion capture.

The inputs are two images with different views, where each image is processed using an inference openpose model to get 2D joint location. Camera calibration is needed to precisely obtain the intrinsic and extrinsic features of the camera. With these features and the 2D joint location results, we obtained the 3D human pose estimation using triangulation method. This system can be carried out on any combination of genders, apparels and poses. The best distance between cameras is 66 cm. The error distance between the camera and object range is 13-21 cm. Error in object distance to the camera can be minimalized by fixing 2D joint location from openpose.

Key words : 2D human pose estimation, 3D human pose pstimation, camera calibration, multview camera, openpose

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji dan syukur kehadiran Allah Subhanahu wa ta'ala atas segala limpahan berkah, rahmat, serta hidayah-Nya, dan serta barokah dari Rasulullah SAW penulis dapat menyelesaikan penelitian dengan judul **MARKERLESS MOTION CAPTURE BERBASIS OPENPOSE MODEL MENGGUNAKAN METODE TRIANGULASI**. Tanpa kehendak-Nya tidak mungkin penulis mampu mewujudkan penelitian ini.

Penelitian ini disusun dalam rangka sebagai persyaratan menyelesaikan Pendidikan Magister Teknik Elektro, Bidang Keahlian Jaringan Cerdas Multimedia, Institut Teknologi Sepuluh Nopember. Selesaiannya penelitian ini tidak lepas dari bantuan, dukungan, dan doa dari beberapa pihak. Sehingga, pada kesempatan ini penulis mengucapkan syukur alhamdulillah dan terimakasih kepada:

1. Keluarga Penulis, Ibu, Mama, Tete, Aang, Ponakan tercinta penulis yang selalu memberikan dukungan dan mendoakan penulis untuk menyelesaikan penelitian ini
2. Partner Penulis dalam segala hal, terimakasih untuk selalu ada dan menemani penulis hingga saat ini.
3. Bapak Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng selaku pembimbing pertama dan Bapak Dr. Eko Mulyanto Yuniarno, ST., MT selaku pembimbing kedua atas saran, bimbingan, dan dukungan selama mengerjakan penelitian hingga terselesaikan penelitian ini.
4. Bapak Dr. Tri Arief Sardjono, ST., MT. selaku Dekan Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember.
5. Bapak Dr. Ardyono Priyadi, ST., M.Eng. selaku Ketua Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember.
6. Bapak-Ibu dosen pengajar Teknik Elektro khususnya Bidang Keahlian Jaringan Cerdas Multimedia atas ilmu dan bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
7. Segenap civitas akademik Departemen Teknik Elektro untuk berbagai dukungannya secara langsung maupun tidak langsung.
8. Seluruh teman-teman di Laboratorium *B201* dan juga teman-teman JCM atas bantuan dalam pengambilan data, beserta saran dan dukungan motivasi kepada penulis untuk menyelesaikan penelitian ini.
9. Teman-teman penulis yang tidak bisa penulis sebutkan satu persatu, terimakasih telah memberikan dukungan dan menjadi teman yang baik hingga saat ini.

Surabaya, Maret 2020

Uti Solichah

Halaman ini sengaja dikosongkan

DAFTAR ISI

LEMBAR PENGESAHAN	v
PERNYATAAN KEASLIAN	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxiii
NOMENKLATUR	xxv
1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Batasan masalah	4
1.5 Kontribusi	4
2 KAJIAN PUSTAKA <i>MARKERLESS MOTION CAPTURE</i>	7
2.1 Penelitian Terkait <i>Human Pose Estimation</i>	7
2.1.1 <i>Pose Estimation</i> berbasis <i>marker</i> dan <i>markerless</i>	7
2.1.2 <i>Pose Estimation multiview</i> dan <i>single view</i>	10
2.2 <i>Stereo Vision</i> dan <i>Depth Perception</i>	10
2.3 Rekonstruksi 3D dan Konsep Triangulasi	11
2.4 Proyeksi Geometri dan Kamera Model	14
2.5 Kalibrasi Kamera	16
2.6 Estimasi 2D <i>Human Pose</i> Menggunakan <i>Openpose</i>	17
2.6.1 Arsitektur <i>Openpose</i>	19
2.6.2 <i>Part Detection</i>	21
2.6.3 <i>Part Association</i>	22
2.7 <i>Confidence Map</i>	23
2.8 Arsitektur <i>VGG-19</i>	24
2.9 <i>Convolutional Neural Network</i>	24
2.9.1 Convolutional Layer	26
2.9.2 <i>Pooling Layer</i>	26

3	METODOLOGI <i>MARKERLESS MOTION CAPTURE</i>	27
3.1	Desain Sistem	27
3.2	Data Input	27
3.3	Estimasi 2D <i>Human Pose</i> Menggunakan <i>Openpose</i>	28
3.3.1	<i>Part Detection</i>	30
3.3.2	<i>Part Association</i>	30
3.4	Kalibrasi Stereo Kamera	32
3.5	Rekonstruksi 3D <i>Human Pose</i>	32
3.5.1	Proyeksi 3D	33
3.5.2	Triangulasi	34
3.5.3	Menghitung Sumbu Koordinat <i>X</i> dan <i>Y</i>	37
3.5.4	<i>Plot 3D Human Pose</i>	38
4	HASIL DAN PEMBAHASAN	41
4.1	Data Input	41
4.2	Hasil Estimasi 2D <i>Human Pose</i> Menggunakan <i>Openpose</i>	42
4.3	Hasil Kalibrasi Stereo Kamera	42
4.4	Hasil Rekonstruksi 3D <i>Human Pose</i>	43
4.4.1	Hasil Proyeksi 3D	44
4.4.2	Hasil Perhitungan Triangulasi	44
4.4.3	Hasil Sumbu Koordinat <i>X</i> dan <i>Y</i>	45
4.4.4	Hasil Plot 3D <i>Human Pose</i>	46
4.5	Percobaan Skenario Jarak Objek ke Kamera	46
4.5.1	Jarak 240 cm dari Objek ke Kamera	47
4.5.2	Jarak 275 cm dari Objek ke Kamera	48
4.5.3	Jarak 310 cm dari Objek ke Kamera	49
4.5.4	Jarak 340 cm dari Objek ke Kamera	52
4.6	Percobaan Skenario Jarak antar Kedua Kamera	53
4.6.1	Jarak antar Kamera 20 cm	53
4.6.2	Jarak antar kamera 43 cm	54
4.6.3	Jarak antar Kamera 66 cm	56
4.6.4	Jarak antar Kamera 98 cm	57
4.7	Percobaan Skenario Gender dan Baju	60
4.8	Percobaan Skenario <i>Pose</i>	62
4.8.1	<i>Pose</i> Sedang Menunjukkan Arah	62
4.8.2	<i>Pose</i> Sedang Diskusi	63
4.8.3	<i>Pose</i> Sedang Makan	64
4.8.4	<i>Pose</i> Sedang Membayar	65
4.8.5	<i>Pose</i> Sedang Berjalan	66
4.8.6	<i>Pose</i> Sedang Menelpon	67
4.9	Percobaan Skenario <i>Motion</i>	68
4.9.1	<i>Motion</i> Random Model A	68
4.9.2	<i>Motion</i> Random Model B	73
4.9.3	<i>Motion</i> Random Model C	79
4.9.4	<i>Motion</i> Random Model D	85

5 PENUTUP	93
5.1 Kesimpulan	93
5.2 Saran	94
DAFTAR PUSTAKA	95
Biografi Penulis	97

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

2.1	Kendala <i>Kinect</i> pada Deteksi <i>joint</i> (Liu et al., 2016)	8
2.2	Hasil estimasi pose penelitian berbasis markerless (Cao et al., 2017)	9
2.3	Triangulasi dari dua <i>view</i> yang berbeda.	12
2.4	Ilustrasi Proyeksi Kamera Model Pinhole (Hartley, 2004)	14
2.5	Contoh Checkerboard	16
2.6	Hasil Estimasi Pose 2D oleh <i>Openpose</i> (Cao et al., 2017)	18
2.7	Beberapa Kendala Estimasi Pose 2D oleh <i>Openpose</i> (Cao et al., 2017)	18
2.8	Ilustrasi Keseluruhan Tahapan <i>Openpose</i> (Cao et al., 2017) . . .	19
2.9	Arsitektur <i>Openpose</i> (Cao et al., 2017)	20
2.10	Ilustrasi Hasil <i>Part Detection</i> (Cao et al., 2017)	22
2.11	Ilustrasi Proses <i>Part Association</i> (Cao et al., 2017)	23
2.12	Contoh Ilustrasi Hasil <i>Confidence Maps</i> (Wei et al., 2016)	24
2.13	Konfigurasi Arsitektur <i>VGG-19</i> (Simonyan and Zisserman, 2014) 25	
3.1	Desain Sistem Penelitian.	28
3.2	Ilustrasi cara pengambilan gambar objek dengan kedua kamera. 28	
3.3	Proses <i>Inference model 2D human pose</i>	29
3.4	Proses <i>Openpose</i> dalam Mendapatkan 2D <i>Human Pose</i>	29
3.5	Tahapan proses Rekonstruksi 3D <i>Human Pose</i>	32
3.6	Triangulasi dari dua <i>view</i> yang berbeda.	35
3.7	Ilustrasi Perbandingan Koordinat X	37
3.8	Ilustrasi Perbandingan Koordinat Y	38
3.9	Visualisasi Plot 3D <i>human pose</i>	39
4.1	Contoh Citra Input.	41
4.2	Hasil Estimasi 2D <i>Human Pose</i> Menggunakan <i>Openpose</i>	42
4.3	Plot Hasil Rekonstruksi 3D human pose	46
4.4	Input dan Hasil <i>Openpose</i> Jarak Objek ke Kamera 240 cm . . .	47
4.5	Hasil 3D Rekonstruksi Jarak Objek ke Kamera 240 cm	47
4.6	Input dan Hasil <i>Openpose</i> Jarak Objek ke Kamera 275 cm . . .	49
4.7	Hasil 3D Rekonstruksi Jarak Objek ke Kamera 275 cm.	49
4.8	Input dan Hasil <i>Openpose</i> Jarak Objek ke Kamera 310 cm . . .	49
4.9	Hasil 3D Rekonstruksi Jarak Objek ke Kamera 310 cm	50
4.10	Input dan Hasil <i>Openpose</i> Jarak Objek ke Kamera 340 cm. . . .	51
4.11	Hasil 3D Rekonstruksi Jarak Objek ke Kamera 340 cm.	52
4.12	Input dan Hasil <i>Openpose</i> Jarak antar Kamera 20 cm.	54
4.13	Hasil 3D Rekonstruksi Jarak antar Kamera 20 cm.	54
4.14	Input dan Hasil <i>Openpose</i> Jarak antar Kamera 43 cm.	55
4.15	Hasil 3D Rekonstruksi Jarak Kamera 43 cm.	56
4.16	Input dan Hasil <i>Openpose</i> Jarak antar Kamera 66 cm.	57
4.17	Hasil 3D Rekonstruksi Jarak antar Kamera 66 cm.	57

4.18	Input dan Hasil Openpose Jarak antar Kamera 98 cm.	58
4.19	Hasil 3D Rekonstruksi Jarak antar Kamera 98 cm.	59
4.20	Citra Input Berbeda Gender dan Baju	60
4.21	Hasil Openpose Berbeda Gender dan Baju pada kamera pertama	61
4.22	Hasil 3D Rekonstruksi Berbeda Gender dan Baju.	61
4.23	Citra Input dan Hasil Openpose Pose Sedang Menunjukkan Arah.	62
4.24	Hasil 3D Rekonstruksi Pose Menunjuk Arah.	63
4.25	Citra Input dan Hasil Openpose Pose Sedang Diskusi.	63
4.26	Hasil 3D Rekonstruksi Pose Sedang Diskusi.	64
4.27	Citra Input dan Hasil Openpose Pose Sedang Makan.	64
4.28	Hasil 3D Rekonstruksi Pose Sedang Makan.	65
4.29	Citra Input dan Hasil Openpose Pose Sedang Bayar.	65
4.30	Hasil 3D Rekonstruksi Pose Sedang Membayar.	66
4.31	Citra Input dan Hasil Openpose Pose Sedang Berjalan.	66
4.32	Hasil 3D Rekonstruksi Pose Berjalan.	67
4.33	Citra Input dan Hasil Openpose Pose Sedang Menelpon.	67
4.34	Hasil 3D Rekonstruksi Pose Sedang Menelpon.	68
4.35	Rangkaian Citra Input Motion Random Model B.	68
4.36	Hasil Openpose dari Gerakan Model A.1.	69
4.37	Hasil 3D Rekonstruksi dari Gerakan Model A.1.	70
4.38	Hasil Openpose dari Gerakan Model A.2.	71
4.39	Hasil 3D Rekonstruksi dari Gerakan Model A.2.	72
4.40	Hasil Openpose dari Gerakan Model A.3.	72
4.41	Hasil 3D Rekonstruksi dari Gerakan Model A.3.	73
4.42	Rangkaian Hasil 3D Rekonstruksi Motion Random Model A . .	74
4.43	Rangkaian Citra Input Motion Random Model B.	74
4.44	Hasil Openpose dari Gerakan Model B.1.	75
4.45	Hasil 3D Rekonstruksi dari Gerakan Model B.1.	76
4.46	Hasil Openpose dari Gerakan Model B.2.	76
4.47	Hasil 3D Rekonstruksi dari Gerakan Model B.2.	77
4.48	Hasil Openpose dari Gerakan Model B.3.	78
4.49	Hasil 3D Rekonstruksi dari Gerakan Model B.3.	78
4.50	Rangkaian Hasil 3D Rekonstruksi Motion Random Model B . .	79
4.51	Rangkaian Citra Input Motion Random Model C.	80
4.52	Hasil Openpose dari Gerakan Model C.1.	81
4.53	Hasil 3D Rekonstruksi dari Gerakan Model C.1.	82
4.54	Hasil Openpose dari Gerakan Model C.2.	82
4.55	Hasil 3D Rekonstruksi dari Gerakan Model C.2.	82
4.56	Hasil Openpose dari Gerakan Model C.3.	84
4.57	Hasil 3D Rekonstruksi dari Gerakan Model C.3.	85
4.58	Rangkaian Hasil 3D Rekonstruksi Motion Random Model C . .	85
4.59	Rangkaian Citra Input Motion Random Model D.	86
4.60	Hasil Openpose dari Gerakan Model D.1.	86
4.61	Hasil 3D Rekonstruksi dari Gerakan Model D.1.	87
4.62	Hasil Openpose dari Gerakan Model D.2.	88

4.63 Hasil 3D Rekonstruksi dari Gerakan Model D.2.	88
4.64 Hasil Openpose dari Gerakan Model D.3.	89
4.65 Hasil 3D Rekonstruksi dari Gerakan Model D.3.	90
4.66 Rangkaian Hasil 3D Rekonstruksi Motion Random Model D . .	91

Halaman ini sengaja dikosongkan

DAFTAR TABEL

3.1	Spesifikasi Kamera Logitech C920	28
3.2	Hasil Estimasi <i>Joint 2D Human Pose</i>	31
3.3	Daftar Plot Joint pada Sistem	39
4.1	Hasil Proyeksi 3D kamera pertama dan kedua	44
4.2	Sumbu Koordinat Dunia Hasil Rekonstruksi 3D	45
4.3	Hasil Triangulasi Koordinat Z dengan Jarak Objek ke Kamera 240 cm	48
4.4	Hasil Triangulasi Koordinat Z dengan Jarak Objek ke Kamera 275 cm	50
4.5	Hasil Triangulasi Koordinat Z dengan Jarak Objek ke Kamera 310 cm	51
4.6	Perbandingan Rata-Rata Error Jarak Objek ke Kamera	52
4.7	Hasil Triangulasi Koordinat Z dengan Jarak Objek ke Kamera 340 cm	53
4.8	Hasil Triangulasi Koordinat Z Desain Jarak 20 cm	55
4.9	Hasil Triangulasi Koordinat Z Desain Jarak 43 cm	56
4.10	Hasil Triangulasi Koordinat Z Desain Jarak 66 cm	58
4.11	Hasil Triangulasi Koordinat Z Desain Jarak 98 cm	59
4.12	Perbandingan Rata-Rata Error Keempat Jarak antar Kamera	60
4.13	Hasil Triangulasi Koordinat Z Model A.1	70
4.14	Hasil Triangulasi Koordinat Z Model A.2	71
4.15	Hasil Triangulasi Koordinat Z Model A.3	73
4.16	Perbandingan Rata-Rata Error Motion Random Model A	73
4.17	Hasil Triangulasi Koordinat Z Model B.1	75
4.18	Hasil Triangulasi Koordinat Z Model B.2	77
4.19	Hasil Triangulasi Koordinat Z Model B.3	79
4.20	Perbandingan Rata-Rata Error Motion Random Model B	79
4.21	Hasil Triangulasi Koordinat Z Model C.1	81
4.22	Hasil Triangulasi Koordinat Z Model C.2	83
4.23	Hasil Triangulasi Koordinat Z Model C.3	84
4.24	Perbandingan Rata-Rata Error Motion Random Model C	85
4.25	Hasil Triangulasi Koordinat Z Model D.1	87
4.26	Hasil Triangulasi Koordinat Z Model D.2	89
4.27	Hasil Triangulasi Koordinat Z Model D.3	90
4.28	Perbandingan Rata-Rata Error Motion Random Model D	91

Halaman ini sengaja dikosongkan

NOMENKLATUR

$2D$	= Bentuk 2 Dimensi.
$3D$	= Bentuk 3 Dimensi.
CNN	= Convolutional Neural Network.
xc_1	= Posisi koordinat x pada kamera 1.
zc_1	= Posisi koordinat y pada kamera 1.
xc_2	= Posisi koordinat x pada kamera 2.
zc_2	= Posisi koordinat y pada kamera 2.
x_1	= Posisi koordinat titik objek x pada kamera 1.
z_1	= Posisi koordinat titik objek y pada kamera 1.
x_2	= Posisi koordinat titik objek x pada kamera 2.
z_2	= Posisi koordinat titik objek y pada kamera 2.
f	= <i>Focal length</i> .
P	= <i>Image plane</i> .
u	= Posisi koordinat titik x pada <i>image plane</i> .
v	= Posisi koordinat titik y pada <i>image plane</i> .
Pc	= Koordinat dunia
X	= Sumbu koordinat dunia X
Y	= Sumbu koordinat dunia Y
Z	= Sumbu koordinat dunia Z
K	= Matriks Intrinsik Kamera
cx	= <i>principal point</i> x pada image
cy	= <i>principal point</i> y pada image
VGG	= Visual Geometry Group.
S	= Himpunan 2D <i>confidence maps</i> lokasi <i>body part</i> .
J	= Jumlah banyaknya <i>confidence maps</i> .
L	= Himpunan 2D <i>vector body part</i> .
C	= Jumlah banyaknya <i>set vector</i> 2D.
F	= <i>Feature maps</i> .
S^1	= <i>Stage</i> pertama pada himpunan 2D <i>confidence maps</i> .
S^t	= <i>Stage</i> ke- t pada himpunan 2D <i>confidence maps</i> .
ρ^t	= Fungsi F pada 2D <i>confidence maps</i> .
L^1	= <i>Stage</i> pertama pada himpunan 2D <i>vector</i> .
L^t	= <i>Stage</i> ke- t pada himpunan 2D <i>vector</i> .
σ^t	= Fungsi F pada 2D <i>vector</i> .
NP	= <i>Nondeterministic polynomial time</i> .
W	= Panjang atau tinggi <i>output</i> .
N	= Panjang atau tinggi <i>filter</i> .
Pd	= <i>Zero padding</i> .
St	= <i>Stride</i> .

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

1.1 Latar belakang

Motion Capture menjadi salah satu topik yang populer di beberapa penelitian. Salah satu pengaplikasian *motion capture* adalah pada *human pose estimation*. Sebelum banyak penelitian yang menerapkan *human pose estimation* dengan berbasis *markerless*, beberapa penelitian masih menggunakan *marker* atau penanda untuk mendapatkan skeleton manusia. *Optical motion* menjadi salah satu alat untuk mendapatkan skeleton manusia, hasil *motion capture* yang dihasilkan memang lebih akurat, namun *optical motion* ini tidak praktis, dimana pengguna yang ditargetkan harus menggunakan baju penanda, selain itu alat ini terbilang mahal dan tidak dapat dijangkau oleh semua pengguna (Liu et al., 2016).

Selain *optical motion* ada beberapa penelitian yang menggunakan *device kinect*. Terkadang *device kinect* dalam pengaplikasiannya mengalami kendala dalam mendapatkan skeleton manusia (Liu et al., 2016). Hal ini terjadi akibat dari *self occlusions* dan juga kesalahan yang berasal dari sensor *kinect* itu sendiri, karena *device kinect* sendiri yang pada dasarnya adalah barang konsumen, sehingga akurasi dan keandalannya lebih rendah.

Deep Learning menjadi topik yang berkembang pesat dalam waktu beberapa tahun terakhir. Pengaplikasian dari *deep learning* sangat beragam, dari masukan berupa teks, gambar maupun video. Tuntutan untuk mendapatkan *human pose estimation* yang lebih baik juga semakin berkembang, beberapa tuntutan diantaranya adalah, pertama bagaimana mendapatkan *human pose estimation* dengan baju yang beragam, bentuk badan yang bervariasi, pose yang bervariasi hingga tuntutan *human pose estimation* yang dapat dilakukan pada orang banyak. Dengan beberapa

tuntutan diatas, dan dengan beberapa kekurangan dari penelitian berbasis *marker*, maka *deep learning* dijadikan sebagai salah satu solusi pendekatan berbasis *markerless motion capture* yang dianggap dapat memenuhi kebutuhan diatas.

Terdapat beberapa poin perbedaan antara penelitian *human pose estimation* berbasis *marker* dan *markerless*. Pertama, pada penelitian berbasis *marker* membutuhkan penanda dan memerlukan waktu untuk setup penanda (Xu et al., 2018), sedangkan pada penelitian berbasis *markerless* tidak memerlukan penanda dan juga setup penanda (Chen and Ramanan, 2017), (Belagiannis et al., 2016), (Lin, 2019). Kedua, lingkungan pada penelitian berbasis *marker*, lebih terbatas, biasanya hanya bisa dilakukan didalam ruangan saja, sedangkan pada penelitian berbasis *markerless* lingkungannya lebih beragam (Cao et al., 2017), bisa didalam ruangan maupu luar ruangan, bahkan dalam lingkungan yang kompleks sekalipun. Ketiga, alat yang digunakan pada penelitian berbasis *marker* terbilang cukup mahal (Liu et al., 2016), sedangkan pada penelitian berbasis *markerless* alat yang digunakan masih dapat dijangkau oleh semua kalangan. Keempat, pada penelitian berbasis *marker* memiliki keterbatasan jika dilakukan pada banyak orang, sedangkan pada penelitian berbasis *markerless* sudah dapat dilakukan pada banyak orang (Cao et al., 2017).

Pada penelitian 3D *human pose estimation*, terdapat dua macam masukan, yaitu *single view* dan *multiple view*. (Ge et al., 2018) menyebutkan, beberapa perbedaan antara *single view* dan *multiple view*. Pertama, pada *single view camera*, eror dari estimasi kedalaman atau *depth* yang dihasilkan lebih besar, sedangkan pada *multiple view camera*, dengan teknik penggabngan dua *confidence maps* pada masing-masing view dapat mengestimasi lokasi *joint* lebih robust dibandingkan dengan *single view camera*. Kedua, pada *single view* tidak dapat memetakan *confidence maps* yang sebenarnya, sedangkan pada *multiple view camera*, ambiguitas dapat dikurangi dengan *confidence maps* yang ada pada view lain, jika pada view

tersebut tidak terdeteksi.

3D *human pose estimation* dapat dihasilkan dari dataset pose 3D ataupun dari lokasi *joint* 2D. Namun, dengan keterbatasan penggunaan dataset pose 3D, dimana sumber dataset pose sedikit (Kasaei and Sanaei, 2018) dan jika ingin membangun dataset pose 3D sendiri akan memerlukan usaha dan waktu yang lebih untuk mendapatkan *ground truth* pada koordinat dunia (Hwang, 2019). (Martinez et al., 2017) juga menyebutkan bahwa dengan 3D *human pose estimation* yang diproses dari lokasi *joint* 2D dapat meminimalkan kemungkinan eror. Dari beberapa kasus di atas, kami dapat mengambil keputusan untuk praktek dan metodologi terbaik pada sistem kami. Kami memilih untuk mengembangkan penelitian berbasis *markerless* dengan *multiple view camera* dan lokasi *joint* 2D sebagai masukan untuk mendapatkan 3D *human pose estimation*.

Masukan dari penelitian ini berupa dua *view* yang berbeda sudut pandangnya, namun bersamaan dalam pengambila objeknya. Dari setiap *view* kamera tersebut, masing-masing dilanjutkan pada proses deteksi lokasi *joint* 2D. Proses kalibrasi kamera dibutuhkan untuk mendapatkan fitur intrinsik dan ekstrinsik kamera. Dengan fitur tersebut dan hasil dari lokasi *joint* 2D, kita akan mendapatkan 3D *human pose estimation* dengan pendekatan metode triangulasi.

1.2 Rumusan Masalah

1. Bagaimana proses *markerless motion capture* berbasis *openpose* dan metode triangulasi didapatkan?
2. Bagaimanakah kehandalan *markerless motion capture* dalam berbeda baju, gender, pose dan percobaan skenario lainnya?

