

SKRIPSI

PEMROGRAMAN KALIBRASI KAMERA IP STEREO DENGAN MENGGUNAKAN PRINSIP KOLINEARITI UNTUK PAPAN KALIBRASI 2D DAN 3D



**Diajukan untuk memenuhi persyaratan
dalam mencapai gelar sarjana S-1 Teknik Geodesi**

Disusun Oleh :

SYAHRUNI

07.25.026

**JURUSAN TEKNIK GEODESI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
INSTITUT TEKNOLOGI NASIONAL
MALANG
2012**

SKRIPSI

PENGOLOHANNYA KALIBARASI KAMENIA IS STERIL
DEWAS MENGGUNAKAN PRINSIP KILNARINTI LUTUK PAKAN
KALIBARASI 20 JAN 20

Disajikan untuk memenuhi persyaratan
dalam mencapai gelar sarjana S-1 Teknik Geologi

Disusun Oleh :
STANISLAUS
07.13.033

LUBERAN TEKNIK GEOMATIKA
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
INSTITUT TEKNOLOGI NASIONAL
MALANG
2012

**LEMBAR PERSETUJUAN
SKRIPSI**

**"Pemrograman Kalibrasi Kamera IP Stereo dengan Menggunakan Prinsip
Kolineariti untuk Papan Kalibrasi 2D dan 3D"**

Diajukan Sebagai Salah Satu Syarat Memperoleh
Gelar Sarjana Teknik Geodesi S-1
Institut Teknologi Nasional Malang

Disusun Oleh :

Syahrani

07.25.026

Menyetujui,

Dosen Pembimbing I



Dr. M. Edwin Tjahjadi, ST., M.Geom.Sc

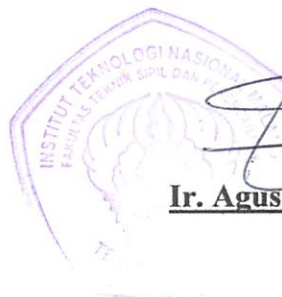
Dosen Pembimbing II



Ir. Leo Patimena, M.Sc

Mengetahui,

Ketua Jurusan Teknik Geodesi S-1

Ir. Agus Darpono, MT

LEMBAR PENGESAHAN

SKRIPSI

"Pemrograman Kalibrasi Kamera IP Stereo dengan Menggunakan Prinsip Kolineariti untuk Papan Kalibrasi 2D dan 3D"

Telah dipertahankan di hadapan Majelis Penguji Sidang Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Rabu

Tanggal : 8 Agustus 2012

Dan diterima untuk memenuhi salah satu persyaratan guna memperoleh gelar Sarjana Teknik (ST).

Disusun Oleh :

Syahrani

07.25.026

Panitia Ujian Tugas Akhir

Ketua



Ir. Agus Darpono, MT

Sekretaris



Silvester Sari Sai, ST., MT

Anggota Penguji

Penguji I



Ir. M. Nurhadi, MT

Penguji II



Dr. M. Edwin Tjahjadi, ST., M.Geom.Sc

Penguji III



Ir. D.K. Sunaryo, MS.Tis

JURUSAN TEKNIK GEODESI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
INSTITUT TEKNOLOGI NASIONAL
MALANG

2012

**PEMROGRAMAN KALIBRASI KAMERA IP STEREO DENGAN
MENGUNAKAN PRINSIP KOLINEARITI UNTUK PAPAN KALIBRASI
2D DAN 3D**

Syahrini 07.25.026

Dosen Pembimbing I : Dr. M. Edwin Tjahjadi, ST., M.Geom.Sc

Dosen Pembimbing II : Ir. Leo Patimena, M.Sc

ABSTRAKSI

Sebelum melakukan pengambilan data di lapangan, kamera IP yang akan digunakan harus dikalibrasi terlebih dahulu agar mendapatkan data foto yang akurat. Kamera IP adalah jenis kamera non-metrik yang tidak stabil dan mempunyai lensa yang tidak sempurna, sehingga proses perekaman yang dilakukan akan memiliki kesalahan (*Hanifa, 2007*). Oleh karena itu, kamera IP perlu dilakukan kalibrasi menggunakan persamaan kolinear agar dapat menentukan besarnya penyimpangan-penyimpangan yang terjadi (*Fraser, 2000*). Kalibrasi kamera dilakukan untuk menentukan parameter internal kamera (IO) yang meliputi *principal distance* (f), titik pusat *fidusial* foto (x_0, y_0), distorsi lensa (K_1, K_2, K_3, P_1 dan P_2), serta distorsi akibat perbedaan penyekalaan dan ketidak ortogonal antara sumbu X dan sumbu Y (b_1, b_2) (*Fraser, 2000*).

Kata kunci : Kamera IP, Kamera non-metrik, Persamaan kolinear, Kalibrasi Kamera

PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan dibawah ini :

Nama : Syahruni
NIM : 07.25.026
Program Studi : Teknik Geodesi S-1
Fakultas : Teknik Sipil Dan Perencanaan

Menyatakan dengan sesungguhnya bahwa Skripsi saya dengan judul :

**"Pemrograman Kalibrasi Kamera IP Stereo Dengan Menggunakan Prinsip
Kolineariti Untuk Papan Kalibrasi 2D Dan 3D"**

Adalah hasil karya saya sendiri, bukan merupakan duplikat, copy, salinan maupun saduran, kecuali beberapa kalimat kutipan dan gambar yang telah disebutkan sumbernya.

Malang, 23 September 2012

Yang membuat pernyataan

Syahruni

07.25.026

THANK'S FOR ALL

Puji syukur kehadiran Allah SWT atas limpahan rahmat dan hidayah-Nya yang telah membimbing saya menyelesaikan skripsi ini. Saya ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

- Kedua orang tua saya yang selalu memberikan kasih sayang, doa, nasehat, dan memberikan support moril dan materil, I LOVE Mama N Abah.
- Keluarga yang selalu memberikan semangat dan doa untuk saya.
- Dosen-dosen Geodesi yang telah memberikan ilmu dan pengalaman yang berharga buat saya.
- Anak-anak Makhluk Laut (ML) : Idi, Yaomi, Icho, dan Arya, makasih telah menjadi teman, sahabat, dan saudara, moga kita semua jadi orang yg sukses n kapan-kapan bisa kumpul-kumpul lagi, Thanks Brow...^_^v
- Geodesi Zero 7 yang gak bisa disebutin namanya satu-satu, Semangat bro...mengejar cita-cita kalian dan moga semua jadi orang-orang sukses.
- Tim Rapid Mapping : Terima kasih atas kerja samanya dan perjuangannnya.
- Ade dan Kakak Tingkat, terima kasih atas bantuan n kerja samanya.
- Makasih buat semuanya yang sudah mendukung sampai terselesaikannya skripsi ini, maaf kalau tidak dapat sebutin satu persatu, semua ku anggap berjasa buat ku.

KATA PENGANTAR

Dengan memanjatkan puji syukur kehadirat Allah S.W.T atas limpahan rahmat serta hidayah-Nya, sehingga dapat terselesaikannya penulisan skripsi ini dengan judul **“Pemrograman Kalibrasi Kamera IP Stereo dengan Menggunakan Prinsip Kolineariti untuk Papan Kalibrasi 2D dan 3D”**.

Skripsi ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknik Geodesi (S-1) di Jurusan Teknik Geodesi Fakultas Teknik Sipil dan Perencanaan Institut Teknologi Nasional Malang.