1.3 Tujuan

Tujuan yang ingin dicapai dari penelitian ini adalah menerapkan konsep *markerless motion capture* dengan penggunaan *openpose* dan pendekatan

metode triangulasi dari *multiple view camera* untuk mendapatkan 3D *motion capture*. Penelitian ini juga akan mengamati sejauh mana kehandalan rekonstruksi *markerless motion capture* yang didapat dari berbagai pose manusia.

1.4 Batasan masalah

Agar penelitian ini menjadi terarah terhadap permasalahan yang akan dibahas, maka perlu diberikan beberapa batasan masalah. Berikut adalah beberapa batasan masalah tersebut:

1. *Markerless motion capture* yang dikembangkan baru berbasis *image*, belum video atau *real-time*
2. Bentuk *markerless motion capture* berbentuk tiga dimensi (3D)
3. Objek manusia yang dapat diproses hanya satu orang
4. Pengambilan citra gambar pada masing-masing kamera diambil bersamaan
5. Dalam satu frame citra gambar, semua *joint* harus terlihat (dari kepala hingga kaki)
6. Proses 2D *Pose Estimation* diperoleh dari *inference model* dari *openpose*
7. Jumlah kamera yang digunakan adalah dua
8. Posisi kamera diberi sudut, tidak paralel untuk mendapatkan hasil 2D *joint* yang baik
9. Posisi wajah seseorang harus lihat kedepan

1.5 Kontribusi

Dengan melakukan implementasi *markerless motion capture* berbasis *openpose model* menggunakan metode triangulasi diharapkan dapat memperoleh *motion capture* yang lebih baik, dimana pada *markerless motion capture* ini sistem tidak perlu mengeluarkan waktu dan tenaga dan juga setup penanda, selain itu pada *markerless motion capture* diharapkan dapat dilakukan di lingkungan yang lebih beragam dan kompleks dan dapat dilakukan pada pose yang beragam.

Penelitian ini juga diharapkan dapat digunakan untuk pengaplikasian dari konsep *markerless motion capture*. Salah satu pengaplikasian yang memungkinkan adalah untuk *mimicking* gerakan manusia yang dilakukan oleh *Robot Humanoid*, selain itu *generate aset* animasi 3D juga dapat diaplikasikan menggunakan pendekatan konsep ini. Sehingga, penelitian ini juga diharapkan dapat diterapkan pada aplikasi selain dua tadi dan dapat dilakukan pengembangan penelitian di kemudian hari.

Halaman ini sengaja dikosongkan

BAB 2

KAJIAN PUSTAKA *MARKERLESS* *MOTION CAPTURE*

2.1 Penelitian Terkait *Human Pose Estimation*

2.1.1 *Pose Estimation* berbasis *marker* dan *markerless*

Cakupan penelitian pada *human pose estimation* dibagi menjadi dua bagian. Pertama adalah penelitian berbasis *marker* dan penelitian berbasis *markerless* atau tanpa penanda. Beberapa penelitian masih menggunakan *marker* untuk mendapatkan skeleton manusia. *Optical motion* menjadi salah satu alat untuk mendapatkan skeleton manusia, hasil *motion capture* yang didapatkan memang lebih akurat, namun *optical motion* ini tidak praktis, karena pengguna harus menggunakan baju penanda dimana membutuhkan waktu dan upaya yang lebih untuk menggunakannya (Liu et al., 2016), selain itu, *optical motion* terbilang cukup mahal dan tidak bisa dijangkau oleh semua pengguna.

Selain *optical motion* ada beberapa penelitian yang menggunakan *device kinect*. Beberapa diantaranya adalah (Agarwal et al., 2016) yang menggunakan *device kinect* untuk mendapatkan skeleton manusia, dimana hasil dari skeleton yang didapat akan ditiru oleh kepala robot. Selain itu, (Hwang et al., 2017) juga menggunakan *device kinect* untuk mendapatkan skeleton tubuh manusia, yang dijadikan referensi lokasi *joint* untuk ditiru oleh Robot Humanoid Nao. *Device kinect* terkadang mengalami kendala dalam mendapatkan skeleton manusia (Liu et al., 2016). Jika kita lihat pada Gambar 2.1, dimana diperlihatkan bahwa hasil deteksi lokasi *joint* yang ada pada orang tersebut masih belum tepat, hal ini biasa terjadi akibat *self occlusions* dari orang tersebut dan juga kesalahan yang berasal dari sensor

kinect itu sendiri (Liu et al., 2016).

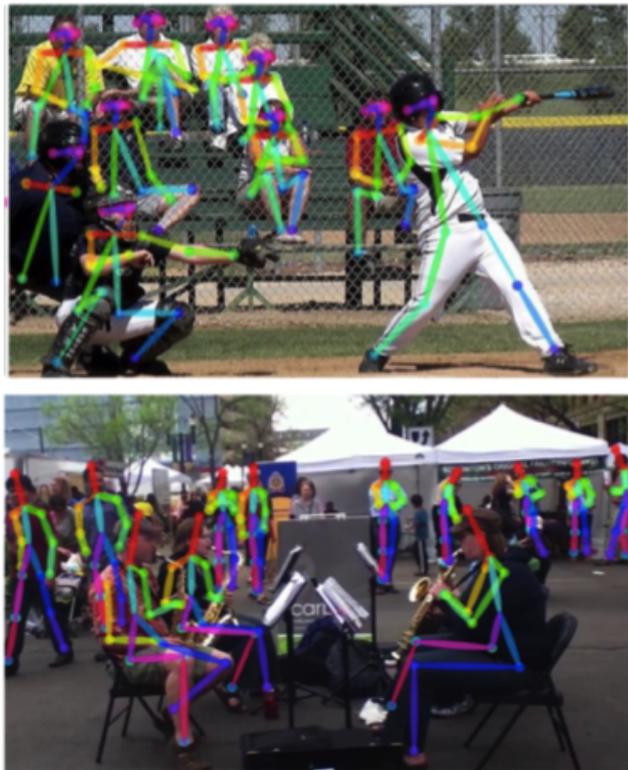


Gambar 2.1: Kendala *Kinect* pada Deteksi *joint*(Liu et al., 2016)

Sedangkan jika kita lihat pada Gambar di 2.2, hasil penelitian *human pose* berbasis *markerless* deteksi *joint* yang didapat lebih baik, kemudian orang yang ada pada gambar tersebut berhasil didapatkan estimasi posenya meskipun hanya berpakaian biasa sehari-hari tanpa penggunaan *marker* dan *setup marker* pada badannya, selain itu, baju yang digunakan pun sangat beragam tidak ada keterbatasan dalam penggunaan baju untuk mendapatkan *human pose estimation* dalam penelitian berbasis *markerless* ini. Selain itu, jika kita perhatikan pada Gambar 2.2, dimana estimasi pose dapat dilakukan pada *background scene* yang lebih kompleks, bisa di dalam ruangan ataupun di luar ruangan, dengan latar belakang taman, lapangan, bahkan latar belakang yang lebih kompleks sekalipun tidak akan mempengaruhi hasil dari estimasi ini. Selain itu, penelitian berbasis *markerless* sudah dapat diterapkan dalam skala yang lebih besar, tidak hanya satu orang, melainkan banyak orang atau *multi-person*.

Dengan hasil penelitian berbasis *marker* dan *markerless* di atas, dapat disimpulkan beberapa poin. Diantaranya adalah, pertama pada penelitian berbasis *marker* membutuhkan penanda dan memerlukan waktu untuk setup penanda (Xu et al., 2018), sedangkan pada penelitian berbasis *markerless* tidak memerlukan penanda dan juga *setup* penanda (Chen and Ramanan,

2017), (Belagiannis et al., 2016), (Lin, 2019). Kedua, pada lingkungan penelitian berbasis *marker* lebih terbatas, biasanya hanya bisa dilakukan dalam ruangan saja, sedangkan pada penelitian berbasis *markerless* lingkungannya lebih beragam (Cao et al., 2017), bisa didalam ruangan maupun luar ruangan, bahkan dapat juga dilingkungan yang kompleks sekalipun. Ketiga, alat yang digunakan penelitian berbasis *marker* terbilang cukup mahal, sedangkan pada penelitian berbasis *markerless* alat yang digunakan masih dapat dijangkau oleh semua kalangan. Keempat, pada penelitian berbasis *marker* memiliki keterbatasan jika dilakukan pada banyak orang, sedangkan pada penelitian berbasis *markerless* sudah dapat dilakukan pada banyak orang (Cao et al., 2017). Dengan beberapa kelebihan penelitian berbasis *markerless* dan kendala yang terkadang dialami oleh *device kinect*, maka pada penelitian kali ini akan menggunakan pendekatan berbasis *markerless* dengan konsep yang akan dijabarkan pada buku ini.



Gambar 2.2: Hasil estimasi pose penelitian berbasis *markerless* (Cao et al., 2017)

2.1.2 Pose Estimation multiview dan single view

Dengan perkembangan *deep learning*, *human pose estimation* semakin berkembang mengikuti pengaplikasian *pose estimation* sendiri yang juga semakin luas, sehingga tuntutan dan tantangan pada penelitian *human pose estimation* semakin bertambah. Selain penelitian berbasis *marker* dan *markerless*. Kategori *human pose estimation* juga bisa dikelompokkan dalam penelitian *multiview* dan *single view* (Chen and Ramanan, 2017) menerapkan *deep learning* untuk membantu dalam menemukan estimasi 3D pose dengan *single view*. Selain itu, (Ge et al., 2018) menerapkan estimasi pose 3D dengan *multiview CNN* dan diterapkan untuk mendapatkan skeleton, namun fokus pengaplikasian estimasi pose 3D pada penelitian ini hanya pada telapak tangan saja.

Pada penelitian (Ge et al., 2018) menyebutkan beberapa kekurangan dari penelitian berbasis *single view camera*, diantaranya disebutkan bahwa pada inputan *single view camera*, eror dari estimasi kedalaman yang dihasilkan pada *single view* akan lebih besar, sedangkan pada *multiple view* dengan teknik *confidence maps* dapat mengestimasi lokasi 3D *joint* lebih handal dibandingkan dengan *single view*. Kelemahan kedua, *single view* tidak dapat menentukan posisi yang sebenarnya diantara beberapa *confidence maps*, sedangkan pada *multiple view*, ambiguitas dapat dikurangi dengan *confidence maps* yang ada pada *view* yang lain, jika tidak terdeteksi di *view* satunya. Kekurangan *single view camera* yang disebutkan menjadi salah satu alasan mengapa penelitian ini menggunakan *multiple view camera*.

2.2 Stereo Vision dan Depth Perception

Pada dasarnya konsep kerja stereo kamera berasal dari konsep cara kerja mata kita dalam melihat sesuatu. Ketika kita melihat sesuatu dengan kedua mata kita, masing-masing retina mata dapat melihat objek tersebut. Dengan posisi mata kanan dan kiri terpisahkan satu sama lain, gambar yang didapat pada masing-masing mata tidak identik (Cyganek and Siebert, 2009).

Dengan demikian untuk mendapatkan seberapa jauh objek tersebut yang ditangkap oleh kedua mata kita, dapat dilakukan dengan dua hal. Pertama, adalah dengan mengetahui bagian-bagian lokal dari suatu citra *stereo-pair* yang cocok dengan bagian-bagian yang terkait dalam gambar lain dari pasangan stereo, untuk menemukan paralaks lokal. Kedua, sifat geometris dan konfigurasi mata yang tepat. Dengan konsep pendekatan dari kedua mata, jika kita melakukannya pada kamera, maka untuk mendapatkan *depth perception* atau jarak relatif dari kamera ke objek diperlukan proses konfigurasi dari kedua kamera melalui proses kalibrasi.

2.3 Rekonstruksi 3D dan Konsep Triangulasi

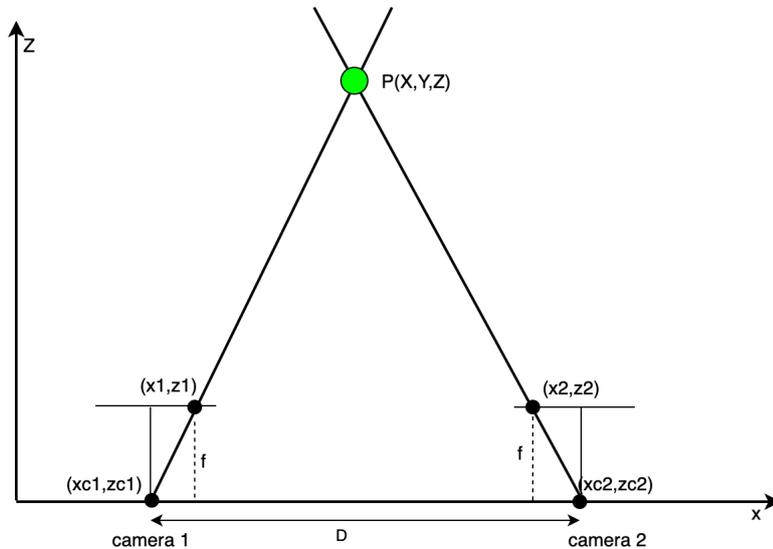
Rekonstruksi 3D pada stereo kamera bisa didapatkan tergantung dengan keadaan pada sistem stereo yang kita pakai. Dengan perbedaan kondisi sistem stereo, berbeda juga dalam proses pendekatan untuk mendapatkan rekonstruksi 3D nya, untuk itu rekonstruksi 3D dapat dibagi menjadi tiga berdasarkan kondisi sistem stereo :

1. Jika parameter intrinsik dan ekstrinsik kamera diketahui. Rekonstruksi 3D didapatkan dari konsep triangulasi (Hartley, 2004).
2. Jika hanya parameter intrinsik saja yang diketahui, maka yang dilakukan adalah dengan memperkirakan parameter ekstrinsik dan menyelesaikan permasalahan rekonstruksi hingga faktor skala yang tidak diketahui
3. Jika parameter intrinsik maupun ekstrinsik tidak diketahui, dan satu-satunya informasi yang tersedia adalah korespondensi piksel. Maka penyelesaian rekonstruksi hanya sampai pada transformasi proyeksi dunia yang tidak diketahui

Berdasarkan penjelasan diatas, sesuai dengan kondisi sistem stereo yang kita pakai, dimana parameter intrinsik dan parameter ekstrinsik diketahui, maka untuk mendapatkan 3D rekonstruksi kita menggunakan pendekatan konsep Triangulasi.

Triangulasi (Hartley, 2004) merupakan konsep pendekatan masalah

untuk menemukan posisi titik dalam ruang, dimana triangulasi dapat dilakukan dengan syarat posisi titik akan didapat dari perbedaan dua sudut pandang gambar. Konsep dasar dari triangulasi adalah perpotongan dari dua sinar dalam sebuah ruang. Untuk lebih jelasnya dapat dilihat pada Gambar 2.3 :



Gambar 2.3: Triangulasi dari dua *view* yang berbeda.

Dimana (x_{c1}, z_{c1}) merupakan posisi koordinat kamera pertama dan (x_{c2}, z_{c2}) merupakan posisi koordinat kamera kedua. (x_1, z_1) merupakan lokasi koordinat titik objek x dan z pada kamera pertama. (x_2, z_2) merupakan lokasi koordinat titik objek x dan z yang didapatkan pada kamera kedua.

Pendekatan konsep triangulasi bisa dilakukan dari konsep perpotongan oleh dua buah garis. jika kita lihat pada Gambar 2.3, titik P didapat dari perpotongan garis dari kamera pertama dan garis dari kamera kedua. Perpotongan dua buah garis didapatkan ketika garis pertama sama dengan garis kedua, untuk itu penyelesaian konsep triangulasi dapat dituliskan pada persamaan berikut :

Untuk garis pertama, persamaan didapatkan dari dua titik, yaitu titik (x_{c1}, z_{c1}) dan (x_1, z_1) , sehingga dapat ditulis seperti Persamaan 2.1, 2.2 dan 2.3:

$$\frac{x - xc_1}{x_1 - xc_1} = \frac{z - zc_1}{z_1 - zc_1} \quad (2.1)$$

$$x(z_1 - zc_1) - xc_1(z_1 - zc_1) = z(x_1 - xc_1) - zc_1(x_1 - xc_1) \quad (2.2)$$

$$x(z_1 - zc_1) - z(x_1 - xc_1) = xc_1(z_1 - zc_1) - zc_1(x_1 - xc_1) \quad (2.3)$$

Sedangkan untuk garis kedua, bentuk persamaannya sama seperti pada garis pertama, yang didapat dari dua titik, yaitu titik (xc_2, zc_2) dan (x_2, z_2) . Berikut 2.4 adalah Persamaan dari garis kedua :

$$x(z_2 - zc_2) - z(x_2 - xc_2) = xc_2(z_2 - zc_2) - zc_2(x_2 - xc_2) \quad (2.4)$$

Setelah mendapatkan persamaan dari kedua garis tersebut. Selanjutnya adalah mencari perpotongan dua buah garis. Dua buah garis dikatakan berpotongan, ketika garis pertama sama dengan garis kedua. Untuk lebih memudahkannya, persamaan dari dua buah garis kita tulis dalam bentuk matriks pada Persamaan 2.5.

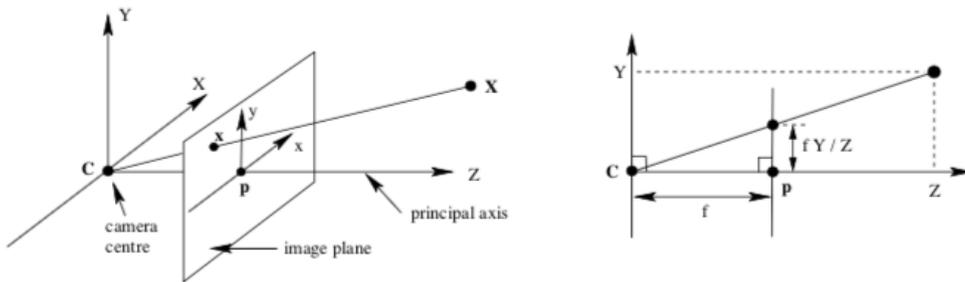
$$\begin{pmatrix} z_1 - zc_1 & x_1 - xc_1 \\ z_2 - zc_2 & x_2 - xc_2 \end{pmatrix} \begin{pmatrix} x \\ z \end{pmatrix} = \begin{pmatrix} xc_1(z_1 - zc_1) & -zc_1(x_1 - xc_1) \\ xc_2(z_2 - zc_2) & -zc_2(x_2 - xc_2) \end{pmatrix} \quad (2.5)$$

Sehingga, untuk mendapatkan titik potong x, z dari dua buah garis persamaan diatas dapat kita sederhanakan dalam bentuk Persamaan 3.7.

$$\begin{pmatrix} x \\ z \end{pmatrix} = \begin{pmatrix} z_1 - zc_1 & x_1 - xc_1 \\ z_2 - zc_2 & x_2 - xc_2 \end{pmatrix}^{-1} \begin{pmatrix} xc_1(z_1 - zc_1) & -zc_1(x_1 - xc_1) \\ xc_2(z_2 - zc_2) & -zc_2(x_2 - xc_2) \end{pmatrix} \quad (2.6)$$

2.4 Proyeksi Geometri dan Kamera Model

Kamera merupakan *mapping* dari 3D koordinat dunia ke dalam bentuk 2D *image*. Untuk lebih memahami hubungan antara 3D koordinat dunia dan 2D *image*, maka gambaran yang paling sederhana yang dapat mewakili adalah dengan mempelajari kamera model dari kamera pinhole dan bagaimana proyeksi geometrinya (Hartley, 2004). Sehingga mendapatkan hubungan antar keduanya. Pada Gambar 2.4 adalah ilustrasi dari proyeksi geometri dari kamera pinhole.



Gambar 2.4: Ilustrasi Proyeksi Kamera Model Pinhole (Hartley, 2004)

Pada Gambar 2.4 terlihat kamera memiliki C sebagai *center projection* dan *principal axis* yang sejajar dengan Z , *image plane* terletak pada koordinat $P(u, v)$ dengan jarak f (*focallength*) dari pusat kamera atau *center projection*. Dan titik $Pc = (X, Y, Z)$ merupakan titik koordinat dunia. Untuk mendapatkan nilai $P(u, v)$ bisa didapat dari perbandingan dua buah segitiga. Sehingga dapat kita ambil persamaan seperti Persamaan 2.7 dan 2.8.

$$\frac{f}{Z} = \frac{u}{X} = \frac{v}{Y} \quad (2.7)$$

$$u = \frac{fX}{Z}, v = \frac{fY}{Z} \quad (2.8)$$

Selain menggunakan konsep tersebut, kita juga dapat menggunakan konsep *homogenous coordinate*, kita dapat menuliskannya seperti Persamaan 2.9 berikut:

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (2.9)$$

Sehingga $P = (u, v, w) = (fX/Z, fY/Z, 1)$. Kemudian, jika asal dari sistem koordinat gambar 2D tidak sesuai dengan sumbu Z yang memotong bidang gambar, maka kita perlu mendefinisikan Pc ke koordinat asal dengan (tu, tv) , sehingga, apabila kita tulis lagi persamaan diatas, akan didapatkan Persamaan 2.10.

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} f & 0 & cx \\ 0 & f & cy \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (2.10)$$

Sehingga, jika kita perhatikan bahwa proyeksi dari 2D ke 3D memerlukan nilai intrinsik kamera K , dengan K seperti pada Persamaan 2.11.

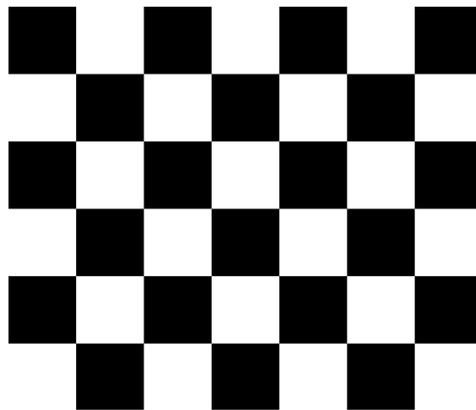
$$K = \begin{pmatrix} f & 0 & cx \\ 0 & f & cy \\ 0 & 0 & 1 \end{pmatrix} \quad (2.11)$$

Sehingga pada akhirnya didapat model proyeksi 2D ke 3D dari kamera pinhole didapat dengan $P = K Pc$

2.5 Kalibrasi Kamera

Kalibrasi kamera merupakan proses untuk memperkirakan parameter lensa dan kamera. Parameter ini bisa digunakan untuk beberapa hal, beberapa diantaranya adalah untuk mengoreksi distorsi lensa, mengukur objek di koordinat dunia, ataupun untuk menentukan lokasi objek terhadap kamera. Beberapa pengaplikasian hasil parameter dari proses kalibrasi kamera ini banyak digunakan pada bidang visi komputer (Zhang and Member, 2000), baik untuk mendeteksi ataupun mengukur objek. Selain itu, pengaplikasian kalibrasi kamera juga digunakan pada bidang robotika, sistem navigasi dan juga rekonstruksi 3D.

Untuk mendapatkan parameter kamera, dibutuhkan koordinat 3D dunia, dan gambar 2D. Sehingga, untuk memudahkan proses kalibrasi, biasanya digunakan *checkerboard* sebagai koordinat 3D dunia, dan gambar *checkerboard* yang diambil dari kamera tersebut. Contoh *checkerboard* dapat dilihat pada Gambar 2.5.



Gambar 2.5: Contoh Checkerboard

Parameter kamera yang dihasilkan ada beberapa (Hartley, 2004), diantaranya adalah parameter intrinsik kamera dan kedua adalah parameter ekstrinsik kamera. Parameter intrinsik kamera berisi dari informasi internal dari kamera itu sendiri. Berikut adalah beberapa parameter intrinsik yang dapat dituliskan dalam bentuk matriks yang dapat kita lihat pada Persamaan 2.12. Dengan $f_x =$ dan f_y merupakan focal length dalam satuan

pixel. cx dan cy merupakan principal point dari kamera.

$$K = \begin{pmatrix} f & 0 & cx \\ 0 & f & cy \\ 0 & 0 & 1 \end{pmatrix} \quad (2.12)$$

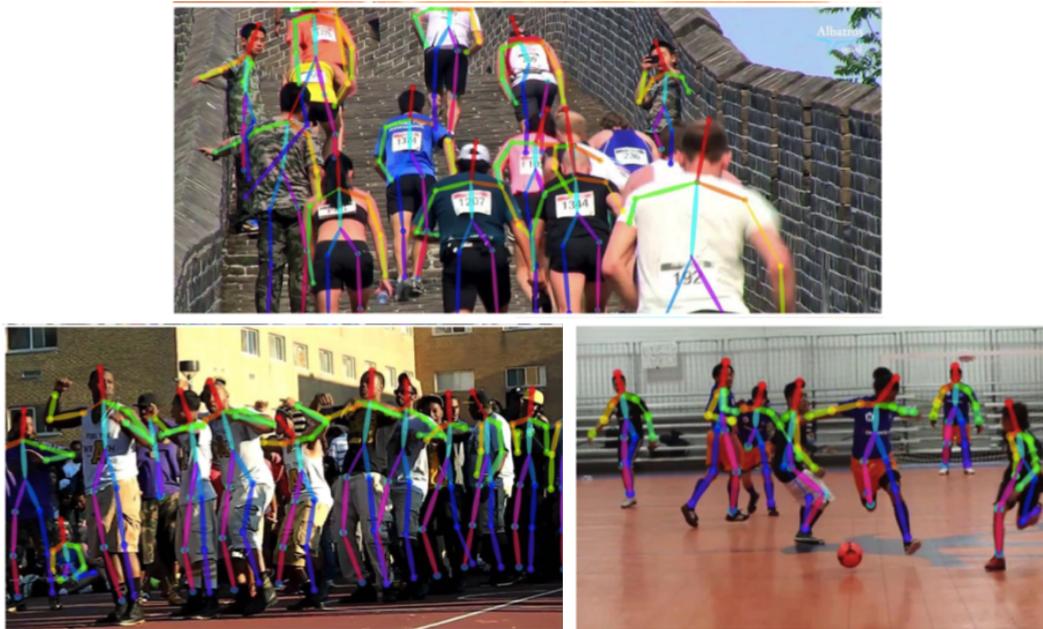
Berbeda dengan parameter intrinsik, parameter ekstrinsik mendeskripsikan posisi dan juga orientasi kamera pada koordinat dunia. Sehingga, jika kita lihat pada Persamaan 2.13 di bawah terdapat parameter ekstrinsik yang dituliskan pada matriks rotasi dan matriks translasi dari kamera yang terkalibrasi.