Dalam kesempatan ini pula, penulis mengucapkan terima kasih yang sebesar-besarnya atas dukungan dan bantuan kepada yang terhormat :

1. Bapak Ir. Soeparno Djiwo, MT selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. A. Agus Santosa, MT selaku Dekan Fakultas Teknik Sipil dan Perencanaan Institut Teknologi Nasional Malang.
3. Bapak Ir. Agus Darpomo, MT selaku Ketua Jurusan Teknik Geodesi Institut Teknologi Nasional Malang.
4. Bapak Dr. M. Edwin Tjahjadi, ST., M.Geom.Sc selaku Dosen Pembimbing I serta sebagai Dosen Penguji.
5. Bapak Ir. Leo Pantimena, M.Sc selaku Dosen Pembimbing II.
6. Bapak Ir. M. Nurhadi, MT selaku Dosen Penguji.
7. Bapak Ir. D. K. Sunaryo, MS.Tis selaku Dosen Penguji.
8. Segenap dosen, staff pengajar dan rekording Jurusan Teknik Geodesi Fakultas Teknik Sipil dan Perencanaan Institut Teknologi Nasional Malang.

9. Orangtua dan keluarga saya yang telah memberikan doa serta semangat baik moril maupun materil.
10. Rekan-rekan Mahasiswa/i dan alumni Teknik Geodesi.
11. Semua pihak yang langsung maupun tidak langsung turut membantu dalam proses penelitian maupun penyusunan laporan Tugas Akhir ini.

Penulis sadari bahwa masih banyak kekurangan dalam penyusunan skripsi ini, sehingga penulis sangat mengharapkan berbagai saran dan kritik dalam perbaikan skripsi ini.

Malang, September 2012

Penulis

DAFTAR ISI

Lembar Persetujuan	i
Lembar Pengesahan.....	ii
Abstraksi	iii
Surat Pernyataan Keaslian Skripsi	iv
Lembar Persembahan.....	v
Kata Pengantar.....	vi
Daftar Isi	viii
Daftar Gambar	xi
Daftar Tabel.....	xiv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Penelitian	1
1.2. Perumusan Penelitian	2
1.3. Tujuan Penelitian	3
1.4. Batasan Masalah.....	3
1.5. Tinjauan Pustaka	4
BAB II DASAR TEORI.....	6
2.1. Sistem Persamaan Dalam Fotogrametri	6
2.1.1. Sistem Persamaan Kolinearitas	6
2.1.2. Sistem Persamaan <i>Least Square</i> Adjustment.....	9
2.2. Ekstraksi Data Koordinat Foto.....	11
2.2.1. Metode Sentroid	11

2.2.2. Konversi Koordinat Piksel Ke Foto	12
2.3. Kalibrasi Kamera	13
2.4. Kalibrasi Kamera Non Metrik.....	15
2.4.1. Parameter x_0 , y_0 , dan Fokus (c)	16
2.4.2. Distorsi Radial.....	17
2.4.3. Distorsi <i>Decentring</i> (Distorsi tangensial).....	19
2.4.4. Distorsi <i>Affinity</i>	20
2.5. <i>Relative Orientation</i>	21
2.6. Stereo Kalibrasi	24
2.6.1. Rotasi.....	24
2.6.2. Translasi	27
2.7. <i>Bundle Adjustment</i>	28
2.7.1. <i>Self Calibrating Bundle Adjustment</i>	31
2.7.2. Model Penilaian <i>Bundle Adjustment</i>	33
BAB III METODOLOGI PENELITIAN	36
3.1. Persiapan Penelitian	36
3.1.1. Materi Penelitian	36
3.1.2. Alat Penelitian.....	37
3.2. Metode Penelitian.....	40
3.2.1. Diagram Alir Penelitian.....	41
3.2.2. Penjelasan Diagram Alir Penelitian.....	42
3.3. Pelaksanaan Penelitian	43
3.3.1. Pengambilan Data Foto Papan Kalibrasi.....	43
3.3.2. Pengolahan Data Foto Papan Kalibrasi.....	51

3.3.3. Pembuatan Program Stereo Kalibrasi	59
BAB IV HASIL DAN PEMBAHASAN.....	67
4.1. Hasil Penelitian	67
4.1.1. Aplikasi dan Listing Kode Pemrograman	67
4.1.2. Prosedur Pengoperasian Program <i>Matlab Stereo Calibration</i>	70
4.1.3. Hasil Perhitungan Kalibrasi Kamera IP Stereo	74
4.2. Pembahasan.....	78
4.2.1. Analisis Hasil Perhitungan Kalibrasi Kamera IP Stereo	79
4.2.2. Keunggulan dan Kelemahan Penggunaan <i>Matlab 2011b</i>	81
BAB V KESIMPULAN DAN SARAN	83
5.1. Kesimpulan	83
5.2. Saran.....	84

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1 Kondisi kolinearitas.....	6
Gambar 2.2 Sistem koordinat piksel vs sistem koordinat foto.....	12
Gambar 2.3 Geometri foto	16
Gambar 2.4 Penyebab distorsi <i>decentring</i> (Atkinson, 2001).....	19
Gambar 2.5 Ilustrasi akibat adanya distorsi lensa dan tidak ortogonalnya sumbu atau <i>affine deformation</i> , (Pullivelli, 2005)	21
Gambar 2.6 Relatif orientasi secara analitik	22
Gambar 2.7 Rotasi pada sumbu x sebesar ω	25
Gambar 2.8 Rotasi pada sumbu y sebesar ϕ	25
Gambar 2.9 Rotasi pada sumbu z sebesar κ	26
Gambar 2.10 Konversi dari objek ke sistem koordinat kamera; titik P pada objek terlihat sebagai titik p pada bidang foto (Bradski & Kaehler, 2008).....	27
Gambar 2.11 Struktur Persamaan Normal	30
Gambar 3.1 Kamera IP GV-BX320D 3M H.264.....	37
Gambar 3.2 Dome kamera	37
Gambar 3.3 HUB D-LINK 8 port	37
Gambar 3.4 Kabel Lan	38
Gambar 3.5 Inverter 700 Watt	38
Gambar 3.6 Papan/ plat kalbrasi 2D	38
Gambar 3.7 Papan/ plat kalbrasi 3D	39
Gambar 3.8 <i>Roof Rack</i> untuk tempat kamera IP	39

Gambar 3.9 Diagram alir penelitian.....	41
Gambar 3.10 <i>Window IP Device Setup</i>	45
Gambar 3.11 <i>Window Scan Camera</i>	45
Gambar 3.12 Kamera terdeteksi.....	45
Gambar 3.13 <i>Query</i> pada kamera.....	46
Gambar 3.14 <i>IP Device Setup</i>	46
Gambar 3.15 Mengubah Status Koneksi Kamera	46
Gambar 3.16 Memulai <i>Monitoring</i> Kamera.....	47
Gambar 3.17 Melihat Hasil <i>Monitoring</i>	47
Gambar 3.18 <i>GV-NVR system</i> untuk melihat dan menyimpan hasil monitoring	47
Gambar 3.19 <i>Save Video</i>	48
Gambar 3.20 <i>Window Save as Avi</i>	48
Gambar 3.21 <i>Window Save Avi File</i>	48
Gambar 3.22 <i>Window Merging List</i>	49
Gambar 3.23 Buka Program <i>Free Video to JPEG Converter</i>	49
Gambar 3.24 <i>Window Free Video to JPEG Converter</i>	50
Gambar 3.25 <i>Add Files Video</i>	50
Gambar 3.26 <i>Window Conversion Progress</i>	51
Gambar 3.27 Hasil Ekstraksi Video (*.Jpg).....	51
Gambar 3.28 Tampilan Australis 6.05	52
Gambar 3.29 <i>Define Project Units</i>	53
Gambar 3.30 <i>Australis Preferences</i>	53
Gambar 3.31 <i>Camera Input</i>	54
Gambar 3.32 <i>Project Dengan Id Camera Yang Sama</i>	55