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}, T = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \quad (2.13)$$

2.6 Estimasi 2D *Human Pose* Menggunakan *Openpose*

Pada dasarnya *openpose* (Cao et al., 2017) dilakukan karena ada dua permasalahan besar yang ingin diselesaikan, pertama yaitu dalam setiap citra gambar yang diambil kamera, terdapat jumlah orang yang tidak dapat diketahui dengan pasti. Kedua, dengan banyaknya jumlah orang yang ada tersebut, dan dengan interaksi yang kompleks antar orang, maka sangat mungkin terjadi oklusi, dan untuk mendapatkan hubungan antar part dalam setiap anggota tubuh dalam setiap orang yang berada pada gambar akan sulit. Oleh karena itu, mendapatkan estimasi pose 2D dengan *multi-person* merupakan suatu tantangan tersendiri.

Pada Gambar 2.6 merupakan contoh dari beberapa hasil *openpose* 2D *pose estimation* pada *multi-person*. Terlepas dari kesuksesan (Cao et al., 2017) ada beberapa kasus tertentu, dimana *openpose* tidak menghasilkan



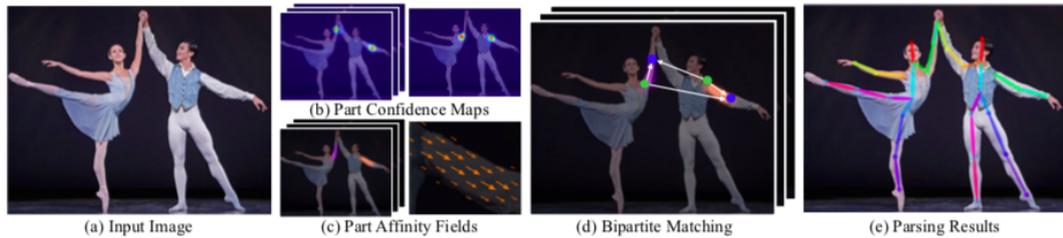
Gambar 2.6: Hasil Estimasi Pose 2D oleh *Openpose*(Cao et al., 2017)

hasil yang baik. Beberapa contoh Gambar 2.7 merupakan kasus-kasus dari *openpose* yang masih belum baik dalam mendapatkan estimasi posenya. Gambar pertama dari 2.7 adalah *openpose* masih mengalami kendala ketika gambar input diambil dari pose yang tidak wajar, dimana tangan di kaki dan kaki di tangan, akibatnya, hasil estimasi pose yang keluar masih keliru. Gambar kedua dari 2.7 *openpose* mengalami kendala ketika dua orang berdiri berdekatan, akibatnya hasil estimasi pose yang didapat tampak samar dan menyatu. Sedangkan pada gambar ketiga dari 2.7, *openpose* mengalami kendala ketika ada inputan terdapat objek patung hewan, dimana sistem *openpose* menganggap patung sebagai manusia.



Gambar 2.7: Beberapa Kendala Estimasi Pose 2D oleh *Openpose*(Cao et al., 2017)

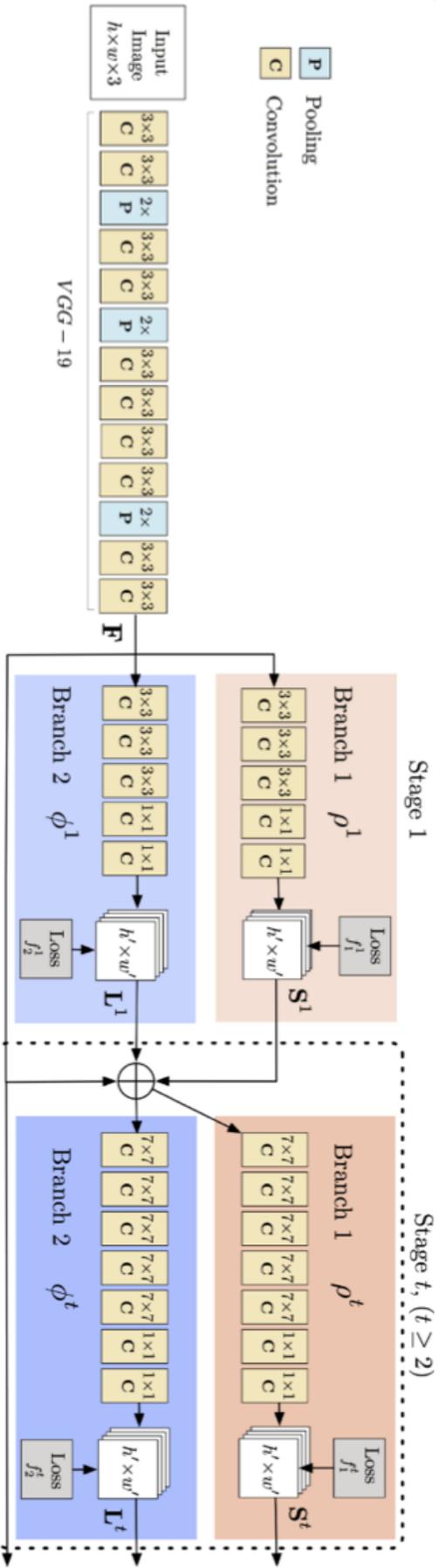
Ilustrasi Keseluruhan dari proses *openpose* dari setiap tahapan dapat kita lihat pada Gambar 2.8. Pertama, citra inputan akan diproses dan mendapatkan dua hasil secara paralel, yaitu mendapatkan hasil dari *confidence maps* untuk setiap *body part* seperti pada Gambar 2.8 bagian (a), dan hasil vector 2D dengan metode *part affinity fields* seperti Gambar 2.8 bagian (b). Dari kedua hasil tersebut, maka akan dilanjutkan pada proses mendapatkan hubungan untuk setiap *joint* di setiap kandidat *joint* dengan pendekatan *bipartate matching*, dan dari proses tersebut akan didapatkan hasil akhir dari 2D *pose estimation*. Untuk lebih jelasnya, di bawah akan di jelaskan lebih terinci.



Gambar 2.8: Ilustrasi Keseluruhan Tahapan *Openpose*(Cao et al., 2017)

2.6.1 Arsitektur *Openpose*

Arsitektur dari *openpose* dapat kita lihat pada Gambar 2.9. Tahapan dari arsitektur tersebut dimulai dari citra gambar input dengan ukuran $w \times h$ *pixel*, citra kemudian diteruskan untuk diproses pada 10 *layer* dari arsitektur *VGG-19*. Hasil dari proses 10 *layer VGG-19* berupa *feature maps F*. *Feature maps F* ini adalah beberapa *feature* yang dibutuhkan untuk proses selanjutnya, dimana proses selanjutnya akan dilakakukan secara simultan yang terdiri dari dua cabang. Cabang pertama dari *feature maps* yang ada akan didapatkan himpunan dari 2D *confidence maps S* dari lokasi *body part* yang dideteksi. $S = \{S_1, S_2, \dots, S_j\}$, dengan J adalah banyak *confidence maps* dari setiap *part* yang ada pada gambar tersebut. Kemudian pada cabang kedua *feature maps* yang ada akan digunakan untuk mendapatkan himpunan dari 2D vector L . $L = \{L_1, L_2, \dots, L_C\}$ dengan C adalah banyaknya set vector 2D dari setiap *limb*.



Gambar 2.9: Arsitektur Openpose(Cao et al., 2017)

Jika kita lihat pada Gambar 2.9 arsitektur diatas. Hasil dari *convolutional network* dari *VGG-19* menghasilkan *feature maps* F , dimana *feature maps* ini yang akan digunakan sebagai input pada *stage* 1 pada masing-masing cabang. Pada *stage* 1, *network* menghasilkan himpunan dari *confidence maps* $S^1 = \rho^1(F)$ dan himpunan dari *vector fields* $L^1 = \sigma^1(F)$. Dan untuk *stage* selanjutnya, hasil prediksi dari kedua cabang pada *stage* sebelumnya digunakan untuk dijadikan input pada *stage* selanjutnya, atau dapat dituliskan dengan Persamaan 2.14 dan 2.14 berikut :

$$S^t = \rho^t(F, S^{t-1}, L^{t-1}), \forall t \geq 2 \quad (2.14)$$

$$L^t = \sigma^t(F, S^{t-1}, L^{t-1}), \forall t \geq 2 \quad (2.15)$$

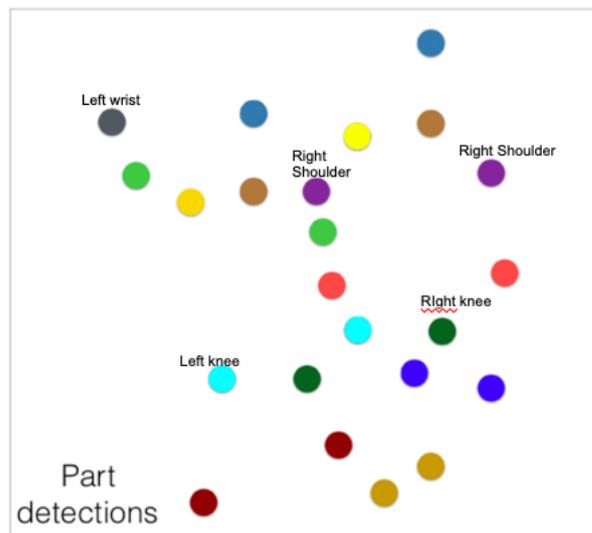
Untuk memperbaiki *confidence maps* dan *vector fields* dari setiap *stage*, dilakukan perhitungan *loss function* di setiap akhir *stage*. *Loss function* digunakan untuk mengukur perbedaan prediksi estimasi yang didapat dengan *ground truth* nya. *Stage* ini berlanjut hingga *stage* ke- t . Jika kita perhatikan dari arsitektur tersebut, masing-masing cabang membagi fokus ke dalam dua hal, yaitu *part detection* dan *part association*. Dibawah ini akan dijelaskan lebih dalam mengenai kedua hal tersebut.

2.6.2 Part Detection

Part Detection atau deteksi bagian tubuh didapatkan dari nilai *confidence maps* dari setiap lokasi *body part* dari tubuh manusia. Setiap nilai *confidence maps* merupakan representasi 2D dari keyakinan bahwa lokasi dari beberapa *body part* berada pada setiap *pixel* tersebut. *Confidence maps* memetakan bahwa lokasi *joint* atau bagian tersebut ada pada *pixel* tertentu dalam gambar. Misal, sebuah citra gambar mendapatkan *confidence maps* dari bahu, maka

representasi nilai dari *confidence maps* bahu ada di skala 0 hingga 1, bahwa itu adalah *body part* bahu kanan yang dicari, dan begitu seterusnya, hingga semua *body part* terdeteksi.

Jika dalam citra inputan terdapat beberapa *confidence maps* dari bagian tubuh, maka sebanyak itu juga hasil dari deteksi *part* atau bagian tubuh yang didapat. Gambar 2.10 merupakan salah satu contoh gambaran dari citra masukkan dalam proses *part detection*.



Gambar 2.10: Ilustrasi Hasil *Part Detection*(Cao et al., 2017)

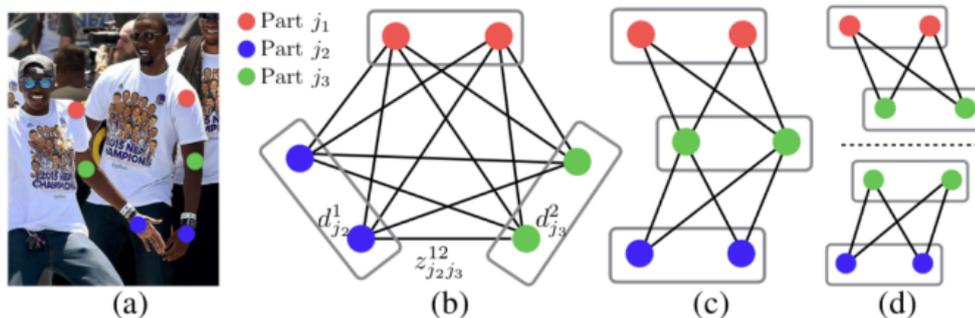
2.6.3 *Part Association*

Untuk mendapatkan *2D pose estimation*, setelah melakukan proses *part detection* atau deteksi part, maka kita harus mengetahui hubungan antar part yang sudah terdeteksi tadi. Konsep *part association* yang dipakai pada *openpose* adalah konsep *part affinity fields*. PAF akan menentukan pembobotan untuk penentuan apakah dua titik *part* yang terdeteksi saling berasosiasi atau tidak.

Secara keseluruhan, cara kerja untuk mendapatkan *2D pose estimation* dimulai dari konsep *K-partite graph*, dimana *nodes* dari *graph* tersebut adalah *body part* yang terdeteksi dan *edges* dari *graph* tersebut adalah semua

kemungkinan hubungan antara *nodes*. *K-partite graph* pada dasarnya akan mempartisi semua node yang terdeteksi menjadi set *k-graph* sehingga tidak ada dua node yang sama berdekatan. Ilustrasi tersebut dapat kita lihat pada Gambar 2.11 bagian (a) dan (b). *Edges* pada *graph* dibentuk berdasarkan *weight* dari *part affinity* masing-masing koneksi, sehingga bisa disederhanakan menjadi bentuk seperti pada Gambar 2.11 bagian (c).

Solusi dari *k-partite graph* ini jika dilihat secara komputasi itu sangat sulit (*NP-Hard*), untuk itu Z.Cao mengusulkan pengembangan dari *k-partite graph* menjadi bentuk set *bipartate graph* seperti Gambar 2.11 bagian (d). Pembagian dengan konsep *bipartate graph* pada dasarnya mengikuti domain dari masalah pemodelan tubuh manusi, sebagai contoh jika kita tahu pinggul tidak dapat terhubung secara langsung dengan kaki, tapi pinggul terhubung dahulu ke lutut, baru kemudian lutut ke kaki.



Gambar 2.11: Ilustrasi Proses *Part Association*(Cao et al., 2017)

2.7 Confidence Map

Confidence maps merupakan kemungkinan sebuah *pixel* adalah objek yang dicari dengan membandingkannya pada *groundtruth* dataset yang sebelumnya sudah diberi label. Sebagai contoh, dapat kita lihat pada Gambar 2.12, dimana pada Gambar 2.12 tersebut diperlihatkan hasil dari *confidence maps* pada lengan kiri dan kanan orang tersebut pada citra. Nilai *confidence maps* berkisar dari 0 hingga 1. Satu menunjukkan nilai kemungkinan yang paling mendekati *groundtrut* sedangkan 0 adalah nilai

confidence maps dari *body part* yang jauh atau tidak mendekati dari *groundtruth*.



Gambar 2.12: Contoh Ilustrasi Hasil *Confidence Maps*(Wei et al., 2016)

2.8 Arsitektur *VGG-19*

Pada awalnya *VGG-19* dibentuk untuk menganalisa dan investigasi efek dari *deep convolutional network* untuk pengenalan gambar pada *large-scale image* (Simonyan and Zisserman, 2014). Kontribusi utama dari *VGG-19* adalah menggunakan *convolutional filter* yang sangat kecil pada setiap peningkatan kedalaman arsitektur, dengan penggunaan *convolution filter* yang sangat kecil ini, jika kita mengevaluasi lebih dalam, arsitektur ini akan menunjukkan hasil yang signifikan pada kedalaman 16 – 19 *layer*.

Dengan konsep seperti ini, tim yang berasal dari *Visual Geometry Group, Department of Engineering Science, University of Oxford*, mengajukan arsitektur *VGG* pada kompetisi *ImageNet Classification*, dimana tim berhasil mendapatkan posisi pertama dan kedua dalam lokalisasi dan klasifikasi iamage. Selain itu, arsitektur *VGG-19* juga menunjukkan bahwa dapat digunakan untuk menggeneralisasi dengan baik pada dataset lain. Pada Gambar 2.13 merupakan beberapa konfigurasi dari arsitektur *VGG*.

2.9 *Convolutional Neural Network*

Convolutional Neural Network (CNN) adalah salah satu jenis *neural network*, dimana cara kerja *CNN* itu sendiri mengikuti dengan cara kerja dari jaringan saraf virtual (Géron, 2017). *CNN* pada dasarnya sudah digunakan sejak tahun 1980. Beberapa tahun terakhir, dengan

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Gambar 2.13: Konfigurasi Arsitektur *VGG-19* (Simonyan and Zisserman, 2014)

berkembangnya kekuatan komputasi, *CNN* semakin banyak digunakan pada beberapa penelitian yang berhubungan dengan gambar, namun tidak hanya itu, *CNN* juga dapat mengatasi dalam hal *voice recognition* dan *natural language processing(NLP)* (Géron, 2017).

Didalam *CNN* sebenarnya tujuan utama adalah mendapatkan *feature extraction*. Fitur yang diinginkan ini tergantung peneliti akan membutuhkan fitur-fitur yang seperti apa, sehingga jika kita lihat hingga kini, semakin berkembang sekali peneliti-peneliti *deep learning* untuk membuat arsitektur sendiri sesuai dengan fitur, komputasi, dan masalah yang ingin diselesaikan. Bagaimanapun bentuk arsitektur, *convolutional neural network* khususnya adalah kumpulan dari *convolutional layer*, *activation layer*, dan *pooling layer*. Berikut beberapa penjelasan dari masing-masing *layer*:

2.9.1 Convolutional Layer

Convolutional layer terdiri dari *neuron* yang tersusun sedemikian rupa sehingga membentuk sebuah filter dengan panjang dan tinggi (*pixel*). Filter ini didenisikan dengan nilai tertentu (secara *random* atau menggunakan teknik tertentu) dan nilai dari filter inilah yang akan menjadi parameter yang akan di *update* dalam proses *learning*. Di bawah ini adalah rumus umum untuk setiap inputan *convolutional layer* dengan filter tertentu menghasilkan *ouput layer* tertentu. *Output* dari *convolutional layer* dapat kita rumuskan menjadi formulas yang tertera pada Persamaan 2.16.

$$output = \frac{W - N + 2Pd}{St} + 1 \quad (2.16)$$

Dimana W adalah panjang atau tinggi input, kemudian N adalah panjang atau tinggi filter, Pd adalah banyak nya *zero padding* dan St adalah jumlah *stride*.

2.9.2 Pooling Layer

Pooling layer biasanya berada setelah *convolutional layer*. Pada prinsipnya, *pooling layer* terdiri dari sebuah filter dengan ukuran dan *stride* tertentu yang akan bergeser pada seluruh area *feature maps*. *Pooling* yang biasa digunakan adalah *max pooling* dan *average pooling*. Sebagai contoh, jika kita menggunakan *max pooling 2x2* dengan *stride 2*, maka pada setiap pergeseran filter, nilai maksimum pada area 2×2 tersebut akan dipilih, sedangkan *average pooling* akan memilih nilai rata-ratanya. Tujuan dari penggunaan *pooling layer* adalah untuk mengurangi dimensi dari *feature maps* yang didapat, sehingga akan mempercepat komputasi.

BAB 3

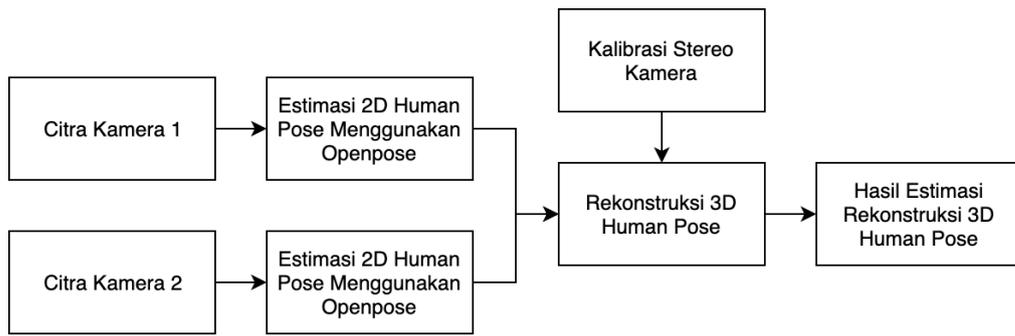
METODOLOGI *MARKERLESS MOTION CAPTURE*

3.1 Desain Sistem

Desain sistem untuk sistem untuk *markerless motion capture* berbasis *openpose model* menggunakan pendekatan metode triangulasi dapat dilihat pada Gambar 3.1. Pada Gambar 3.1, dapat dilihat bahwa sistem membutuhkan dua input citra. Input pertama adalah objek citra dari kamera pertama, dan input kedua adalah objek citra dari kamera kedua. Masing-masing citra kamera diambil secara bersamaan dalam pengambilan objeknya. Setiap input akan dilanjutkan pada proses estimasi 2D *human pose* dengan menggunakan proses *inference model* dari *openpose*. Dari hasil proses ini dihasilkan estimasi 2D *joint human pose* dari tiap-tiap input citra. Selain proses estimasi 2D *joint human pose*, sistem juga memerlukan proses kalibrasi stereo kamera, hal ini dikarenakan untuk mendapatkan fitur intrinsik dan fitur ekstrinsik kamera yang memudahkan kita dalam memproyeksikan 2D *human pose* ke 3D *human pose*. Dengan fitur tersebut dan hasil dari 2D *human pose estimation* kita akan mendapatkan 3D *human pose estimation* dengan pendekatan metode triangulasi. Detail dari setiap tiap-tiap proses pada diagram akan dijelaskan dibawah ini.

3.2 Data Input

Data Input dari sistem ini berupa citra kamera. Citra input pada kamera pertama dan citra input pada kamera kedua menggunakan spesifikasi kamera yang sama, yaitu kamera Logitech C920. Spesifikasi kamera *webcam* Logitech C920 dapat dilihat pada Tabel 3.1. Peletakkan antara kamera pertama dan kamera kedua diberi jarak tertentu, sesuai dengan jarak

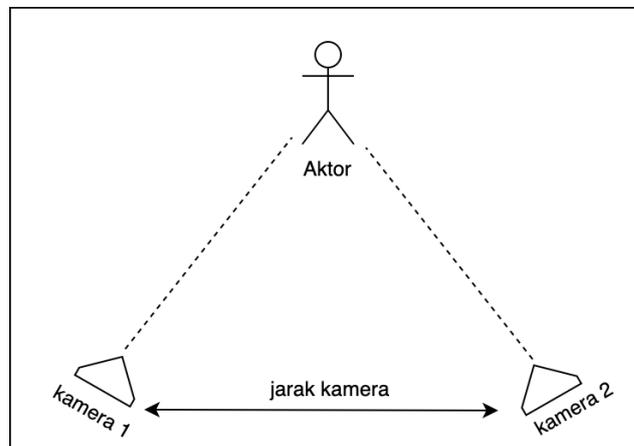


Gambar 3.1: Desain Sistem Penelitian.

skenario yang diinginkan. Pada Gambar 3.2 diilustrasikan bagaimana pengambilan posisi objek manusia terhadap kamera dan juga posisi kamera pertama dan kamera kedua terhadap objek.

Tabel 3.1: Spesifikasi Kamera Logitech C920

Spesifikasi Kamera Logitech C920	
Dimensi	29 mm x 94 mm x 24 mm
Resolusi	1080p/30 fps - 720p/30 fps
Panjang Kabel	1, 5 m



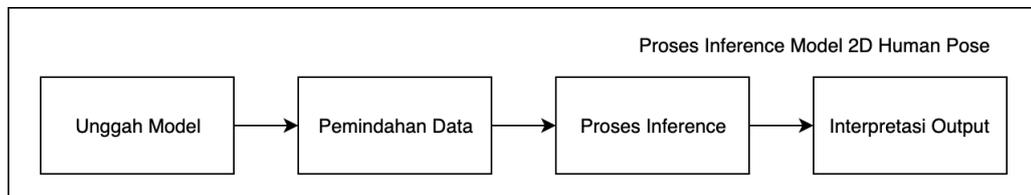
Gambar 3.2: Ilustrasi cara pengambilan gambar objek dengan kedua kamera.