Gambar 3.33 Penyimpanan <i>Project</i>	55
Gambar 3.34 Penginputan Data Foto	56
Gambar 3.35 Proses Ekstraksi Data Koordinat Foto	56
Gambar 3.36 Proses <i>Relative Orientation</i>	57
Gambar 3.37 <i>File Output Relative Orientation</i>	58
Gambar 3.38 Proses <i>Intersection</i>	58
Gambar 3.39 <i>File Output Intersection</i>	58
Gambar 3.40 Proses <i>Resection</i>	59
Gambar 3.41 Diagram Alir Program Stereo Kalibrasi (1)	60
Gambar 3.42 Diagram Alir Program Stereo Kalibrasi (2)	61
Gambar 4.1 Tampilan aplikasi perhitungan kalibrasi kamera IP stereo	70
Gambar 4.2 Proses penginputan data	71
Gambar 4.3 Susunan data input parameter EO awal.....	72
Gambar 4.4 Susunan data input parameter koordinat <i>object space</i> awal.....	72
Gambar 4.5 Susunan data input parameter koordinat foto masing-masing kamera	72
Gambar 4.6 Hasil keluaran dari proses perhitungan kalibrasi kamera IP stereo.	73
Gambar 4.7 Hasil penyimpanan perhitungan kalibrasi kamera IP stereo	74

DAFTAR TABEL

Tabel 4.1 Nama Fungsi, Input Data dan Output Data yang Telah Dibuat	68
Tabel 4.2 Parameter Kalibrasi Kamera (Papan 2D).....	74
Tabel 4.3 Parameter <i>exterior orientation</i> (Papan 2D).....	75
Tabel 4.4 Akurasi Parameter <i>exterior orientation</i> (Papan 2D).....	75
Tabel 4.5 Parameter Koordinat <i>Object Space Point</i> (Papan 2D).....	75
Tabel 4.6 Akurasi Koordinat <i>Object Space Point</i> (Papan 2D).....	76
Tabel 4.7 Parameter Kalibrasi Kamera (Papan 3D).....	76
Tabel 4.8 Parameter <i>exterior orientation</i> (Papan 3D).....	77
Tabel 4.9 Akurasi Parameter <i>exterior orientation</i> (Papan 3D).....	77
Tabel 4.10 Parameter Koordinat <i>Object Space Point</i> (Papan 3D).....	77
Tabel 4.11 Akurasi Koordinat <i>Object Space Point</i> (Papan 3D).....	78
Tabel 4.12 Hasil Selisih Parameter EO Awal dengan Parameter EO terkoreksi untuk Papan 2D	79
Tabel 4.13 Hasil Selisih Parameter Koordinat <i>Object Space Point</i> Awal dengan Parameter Koordinat <i>Object Space Point</i> Terkoreksi untuk Papan 2D ..	79
Tabel 4.14 Hasil Selisih Parameter EO Awal dengan Parameter EO terkoreksi untuk Papan 3D	80
Tabel 4.15 Hasil Selisih Parameter Koordinat <i>Object Space Point</i> Awal dengan Parameter Koordinat <i>Object Space Point</i> Terkoreksi untuk Papan 3D ..	81

BAB I

PENDAHULUAN

1.1. Latar Belakang Penelitian

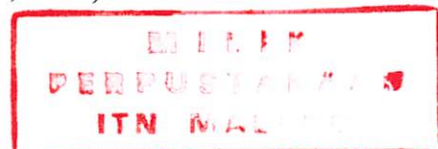
Pada era pembangunan dan informasi seperti sekarang ini, perkembangan teknologi dan dunia digital semakin pesat. Perkembangan tersebut ditandai oleh perkembangan sensor (kamera, *scanner*, hingga *hyperspectral*). Salah satu dari dampak perkembangan tersebut adalah pada bidang fotogrametri (*Hartono, 2004*).

Perkembangan di bidang fotogrametri sangatlah mempengaruhi dalam sistem pengukuran. Sistem pengukuran tersebut dapat dilakukan dengan cepat, akurat, efisien dan ekonomis. Salah satu sistem pengukuran tersebut adalah sistem pemetaan cepat jalan raya. Sistem ini terdiri sepasang kamera IP yang diintegrasikan dengan GPS dan ditempatkan pada wahana bergerak (*Tjahjadi, 2008*).

Sebelum melakukan pengambilan data di lapangan, kamera IP yang akan digunakan harus dikalibrasi terlebih dahulu agar mendapatkan data foto yang akurat. Kamera IP adalah jenis kamera non-metrik yang tidak stabil dan mempunyai lensa yang tidak sempurna, sehingga proses perekaman yang dilakukan akan memiliki kesalahan (*Hanifa, 2007*). Oleh karena itu, kamera IP perlu dilakukan kalibrasi menggunakan persamaan kolinear agar dapat menentukan besarnya penyimpangan-penyimpangan yang terjadi (*Fraser, 2000*).

Kalibrasi kamera dilakukan untuk menentukan parameter internal kamera (IO) yang meliputi *principal distance* (f), titik pusat *fidusial* foto (x_0 , y_0), distorsi lensa (K_1 , K_2 , K_3 , P_1 dan P_2), serta distorsi akibat perbedaan penyekalaan dan ketidak ortogonal antara sumbu X dan sumbu Y (b_1 , b_2) (Fraser, 2000).

Pada kamera non-metrik, titik pusat fidusial foto elemen dari *principle point* (x_0 , y_0) dan *principal distance* (f) harus ditentukan, hal ini dikarenakan semua sistem persamaan matematis yang digunakan dalam fotogrametri bergantung dari ketiga parameter ini (Wolf dan Dewitt, 2000). Distorsi lensa dapat menyebabkan bergesernya titik pada foto dari posisi yang sebenarnya, sehingga memberikan ketelitian pengukuran yang tidak baik, namun tidak mempengaruhi kualitas ketajaman citra yang dihasilkan. Distorsi lensa dapat dibagi menjadi distorsi radial (K_1 , K_2 , K_3) dan distorsi tangensial (P_1 dan P_2) dan distorsi *affinity* (b_1 , b_2) (Wolf dan Dewitt, 2000).



1.2. Perumusan Penelitian

Rumusan masalah yang akan dibahas dalam penelitian ini adalah bagaimana desain program kalibrasi kamera IP stereo dengan menggunakan prinsip kolineariti untuk menentukan parameter kalibrasi kamera yang meliputi internal kamera (IO) yaitu *principal distance* (f), titik pusat fidusial foto (x_0 , y_0), distorsi lensa (K_1 , K_2 , K_3 , P_1 dan P_2) dan distorsi *affinity* (b_1 , b_2), serta menentukan parameter EO (X_L , Y_L , Z_L , ω , ϕ , κ) yang terkoreksi sehingga didapat hasil koordinat *object space* (X_A , Y_A , Z_A) yang seakurat mungkin.

1.3. Tujuan Penelitian

Adapun maksud dari tujuan penelitian ini adalah sebagai berikut :

1. Membuat desain program menggunakan *software Matlab R2011b* untuk mengkalibrasi kamera IP stereo dengan menggunakan prinsip kolineariti.
2. Mengkalibrasi kamera IP stereo untuk dapat menentukan parameter internal kamera (IO), distorsi lensa (K_1 , K_2 , K_3 , P_1 dan P_2) dan distorsi *affinity* (b_1 , b_2) serta menentukan nilai parameter orientasi luar (EO) yang terkoreksi dari masing-masing kamera, yaitu tiga sudut rotasi (ω , ϕ , κ), dan koordinat posisi pemotretan (X_L , Y_L , Z_L) dan nilai parameter koordinat *object space* (X_A , Y_A , Z_A) yang terkoreksi.

1.4. Batasan Masalah

Adapun batasan-batasan masalah yang akan dibahas yaitu antara lain:

1. Kamera yang digunakan adalah kamera non-metrik yang terdiri dari dua buah kamera IP dengan resolusi yang sama.
2. Membuat desain program menggunakan *software Matlab R2011b* untuk kalibrasi kamera IP stereo dengan menggunakan persamaan kolineariti.
3. Mencari parameter kalibrasi kamera yaitu parameter internal kamera (IO), distorsi lensa (K_1 , K_2 , K_3 , P_1 dan P_2) dan distorsi *affinity* (b_1 , b_2) untuk masing-masing kamera IP.
4. Mencari nilai parameter EO (X_L , Y_L , Z_L , ω , ϕ , κ) yang terkoreksi untuk masing-masing kamera IP dan koordinat *object space* (X_A , Y_A , Z_A) yang terkoreksi.
5. Papan 2-Dimensi dan 3-Dimensi.

1.5. Tinjauan Pustaka

Dalam situasi tertentu dimana persyaratan akurasi dan anggaran rendah, fotogrametri dapat menggunakan kamera non-metrik untuk akuisisi citra. Kamera non-metrik dicirikan oleh jarak utama diatur, film tidak merata atau tanda fidusial (*fiducial marks*), dan lensa dengan distorsi relatif besar. Kalibrasi kamera non-metrik memungkinkan setidaknya kompensasi harus dibuat untuk kesalahan sistematis ini. Mungkin aspek yang paling sulit dari kalibrasi kamera non-metrik adalah berurusan dengan distorsi film. Pertama, menurut definisi, kamera non-metrik tidak memiliki fidusial yang dapat digunakan sebagai titik kontrol untuk mengoreksi distorsi film. Kedua, karena tidak ada mekanisme film merata, distorsi film bisa sangat bervariasi (*Wolf dan Dewiit, 2000*).

Di dalam bidang fotogrametri, proses kalibrasi kamera perlu dilakukan sebelum kamera tersebut digunakan. Menurut Ariff (2005), proses kalibrasi perlu dilakukan untuk menentukan dua hal dasar yang perlu diketahui terkait dengan kamera yang akan digunakan. Hal dasar tersebut adalah elemen-elemen orientasi internal (IO) dan parameter-parameter distorsi lensa. Untuk sebuah kamera metrik dan non-metrik, elemen-elemen orientasi internal yang dikatakan adalah terdiri dari koordinat titik utama (x_0, y_0) dan jarak fokus (f). Sedangkan parameter-parameter distorsi lensa terdiri dari dua jenis yaitu parameter distorsi radial yaitu K_1, K_2 dan K_3 dan parameter distorsi tangensial yaitu P_1 dan P_2 . Semua elemen dan parameter ini hanya tersedia dan diselesaikan melalui proses kalibrasi.

Menurut Wolf (1993), proses kalibrasi stereo dengan menggunakan prinsip kolinearitas merupakan suatu teknik yang umum digunakan untuk menghitung nilai-nilai parameter IO (Interior Orientasi). Persamaan dasar dari kondisi kolinearitas bersifat nonlinier dan dilinearkan dengan menggunakan *Teorema Taylor* dengan pendekatan awal bagi semua unsur orientasi luar yang tidak diketahui. Dua persamaan menunjukkan kondisi kolinearitas untuk setiap titik pada foto, satu persamaan untuk koordinat foto x dan persamaan yang lain untuk koordinat foto y (*Wolf dan Dewitt, 2000*).

BAB II

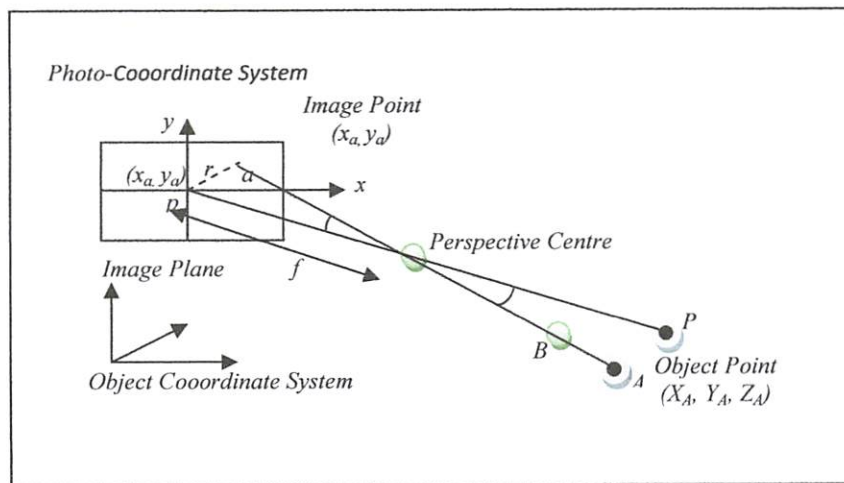
DASAR TEORI

2.1. Sistem Persamaan Dalam Fotogrametri

2.1.1. Sistem Persamaan Kolinearitas

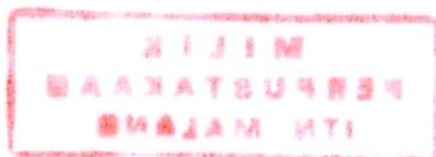


Kolinearitas merupakan kondisi dimana stasiun pemotretan foto, titik objek, dan gambar foto, semua terletak pada satu garis lurus. Persamaan yang menyatakan kondisi ini dinamakan persamaan kondisi kolinearitas (*Hanifa, 2007*). Kondisi kolinearitas diilustrasikan seperti gambar di bawah ini dimana A , o dan a terletak pada satu garis lurus :



Gambar 2.1 Kondisi kolinearitas

Dalam fotogrameteri sistem persamaan yang digunakan adalah sistem persamaan kolinear, yang merupakan suatu sistem persamaan yang bersifat nonlinier dan dilinierkan dengan menggunakan *Teorema Taylor*. Penggunaan *Teorema Taylor* untuk menyelesaikan kolinearitas memerlukan pendekatan awal bagi semua unsur orientasi luar yang tidak diketahui. Dua persamaan menunjukkan kondisi kolinearitas untuk



2.1. *Protein Termination - (Protein Termination)*

2.2. *Protein Termination - (Protein Termination)*

The first step in the process of protein termination is the binding of the ribosome to the mRNA. This is followed by the binding of the initiator tRNA to the start codon. The ribosome then moves along the mRNA, translating the code into a sequence of amino acids. This process continues until a stop codon is reached, at which point the protein is released from the ribosome.