3.3 Estimasi 2D *Human Pose* Menggunakan *Openpose*

Proses estimasi 2D *joint human pose* dibutuhkan sebagai sebuah input untuk mendapatkan rekonstruksi 3D *human pose*. Estimasi 2D *joint human pose* ini pada dasarnya mengadaptasikan konsep dari *openpose* (Cao et al., 2017) seperti penjelasan pada Bab 2. Dengan kesuksesan dan kecukupan dari

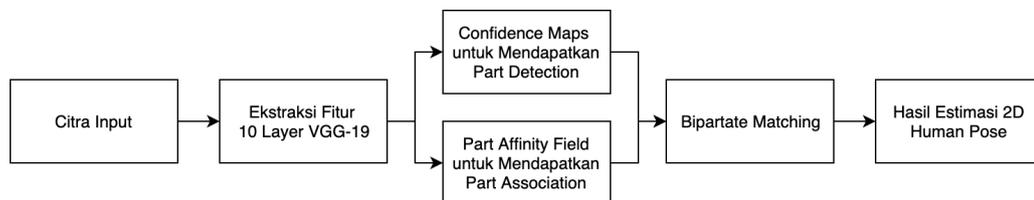
hasil estimasi 2D *human pose* yang dilakukan oleh *openpose* (Cao et al., 2017), maka untuk tahapan ini, kita akan menggunakan *inference model* dari metode *openpose*.

Inference model yaitu mengestimasi input sebagaimana sehingga akan menghasilkan *output* yang sesuai dengan *output* yang sudah di *training* pada metode *openpose* tersebut. Proses *inference model* itu sendiri jika dijabarkan dimulai dari mengunggah model, yaitu memasukkan model yang akan dipakai ke dalam sistem, setelah model sudah diunggah, dilanjutkan pada proses pemindahan data, maksudnya adalah input yang akan digunakan pada sistem ini akan di masukkan kedalam model tersebut. Setelah itu, proses *inference model*, yaitu sistem memproses input tadi ke dalam model yang akan digunakan, dan terakhir adalah menginterpretasi *output*, atau bisa dibilang adalah mengestimasi *output* dari inputan yang kita masukkan pada model tersebut. Deskripsi proses *inference model* dapat dilihat pada Gambar 3.3.



Gambar 3.3: Proses *Inference model 2D human pose*

Cara kerja dari *openpose* itu sendiri akan dideskripsikan beberapa dibawah ini. Pada Gambar 3.4 merupakan tahapan *openpose* untuk mendapatkan 2D *human pose estimation*.



Gambar 3.4: Proses *Openpose* dalam Mendapatkan 2D *Human Pose*

Openpose dimulai dari citra gambar 2D yang kemudian diteruskan untuk di proses pada 10 layer dari *VGG-19*. Hasil dari proses 10 layer *VGG-19*

berupa *feature maps* yang digunakan sebagai input pada proses selanjutnya. Proses selanjutnya terdiri dari dua cabang yang dilakukan secara bersamaan. Pada cabang pertama, dengan *feature maps* yang ada akan didapatkan himpunan dari 2D *confidence maps* S dari lokasi *body part* yang dideteksi. Kemudian, pada cabang kedua *feature maps* yang ada akan digunakan untuk mendapatkan himpunan dari 2D *vector fields* L dari setiap limb yang ada. Arsitektur dari *openpose* dapat dilihat pada 2.9.

3.3.1 *Part Detection*

Part Detection atau deteksi bagian tubuh didapatkan dari nilai *confidence maps* dari setiap lokasi *body part* dari tubuh manusia. Setiap nilai *confidence maps* merupakan representasi 2D dari keyakinan bahwa lokasi dari beberapa *body part* berada pada setiap pixel tersebut. Ilustrasi hasil dari *part detection* dapat dilihat pada Gambar 2.10.

3.3.2 *Part Association*

Untuk mendapatkan 2D *human pose estimation*, maka setelah kita mendapatkan deteksi untuk seluruh *body part* yang didapatkan, proses selanjutnya adalah bagaimana untuk mendapatkan hubungan antar *part* yang sudah didapatkan tadi diawal. Proses untuk mendapatkan hubungan antar *part* didefinisikan sebagai *part association*.

Secara keseluruhan, cara kerja untuk mendapatkan hubungan antar *part* ini dimulai dengan melakukan pendekatan pada konsep *k-partite graph*. Dimana *nodes* dari *graph* tersebut adalah *body part* yang terdeteksi dan *edges* dari *graph* tersebut adalah semua kemungkinan hubungan antar *nodes*. *K-partite graph* pada dasarnya akan mempartisi semua *nodes* yang terdeteksi menjadi set *k-graph* sehingga tidak ada dua node yang sama berdekatan. Kemudian dari semua *edges* yang ada diseleksi oleh konsep *part affinity fields* untuk memberikan bobot di setiap *edges* yang didapatkan, sehingga, *graph* yang ada menjadi lebih sederhana dengan *edges* yang memenuhi persyaratan dari PAF tersebut.

Tabel 3.2: Hasil Estimasi *Joint 2D Human Pose*

No	Id_join	Nama Join
1	0	<i>Nose</i>
2	1	<i>Neck</i>
3	2	<i>Right Shoulder</i>
4	3	<i>Right Elbow</i>
5	4	<i>Right Wrist</i>
6	5	<i>Left Shoulder</i>
7	6	<i>Left Elbow</i>
8	7	<i>Left Wrist</i>
9	8	<i>Right Hip</i>
10	9	<i>Right Knee</i>
11	10	<i>Left Hip</i>
12	11	<i>Left Knee</i>
13	12	<i>Left Ankle</i>
14	13	<i>Right Eye</i>
15	14	<i>Right Ear</i>
16	15	<i>Left Eye</i>
17	16	<i>Left Ear</i>
18	17	<i>Background</i>

Kemudian dari penyederhanaan *graph* oleh PAF. Selanjutnya adalah mengubah *graph* yang ada menjadi bentuk *bipartite graph*. Pembagian konsep *bipartite graph* pada dasarnya mengikuti domain dari masalah pemodelan tubuh manusia, dimana hubungan antar *part* dapat terhubung jika *part* sebelumnya sudah terhubung. Untuk lebih jelasnya dapat kita lihat proses dan ilustrasi dari *part ssociation* pada 2.11

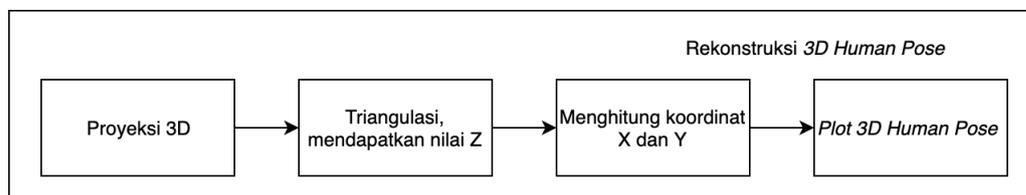
Setelah membaca penjelasan singkat dari proses bagaimana *openpose* mengestimasi 2D *human pose*. Kita kembali lagi pada sistem yang diterapkan pada penelitian kali ini, dimana yang kita lakukan adalah menggunakan *inference model* yang ada dari *openpose*. Dari hasil *inference model* tersebut, tiap-tiap citra kamera, baik citra kamera pertama maupun citra kamera kedua akan menghasilkan estimasi *joint 2D human pose* dengan jumlah 18 lokasi *joint* dari manusia dan satu lokasi *background*. Rincian ke-18 lokasi *joint* yang didapat pada proses estimasi 2D *human pose* dapat dilihat pada Tabel 3.2

3.4 Kalibrasi Stereo Kamera

Pada rekonstruksi 3D *human pose*, ada dua hal yang dibutuhkan pada sistem ini, pertama adalah fitur intrinsik dan ekstrinsik yang diperoleh dari proses kalibrasi stereo kamera, dan yang kedua adalah hasil estimasi 2D *joint human pose* dari masing-masing citra. Dengan begitu, proses kalibrasi sangat berperan penting untuk mendapatkan rekonstruksi 3D *human pose* yang baik.

Proses kalibrasi stereo kamera dibutuhkan untuk mengetahui bagaimana hubungan dari satu kamera dengan kamera lainnya. Hasil dari proses kamera kalibrasi akan mendapatkan beberapa fitur intrinsik dan fitur ekstrinsik dari masing-masing kamera. Hasil fitur yang didapatkan dari kalibrasi stereo kamera ini nantinya akan digunakan untuk membantu memproyeksikan *joint 2D human pose* ke dalam bentuk 3D *human pose* yang nanti akan dibahas di bawah.

3.5 Rekonstruksi 3D *Human Pose*



Gambar 3.5: Tahapan proses Rekonstruksi 3D *Human Pose*.

Dengan fitur intrinsik dan ekstrinsik yang didapatkan dan juga dengan hasil estimasi *joint 2D human pose* dari tiap-tiap citra inputan, maka proses selanjutnya adalah melakukan proyeksi dari bentuk 2D ke dalam bentuk 3D *human pose*. Tahapan proses rekonstruksi 3D *human pose* dapat diilustrasikan pada Gambar 3.5. Dimana, jika kita lihat pada Gambar 3.5 tersebut, rekonstruksi dari bentuk 2D ke bentuk 3D memerlukan proses proyeksi 3D kemudian dilanjutkan pada pengaplikasian pendekatan metode triangulasi, dimana dengan pendekatan ini akan didapatkan titik potong atau *depth Z* dari setiap titik *joint*. Ketika sudah mendapatkan sumbu koordinat *Z* dalam koordinat *world space*, maka sumbu koordinat *X* dan *Y* juga dapat

kita hitung. Sehingga, akhirnya estimasi rekonstruksi untuk 3D *human pose* bisa dihasilkan. Berikut dibawah ini penjelasan detail mengenai proses rekonstruksi 3D *human pose*.

3.5.1 Proyeksi 3D

Proses proyeksi kamera dilakukan untuk mendapatkan sumbu koordinat dunia X , Y dan Z dalam citra. Pada persamaan 3.1 dijelaskan bahwa matriks K_1 , merupakan matriks intrinsik dari kamera pertama. Matriks I adalah matriks Identitas 4x4. Mengapa dikali dengan matriks Identitas, karena kita mendefinisikan bahwa yang menjadi titik acuan adalah kamera pertama, untuk itu nilai matrik yang seharusnya berisi rotasi dan translasi kita isi dengan matriks identitas.

$$K_1 = \begin{bmatrix} fx & 0 & cx & 0 \\ 0 & fy & cy & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

Matriks $S_p = [u, v, 1, 1]^T$, dimana u, v merupakan koordinat x dan y *pixel* suatu objek dalam gambar. Matriks $P = [X, Y, Z, 1]^T$ dimana X, Y , dan Z merupakan sumbu koordinat dunia yang ingin kita cari, atau bisa dibilang adalah koordinat 3D. Untuk mendapatkan koordinat 3D, hal yang dilakukan adalah melakukan seperti pada persamaan 3.2, yaitu dengan menginverskan matriks I, K_1 , kemudian dikali dengan matriks S_p .

$$P = I^{-1}K_1^{-1}S_p \quad (3.2)$$

Berbeda dengan kamera pertama, karena kita mendefinisikan bahwa kamera pertama adalah acuan, maka untuk mendapatkan proyeksi 3D X, Y ,

Z untuk kamera kedua, untuk matriks ekstrinsik di definisikan dengan matriks rotasi dan translasi dari kamera kedua ke kamera pertama. Untuk lebih jelasnya, dapat dilihat pada Persamaan 3.3.

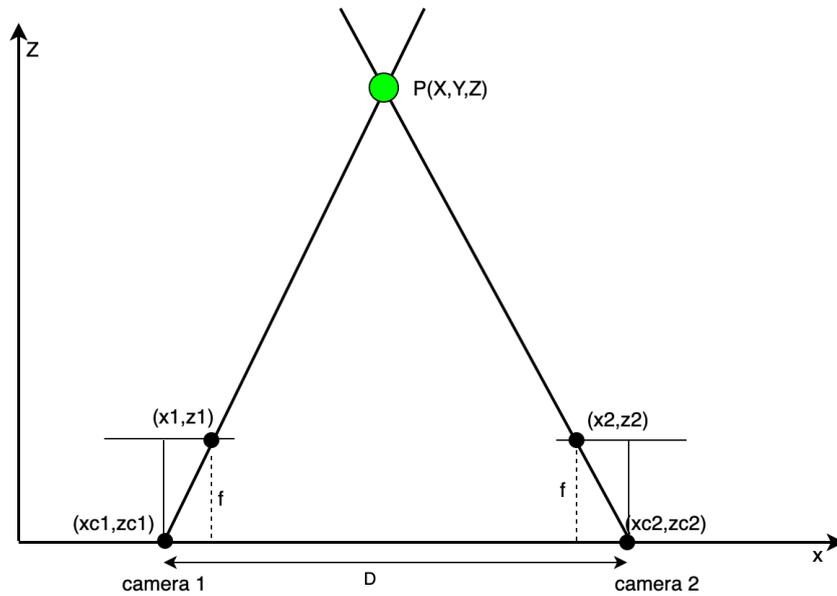
$$K_2 = \begin{bmatrix} fx & 0 & cx & 0 \\ 0 & fy & cy & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R|t = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

Selebihnya sama, $S_p = [u, v, 1, 1]^T$, dimana u, v merupakan koordinat *pixel* suatu objek pada gambar yang diambil oleh kamera kedua. Matriks $R|t$ merupakan matriks ekstrinsik kamera yang berisi rotasi dan translasi dari kamera kedua ke kamera pertama. Matriks $P = [X, Y, Z, 1]^T$ merupakan koordinat 3D yang ingin kita cari. Untuk lebih jelasnya dapat dilihat pada equation 3.4.

$$P = R|t^{-1}K_2^{-1}S_p \quad (3.4)$$

3.5.2 Triangulasi

Metode triangulasi (Hartley, 2004) dilakukan untuk mendapatkan *depth* dari sebuah objek. Triangulasi pada dasarnya adalah perpotongan dua buah garis. Dimana, garis tersebut jika dipanjangkan akan bertemu pada suatu titik tertentu. Berdasarkan konsep dari *multiple view geometry*, jarak objek ke kamera dapat diperoleh dari dua *view* yang berbeda, dengan konsep seperti ini, maka pada sistem ini kita menggunakan pendekatan triangulasi untuk menemukan titik potong *depth* datau koordinat sumbu Z dari suatu objek terhadap dua *view* berbeda. Ilustrasi metode triangulasi dapat dilihat pada Gambar 3.6.



Gambar 3.6: Triangulasi dari dua *view* yang berbeda.

Dimana (x_{c1}, z_{c1}) merupakan posisi koordinat kamera pertama dan (x_{c2}, z_{c2}) merupakan posisi koordinat kamera kedua. (x_1, z_1) merupakan koordinat titik objek x dan z yang didapatkan dari proses 3D proyeksi pada kamera pertama. (x_2, z_2) merupakan koordinat titik objek x dan z yang didapatkan dari proses 3D proyeksi pada kamera kedua.

Dengan pendekatan metode triangulasi, kita akan mempunyai empat titik. Dimana dua titik dari kamera pertama dan dua titik lainnya pada kamera kedua. Dua titik pada kamera pertama, yaitu terdiri dari (x_{c1}, z_{c1}) dan (x_1, z_1) , dimana dari kedua titik tersebut dapat kita gambarkan sebagai garis lurus yang melewati kedua titik tersebut. Begitulah garis pertama didapatkan, begitupun untuk garis kedua pada kamera kedua terbentuk dari dua titik, yaitu (x_{c2}, z_{c2}) dan (x_2, z_2) . Dengan diperolehnya dua garis, maka kita dapat menerapkan hitungan dibawah ini untuk mendapatkan titik potong $P(X, Y, Z)$. Dimana, titik P ini adalah titik 3D dari suatu joint. Sehingga, untuk mendapatkan rekonstruksi 3D *human pose* dimana satu manusia pada sistem ini terdiri dari 18 *joint*, maka proses triangulasi dilakukan juga sebanyak 18 kali, untuk mendapatkan lokasi 3D ke semua *joint* pada manusia.

Untuk lebih memahami lebih mendalam penjelasan diatas, untuk menerapkan konsep metode triangulasi, dimana suatu titik didapatkan dari perpotongan dua buah garis. Maka untuk mendapatkan titik potong tersebut Z atau *depth* dari setiap lokasi *joint* diperlukan dua buah garis. Untuk garis pertama, seperti konsep triangulasi yang sudah dijabarkan pada persamaan 2.1, sehingga dapat ditulis seperti Persamaan 3.5.

$$x(z_1 - zc_1) - z(x_1 - xc_1) = xc_1(z_1 - zc_1) - zc_1(x_1 - xc_1) \quad (3.5)$$

Sedangkan, untuk garis kedua, bentuk persamaannya sama seperti pada garis pertama, yang didapat dari dua titik pada kamera tersebut. Berikut Persamaan 3.6 adalah persamaan dari garis kedua

$$x(z_2 - zc_2) - z(x_2 - xc_2) = xc_2(z_2 - zc_2) - zc_2(x_2 - xc_2) \quad (3.6)$$

Setelah mendapatkan persamaan dari kedua garis tersebut, selanjutnya adalah mencari perpotongan dua buah garis. Dua buah garis dikatakan berpotongan, ketika garis pertama sama dengan garis kedua. Untuk lebih memudahkannya, persamaan dari buah garis kita tulis dalam bentuk matriks pada Persamaan 2.5 dan apabila kita lihat Persamaan 3.7, kita sudah bisa untuk mendapatkan koordinat x dan z , dimana sumbu koordinat z disini menjadi sumbu koordinat Z yang kita gunakan untuk *plot* rekonstruksi 3D *human pose*.

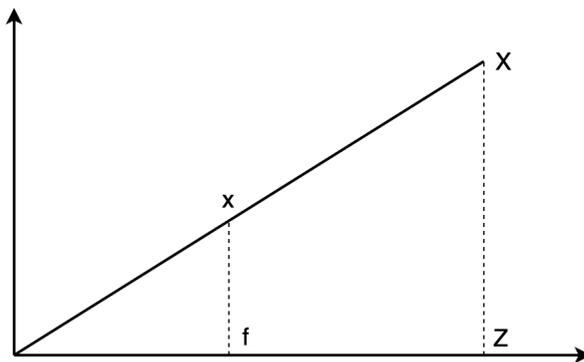
$$\begin{pmatrix} x \\ z \end{pmatrix} = \begin{pmatrix} z_1 - zc_1 & x_1 - xc_1 \\ z_2 - zc_2 & x_2 - xc_2 \end{pmatrix}^{-1} \begin{pmatrix} xc_1(z_1 - zc_1) - zc_1(x_1 - xc_1) \\ xc_2(z_2 - zc_2) - zc_2(x_2 - xc_2) \end{pmatrix} \quad (3.7)$$

3.5.3 Menghitung Sumbu Koordinat X dan Y

Setelah mendapatkan titik potong atau nilai *depth* Z dari kedua lokasi *joint* dari suatu objek yang di representasikan oleh garis. Maka, proses selanjutnya adalah menghitung sumbu koordinat X dan Y dari objek tersebut.

a. Koordinat X

Nilai X didapat dari perbandingan dua buah segitiga seperti pada Gambar 3.7. Dimana x sebanding dengan X dan f sebanding dengan Z , ketika nilai x sudah diketahui, f sudah diketahui, dan Z sudah diketahui. Maka nilai X bisa didapat seperti pada Persamaan 3.8 dan 3.9:



Gambar 3.7: Ilustrasi Perbandingan Koordinat X

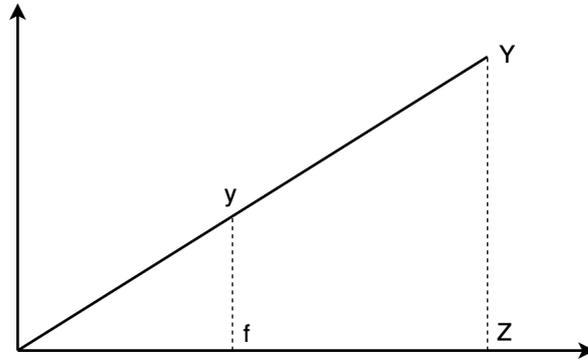
$$\frac{x}{f} = \frac{X}{Z} \quad (3.8)$$

$$X = \frac{x \times Z}{f} \quad (3.9)$$

b. Koordinat Y

Sama halnya dengan proses mendapatkan nilai X , sumbu koordinat Y juga didapat dari konsep yang sama, yaitu perbandingan dua buah segitiga

seperti pada Gambar 3.8. Sehingga, sumbu koordinat Y bisa didapat seperti pada Persamaan 3.10 dan 3.11



Gambar 3.8: Ilustrasi Perbandingan Koordinat Y

$$\frac{y}{f} = \frac{Y}{Z} \quad (3.10)$$

$$Y = \frac{y \times Z}{f} \quad (3.11)$$

3.5.4 Plot 3D Human Pose

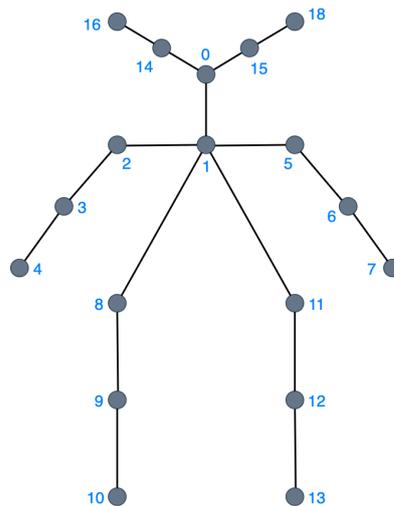
Dari proses sebelumnya, yaitu mendapatkan koordinat X , Y dan Z untuk setiap lokasi *joint* yang ada di manusia, maka proses selanjutnya adalah melakukan plot untuk mempermudah visualisasi dari hasil estimasi rekonstruksi 3D *human pose* yang kita dapatkan pada sistem ini.

Daftar *joint* yang didapat dapat dilihat pada Tabel 3.3. Jika kita lihat pada Tabel 3.3 terdapat 18 total *joint* yang didapatkan, dari kepala hingga kaki. Pada proses *plot* ini dilakukan secara manual, maksudnya adalah mendefinisikan bahwa contoh *joint neck* berasosiasi dengan *right shoulder* dan *left shoulder*, dan *joint right shoulder* berhubungan dengan *joint right elbow*, dan begitu seterusnya sehingga, kesemua *joint* berpasangan atau

Tabel 3.3: Daftar Plot Joint pada Sistem

Id Joint	Nama Joint
0	<i>Nose</i>
1	<i>Neck</i>
2	<i>Right Shoulder</i>
3	<i>Right Elbow</i>
4	<i>Right Wrist</i>
5	<i>Left Shoulder</i>
6	<i>Left Elbow</i>
7	<i>Left Wrist</i>
8	<i>Right Hip</i>
9	<i>Right Knee</i>
10	<i>Right Ankle</i>
11	<i>Left Hip</i>
12	<i>Left Knee</i>
13	<i>Left Ankle</i>
14	<i>Right Eye</i>
15	<i>Right Ear</i>
16	<i>Left Eye</i>
17	<i>Left Ear</i>

berasosiasi. Untuk lebih jelasnya, dapat kita lihat ilustrasi pasangan *joint* yang kita definisikan pada sistem ini pada Gambar 3.9.



Gambar 3.9: Visualisasi Plot 3D *human pose*.

Halaman ini sengaja dikosongkan

BAB 4

HASIL DAN PEMBAHASAN

Pada bab ini akan dipaparkan hasil dan analisis dari penelitian yang dilakukan beserta beberapa skenario yang di uji coba kan pada sistem ini. Pertama-tama akan dipaparkan hasil dan analisis dari penelitian ini.

4.1 Data Input

Pembahasan pertama adalah proses mempersiapkan data input. Seperti yang kita ketahui, bahwa data input untuk penelitian ini membutuhkan dua citra input. Kamera yang digunakan adalah sama menggunakan kamera logitech C920. Gambar 4.1 merupakan salah satu contoh bagaimana masing-masing kamera melihat objek dari masing-masing view kameranya, dengan desain kamera 60 cm. Gambar 4.1 bagian a merupakan contoh hasil pengambilan gambar objek dari kamera pertama, dan gambar 4.1 bagian b merupakan hasil pengambilan gambar objek dari kamera kedua.

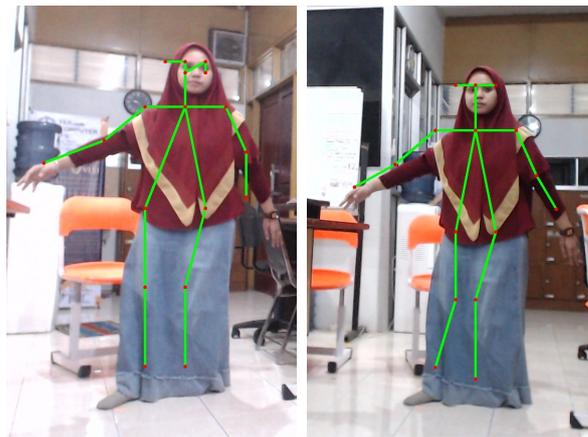


(a) Kamera Pertama (b) Kamera Kedua

Gambar 4.1: Contoh Citra Input.

4.2 Hasil Estimasi 2D *Human Pose* Menggunakan *Openpose*

Pada sub ini penulis akan menjelaskan alasan tentang hasil dari proses 2D *joint human pose estimation* dari tiap kamera. Jika kita lihat pada Gambar 4.2, pada Gambar 4.2 bagian a merupakan hasil proses *inference model* dari *openpose* yang berupa lokasi 2D *joint* yang diambil dari kamera pertama, begitu pun pada Gambar 4.2 bagian b merupakan hasil dari proses *inference model* dengan menggunakan *openpose model* yang berupa lokasi 2D *joint*. Jika kita lihat dari kedua Gambar, masing-masing berhasil menemukan lokasi *joint* dari 18 total *joint* yang ingin kita dapatkan.