Figure 1. Protein Termination

The second step in the process of protein termination is the binding of the ribosome to the mRNA. This is followed by the binding of the initiator tRNA to the start codon. The ribosome then moves along the mRNA, translating the code into a sequence of amino acids. This process continues until a stop codon is reached, at which point the protein is released from the ribosome.

setiap titik pada foto, satu persamaan untuk koordinat foto x dan persamaan yang lain untuk koordinat foto y (Wolf dan Dewitt, 2000).

$$x_a = x_0 - f \left[\frac{m_{11}(X_A - X_L) + m_{12}(Y_A - Y_L) + m_{13}(Z_A - Z_L)}{m_{31}(X_A - X_L) + m_{32}(Y_A - Y_L) + m_{33}(Z_A - Z_L)} \right] \dots\dots (2.1)$$

$$y_a = y_0 - f \left[\frac{m_{21}(X_A - X_L) + m_{22}(Y_A - Y_L) + m_{23}(Z_A - Z_L)}{m_{31}(X_A - X_L) + m_{32}(Y_A - Y_L) + m_{33}(Z_A - Z_L)} \right] \dots\dots (2.2)$$

Dimana :

x_0, y_0 : Koordinat foto titik a

x_a, y_a : Koordinat foto yang diukur

X_A, Y_A, Z_A : Koordinat *Object Space* untuk titik A

X_L, Y_L, Z_L : Koordinat stasiun pemotretan

f : Panjang fokus kamera

m : 3 sudut matrik rotasi ortogonal (ω, ϕ, κ)

Persamaan (2.1) dan (2.2) merupakan persamaan nonlinear, dalam melinearkan persamaan kolinear, persamaan (2.1) dan (2.2) dituliskan lagi sebagai berikut :

$$F = 0 = qx_a + rf \dots\dots\dots (2.3)$$

$$G = 0 = qy_a + sf \dots\dots\dots (2.4)$$

Dimana :

$$q = m_{31}(X_A - X_L) + m_{32}(Y_A - Y_L) + m_{33}(Z_A - Z_L)$$

$$r = m_{11}(X_A - X_L) + m_{12}(Y_A - Y_L) + m_{13}(Z_A - Z_L) \dots\dots\dots (2.5)$$

$$s = m_{21}(X_A - X_L) + m_{22}(Y_A - Y_L) + m_{23}(Z_A - Z_L)$$

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \dots\dots\dots (2.6)$$

Dimana :

$$\begin{aligned}
 m_{11} &= \cos \varphi \cos \kappa \\
 m_{12} &= \sin \omega \sin \varphi \cos \kappa + \cos \omega \sin \kappa \\
 m_{13} &= -\cos \omega \sin \varphi \cos \kappa + \sin \omega \sin \kappa \\
 m_{21} &= -\cos \varphi \sin \kappa \\
 m_{22} &= -\sin \omega \sin \varphi \sin \kappa + \cos \omega \cos \kappa \dots\dots\dots (2.7) \\
 m_{23} &= \cos \omega \sin \varphi \sin \kappa + \sin \omega \cos \kappa \\
 m_{31} &= \sin \varphi \\
 m_{32} &= -\sin \omega \cos \varphi \\
 m_{33} &= \cos \omega \cos \varphi
 \end{aligned}$$

Menurut teori *Taylor*, persamaan (2.3) dan (2.4) dapat dinyatakan dalam bentuk dilinearisasikan oleh turunan parsial sebagai berikut :

$$\begin{aligned}
 b_{11}d\omega + b_{12}d\varphi + b_{13}d\kappa - b_{14}dX_L - b_{15}dY_L - b_{16}dZ_L + b_{14}dX_A + \\
 b_{15}dY_A + b_{16}dZ_A = J + v_{xa} \dots\dots\dots (2.8)
 \end{aligned}$$

$$\begin{aligned}
 b_{21}d\omega + b_{22}d\varphi + b_{23}d\kappa - b_{24}dX_L - b_{25}dY_L - b_{26}dZ_L + b_{24}dX_A + \\
 b_{25}dY_A + b_{26}dZ_A = K + v_{ya} \dots\dots\dots (2.9)
 \end{aligned}$$

Pada persamaan (2.8) dan (2.9), *J* dan *K* adalah *F* dan *G* dari persamaan (2.3) dan (2.4), *b* merupakan koefisien yang sama dengan jabaran parsial.

$$\begin{aligned}
 b_{11} &= \frac{f}{q^2} [r(-m_{33}\Delta Y + m_{32}\Delta Z) - q(-m_{13}\Delta Y + m_{12}\Delta Z)] \\
 b_{12} &= \frac{f}{q^2} [r(\cos \varphi \Delta X + \sin \omega \sin \varphi \Delta Y - \cos \omega \sin \varphi \Delta Z) - \\
 &\quad q(-\sin \varphi \cos \kappa \Delta X + \\
 &\quad \sin \omega \cos \varphi \cos \kappa \Delta Y - \cos \omega \cos \varphi \cos \kappa \Delta Z)]
 \end{aligned}$$

$$\begin{aligned}
b_{13} &= \frac{-f}{q} (m_{21}\Delta X + m_{22}\Delta Y + m_{23}\Delta Z) \\
b_{14} &= \frac{f}{q^2} (rm_{31} - qm_{11}) \\
b_{15} &= \frac{f}{q^2} (rm_{32} - qm_{12}) \\
b_{16} &= \frac{f}{q^2} (rm_{33} - qm_{13}) \\
b_{21} &= \frac{f}{q^2} [s(-m_{33}\Delta Y + m_{32}\Delta Z) - q(-m_{23}\Delta Y + m_{22}\Delta Z)] \\
b_{22} &= \frac{f}{q^2} [s(\cos \varphi \Delta X + \sin \omega \sin \varphi \Delta Y - \cos \omega \sin \varphi \Delta Z) - \\
&\quad q(\sin \varphi \cos \kappa \Delta X - \\
&\quad \sin \omega \cos \varphi \sin \kappa \Delta Y + \cos \omega \cos \varphi \sin \kappa \Delta Z)] \\
b_{23} &= \frac{f}{q} (m_{11}\Delta X + m_{12}\Delta Y + m_{13}\Delta Z) \\
b_{24} &= \frac{f}{q^2} (sm_{31} - qm_{21}) \dots\dots\dots (2.10) \\
b_{25} &= \frac{f}{q^2} (sm_{32} - qm_{22}) \\
b_{26} &= \frac{f}{q^2} (sm_{33} - qm_{23}) \\
J &= x_a - x_o + f \frac{r}{q} \\
K &= y_a - y_o + f \frac{s}{q}
\end{aligned}$$

2.1.2. Sistem Persamaan *Least Square Adjustment*

Untuk menerapkan persamaan (2.8) dan (2.9) dalam bentuk sederhana dari persamaan kolinearitas, pendekatan aljabar dalam bentuk matriks dan dituliskan sebagai berikut :

$${}_m V_1 = {}_m A_{n \times n} X_1 - {}_m L_1 \dots\dots\dots (2.11)$$

Dimana m adalah jumlah persamaan, n adalah jumlah unsur yang tidak diketahui, V merupakan matriks kesalahan residual dalam koordinat foto x dan y terukur, A ialah matriks b , koefisien unsur yang tidak diketahui, X ialah matriks koreksi unsur tidak diketahui untuk perkiraan awal, dan L ialah matriks tetapan J dan K .

Dengan memperoleh penyajian matrik penyelesaian kuadrat terkecil bagi unsur yang tidak diketahui, akan terlihat bahwa persamaan normal dapat diperoleh dengan sebagai berikut :

$$A^TAX = A^TL \dots\dots\dots (2.12)$$

Menjadi

$$X = (A^TA)^{-1} A^TL \dots\dots\dots (2.13)$$

Bagi suatu sistem pengamatan berbobot :

$$X = (A^TPA)^{-1} A^TPL \dots\dots\dots (2.14)$$

Didalam persamaan matriks identik terhadap persamaan bobot, kecuali bahwa matriks P merupakan matriks diagonal bobot. Melakukan proses iterasi apabila nilai residu belum sesuai. Adapun persamaan matrik untuk menghitung nilai residu setelah penyesuaian, sebagai berikut :

$$V = AX - L \dots\dots\dots (2.15)$$

Rumus standar deviasinya adalah :

$$So = \frac{\sqrt{V^TV}}{r} \dots\dots\dots (2.16)$$

Iterasi berhenti apabila besarnya nilai koreksi parameter dicari (*unknown*) yang didapat paling kecil.