(a) Kamera Pertama

(b) Kamera Kedua

Gambar 4.2: Hasil Estimasi 2D *Human Pose* Menggunakan *Openpose*

4.3 Hasil Kalibrasi Stereo Kamera

Setiap desain stereo kamera berubah, atau posisi antar kamera berubah, maka setiap kali itu juga kita membutuhkan untuk melakukan proses kalibrasi stereo kamera. Jika, posisi desain kamera berubah tapi kita masih menggunakan nilai intrinsik dan ekstrinsik pada desain kamera yang sebelumnya. Maka, dapat dipastikan hasil rekonstruksi 3D *human pose* nya akan tidak baik.

Dengan desain jarak kamera 60cm, dan setelah melakukan proses kalibrasi

stereo kamera, kita mendapatkan beberapa matriks. Pada sistem ini, hanya beberapa matriks yang dibutuhkan, yaitu matriks intrinsik dan ekstrinsik tiap kamera seperti pada Persamaan berikut 4.1, 4.2, dan 4.3. K_1 adalah matriks intrinsik dari kamera pertama. K_2 adalah matriks intrinsik dari kamera kedua. R adalah matriks rotasi dari kamera kedua ke kamera pertama atau kamera acuan. T adalah matriks translasi dari kamera kedua ke kamera pertama.

$$K_1 = \begin{bmatrix} 1.39 \times 10^3 & 0 & 9.51 \times 10^2 \\ 0 & 1.39 \times 10^3 & 4.97 \times 10^2 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

$$K_2 = \begin{bmatrix} 1.39 \times 10^3 & 0 & 9.49 \times 10^2 \\ 0 & 1.39 \times 10^3 & 5.19 \times 10^2 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

$$R = \begin{bmatrix} 7.82 \times 10^{-1} & 2.86 \times 10^{-2} & 6.18 \times 10^{-1} \\ -4.66 \times 10^{-2} & 9.98 \times 10^{-1} & 1.30 \times 10^{-2} \\ -6.17 \times 10^{-1} & -3.90 \times 10^{-2} & 7.85 \times 10^{-1} \end{bmatrix}, T = \begin{bmatrix} -6.61 \times 10^1 \\ -2.37 \times 10^{-1} \\ 1.54 \times 10^1 \end{bmatrix} \quad (4.3)$$

4.4 Hasil Rekonstruksi 3D *Human Pose*

Setelah masing-masing view kamera diproses pada *inference openpose model* dan mendapatkan lokasi *joint 2D*, kemudian juga proses kalibrasi stereo kamera sudah dilakukan, langkah selanjutnya adalah melakukan rekonstruksi 3D *human pose*. Seperti tahapan pada Gambar 3.5, di bawah ini akan dijabarkan hasil dari tiap proses yang didapat.

Tabel 4.1: Hasil Proyeksi 3D kamera pertama dan kedua

Id Joint	Citra 1 (pixel)	Citra 2 (pixel)	3D Proyeksi Citra 1			3D Proyeksi Citra 2		
			X	Y	Z	X	Y	Z
0	500, 281	1126, 328	-0.323	-0.155	1	60.969	2.561	29.605
1	500, 352	1126, 422	-0.323	-0.104	1	60.966	2.629	29.606
2	417, 352	1043, 422	-0.383	-0.104	1	60.919	2.627	29.569
3	333, 422	960, 493	-0.443	-0.054	1	60.87	2.676	29.533
4	208, 469	876, 540	-0.532	-0.020	1	60.821	2.708	29.496
5	584, 352	1210, 422	-0.263	-0.104	1	61.013	2.63	29.643
6	626, 446	1252, 516	-0.233	-0.037	1	61.033	2.698	29.662
7	626, 540	1293, 586	-0.233	0.030	1	61.054	2.749	29.681
8	417, 563	1085, 633	-0.383	0.046	1	60.936	2.779	29.589
9	417, 727	1085, 774	-0.383	0.164	1	60.931	2.88	29.591
10	417, 892	1043, 915	-0.383	0.282	1	60.902	2.98	29.573
11	542, 563	1168, 633	-0.293	0.046	1	60.982	2.78	29.626
12	500, 727	1126, 774	-0.323	0.164	1	60.954	2.88	29.609
13	500, 892	1126, 939	-0.323	0.282	1	60.948	2.999	29.61
14	500, 258	1085, 328	-0.323	-0.171	1	60.946	2.56	29.587
15	542, 258	1126, 328	-0.293	-0.171	1	60.969	2.561	29.605
16	459, 258	1085, 328	-0.352	-0.171	1	60.946	2.56	29.587
17	542, 281	1168, 328	-0.293	-0.155	1	60.992	2.562	29.623

4.4.1 Hasil Proyeksi 3D

Jika kita lihat pada Tabel 4.1 ini, terpampang ke 18 hasil 3D proyeksi dari gambar kamera pertama dan juga hasil 3D proyeksi dari gambar kamera kedua, dimana masing-masing kolom mewakili lokasi *pixel x, y* dari *joint* yang di dapat pada proses 2D *human pose estimation*, dan juga mewakili hasil dari 3D proyeksi yang berupa koordinat sumbu dunia X , Y , dan Z dari *pixel x, y* tersebut.

4.4.2 Hasil Perhitungan Triangulasi

Proses untuk mendapatkan rekonstruksi 3D *human pose* dimulai dari memproyeksikan 2D ke dalam 3D yang akan dilanjutkan pada tahapan menghitung sumbu koordinat Z menggunakan pendekatan konsep triangulasi. Seperti yang telah dijabarkan ada Persamaan 3.5, 3.6, dan ?? pada Tabel dibawah ini didapatkan dari hasil sumbu koordinat Z dari masing-masing *joint*.

Tabel 4.2: Sumbu Koordinat Dunia Hasil Rekonstruksi 3D

<i>Id Joint</i>	<i>Nama Joint</i>	Sumbu Koordinat Dunia (cm)		
		X	Y	Z
0	<i>Nose</i>	-94.083	-45.164	290.76
1	<i>Neck</i>	-93.14	-30.067	287.85
2	<i>Right shoulder</i>	-104.564	-28.514	272.98
3	<i>Right Elbow</i>	-112.482	-13.779	253.77
4	<i>Right Wrist</i>	-134.657	-5.211	252.73
5	<i>Left Shoulder</i>	-78.564	-31.158	298.29
6	<i>Left Elbow</i>	-69.584	-11.066	298.28
7	<i>Left Wrist</i>	-78.639	10.198	337.09
8	<i>Right Hip</i>	-117.521	14.339	306.81
9	<i>Right Knee</i>	-115.691	49.608	302.03
10	<i>Right Ankle</i>	-99.757	73.567	260.43
11	<i>Left Hip</i>	-84.262	13.418	287.11
12	<i>Left Knee</i>	-89.803	45.585	277.54
13	<i>Left Ankle</i>	-88.337	77.119	273
14	<i>Right Eye</i>	-83.119	-44.134	256.88
15	<i>Right Ear</i>	-76.779	-44.948	261.61
16	<i>Left Eye</i>	-100.148	-48.75	283.74
17	<i>Left Ear</i>	-87.098	-46.098	296.78

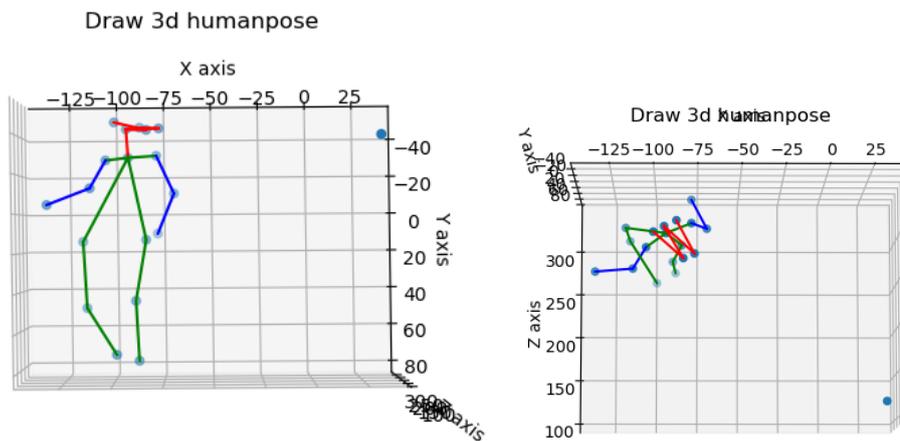
4.4.3 Hasil Sumbu Koordinat X dan Y

Setelah mendapatkan nilai dari sumbu koordinat dunia Z atau kedalaman dari perhitungan triangulasi kedua lokasi *joint* sebuah objek, maka sumbu koordinat X dan Y juga dapat kita hitung, sesuai dengan konsep dan perhitungan pada Persamaan 3.8, dan 3.9, sumbu X dan Y didapatkan. Pada Tabel 4.2 ini, jika kita lihat merupakan sederet daftar sumbu koordinat dunia X , koordinat Y dan koordinat Z dari masing-masing lokasi *joint* yang didapat. Nilai tersebut merupakan nilai *cm* dalam dunia nyata. Jika kita lihat pada baris pertama, lokasi dengan *Id joint* 0, berada di sumbu koordinat $X = -94.083cm$, $Y = -45.164cm$, dan $Z = 290.76cm$, dan begitupun sumbu koordinat dunia dari tiap *joint* selanjutnya.

]

4.4.4 Hasil Plot 3D Human Pose

Dari daftar deret sumbu koordinat dunia pada Tabel 4.2 yang didapat, maka selanjutnya adalah melakukan plot untuk mempermudah visualisasi dari hasil 3D *human pose estimation* yang ingin kita dapatkan pada sistem ini. Gambar 4.3 adalah salah satu contoh hasil plot rekonstruksi 3D *human pose* dari input dan hasil *openpose* dari Gambar 4.1. Bisa kita lihat pada Gambar tersebut terdiri dari dua sudut pandang yang berbeda, pada gambar sebelah kanan adalah sudut pandang 3D plot dari tampak depan, dan sebelah kiri adalah sudut pandang 3D plot dari bagain atas.



Gambar 4.3: Plot Hasil Rekonstruksi 3D human pose

4.5 Percobaan Skenario Jarak Objek ke Kamera

Pada pengujian ini akan dilakukan percobaan skenario jarak objek yang berbeda beda pada kamera, untuk memperkecil parameter evaluasi pada sub-bab ini, objek nya kita definisikan menggunakan orang yang sama. Terdapat empat perbedaan jarak, pertama adalah menempatkan objek dengan jarak 240cm dari kamera. Kedua adalah menempatkan objek dengan jarak 275cm dari kamera. Ketiga adalah menempatkan objek dengan jarak 310cm dari kamera. Dan terakhir, menempatkan objek dengan jarak 34cm dari kamera. Masing-masing hasil dan pembahasan akan dijelaskan di bawah ini.

4.5.1 Jarak 240 cm dari Objek ke Kamera

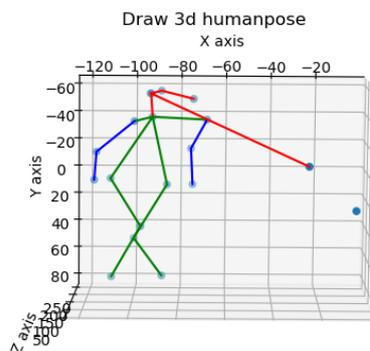
Pada pengambilan gambar, jarak objek yang kita ukur dari kamera adalah 240 cm. Pada Gambar 4.4 bagian a merupakan citra input, sedangkan Gambar 4.4 bagian b adalah hasil openpose dari citra kamera pertama dan 4.4 bagian c adalah hasil openpose dari citra kamera kedua.



(a) Citra Input (b) Kamera Pertama (c) Kamera Kedua

Gambar 4.4: Input dan Hasil Openpose Jarak Objek ke Kamera 240 cm

Pada Gambar 4.5 dapat dilihat hasil ke 18 3D *human pose estimation* berhasil didapatkan pada sistem ini, meskipun estimasi pose 3D masih belum sempurna di bagian salah satu *joint* mata. Performa keseluruhan sistem dapat kita lihat pad Tabel 4.3, dimana rata-rata error jarak dari seluruh *joint* yang terdeteksi adalah 20.10493141 cm.



Gambar 4.5: Hasil 3D Rekonstruksi Jarak Objek ke Kamera 240 cm

Tabel 4.3: Hasil Triangulasi Koordinat Z dengan Jarak Objek ke Kamera 240 cm

Id_Joint	Nilai Koordinat Z	Error dari Jarak Sebenarnya (240 cm)
0	243.8013487	3.801348683
1	241.834709	1.834709016
2	226.5788366	13.42116343
3	233.205755	6.794245043
4	221.3304966	18.66950337
5	233.2467561	6.753243894
6	257.5587715	17.55877149
7	254.7723119	14.77231194
8	251.1633536	11.16335365
9	219.807067	20.19293301
10	233.7859109	6.21408906
11	243.8049215	3.804921506
12	263.8761739	23.8761739
13	229.9703657	10.02963426
14	244.2887479	4.288747931
15	251.5224882	11.52248817
16	62.77924701	177.220753
17	230.029626	9.970374037
Rata-rata Error (cm)		20.10493141

4.5.2 Jarak 275 cm dari Objek ke Kamera

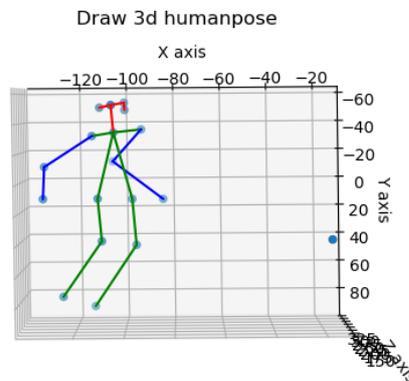
Jarak objek yang kita ukur dari kamera adalah 275 cm. Pada Gambar 4.6 bagian a merupakan citra input, Gambar 4.6 bagian b adalah hasil openpose dari citra kamera pertama dan Gambar 4.6 bagian c adalah hasil openpose dari citra kamera kedua.

Pada Gambar 4.7 dapat dilihat ke 18 3D *human pose estimation* yang berhasil didapatkan pada sistem ini dan estimasi pose 3D masih sudah cukup sempurna hampir sama dengan hasil dari openpose yang ada pada Gambar 4.6. Performa keseluruhan sistem dapat kita lihat pad Tabel 4.4, dimana rata-rata error jarak dari seluruh *joint* yang terdeteksi adalah 13.65912052 cm.



(a) Citra Input (b) Kamera Pertama (c) Kamera Kedua

Gambar 4.6: Input dan Hasil Openpose Jarak Objek ke Kamera 275 cm



Gambar 4.7: Hasil 3D Rekonstruksi Jarak Objek ke Kamera 275 cm.

4.5.3 Jarak 310 cm dari Objek ke Kamera



(a) Citra Input (b) Kamera Pertama (c) Kamera Kedua

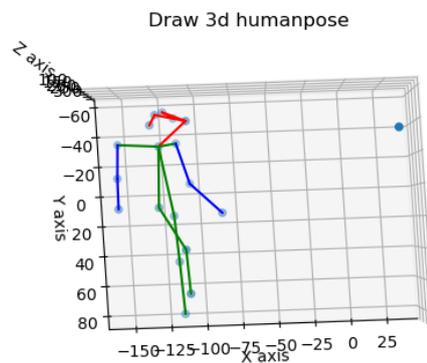
Gambar 4.8: Input dan Hasil Openpose Jarak Objek ke Kamera 310 cm

Jarak objek yang kita ukur dari kamera adalah 310 cm. Pada Gambar

Tabel 4.4: Hasil Triangulasi Koordinat Z dengan Jarak Objek ke Kamera 275 cm

Id_Joint	Nilai Koordinat Z	Error dari Jarak Sebenarnya (275 cm)
0	276.1849214	1.184921377
1	273.6049777	1.395022329
2	255.9292151	19.07078491
3	265.7256714	9.274328578
4	251.2137316	23.78626842
5	288.5584694	13.55846937
6	327.6888032	52.68880322
7	287.8213953	12.82139532
8	251.1633536	23.83664635
9	247.6104551	27.3895449
10	267.5187541	7.481245919
11	275.6906659	0.6906659145
12	271.2554986	3.744501419
13	296.6602243	21.66022428
14	276.1849214	1.184921377
15	284.4322194	9.432219354
16	267.085182	7.914818041
17	283.7493883	8.749388304
Rata-rata Error (cm)		13.65912052

4.8 bagian a merupakan citra input, sedangkan Gambar 4.8 bagian b adalah hasil openpose dari citra kamera pertama dan 4.8 bagian c adalah hasil openpose dari citra kamera kedua, dan Gambar 4.9 merupakan hasil output dari rekonstruksi 3D *human pose* yang didapatkan.



Gambar 4.9: Hasil 3D Rekonstruksi Jarak Objek ke Kamera 310 cm

Pada Gambar 4.9 dapat dilihat ke 18 3D *human pose estimation* yang berhasil didapatkan pada sistem ini dan estimasi pose 3D masih sudah cukup

Tabel 4.5: Hasil Triangulasi Koordinat Z dengan Jarak Objek ke Kamera 310 cm

Id_Joint	Nilai Koordinat Z	Error dari Jarak Sebenarnya (310 cm)
0	274.8875049	35.11249514
1	303.7855797	6.214420274
2	326.4187811	16.41878115
3	308.8158122	1.184187803
4	291.1621099	18.83789005
5	322.5394347	12.5394347
6	327.6888032	17.68880322
7	287.8213953	22.17860468
8	287.3928308	22.60716922
9	248.1010523	61.89894775
10	245.1316608	64.86833918
11	307.6077764	2.392223558
12	302.7985543	7.201445671
13	298.1665349	11.83346515
14	307.0043598	2.995640223
15	317.8001209	7.800120912
16	294.542742	15.45725805
17	325.2905375	15.2905375
Rata-rata Error (cm)		19.02887579



(a) Citra Input (b) Kamera Pertama (c) Kamera Kedua

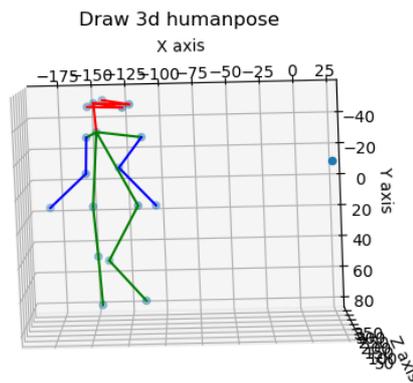
Gambar 4.10: Input dan Hasil Openpose Jarak Objek ke Kamera 340 cm.

sempurna hampir sama dengan hasil dari openpose yang ada pada Gambar 4.8. Performa keseluruhan sistem dapat kita lihat pad Tabel 4.5, dimana rata-rata error jarak dari seluruh *joint* yang terdeteksi adalah 19.02887579 cm.

4.5.4 Jarak 340 cm dari Objek ke Kamera

Jarak objek yang kita ukur dari kamera adalah 340 cm. Pada Gambar 4.10 bagian a merupakan citra input, sedangkan Gambar 4.10 bagian b adalah hasil openpose dari citra kamera pertama dan 4.10 bagian c adalah hasil openpose dari citra kamera kedua.

Pada Gambar 4.11 dapat dilihat ke 18 3D *human pose estimation* yang berhasil didapatkan pada sistem ini dan estimasi pose 3D masih sudah cukup sempurna hampir sama dengan hasil dari openpose yang ada pada Gambar 4.10. Performa keseluruhan sistem dapat kita lihat pad Tabel 4.7, dimana rata-rata error jarak dari seluruh *joint* yang terdeteksi adalah 21.93333765 cm.



Gambar 4.11: Hasil 3D Rekonstruksi Jarak Objek ke Kamera 340 cm.

Dari Keempat perbedaan jarak dari objek ke kamera dan berdasarkan Tabel 4.16, rata-rata error yang paling kecil adalah pada skenario jarak sekian 13 cm dari objek ke kamera, dan rata-rata error paling besar adalah 21 cm.

Tabel 4.6: Perbandingan Rata-Rata Error Jarak Objek ke Kamera

	Jarak Objek ke Kamera			
	240 cm	275 cm	310 cm	340 cm
Rata-Rata Error (cm)	20.10493141	13.65912052	19.02887579	21.93333765

Tabel 4.7: Hasil Triangulasi Koordinat Z dengan Jarak Objek ke Kamera 340 cm

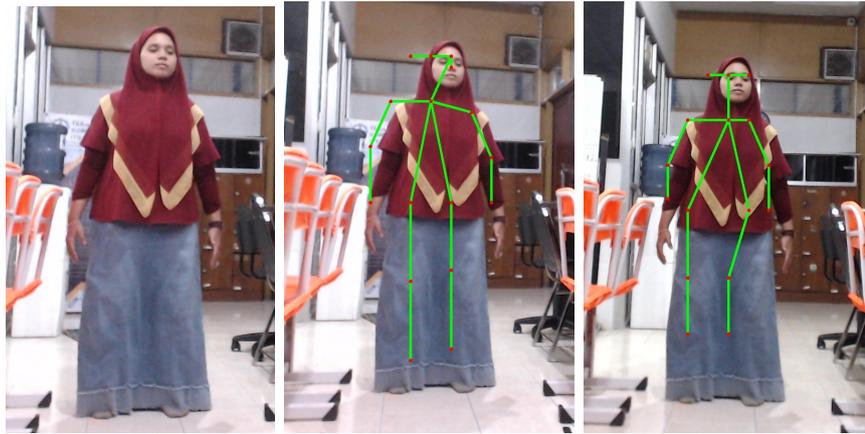
Id_Joint	Nilai Koordinat Z	Error dari Jarak Sebenarnya (340 cm)
0	358.8466754	18.84667545
1	355.4494685	15.44946852
2	325.4864059	14.51359412
3	307.9961172	32.00388276
4	343.3120218	3.312021814
5	321.6088553	18.39114465
6	372.8356036	32.8356036
7	323.947195	16.05280499
8	319.2814018	20.71859824
9	314.1680848	25.83191521
10	310.0717863	29.92821372
11	307.6077764	32.39222356
12	341.6722827	1.672282716
13	298.9428728	41.0571272
14	306.1747497	33.8252503
15	316.9012258	23.09877416
16	344.4814162	4.48141616
17	370.3890805	30.38908051
Rata-rata Error (cm)		21.93333765

4.6 Percobaan Skenario Jarak antar Kedua Kamera

Pada sub-bab ini akan menguji bagaimana hasil sistem jika dilakukan dengan berbagai jarak antar kamera yang berbeda, dan melihat apa pengaruhnya terhadap hasil rekonstruksi 3D *human pose*. Terdapat empat desain jarak kamera yang akan dicoba pada sistem ini. Pertama adalah jarak antar kamera dengan jarak 40 cm. Kedua adalah jarak antar kamera dengan jarak 60 cm. Ketiga adalah mendesain jarak antar kamera dengan jarak 80 cm. Keempat adalah mendesain jarak antar kamera dengan jarak 100 cm.

4.6.1 Jarak antar Kamera 20 cm

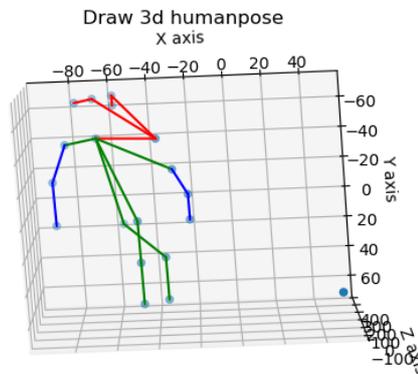
Pertama adalah dengan desain jarak antar kamera sebesar 20 cm. Pada Gambar 4.12 bagian a merupakan citra input pada sistem, Gambar 4.12 bagian b dan Gambar 4.12 bagian c merupakan hasil openpose 2D lokasi *joint* dari masing-masing citra kamera pertama dan kedua.



(a) Citra Input (b) Kamera Pertama (c) Kamera Kedua

Gambar 4.12: Input dan Hasil Openpose Jarak antar Kamera 20 cm.

Jika kita perhatikan, pada hasil rekonstruksi 3D yang dihasilkan pada Gambar 4.13 memberikan hasil yang kurang baik, hal ini dapat dilihat pada Tabel 4.8, dimana nilai Z dari hasil triangulasi yang didapatkan memiliki rata-rata error sebesar 58.70 cm, sehingga, sumbu koordinat untuk rekonstruksi 3D yang didapatkan juga kurang baik.



Gambar 4.13: Hasil 3D Rekonstruksi Jarak antar Kamera 20 cm.

4.6.2 Jarak antar kamera 43 cm

Kedua adalah dengan desain jarak antar kamera sebesar 43 cm. Pada Gambar 4.14 bagian a merupakan citra input pada sistem, Gambar 4.14 bagian b dan Gambar 4.14 bagian c merupakan hasil openpose 2D lokasi *joint* dari masing-masing citra kamera pertama dan kedua.