2.2. Ekstraksi Data Koordinat Foto

Suatu foto dalam format digital merupakan kuantitas nilai-nilai tingkat keabuan (*greyscale*) yang ditampilkan dalam sebuah susunan matriks atau *array*, dimana nilai baris dan kolom dari matriks tersebut merupakan koordinat piksel. Dengan kelebihan yang dimiliki oleh foto dalam format digital ini, maka dapat dengan mudah menentukan nilai suatu koordinat objek dalam suatu sistem koordinat foto. Secara umum metode penentuan nilai koordinat objek pada foto digital yang sering digunakan dalam proses fotogrametri antara lain sebagai berikut :

2.2.1. Metode Sentroid

Dalam penentuan koordinat foto menggunakan teknik ini, hanya terbatas pada target atau objek yang berbentuk lingkaran atau elips. Hal ini dikarenakan, pada metode ini bertujuan menghitung nilai tengah (*center of gravity*) atau pusat *centroid* pada sebuah target dengan memperhitungkan nilai tingkat keabuan pada tiap piksel. Seperti yang dikemukakan oleh *Ganci* dan *Shortis* dalam *Ahmad*, metode ini dapat digunakan untuk menentukan sentroid hingga ketepatan $\pm 0,03$ piksel.

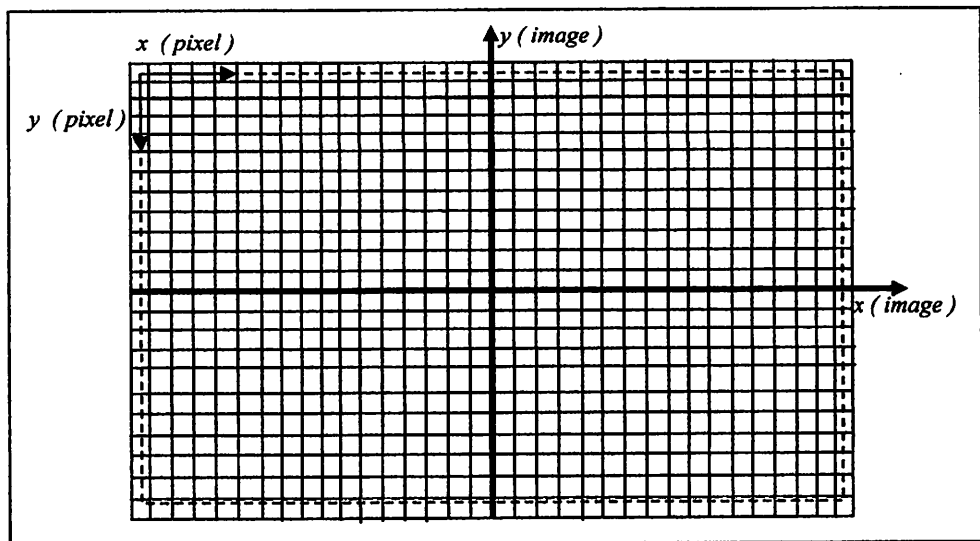
Untuk lebih jelasnya persamaan penentuan *centroid* secara umum dijabarkan sebagai berikut (*Shortis, et al, 1994*):

$$x = \frac{\sum_{j=1}^n \sum_{i=1}^m j \times l_{ij}}{\sum_{j=1}^n \sum_{i=1}^m l_{ij}}, y = \frac{\sum_{j=1}^n \sum_{i=1}^m i \times l_{ij}}{\sum_{j=1}^n \sum_{i=1}^m l_{ij}} \dots\dots\dots (2.17)$$

Dimana l_{ij} merupakan nilai tingkat keabuan (*greyscale*) piksel dalam arah x dan y , m jumlah kolom, n jumlah baris dan masing-masing i dan j adalah nilai baris dan kolom piksel.

2.2.2. Konversi Koordinat Piksel Ke Foto

Pada kamera digital sistem koordinat yang dipakai adalah sistem koordinat piksel, sedangkan dalam proses perhitungan secara analitik, sistem yang dipakai adalah sistem koordinat kartesian (metrik). Sehingga dalam hal ini harus dilakukan transformasi koordinat dari sistem piksel kedalam sistem kartesian foto. Adapun persamaan yang digunakan adalah (*Australis, User Manual, 2004*).



Gambar 2.2 Sistem koordinat piksel vs sistem koordinat foto

$$\begin{aligned} X &= (X' - X_c') \times X_{PixelSize} \\ Y &= (Y_c' - Y') \times Y_{PixelSize} \dots\dots\dots (2.18) \end{aligned}$$

Dimana :

$$X_c' = \left(\frac{nx'}{2} \right) - 0,5$$

$$y_c' = \left(\frac{ny'}{2}\right) - 0,5 \dots\dots\dots (2.19)$$

Dalam hal ini x, y merupakan koordinat foto dalam sistem koordinat metrik, x', y' koordinat dalam piksel, x_c, y_c , *principle point* dalam piksel, $x_{PixelSize}, y_{PixelSize}$ ukuran satu piksel dalam metrik dan nx', ny' merupakan resolusi dari foto dalam piksel.

2.3. Kalibrasi Kamera

Fotogrametri merupakan teknik untuk memperoleh informasi mengenai posisi, ukuran dan bentuk objek melalui suatu pengukuran foto sebagai pengganti pengukuran yang dilakukan secara langsung (*Atkinson, 2001*). Istilah fotogrametri berasal dari kata *photos* (sinar), *gramma* (sesuatu yang tergambar) dan *metron* (mengukur). Secara sederhana maka fotogrametri dapat diartikan sebagai pengukuran secara grafis dengan menggunakan sinar. Dari definisi tersebut dapat dimengerti bahwa fotogrametri meliputi (*Wolf dan Dewitt, 2000*) :

- Perekaman objek (pemotretan).
- Pengukuran gambar objek pada foto udara.
- Pemotretan hasil ukuran untuk dijadikan bentuk yang bermanfaat (peta).

Salah satu karakteristik fotogrametri adalah pengukuran terhadap objek yang dilakukan tanpa berhubungan, perlu berhubungan, ataupun bersentuhan secara langsung dengannya. Pengukuran terhadap objek tersebut dilakukan melalui data yang diperoleh pada sistem sensor yang digunakan.

Dalam bidang fotogrametri, terdapat proses kalibrasi kamera yang perlu dilakukan sebelum kamera tersebut digunakan. Menurut Fraser (2000), kalibrasi kamera dilakukan untuk menentukan parameter internal kamera (IO) yang meliputi *principle distance* (f), titik pusat fidusial foto (x_0, y_0), distorsi lensa (K_1, K_2, K_3, P_1 dan P_2), serta distorsi akibat perbedaan penyekalaan dan ketidak ortogonal antara sumbu X dan Y (b_1, b_2).

Kalibrasi kamera dapat dilakukan dengan berbagai metode. Secara umum kalibrasi kamera biasa dilakukan dengan tiga metode, yaitu *laboratory calibration*, *on the job calibration* dan *self calibration* (Atkinson, 1987). *Laboratory calibration* dilakukan di laboratorium, terpisah dengan proses pemotretan objek. Metode yang termasuk di dalamnya antara lain *optical laboratory* dan *test range calibration*. Secara umum metode ini sesuai untuk kamera jenis metrik. *On the job calibration* merupakan teknik penentuan parameter kalibrasi lensa dan kamera dilakukan bersamaan dengan pelaksanaan pemotretan objek. Pada *self calibration* pengukuran titik-titik target pada objek pengamatan digunakan sebagai data untuk penentuan titik objek sekaligus untuk menentukan parameter kalibrasi kamera.