Tabel 4.8: Hasil Triangulasi Koordinat Z Desain Jarak 20 cm

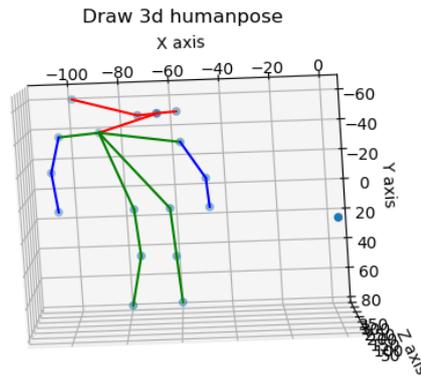
Id_Joint	Nilai Koordinat Z	Error dari Jarak Sebenarnya (280 cm)
0	239.6782966	40.32170341
1	373.1025267	93.10252668
2	346.1481179	66.1481179
3	335.5107624	55.51076241
4	334.287966	54.28796605
5	241.316941	38.68305904
6	241.6395416	38.36045837
7	240.3300263	39.66997371
8	228.4419194	51.55808055
9	227.1870706	52.81292939
10	226.1593206	53.84067941
11	375.261399	95.26139902
12	234.7593968	45.24060322
13	233.4426	46.55740002
14	375.8546175	95.85461753
15	384.433084	104.433084
16	365.0669071	85.06690709
17	384.433084	104.433084
Rata-Rata Error (cm)		58.70612599



(a) Citra Input (b) Kamera Pertama (c) Kamera Kedua

Gambar 4.14: Input dan Hasil Openpose Jarak antar Kamera 43 cm.

Rekonstruksi 3D pada Gambar 4.15 memberikan hasil yang kurang baik, hal ini dapat dilihat pada Tabel 4.9, dimana nilai Z dari hasil triangulasi yang didapatkan memiliki rata-rata error sebesar 19.86 cm, sehingga, sumbu koordinat untuk rekonstruksi 3D yang didapatkan juga kurang baik.



Gambar 4.15: Hasil 3D Rekonstruksi Jarak Kamera 43 cm.

Tabel 4.9: Hasil Triangulasi Koordinat Z Desain Jarak 43 cm

Id_Joint	Nilai Koordinat Z	Error dari Jarak Sebenarnya (280 cm)
0	282.8193872	2.819387152
1	340.8614548	60.86145482
2	325.5035242	45.50352415
3	308.0543485	28.05434852
4	302.624711	22.62471102
5	283.3223777	3.322377655
6	280.8511724	0.851172408
7	276.2951221	3.704877861
8	262.2560745	17.74392554
9	256.7689837	23.2310163
10	245.8202548	34.17974523
11	271.1129668	8.887033238
12	265.2119602	14.78803978
13	259.6202561	20.37974391
14	278.9795125	1.020487507
15	283.7474269	3.747426875
16	338.6500174	58.65001743
17	287.1729563	7.172956316
Rata-rata Error (cm)		19.86345809

4.6.3 Jarak antar Kamera 66 cm

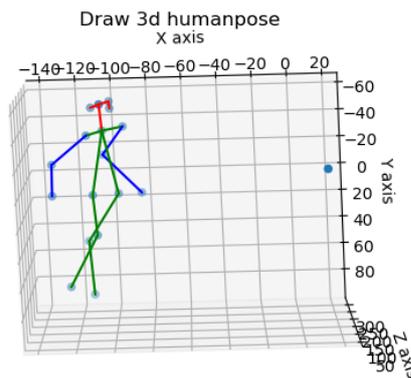
Ketiga adalah dengan desain jarak antar kamera sebesar 66 cm. Pada Gambar 4.16 bagian a merupakan citra input pada sistem, Gambar 4.16 bagian b dan Gambar 4.16 bagian c merupakan hasil openpose 2D lokasi *joint* dari masing-masing citra kamera pertama dan kedua.



(a) Citra Input (b) Kamera Pertama (c) Kamera Kedua

Gambar 4.16: Input dan Hasil Openpose Jarak antar Kamera 66 cm.

Jika kita perhatikan, pada hasil rekonstruksi 3D yang dihasilkan memberikan hasil yang lebih baik dari kedua desain jarak kamera di awal. Hal ini dapat dilihat pada Tabel 4.10, dimana nilai Z dari hasil triangulasi yang didapatkan memiliki rata-rata error yang lebih kecil dibandingkan dengan rata-rata error dari desain jarak kamera lainnya, yaitu 15.76 cm, sehingga, sumbu koordinat untuk rekonstruksi 3D yang didapatkan juga lebih baik dari yang lainnya. Untuk itu, pada percobaan skenario berikutnya menggunakan desain jarak kamera sebesar 66 cm untuk menguji hal lainnya.



Gambar 4.17: Hasil 3D Rekonstruksi Jarak antar Kamera 66 cm.

4.6.4 Jarak antar Kamera 98 cm

Keempat adalah dengan desain jarak antar kamera sebesar 98 cm. Pada Gambar 4.19 bagian a merupakan citra input pada sistem, Gambar 4.19 bagian

Tabel 4.10: Hasil Triangulasi Koordinat Z Desain Jarak 66 cm

Id_Joint	Nilai Koordinat Z	Error dari Jarak Sebenarnya (280 cm)
0	276.1849214	3.815078623
1	273.6049777	6.395022329
2	255.9292151	24.07078491
3	265.7256714	14.27432858
4	251.2137316	28.78626842
5	288.5584694	8.558469368
6	327.6888032	47.68880322
7	287.8213953	7.821395321
8	251.1633536	28.83664635
9	247.6104551	32.3895449
10	267.5187541	12.48124592
11	275.6906659	4.309334085
12	302.0311448	22.03114476
13	297.4270995	17.42709953
14	276.1849214	3.815078623
15	284.4322194	4.432219354
16	267.085182	12.91481804
17	283.7493883	3.749388304
Rata-Rata Error (cm)		15.7664817

b dan Gambar 4.19 bagian c merupakan hasil openpose 2D lokasi *joint* dari masing-masing citra kamera pertama dan kedua.



(a) Citra Input

(b) Kamera Pertama

(c) Kamera Kedua

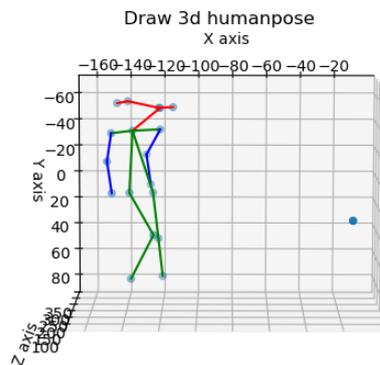
Gambar 4.18: Input dan Hasil Openpose Jarak antar Kamera 98 cm.

Jika kita perhatikan, pada hasil rekonstruksi 3D yang dihasilkan memberikan hasil yang kurang baik, hal ini dapat dilihat pada Tabel 4.11,

Tabel 4.11: Hasil Triangulasi Koordinat Z Desain Jarak 98 cm

Id_joint	Nilai Koordinat Z	Error dari jarak sebenarnya (280cm)
0	289.1461555	9.146155504
1	305.4574446	25.45744464
2	293.5559427	13.55594267
3	281.2225142	1.222514221
4	276.0548917	3.945108272
5	311.6122152	31.61221521
6	332.9153261	52.91532615
7	326.3449255	46.34492551
8	290.1468171	10.14681707
9	260.3690033	19.63099673
10	270.7369884	9.26301157
11	298.9376113	18.93761135
12	291.4161796	11.41617963
13	285.287728	5.287728036
14	311.580139	31.58013896
15	290.2320689	10.23206888
16	305.0459923	25.04599231
17	292.1875113	12.18751127
Rata- Rata Error (cm)		18.77376044

dimana nilai Z dari hasil triangulasi yang didapatkan memiliki rata-rata error sebesar 18.77 cm, sehingga, sumbu koordinat untuk rekonstruksi 3D yang didapatkan juga kurang baik.



Gambar 4.19: Hasil 3D Rekonstruksi Jarak antar Kamera 98 cm.

Dari keempat perbedaan jarak antara kedua kamera, Jika kita perhatikan pada Tabel 4.12, pada hasil rekonstruksi 3D yang dihasilkan dari

Tabel 4.12: Perbandingan Rata-Rata Error Keempat Jarak antar Kamera

	Desain Jarak Antar Kamera			
	20 cm	43 cm	66 cm	98 cm
Rata-Rata Error (cm)	58.70612599	19.86345809	15.7664817	18.77376044

beberapa desain jarak antar kamera, hasil 3D rekonstruksi yang paling baik adalah pada desain kamera dengan jarak antar keduanya sebesar 66cm, dan hasil 3D rekonstruksi yang kurang baik adalah pada desain jarak 20cm . Hal ini dikarenakan jarak yang terlalu dekat menyebabkan nilai dari hasil triangulasi yang didapatkan kurang tepat, sehingga, sumbu koordinat untuk rekonstruksi 3D yang didapatkan juga tidak maksimal.

4.7 Percobaan Skenario Gender dan Baju



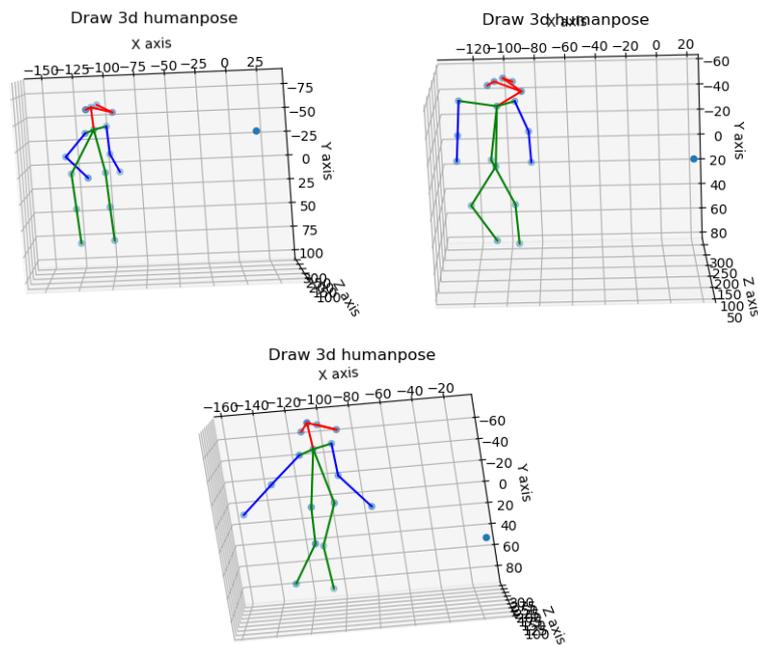
Gambar 4.20: Citra Input Berbeda Gender dan Baju

Selain percobaan skenario diatas, sistemakan dicoba untuk dilakukan pada perbedaan gender dan juga perbedaan pada baju yang dipakai. Percobaan skenario ini akan menguji seberapa baik 3D rekonstruksi yang dihasilkan pada kondisi berbeda gender dan baju yang dipakai. Dalam uji coba ini akan digunakan beberapa aktor laki-laki dan perempuan, dengan menggunakan pakaian yang beragam.

Pada Gambar 4.20 bagian a merupakan input dari aktor pertama, Gambar 4.20 bagian b merupakan input dari aktor kedua, dan Gambar 4.20 bagian c merupakan input dari aktor ketiga.



Gambar 4.21: Hasil Openpose Berbeda Gender dan Baju pada kamera pertama



Gambar 4.22: Hasil 3D Rekonstruksi Berbeda Gender dan Baju.

Pada Gambar 4.21 bagian a merupakan hasil openpose kamera pertama dari aktor pertama, Gambar 4.21 bagian b adalah hasil openpose kamera pertama dari aktor kedua, dan Gambar 4.21 bagian c adalah hasil openpose kamera pertama dari aktor ketiga.

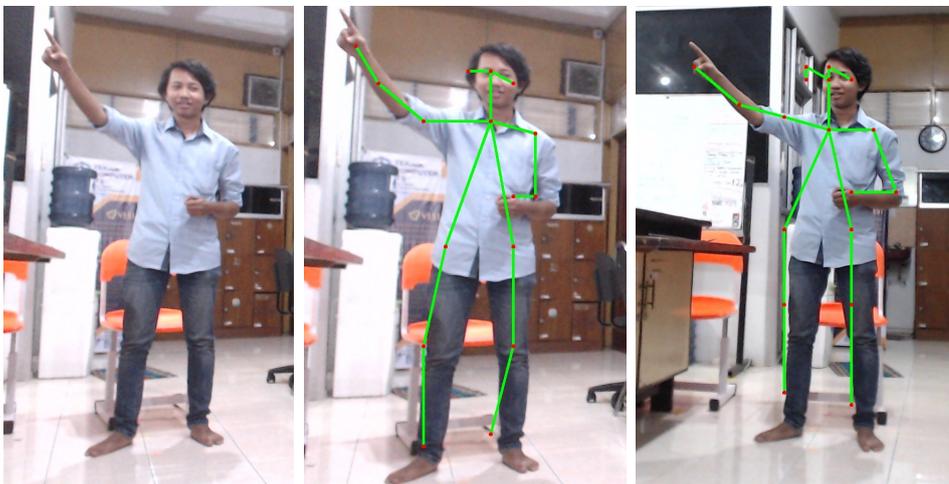
Gambar 4.22 merupakan hasil output dari 3D rekonstruksi *human pose* pada masing masing aktor. Jika kita lihat perhatikan, dari Gambar kedua, sistem dapat dilakukan di kondisi berbagai perbedaan gender, baik laki-laki maupun perempuan dan juga sistem dapat dilakukan pada kondisi baju yang berbeda.

4.8 Percobaan Skenario *Pose*

Pada sub-bab ini akan dijelaskan bagaimana jika sistem ini dilakukan dalam berbagai pose. Terdapat enam skenario perbedaan pose yang akan dilakukan oleh tiga orang. Dimana, masing-masing orang akan berpose pada masing-masing perbedaan pose yang didefinisikan diawal. Perbedaan pose yang dilakukan pada bab ini mengikuti evaluasi pada datasat *Human 3.6* (Ionescu et al., 2014). Berikut beberapa pose yang diadaptasikan pada dataset *Human 3.6*.

4.8.1 Pose Sedang Menunjukkan Arah

Pose yang pertama adalah pose sedang menunjukkan arah. Perbedaan pose akan dilakukan oleh aktor, dengan kebebasan aktor untuk berpose sesuai dengan tema pose menunjuk arah yang ditentukan. Gambar 4.23 bagian a adalah citra input kamera dari aktor. Gambar 4.23 bagian b dan c adalah hasil openpose lokasi *joint* dari kamera pertama dan kedua.



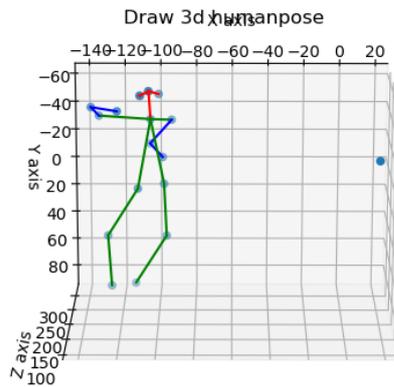
(a) Citra Input

(b) Kamera Pertama

(c) Kamera Kedua

Gambar 4.23: Citra Input dan Hasil Openpose Pose Sedang Menunjukkan Arah.

Gambar 4.24 adalah hasil rekonstruksi 3D *human pose*. Jika kita perhatikan pada hasil rekonstruksi 3D *human pose* yang didapatkan, tiga aktor dapat melakukan pose menunjukkan arah.



Gambar 4.24: Hasil 3D Rekonstruksi Pose Menunjuk Arah.

4.8.2 Pose Sedang Diskusi

Pose yang kedua adalah pose sedang diskusi. Perbedaan pose akan dilakukan oleh aktor, dengan kebebasan aktor untuk berpose sesuai dengan tema pose sedang diskusi yang ditentukan. Gambar 4.25 bagian a adalah citra input kamera dari aktor. Gambar 4.25 bagian b dan c adalah hasil openpose lokasi *joint* dari kamera pertama dan kedua.



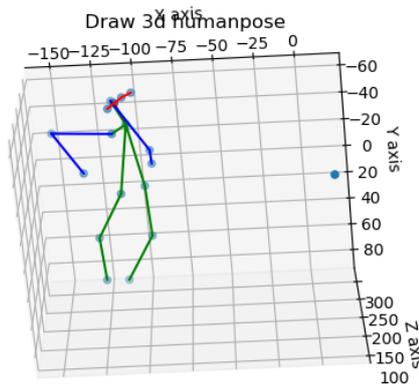
(a) Citra Input

(b) Kamera Pertama

(c) Kamera Kedua

Gambar 4.25: Citra Input dan Hasil Openpose Pose Sedang Diskusi.

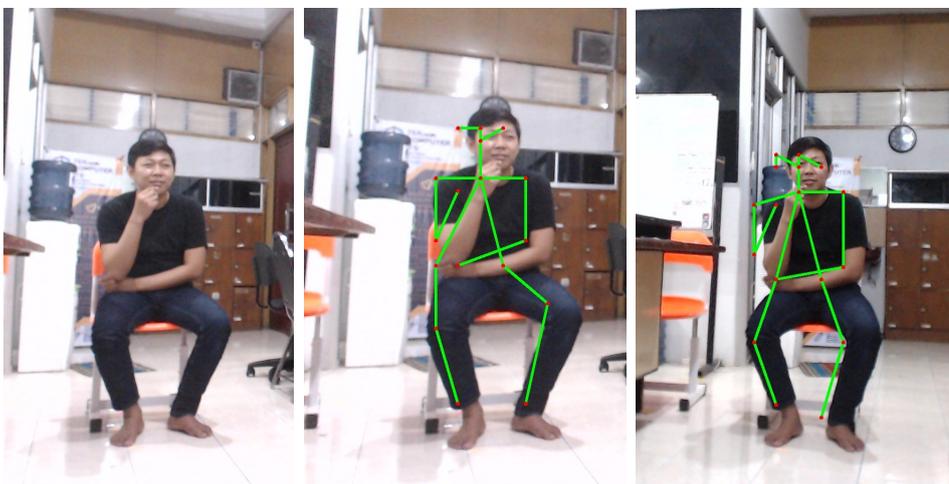
Gambar 4.25 adalah citra input kamera dari aktor. Gambar adalah hasil rekonstruksi 3D *human pose*. Pada hasil rekonstruksi 3D *human pose* yang didapatkan, aktor dapat melakukan pose sedang diskusi.



Gambar 4.26: Hasil 3D Rekonstruksi Pose Sedang Diskusi.

4.8.3 Pose Sedang Makan

Pose yang ketiga adalah pose sedang makan. Pose akan dilakukan aktor untuk berpose sesuai dengan tema pose sedang makan yang sudah ditentukan. Gambar 4.27 bagian a adalah citra input kamera dari aktor. Gambar 4.27 bagian b dan c adalah hasil openpose lokasi *joint* dari kamera pertama dan kedua.



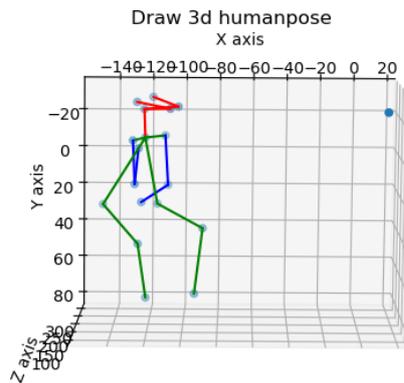
(a) Citra Input

(b) Kamera Pertama

(c) Kamera Kedua

Gambar 4.27: Citra Input dan Hasil Openpose Pose Sedang Makan.

Gambar adalah hasil rekonstruksi 3D *human pose*. Jika kita perhatikan pada hasil rekonstruksi 3D *human pose* yang didapatkan, aktor dapat melakukan pose sedang makan.



Gambar 4.28: Hasil 3D Rekonstruksi Pose Sedang Makan.

4.8.4 Pose Sedang Membayar

Pose yang keempat adalah pose sedang membayar. Perbedaan pose akan dilakukan oleh aktor, dengan kebebasan aktor untuk berpose sesuai dengan tema pose sedang membayar yang ditentukan. Gambar 4.29 bagian a adalah citra input kamera dari aktor. Gambar 4.29 bagian b dan c adalah hasil *openpose* lokasi *joint* dari kamera pertama dan kedua.



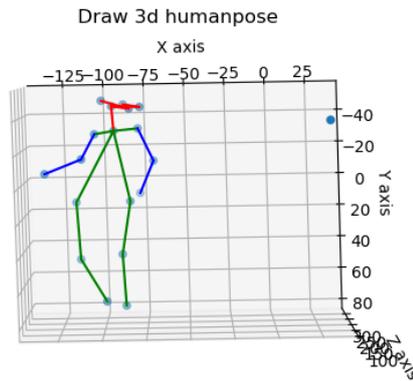
(a) Citra Input

(b) Kamera Pertama

(c) Kamera Kedua

Gambar 4.29: Citra Input dan Hasil Openpose Pose Sedang Bayar.

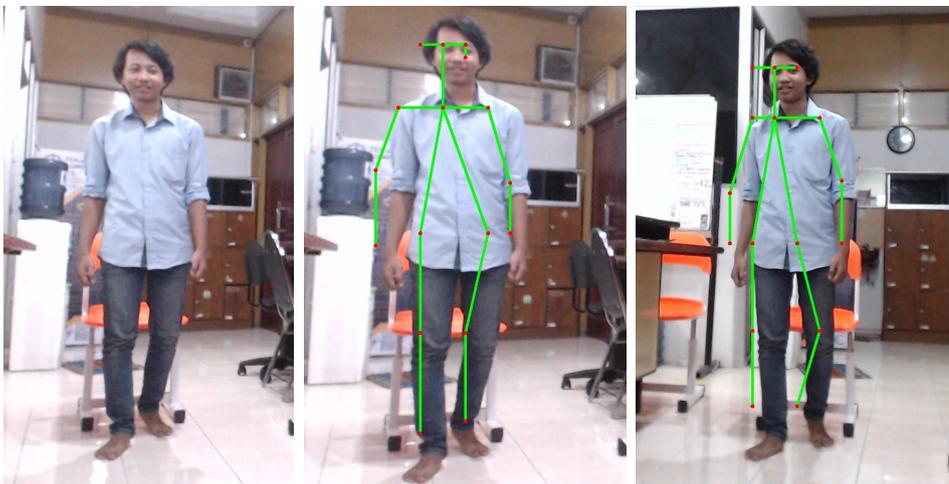
Gambar adalah hasil rekonstruksi 3D *human pose*. Jika kita perhatikan pada hasil rekonstruksi 3D *human pose* yang didapatkan, aktor dapat melakukan pose sedang membayar.



Gambar 4.30: Hasil 3D Rekonstruksi Pose Sedang Membayar.

4.8.5 Pose Sedang Berjalan

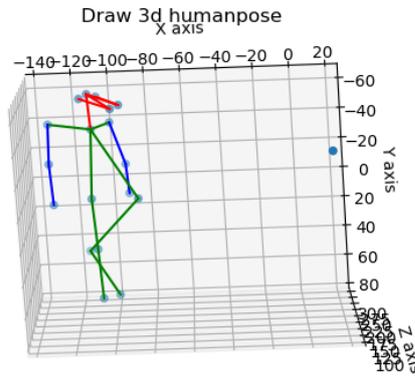
Pose yang kelima adalah pose sedang berjalan. Perbedaan pose akan dilakukan oleh aktor yang berbeda, dengan kebebasan aktor untuk berpose sesuai dengan tema pose sedang berjalan yang ditentukan. Gambar 4.31 bagian a adalah citra input kamera dari aktor. Gambar 4.31 bagian b dan c adalah hasil openpose lokasi *joint* dari kamera pertama dan kedua.



(a) Citra Input (b) Kamera Pertama (c) Kamera Kedua

Gambar 4.31: Citra Input dan Hasil Openpose Pose Sedang Berjalan.

Gambar adalah hasil rekonstruksi 3D *human pose*. Jika kita perhatikan pada hasil rekonstruksi 3D *human pose* yang didapatkan, aktor dapat melakukan pose sedang berjalan.



Gambar 4.32: Hasil 3D Rekonstruksi Pose Berjalan.

4.8.6 Pose Sedang Menelpon

Pose yang keenam adalah pose sedang menelpon. Perbedaan pose akan dilakukan aktor, dengan kebebasan aktor untuk berpose sesuai dengan tema pose sedang menelpon yang ditentukan. Gambar 4.33 bagian a adalah citra input kamera dari aktor. Gambar 4.33 bagian b dan c adalah hasil openpose lokasi *joint* dari kamera pertama dan kedua.



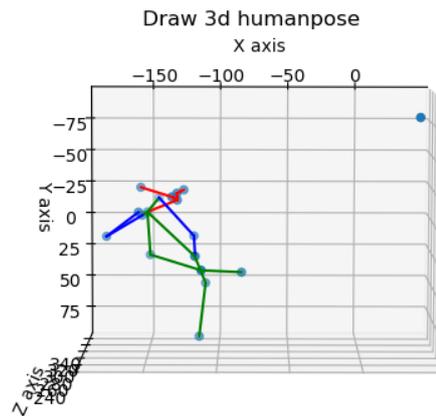
(a) Citra Input

(b) Kamera Pertama

(c) Kamera Kedua

Gambar 4.33: Citra Input dan Hasil Openpose Pose Sedang Menelpon.

Gambar 4.34 adalah hasil rekonstruksi 3D *human pose*. Jika kita perhatikan pada hasil rekonstruksi 3D *human pose* yang didapatkan, aktor dapat melakukan pose sedang menelpon.



Gambar 4.34: Hasil 3D Rekonstruksi Pose Sedang Menelpon.