Dalam penelitian ini kalibrasi kamera akan dilakukan dengan menggunakan metode *self calibrating bundle adjustment* yang berdasarkan pada persamaan kolinear (Fraser, 1995) :

$$x - x_0 + \Delta x = -c \frac{R_1}{R_3}$$

$$y - y_0 + \Delta y = -c \frac{R_2}{R_3} \dots\dots\dots (2.20)$$

Dimana :



$$\begin{aligned}
R_3 &= m_{31}(X_A - X_L) + m_{32}(Y_A - Y_L) + m_{33}(Z_A - Z_L) \\
R_2 &= m_{11}(X_A - X_L) + m_{12}(Y_A - Y_L) + m_{13}(Z_A - Z_L) \dots\dots\dots (2.21) \\
R_1 &= m_{21}(X_A - X_L) + m_{22}(Y_A - Y_L) + m_{23}(Z_A - Z_L)
\end{aligned}$$

Dimana :

$$\begin{aligned}
\Delta x &= -x_0 - \frac{x}{c}\Delta c + \bar{x}r^2k_1 + \bar{x}r^4k_2 + \bar{x}r^6k_3 + (r^2 + 2\bar{x}^2)p_1 + \\
&2p_2\bar{x}\bar{y} + b_1\bar{x} + b_2\bar{y} \dots\dots\dots (2.22) \\
\Delta y &= -y_0 - \frac{y}{c}\Delta c + \bar{y}r^2k_1 + \bar{y}r^4k_2 + \bar{y}r^6k_3 + 2p_1\bar{x}\bar{y} + (r^2 + 2\bar{y}^2)p_2
\end{aligned}$$

Dimana :

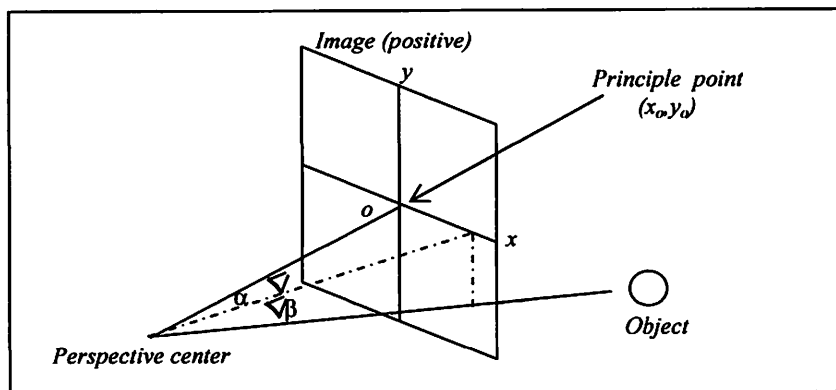
$$\begin{aligned}
r &= \sqrt{\bar{x}^2 + \bar{y}^2} \\
\bar{x} &= x - x_0 \dots\dots\dots (2.23) \\
\bar{y} &= y - y_0
\end{aligned}$$

2.4. Kalibrasi Kamera Non Metrik

Kamera non-metrik tidak mempunyai lensa yang sempurna, sehingga proses perekaman yang dilakukan akan memiliki kesalahan (*Fraser, 2000*). Oleh karena itu perlu dilakukan kalibrasi kamera untuk dapat menentukan besarnya penyimpangan-penyimpangan yang terjadi. Kalibrasi kamera ditujukan untuk memodelkan dan menentukan nilai distorsi dan konstanta sistem optik yang ada pada kamera. Kalibrasi kamera dilakukan untuk menentukan parameter distorsi, meliputi distorsi radial dan distorsi tangensial (*decentring*), serta parameter-parameter lensa lainnya, termasuk juga *principal distance* (*c*), serta titik pusat fidusial foto.

2.4.1. Parameter x_o, y_o , dan Fokus (c)

Dalam berbagai kasus fotogrametri, elemen dari *principle point* (x_o, y_o) dan *perspektif distance* (panjang fokus) harus ditentukan, hal ini dikarenakan semua sistem persamaan matematis yang digunakan dalam fotogrametri bergantung dari ketiga parameter ini. Secara geometris hubungan antara ketiga parameter ini dapat di lihat pada gambar dibawah ini.



Gambar 2.3 Geometri foto

Dari gambar diatas posisi *principle point* (x_o, y_o) merupakan proyeksi garis lurus dari letak *perspective center* ke bidang foto dan jarak dari *principle point* ke *perspective center* merupakan panjang fokus (c). Secara praktis panjang fokus kamera dan letak *principle point* tidak mutlak berada di tengah-tengah pusat foto, permasalahan ini disebabkan oleh kurang stabilnya susunan lensa dan CCD yang berguna untuk merekam bayangan objek pada saat perakitan. Sehingga perubahan posisi *principle point* dan panjang fokus dapat dimodelkan menggunakan persamaan sebagai berikut (Dorstel, 2004) :

$$\begin{aligned}\Delta x_i &= \Delta x_p - \frac{\bar{x}}{z} \Delta f \\ \Delta y_i &= \Delta y_p - \frac{\bar{y}}{z} \Delta f\end{aligned} \dots\dots\dots (2.24)$$

Dimana $\Delta x_i, \Delta y_i$ merupakan total koreksi dari parameter (x_0, y_0) dan fokus, $\Delta x_p, \Delta y_p$ koreksi untuk parameter *principle point* dan Δf koreksi untuk nilai parameter fokus, dengan nilai koordinat foto didefinisikan sebagai berikut (Dorstel, 2004) :

$$\begin{aligned}\bar{x} &= x - x_p \\ \bar{y} &= y - y_p \\ \bar{z} &= -c\end{aligned} \dots\dots\dots (2.25)$$

2.4.2. Distorsi Radial

Distorsi radial menyebabkan semua bagian gambar diubah letaknya menurut jari-jari, bermula dari sumbu optik. Keadaan ini disebabkan oleh kesalahan dalam pengasahan bagian-bagian lensa. Nilai distorsi radial merupakan perpidahan secara radial suatu titik dari posisi sebenarnya terhadap posisi dari *principle point* (x_0, y_0) . dengan indikator apabila nilainya positif maka pergeserannya mengarah keluar dan jika nilainya negatif maka pergeserannya mengarah kedalam (Wolf dan Dewitt, 2000).

Secara teoritik koreksi radial dilakukan setelah dilakukan reduksi gambar ke titik utama dan koreksi oleh kesalahan pengkerutan atau pemekaran. Jumlah pengkerutan dan pemekaran pada foto ditentukan di

dalam kalibrasi kamera. Sehingga koordinat foto dapat dikoreksi. Bila x_m dan y_c merupakan ukuran fidusial pada positif, sedangkan x_c dan y_c merupakan jarak fidusial terkalibrasi, maka koordinat foto pada titik “a” dapat dihitung sebagai berikut (Wolf dan Dewitt, 2000) :

$$\begin{aligned}x'_a &= \left(\frac{x_c}{x_m}\right) x_a \\y'_a &= \left(\frac{y_c}{y_m}\right) y_a \dots\dots\dots (2.26)\end{aligned}$$

Dimana x_a' dan y_a' adalah koordinat foto terkoreksi dan x_a dan y_a merupakan koordinat terukur. Persamaan polinomial berdasarkan pada teori desain kamera adalah sebagai berikut (Wolf dan Dewitt, 2000) :

$$\begin{aligned}\Delta x_r &= k_1 r^3 + k_2 r^5 + k_3 r^7 \\ \Delta y_r &= k_1 r^3 + k_2 r^5 + k_3 r^7 \dots\dots\dots (2.27)\end{aligned}$$

Atau

$$d_r = k_1 r^3 + k_2 r^5 + k_3 r^7 + \dots\dots\dots (2.28)$$

Dimana istilah dari K_1 , K_2 , dan K_3 merupakan koefisien dari distorsi radial lensa dan r merupakan jarak radial dari pusat foto terkalibrasi yang didapat dari (Wolf dan Dewitt, 2000) :

$$r = \sqrt{\bar{x}^2 + \bar{y}^2} \dots\dots\dots (2.29)$$

Efek distorsi radial adalah sekitar 1 sampai 2 piksel di perbatasan CCD sensor. Dalam kaitannya dengan definisi distorsi radial, ada korelasi besar antara koefisien distorsi itu sendiri K_1 , K_2 , K_3 dan antara *principle distance*. Hubungan antara distorsi radial dengan *principle distance* adalah dalam kaitannya sesuai *principle distance* dengan deviasi rata-rata akan dihitung menjadi minimum.