4.9 Percobaan Skenario *Motion*

Selain percobaan skenario pose, pada percobaan skenario kali ini kami akan menguji bagaimana sistem dalam mendapatkan alur gerakan atau *motion* yang berurutan waktu diambilnya. Kami mendesain aktor akan bergerak bebas dari gerakan satu ke gerakan lainnya, kamera akan menyimpan atau mengambil gambar setiap 100 hitungan. Aktor pada sistem kali ini ada dua, perempuan dan laki-laki.

4.9.1 Motion Random Model A



(a) Model A.1

(b) Model A.2

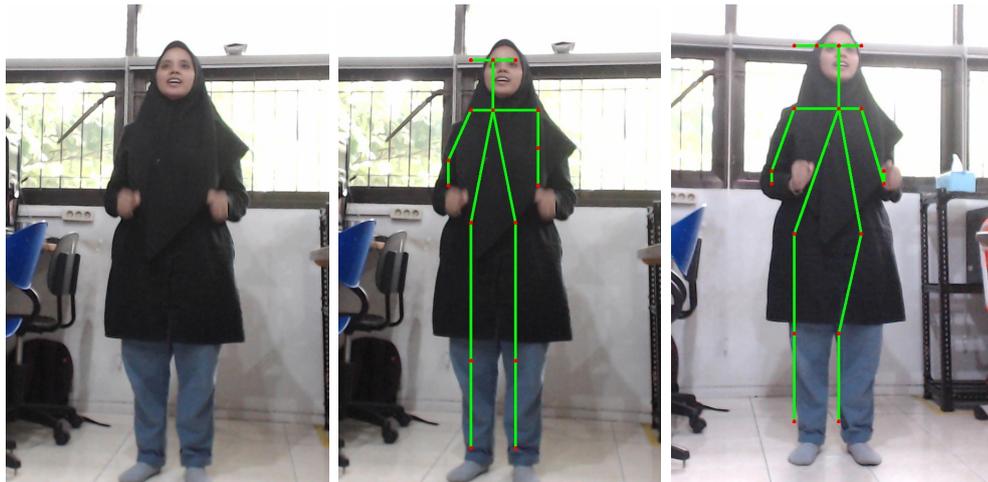
(c) Model A.3

Gambar 4.35: Rangkaian Citra Input Motion Random Model B.

Jika kita lihat Gambar 4.35 merupakan rangkaian gerakan yang dilakukan oleh aktor yang kemudian disimpan oleh kamera, untuk memudahkan menganalisa dari setiap gambar, kami akan memberikan hasil estimasi lokasi 2D *joint* dari kamera pertama dan hasil estimasi lokasi 2D *joint* dari kamera kedua dan juga hasil triangulasi koordinat Z dari setiap gambar.

Model A.1

Gerakan random Model A pertama dilakukan seperti pada Gambar 4.36, dengan jarak antar kamera 69 cm, dan jarak objek ke kamera sebesar 250 cm. Gambar 4.36 bagian a adalah gambar input. Gambar 4.36 bagian b dan c merupakan hasil *openpose* estimasi lokasi 2D *joint* dari kamera pertama dan kamera kedua.



(a) Citra Input (b) Kamera Pertama (c) Kamera Kedua

Gambar 4.36: Hasil Openpose dari Gerakan Model A.1.

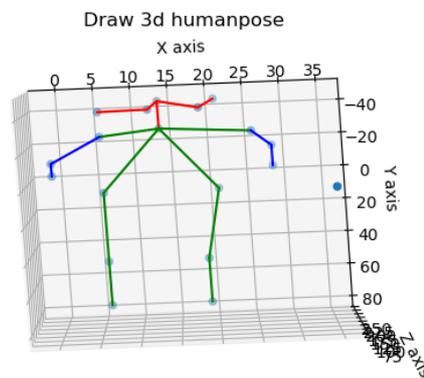
Seperti kita lihat pada Tabel 4.13 rata-rata error jarak dari seluruh joint yang terdeteksi adalah sekian 18.756 cm. Gambar 4.37 adalah hasil rekonstruksi 3D *human pose* dari Model A.1.

Model A.2

Gerakan Random Model A kedua dilakukan seperti pada Gambar 4.38, dengan jarak antar kamera 69 cm, dan jarak objek ke kamera sebesar 260 cm.

Tabel 4.13: Hasil Triangulasi Koordinat Z Model A.1

Id Joint	Nilai Z	Error dari jarak sebenarnya (250 cm)
0	247.054	2.946
1	247.211	2.789
2	222.157	27.843
3	221.176	28.824
4	221.274	28.726
5	224.315	25.685
6	247.653	2.347
7	247.679	2.321
8	222.882	27.118
9	223.45	26.55
10	223.947	26.053
11	247.758	2.242
12	224.408	25.592
13	224.58	25.42
14	223.501	26.499
15	223.832	26.168
16	221.788	28.212
17	247.727	2.273
Rata-rata Error (cm)		18.756



Gambar 4.37: Hasil 3D Rekonstruksi dari Gerakan Model A.1.

Gambar 4.38 bagian a adalah gambar input. Gambar 4.38 bagian b dan c merupakan hasil *openpose* estimasi lokasi 2D *joint* dari kamera pertama dan kamera kedua.

Seperti kita lihat pada Tabel 4.14 rata-rata error jarak dari seluruh joint yang terdeteksi adalah sekian 18.152 cm. Gambar 4.39 adalah hasil rekonstruksi 3D *human pose* dari Model A.2.



(a) Citra Input (b) Kamera Pertama (c) Kamera Kedua

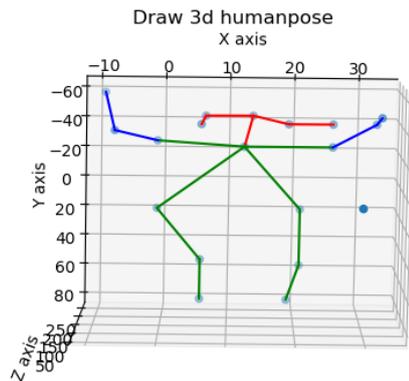
Gambar 4.38: Hasil Openpose dari Gerakan Model A.2.

Tabel 4.14: Hasil Triangulasi Koordinat Z Model A.2

Id Joint	Nilai Z	Error dari jarak sebenarnya (250 cm)
0	247.054	2.946
1	223.699	26.301
2	244.741	5.259
3	218.387	31.613
4	271.993	-21.993
5	224.313	25.687
6	224.892	25.108
7	188.996	61.004
8	245.715	4.285
9	223.45	26.55
10	223.947	26.053
11	247.758	2.242
12	247.774	2.226
13	224.58	25.42
14	245.955	4.045
15	223.832	26.168
16	221.864	28.136
17	224.308	25.692
Rata-rata Error (cm)		18.152

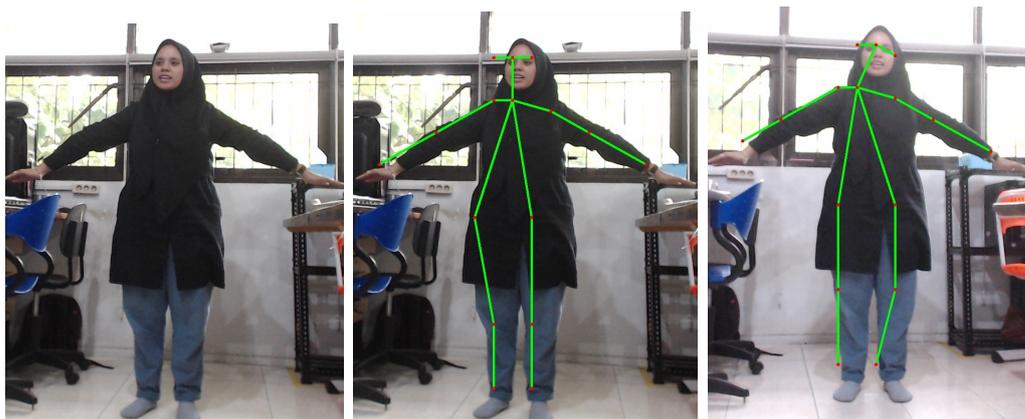
Model A.3

Gerakan random Model A ketiga dilakukan seperti pada Gambar 4.40, dengan jarak antar kamera 69 cm, dan jarak objek ke kamera sebesar 260 cm.



Gambar 4.39: Hasil 3D Rekonstruksi dari Gerakan Model A.2.

Gambar 4.40 bagian a adalah gambar input. Gambar 4.40 bagian b dan c merupakan hasil *openpose* estimasi lokasi 2D *joint* dari kamera pertama dan kamera kedua.



(a) Citra Input (b) Kamera Pertama (c) Kamera Kedua

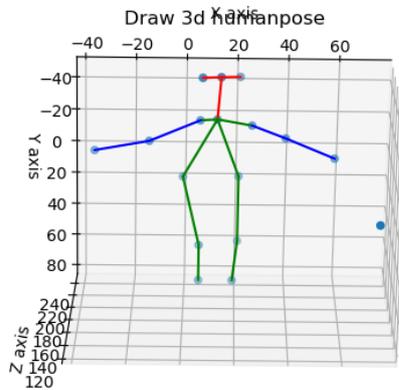
Gambar 4.40: Hasil Openpose dari Gerakan Model A.3.

Seperti kita lihat pada Tabel 4.15 rata-rata error jarak dari seluruh joint yang terdeteksi adalah sekian 16.029 cm. Gambar 4.42 adalah hasil rekonstruksi 3D *human pose* dari Model A.2.

Sehingga dapat kita ambil kesimpulan bahwa sistem dapat dilakukan pada percobaan motion random model A, dengan rentang error seperti pada Tabel 4.20 sebesar 16.029 hingga 18.756 cm. Gambar merupakan rangkaian hasil 3D dari rangkaian input Gambar 4.35.

Tabel 4.15: Hasil Triangulasi Koordinat Z Model A.3

Id Joint	Nilai Z	Error dari jarak sebenarnya (250 cm)
0	247.054	2.946
1	223.699	26.301
2	222.082	27.918
3	217.507	32.493
4	232.366	17.634
5	224.315	25.685
6	223.841	26.159
7	220.69	29.31
8	245.715	4.285
9	223.45	26.55
10	223.947	26.053
11	247.758	2.242
12	247.774	2.226
13	224.58	25.42
14	245.955	4.045
15	247.054	2.946
16	245.955	4.045
17	247.729	2.271
Rata-rata Error (cm)		16.029



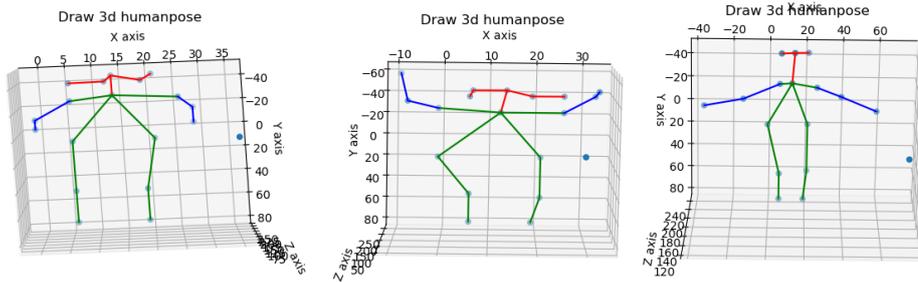
Gambar 4.41: Hasil 3D Rekonstruksi dari Gerakan Model A.3.

Tabel 4.16: Perbandingan Rata-Rata Error Motion Random Model A

	Model A.1	Model A.2	Model A.3
Rata-Rata Error (cm)	18.756	18.152	16.029

4.9.2 Motion Random Model B

Jika kita lihat Gambar 4.43 merupakan rangkaian gerakan yang dilakukan oleh aktor yang kemudian disimpan oleh kamera, untuk



(a) Model A.1

(b) Model A.2

(c) Model A.3

Gambar 4.42: Rangkaian Hasil 3D Rekonstruksi Motion Random Model A

memudahkan menganalisa dari setiap gambar, kami akan memberikan hasil estimasi lokasi 2D *joint* dari kamera pertama dan hasil estimasi lokasi 2D *joint* dari kamera kedua dan juga hasil triangulasi koordinat Z dari setiap gambar.



(a) Model B.1

(b) Model B.2

(c) Model B.3

Gambar 4.43: Rangkaian Citra Input Motion Random Model B.

Model B.1

Gerakan random Model B pertama dilakukan seperti pada Gambar 4.44, dengan jarak antar kamera 69 cm, dan jarak objek ke kamera sebesar 250 cm. Gambar 4.44 bagian a adalah gambar input. Gambar 4.44 bagian b dan c merupakan hasil *openpose* estimasi lokasi 2D *joint* dari kamera pertama dan kamera kedua.



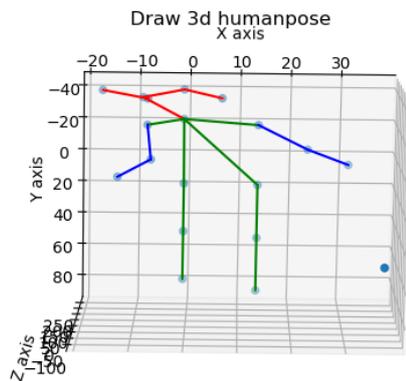
(a) Citra Input (b) Kamera Pertama (c) Kamera Kedua

Gambar 4.44: Hasil Openpose dari Gerakan Model B.1.

Seperti kita lihat pada Tabel 4.17 rata-rata error jarak dari seluruh joint yang terdeteksi adalah 17.538 cm. Gambar 4.45 adalah hasil rekonstruksi 3D *human pose* dari Model B.1.

Tabel 4.17: Hasil Triangulasi Koordinat Z Model B.1

Id Joint	Nilai Z	Error dari jarak sebenarnya (250 cm)
0	272.326	22.326
1	274.62	24.62
2	243.078	6.922
3	219.593	30.407
4	218.506	31.494
5	247.241	2.759
6	276.695	26.695
7	275.322	25.322
8	221.658	28.342
9	222.324	27.676
10	223.079	26.921
11	247.518	2.482
12	247.73	2.27
13	247.969	2.031
14	242.605	7.395
15	274.321	24.321
16	269.787	19.787
17	246.075	3.925
Rata-rata Error (cm)		17.538



Gambar 4.45: Hasil 3D Rekonstruksi dari Gerakan Model B.1.

Model B.2

Gerakan random Model B kedua dilakukan seperti pada Gambar 4.46, dengan jarak antar kamera 69 cm, dan jarak objek ke kamera sebesar 260 cm. Gambar 4.46 bagian a adalah gambar input. Gambar 4.46 bagian b dan c merupakan hasil *openpose* estimasi lokasi 2D *joint* dari kamera pertama dan kamera kedua.



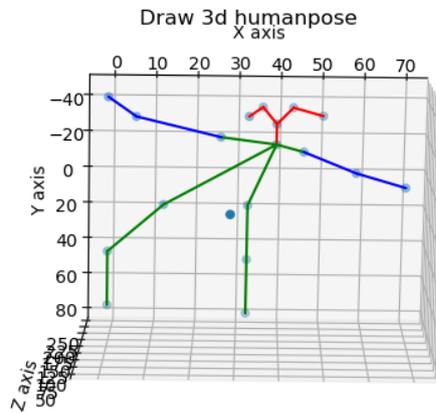
(a) Citra Input (b) Kamera Pertama (c) Kamera Kedua

Gambar 4.46: Hasil Openpose dari Gerakan Model B.2.

Seperti kita lihat pada Tabel 4.18 rata-rata error jarak dari seluruh joint yang terdeteksi adalah 23.242 cm. Gambar 4.47 adalah hasil rekonstruksi 3D *human pose* dari Model B.2.

Tabel 4.18: Hasil Triangulasi Koordinat Z Model B.2

Id Joint	Nilai Z	Error dari jarak sebenarnya (250cm)
0	224.014	25.986
1	223.885	26.115
2	224.315	25.685
3	221.935	28.065
4	220.109	29.891
5	223.065	26.935
6	220.586	29.414
7	216.079	33.921
8	224.236	25.764
9	222.324	27.676
10	222.983	27.017
11	224.573	25.427
12	224.427	25.573
13	224.284	25.716
14	247.552	2.448
15	246.8	3.2
16	224.849	25.151
17	245.622	4.378
Rata-rata Error (cm)		23.242



Gambar 4.47: Hasil 3D Rekonstruksi dari Gerakan Model B.2.

Model B.3

Gerakan random Model B ketiga dilakukan seperti pada Gambar 4.48, dengan jarak antar kamera 69 cm, dan jarak objek ke kamera sebesar 260 cm. Gambar 4.48 bagian a adalah gambar input. Gambar 4.48 bagian b dan c merupakan hasil *openpose* estimasi lokasi 2D *joint* dari kamera pertama dan

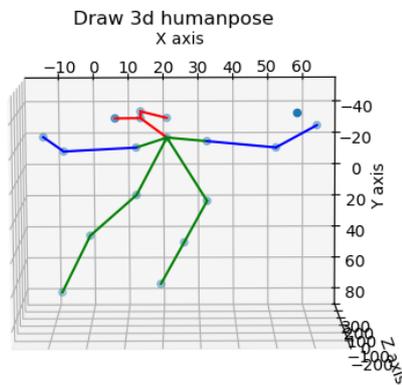
kamera kedua.



(a) Citra Input (b) Kamera Pertama (c) Kamera Kedua

Gambar 4.48: Hasil Openpose dari Gerakan Model B.3.

Seperti kita lihat pada Tabel 4.19 rata-rata error jarak dari seluruh joint yang terdeteksi adalah 15.348 cm. Gambar 4.58 adalah hasil rekonstruksi 3D *human pose* dari Model B.3.



Gambar 4.49: Hasil 3D Rekonstruksi dari Gerakan Model B.3.

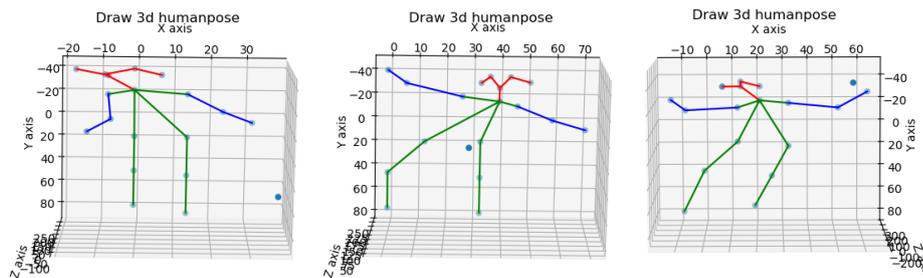
Sehingga dapat kita ambil kesimpulan bahwa sistem dapat dilakukan pada percobaan motion random model B, dengan rentang error seperti pada Tabel 4.20 sebesar 15.348 hingga 23.242 cm. Gambar 4.58 merupakan rangkaian hasil 3D dari rangkaian input Gambar 4.43.

Tabel 4.19: Hasil Triangulasi Koordinat Z Model B.3

Id Joint	Nilai Z	Error dari jarak sebenarnya (250 cm)
0	247.148	2.852
1	247.741	2.259
2	223.847	26.153
3	243.314	6.686
4	217.072	32.928
5	224.785	25.215
6	221.947	28.053
7	241.82	8.18
8	224.283	25.717
9	246.325	3.675
10	275.167	25.167
11	224.573	25.427
12	224.345	25.655
13	224.58	25.42
14	246.137	3.863
15	247.116	2.884
16	246.137	3.863
17	247.733	2.267
Rata-rata Error (cm)		15.348

Tabel 4.20: Perbandingan Rata-Rata Error Motion Random Model B

	Model B.1	Model B.2	Model B.3
Rata-Rata Error (cm)	17.538	23.242	15.348



(a) Model B.1

(b) Model B.2

(c) Model B.3

Gambar 4.50: Rangkaian Hasil 3D Rekonstruksi Motion Random Model B

4.9.3 Motion Random Model C

Jika kita lihat Gambar 4.51 ini, ini adalah rangkaian gerakan yang dilakukan oleh aktor yang kemudian disimpan oleh kamera, untuk

memudahkan menganalisa dari setiap gambar, kami akan memberikan hasil estimasi lokasi 2D *joint* dari kamera pertama dan hasil estimasi lokasi 2D *joint* dari kamera kedua dan juga hasil triangulasi koordinat Z dari setiap gambar.



(a) Model C.1

(b) Model C.2

(c) Model C.3

Gambar 4.51: Rangkaian Citra Input Motion Random Model C.

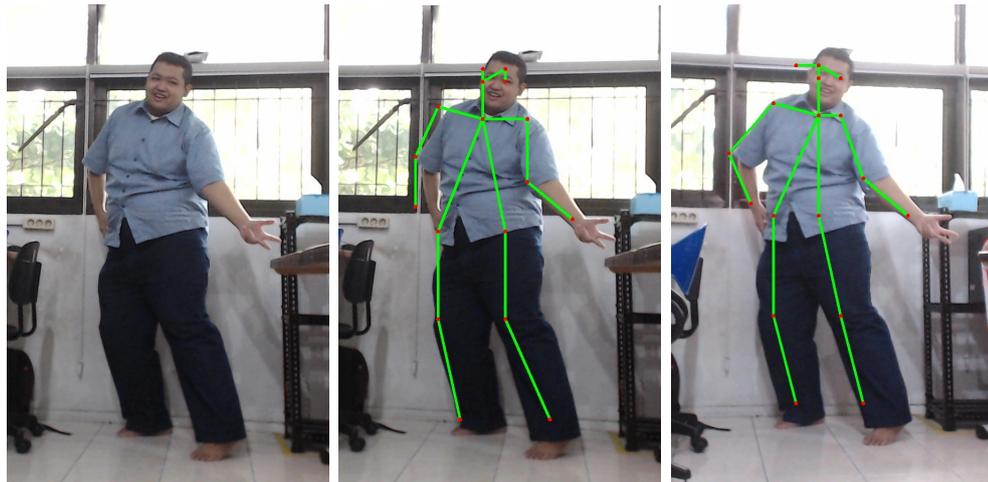
Model C.1

Gerakan random Model C pertama dilakukan seperti pada Gambar 4.52, dengan jarak antar kamera 69 cm, dan jarak objek ke kamera sebesar 260 cm. Gambar 4.52 bagian a adalah gambar input. Gambar 4.52 bagian b dan c merupakan hasil *openpose* estimasi lokasi 2D *joint* dari kamera pertama dan kamera kedua.

Seperti kita lihat pada Tabel 4.21 rata-rata error jarak dari seluruh *joint* yang terdeteksi adalah 61.45 cm. Gambar 4.53 adalah hasil rekonstruksi 3D *human pose* dari Model C.1.

Model C.2

Gerakan random Model C kedua dilakukan seperti pada Gambar 4.54, dengan jarak antar kamera 69 cm, dan jarak objek ke kamera sebesar 260 cm. Gambar 4.54 bagian a adalah gambar input. Gambar 4.54 bagian b dan c merupakan hasil *openpose* estimasi lokasi 2D *joint* dari kamera pertama dan



(a) Citra Input (b) Kamera Pertama (c) Kamera Kedua

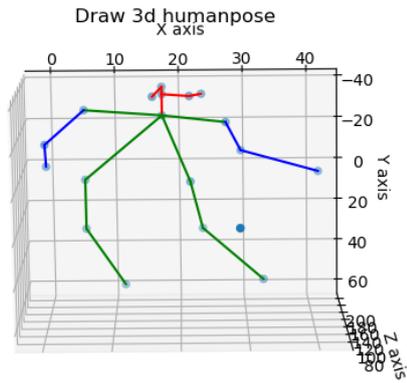
Gambar 4.52: Hasil Openpose dari Gerakan Model C.1.

Tabel 4.21: Hasil Triangulasi Koordinat Z Model C.1

Id Joint	Nilai Z	Error dari jarak sebenarnya (260 cm)
0	204.429	55.571
1	204.552	55.448
2	202.001	57.999
3	184.437	75.563
4	201.303	58.697
5	188.839	71.161
6	205.367	54.633
7	204.243	55.757
8	202.728	57.272
9	203.367	56.633
10	205.058	54.942
11	188.699	71.301
12	205.019	54.981
13	189.028	70.972
14	204.387	55.613
15	188.285	71.715
16	187.526	72.474
17	204.622	55.378
Rata-rata Error (cm)		61.45

kamera kedua.

Seperti kita lihat pada Tabel 4.18 rata-rata error jarak dari seluruh joint yang terdeteksi adalah 64.1635 cm. Gambar 4.55 adalah hasil rekonstruksi 3D



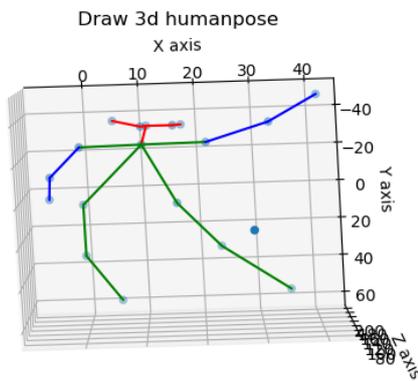
Gambar 4.53: Hasil 3D Rekonstruksi dari Gerakan Model C.1.



(a) Citra Input (b) Kamera Pertama (c) Kamera Kedua

Gambar 4.54: Hasil Openpose dari Gerakan Model C.2.

human pose dari Model C.2.



Gambar 4.55: Hasil 3D Rekonstruksi dari Gerakan Model C.2.

Tabel 4.22: Hasil Triangulasi Koordinat Z Model C.2

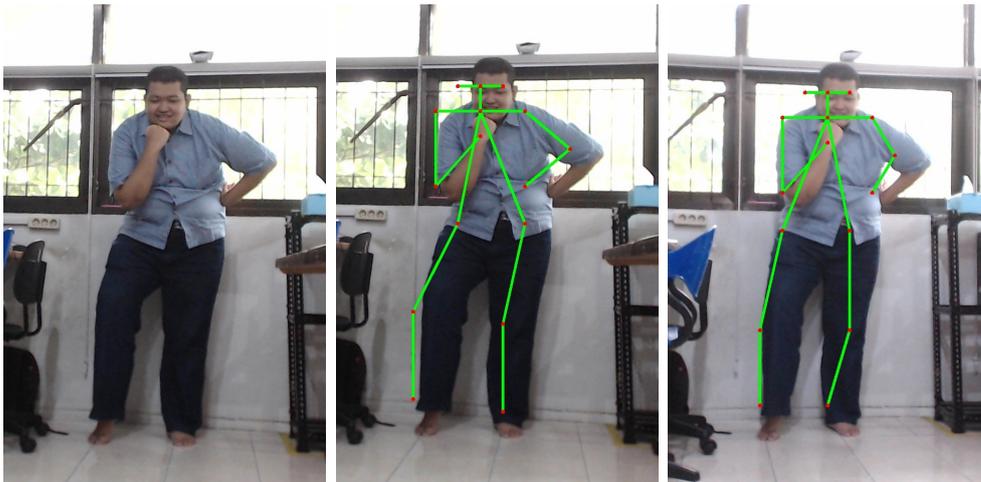
Id Joint	Nilai Z	Error dari jarak sebenarnya (260 cm)
0	203.428	56.572
1	186.711	73.289
2	184.026	75.974
3	182.773	77.227
4	200.093	59.907
5	188.39	71.61
6	188.996	71.004
7	204.781	55.219
8	201.503	58.497
9	202.2	57.8
10	203.919	56.081
11	188.147	71.853
12	205.019	54.981
13	204.902	55.098
14	186.501	73.499
15	187.526	72.474
16	201.755	58.245
17	204.387	55.613
Rata-rata Error (cm)		64.1635

Model C.3

Gerakan random Model C ketiga dilakukan seperti pada Gambar 4.56, dengan jarak antar kamera 69 cm, dan jarak objek ke kamera sebesar 260 cm. Gambar 4.56 bagian a adalah gambar input. Gambar 4.56 bagian b dan c merupakan hasil *openpose* estimasi lokasi 2D *joint* dari kamera pertama dan kamera kedua.

Seperti kita lihat pada Tabel 4.23 rata-rata error jarak dari seluruh joint yang terdeteksi adalah 57.255 cm. Gambar 4.57 adalah hasil rekonstruksi 3D *human pose* dari Model C.3.

Sehingga dapat kita ambil kesimpulan bahwa sistem dapat dilakukan pada percobaan motion random model C, dengan rentang error seperti pada Tabel 4.24 sebesar 57.255 hingga 64.1635 cm. Error yang lebih besar dari sebelumnya karena posisi sudut antar kamera nya terbilang kecil, bahkan saat mengambil gambar ini, posisi kamera terbilang paralel, berbeda dengan



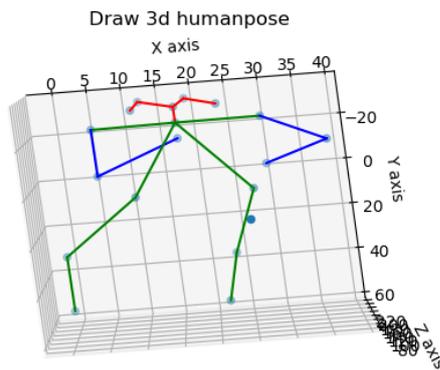
(a) Citra Input (b) Kamera Pertama (c) Kamera Kedua

Gambar 4.56: Hasil Openpose dari Gerakan Model C.3.

Tabel 4.23: Hasil Triangulasi Koordinat Z Model C.3

Id Joint	Nilai Z	Error dari jarak sebenarnya (250 cm)
0	204.47	55.53
1	204.552	55.448
2	202.081	57.919
3	202.567	57.433
4	204.634	55.366
5	205.36	54.64
6	188.853	71.147
7	205.369	54.631
8	187.399	72.601
9	202.297	57.703
10	202.89	57.11
11	189	71
12	205.039	54.961
13	189.204	70.796
14	223.6	36.4
15	223.883	36.117
16	203.552	56.448
17	204.643	55.357
Rata-rata Error (cm)		57.255

percobaan skenario sebelumnya, yang meletakkan posisi kamera memiliki sudut, sehingga hasil deteksi *joint* 2D yang didapat lebih besar perbedaan antara kedua *joint* tersebut, dan hasil triangulasi nantinya juga akan lebih

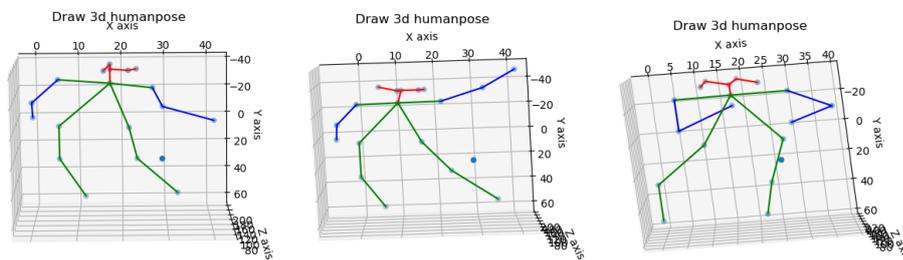


Gambar 4.57: Hasil 3D Rekonstruksi dari Gerakan Model C.3.

baik. Gambar 4.57 merupakan rangkaian hasil 3D dari rangkaian input Gambar 4.51.

Tabel 4.24: Perbandingan Rata-Rata Error Motion Random Model C

	Model C.1	Model C.2	Model C.3
Rata-Rata Error (cm)	61.45	64.1635	57.255



(a) Model C.1

(b) Model C.2

(c) Model C.3

Gambar 4.58: Rangkaian Hasil 3D Rekonstruksi Motion Random Model C

4.9.4 Motion Random Model D

Jika kita lihat Gambar 4.59 ini, ini adalah rangkaian gerakan yang dilakukan oleh aktor yang kemudian disimpan oleh kamera, untuk memudahkan menganalisa dari setiap gambar, kami akan memberikan hasil estimasi lokasi 2D *joint* dari kamera pertama dan hasil estimasi lokasi 2D *joint* dari kamera kedua dan juga hasil triangulasi koordinat Z dari setiap gambar.



(a) Model D.1

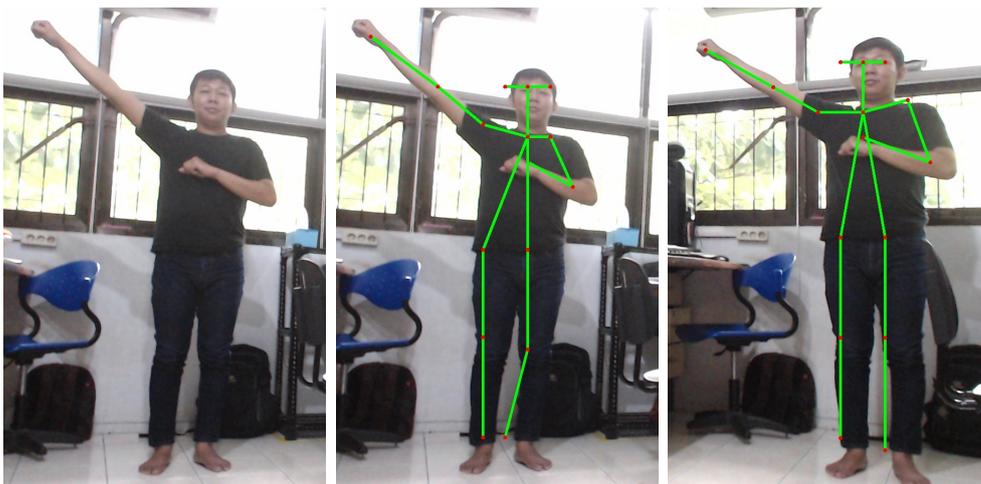
(b) Model D.2

(c) Model D.3

Gambar 4.59: Rangkaian Citra Input Motion Random Model D.

Model D.1

Gerakan random Model D pertama dilakukan seperti pada Gambar 4.60, dengan jarak antar kamera 57 cm, dan jarak objek ke kamera sebesar 300 cm. Gambar 4.60 bagian a adalah gambar input. Gambar 4.60 bagian b dan c merupakan hasil *openpose* estimasi lokasi 2D *joint* dari kamera pertama dan kamera kedua.



(a) Citra Input

(b) Kamera Pertama

(c) Kamera Kedua

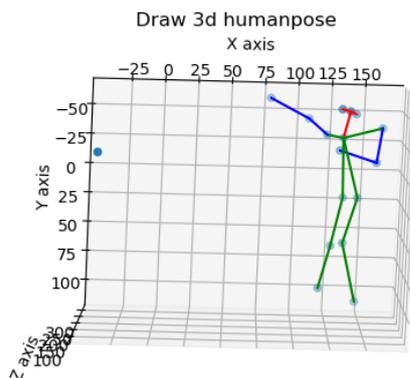
Gambar 4.60: Hasil Openpose dari Gerakan Model D.1.

Seperti kita lihat pada Tabel 4.25 rata-rata error jarak dari seluruh joint

yang terdeteksi adalah 12.001 cm. Gambar 4.61 adalah hasil rekonstruksi 3D *human pose* dari Model D.1.

Tabel 4.25: Hasil Triangulasi Koordinat Z Model D.1

Id Joint	Nilai Z	Error dari jarak sebenarnya (300 cm)
0	305.946	5.946
1	295.398	4.602
2	310.85	10.85
3	325.515	25.515
4	326.671	26.671
5	318.684	18.684
6	293.117	6.883
7	290.347	9.653
8	321.855	21.855
9	302.874	2.874
10	283.285	16.715
11	303.718	3.718
12	282.645	17.355
13	306.666	6.666
14	313.834	13.834
15	305.946	5.946
16	313.834	13.834
17	295.579	4.421
Rata-rata Error (cm)		12.001

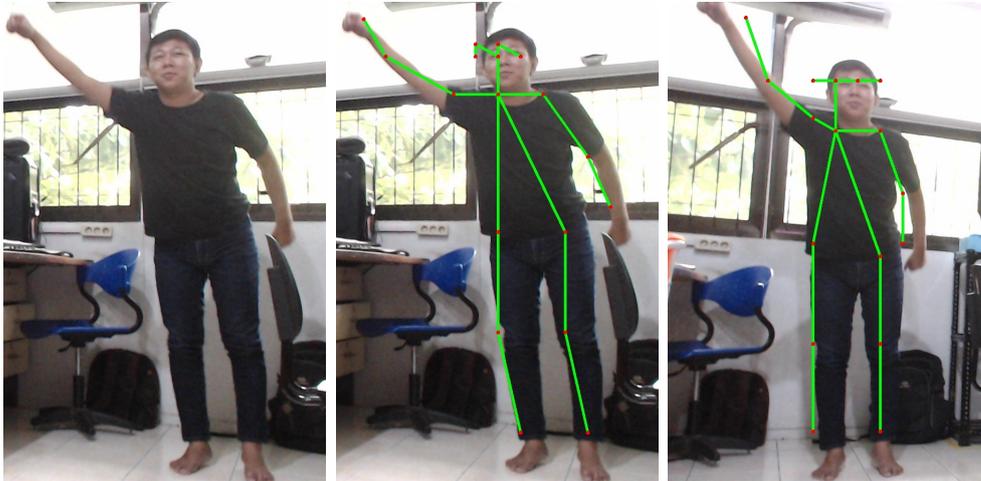


Gambar 4.61: Hasil 3D Rekonstruksi dari Gerakan Model D.1.

Model D.2

Gerakan random Model D kedua dilakukan seperti pada Gambar 4.62, dengan jarak antar kamera 57 cm, dan jarak objek ke kamera sebesar 300 cm.

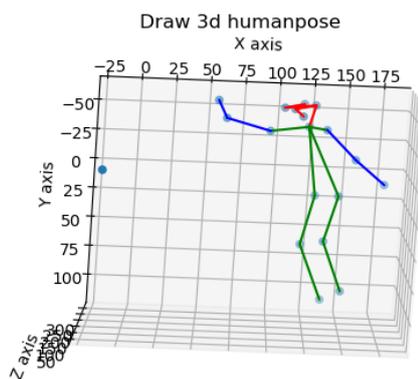
Gambar 4.62 bagian a adalah gambar input. Gambar 4.62 bagian b dan c merupakan hasil *openpose* estimasi lokasi 2D *joint* dari kamera pertama dan kamera kedua.



(a) Citra Input (b) Kamera Pertama (c) Kamera Kedua

Gambar 4.62: Hasil Openpose dari Gerakan Model D.2.

Seperti kita lihat pada Tabel 4.26 rata-rata error jarak dari seluruh joint yang terdeteksi adalah 18.455 cm. Gambar 4.63 adalah hasil rekonstruksi 3D *human pose* dari Model D.2.



Gambar 4.63: Hasil 3D Rekonstruksi dari Gerakan Model D.2.

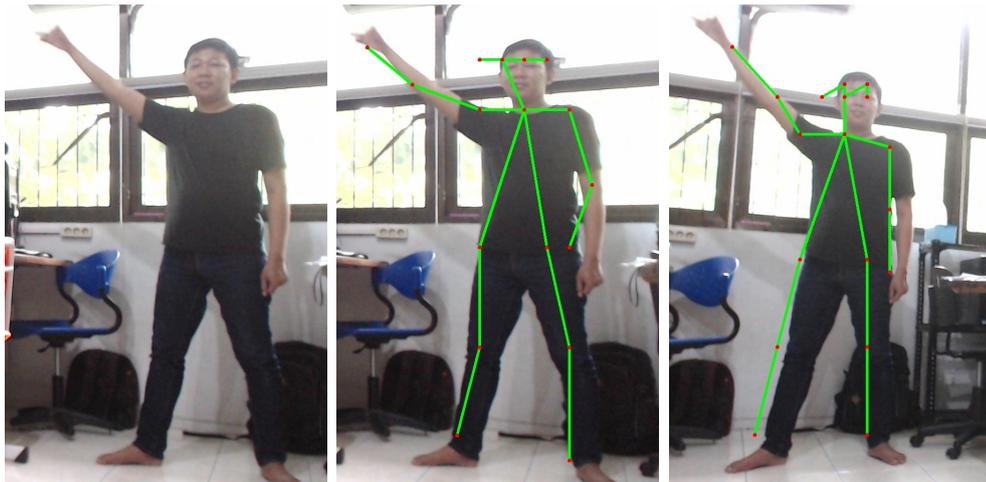
Model D.3

Gerakan random Model D ketiga dilakukan seperti pada Gambar 4.64, dengan jarak antar kamera 57 cm, dan jarak objek ke kamera sebesar 300 cm. Gambar 4.64 bagian a adalah gambar input. Gambar 4.64 bagian b dan c

Tabel 4.26: Hasil Triangulasi Koordinat Z Model D.2

Id Joint	Nilai Z	Error dari jarak sebenarnya (250 cm)
0	321.392	21.392
1	310.85	10.85
2	283.56	16.44
3	260.183	39.817
4	265.452	34.548
5	297.923	2.077
6	306.437	6.437
7	330.6	30.6
8	331.052	31.052
9	309.333	9.333
10	328.733	28.733
11	303.718	3.718
12	285.104	14.896
13	297.495	2.505
14	286.158	13.842
15	282.63	17.37
16	325.263	25.263
17	276.672	23.328
Rata-rata Error (cm)		18.455

merupakan hasil *openpose* estimasi lokasi 2D *joint* dari kamera pertama dan kamera kedua.



(a) Citra Input (b) Kamera Pertama (c) Kamera Kedua

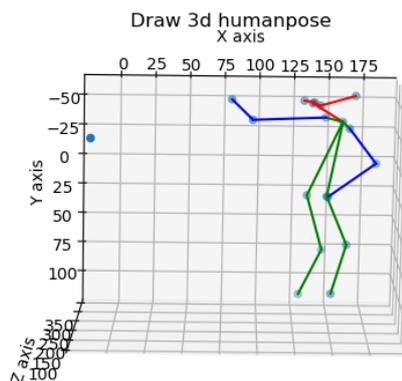
Gambar 4.64: Hasil Openpose dari Gerakan Model D.3.

Seperti kita lihat pada Tabel 4.27 rata-rata error jarak dari seluruh joint

yang terdeteksi adalah 18.444 cm. Gambar 4.66 adalah hasil rekonstruksi 3D *human pose* dari Model D.3.

Tabel 4.27: Hasil Triangulasi Koordinat Z Model D.3

Id Joint	Nilai Z	Error dari jarak sebenarnya (250 cm)
0	303.194	3.194
1	330.972	30.972
2	349.502	49.502
3	285.533	14.467
4	293.857	6.143
5	301.819	1.819
6	320.813	20.813
7	274.185	25.815
8	319.076	19.076
9	347.53	47.53
10	332.384	32.384
11	289.488	10.512
12	301.338	1.338
13	281.365	18.635
14	305.946	5.946
15	295.579	4.421
16	311.078	11.078
17	328.352	28.352
Rata-rata Error (cm)		18.444



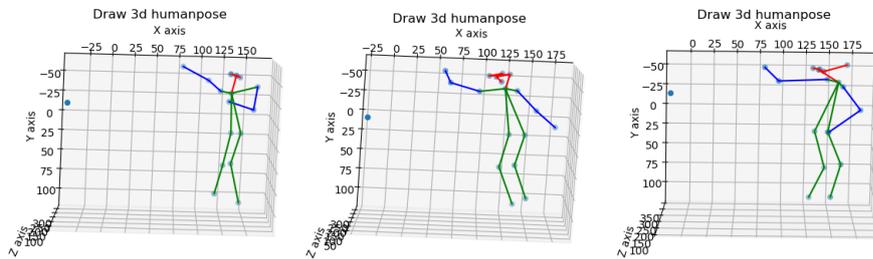
Gambar 4.65: Hasil 3D Rekonstruksi dari Gerakan Model D.3.

Sehingga dapat kita ambil kesimpulan bahwa sistem dapat dilakukan pada percobaan motion random model D, dengan rentang error seperti pada Tabel 4.28 sebesar 12.001 hingga 18.455 cm. Error yang lebih kecil dari

sebelumnya karena posisi sudut antar kameranya dibesarkan hingga memiliki sudut kemiringan dari tiap kamera, sehingga hasil deteksi *joint* 2D yang didapat lebih besar perbedaan antara kedua *joint* tersebut, dan hasil triangulasi nantinya juga akan lebih baik. Gambar 4.66 merupakan rangkaian hasil 3D dari rangkaian input Gambar 4.59.

Tabel 4.28: Perbandingan Rata-Rata Error Motion Random Model D

	Model D.1	Model D.2	Model D.3
Rata-Rata Error (cm)	12.001	18.455	18.444



(a) Model D.1

(b) Model D.2

(c) Model D.3

Gambar 4.66: Rangkaian Hasil 3D Rekonstruksi Motion Random Model D

Halaman ini sengaja dikosongkan

BAB 5

PENUTUP

5.1 Kesimpulan

Dari hasil pengujian yang telah dilakukan dengan berbagai kondisi percobaan sistem, dapat ditarik beberapa kesimpulan, diantaranya sebagai berikut :

1. *Markerless motion capture* berbasis *openpose* model menggunakan metode triangulasi dapat diterapkan dengan entang rata-rata error kedalaman 13 - 21 cm. Hasil lokasi 2D *joint* dari *openpose* berpangaruh besar terhadap baik atau tidaknya triangulasi.
2. *Markerless motion capture* dapat dilakukan pada berbagai kondisi gender, bentuk tubuh dan baju yang berbeda. Desain jarak antar kamera yang memberikan *motion capture* lebih baik ada pada desain jarak antar kamera 66 cm.

5.2 Saran

Pengembangan sistem ini masih belum sempurna, untuk itu terdapat beberapa hal yang masih dapat dilakukan, hal ini dilakukan untuk pengembangan *markerless motion capture* yang lebih baik dan penggunaan *markerless motion capture* yang lebih berguna. Berikut adalah beberapa yang dapat dilakukan:

1. Perbaiki dan analisa untuk mendapatkan error minimal pada hasil triangulasi atau depth
2. Menambahkan jumlah kamera lebih dari dua untuk mendapatkan triangulasi yang lebih baik
3. Mencoba sistem pada kondisi luar ruangan
4. Mencoba sistem pada aktor yang lebih beragam dan pose yang lebih beragam
5. Menambahkan jumlah orang yang diproses
6. Melakukan asosiasi *joint* tidak secara manual
7. Pengembangan selanjutnya dapat dilakukan pada peniruan gerakan manusia pada *Robot Humanoid*, ataupun generate aset 3D

DAFTAR PUSTAKA

- Agarwal, P., Al Moubayed, S., Alspach, A., Kim, J., Carter, E. J., Lehman, J. F., and Yamane, K. (2016). Imitating human movement with teleoperated robotic head. In 25th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2016. (Dikutip pada halaman 7).
- Belagiannis, V., Amin, S., and Andriluka, M. (2016). 3D Pictorial Structures Revisited : Multiple Human Pose Estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(10):1929–1942. (Dikutip pada halaman 2, 9).
- Cao, Z., Simon, T., Wei, S. E., and Sheikh, Y. (2017). Realtime multi-person 2D pose estimation using part affinity fields. In Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. (Dikutip pada halaman xix, 2, 9, 17, 18, 19, 20, 22, 23, 28, 29).
- Chen, C. H. and Ramanan, D. (2017). 3D human pose estimation = 2D pose estimation + matching. In Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. (Dikutip pada halaman 2, 8, 10).
- Cyganek, B. and Siebert, J. P. (2009). An Introduction to 3D Computer Vision Techniques and Algorithms. (Dikutip pada halaman 10).
- Ge, L., Liang, H., Yuan, J., and Thalmann, D. (2018). Robust 3D Hand Pose Estimation from Single Depth Images Using Multi-View CNNs. In IEEE Transactions on Image Processing. (Dikutip pada halaman 2, 10).
- Géron, A. (2017). Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, 1 edition edition. (Dikutip pada halaman 24, 25).
- Hartley, R. (2004). Multiple View Geometry in Computer Vision. Cambridge University Press, 2 edition. (Dikutip pada halaman xix, 11, 14, 16, 34).
- Hwang, D.-h. (2019). MonoEye : Monocular Fisheye Camera-based 3D Human Pose Estimation. pages 988–989. (Dikutip pada halaman 3).
- Hwang, K. S., Jiang, W. C., Chen, Y. J., and Shi, H. (2017). Motion Segmentation and Balancing for a Biped Robot's Imitation Learning. IEEE Transactions on Industrial Informatics. (Dikutip pada halaman 7).
- Ionescu, C., Papava, D., Olaru, V., and Sminchisescu, C. (2014). Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. pages 1–15. (Dikutip pada halaman 62).
- Kasaei, S. and Sanaei, E. (2018). Regression-based convolutional 3D pose estimation from single image. 54(5):292–293. (Dikutip pada halaman 3).

- Lin, G.-h. (2019). Fully Convolutional Network for 3D Human Skeleton Estimation from a Single View for Action Analysis. 2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), pages 1–6. (Dikutip pada halaman 2, 9).
- Liu, Z., Zhou, L., Leung, H., Hubert, P., and Shum, H. (2016). Kinect Posture Reconstruction Based on a Local Mixture of Gaussian Process Models. IEEE Transactions on Visualization and Computer Graphics, 22(11):2437–2450. (Dikutip pada halaman xix, 1, 2, 7, 8).
- Martinez, J., Hossain, R., Romero, J., and Little, J. J. (2017). A simple yet effective baseline for 3d human pose estimation. (Dikutip pada halaman 3).
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556. (Dikutip pada halaman xix, 24, 25).
- Wei, S. E., Ramakrishna, V., Kanade, T., and Sheikh, Y. (2016). Convolutional pose machines. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (Dikutip pada halaman xix, 24).
- Xu, L., Liu, Y., Cheng, W., Guo, K., Zhou, G., Dai, Q., and Fang, L. (2018). FlyCap: Markerless Motion Capture Using Multiple Autonomous Flying Cameras. IEEE Transactions on Visualization and Computer Graphics. (Dikutip pada halaman 2, 8).
- Zhang, Z. and Member, S. (2000). A Flexible New Technique for Camera Calibration æ. 22(11):1330–1334. (Dikutip pada halaman 16).

BIOGRAFI PENULIS



Uti Solichah, lahir di Indramayu pada tanggal 30 April 1994. Penulis menempuh pendidikan Strata Satu di Teknik Informatika Institut Teknologi Sepuluh Nopember sejak tahun 2012 hingga 2016. Semasa perkuliahan Strata Satu penulis aktif menjadi staff Hubungan Luar Himpunan Mahasiswa Teknik Computer periode 2013/2014, sebagai staff Riset dan dan Teknologi Unit Kegiatan Mahasiswa Robotika. Selain itu, penulis juga aktif menjadi anggota Tim Robotika ITS sejak tahun 2013 hingga tahun 2016, dan mengikuti berbagai perlombaan Kontes Robot Indonesia Regional maupun Nasional dari 2013 hingga 2016. Penulis menempuh Pendidikan Magister pada Jurusan Teknik Elektro Bidang Minat Jaringan Cerdas Multimedia sejak tahun 2018. Penulis dapat dihubungi melalui email: uti.solichah@gmail.com.

Halaman ini sengaja dikosongkan