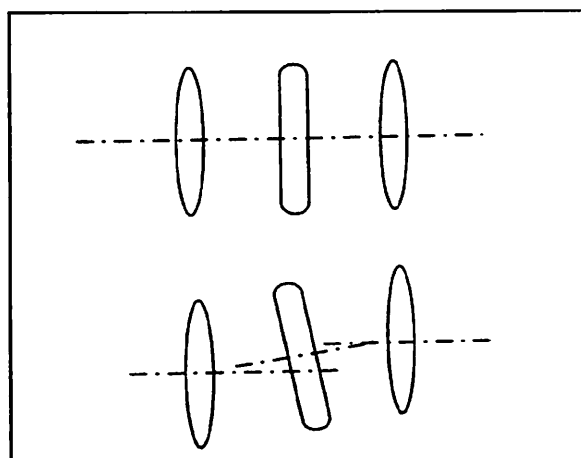
2.4.3. Distorsi *Decentring* (Distorsi tangensial)

Distorsi tangensial atau distorsi *decentring* adalah pergeseran linier titik di foto pada arah normal (tegak lurus) garis radial melalui titik foto tersebut. Distorsi tangensial disebabkan kesalahan *centering* elemen-elemen lensa dalam satu gabungan lensa dimana titik pusat elemen-elemen lensa dalam gabungan lensa tersebut tidak terletak pada satu garis lurus. Pergeseran ini biasa dideskripsikan dengan 2 persamaan polonomial untuk pergeseran pada arah x (dx) dan y (dy) (Atkinson, 2001).

$$d_x = P_1 [r^2 + 2(x - x_0)^2] + 2P_2(x - x_0)(y - y_0)$$

$$d_y = P_2 [r^2 + 2(y - y_0)^2] + 2P_1(x - x_0)(y - y_0) \dots\dots\dots (2.30)$$

Dimana P_1 dan P_2 merupakan koefesien dari parameter distorsi *decentring* yang nilainya tergantung dari nilai panjang fokus kamera. Efek dari distorsi *decentring* ini akan menyebabkan kesan hiperbolik pada foto yang terekam oleh kamera.



Gambar 2.4 Penyebab distorsi *decentring* (Atkinson, 2001)

2.4.4. Distorsi Affinity

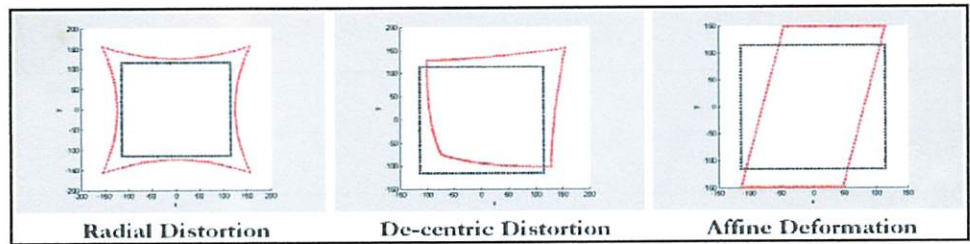
Parameter distorsi *affinity* secara umum dapat dimodelkan dalam bentuk persamaan polinomial dan gabungan dari koreksi koordinat foto (Fraser, 1995).

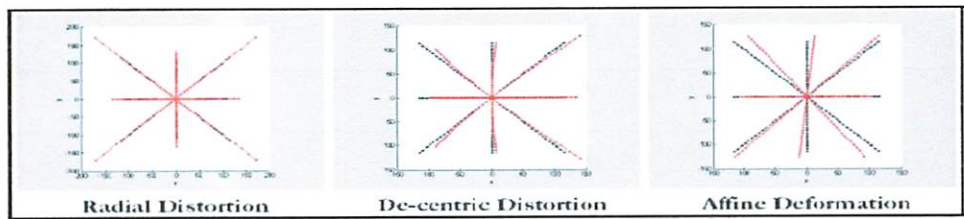
$$\begin{Bmatrix} \Delta x_u \\ \Delta y_u \end{Bmatrix} \begin{Bmatrix} \bar{x}/r \\ \bar{y}/r \end{Bmatrix} \sum_{i=0}^n \sum_{j=0}^i a_{ij} \bar{x}^{(i-j)} \bar{y}^{(j)} \dots\dots\dots (2.31)$$

Dari persamaan (2.31) dapat digunakan untuk memodelkan bidang yang tidak rata, dimana model koreksi pada bidang distorsi dapat dikurangi menjadi dua aturan hanya untuk koordinat *x*, satu untuk menghitung perbedaan skala antar selang piksel horisontal dan vertikal, dan satu untuk memodelkan *non-ortogonalitas* antara sumbu *x* dan sumbu *y* (Fraser, 1995) :

$$\begin{matrix} \Delta x_f = b_1x + b_2y \\ \Delta y_f = 0 \end{matrix} \dots\dots\dots (2.32)$$

Distorsi *affinity* ini terjadi akibat kurang sikunya bidang CCD atau CMOS yang digunakan untuk merekam bayangan objek, sehingga frame dari foto tidak akan benar-benar berbentuk sebuah bujur sangkar ataupun persegi panjang akan tetapi akan membentuk sebuah jajaran genjang.



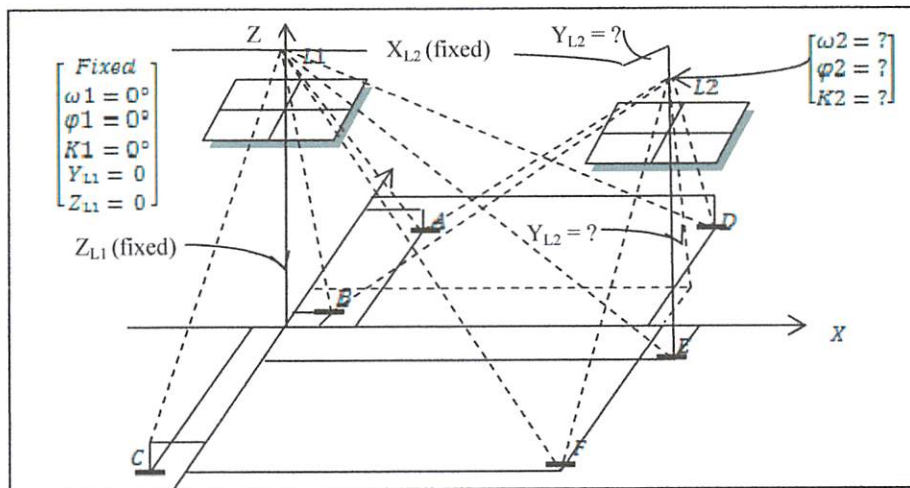


Gambar 2.5 Ilustrasi akibat adanya distorsi lensa dan tidak ortogonalnya sumbu atau *affine deformation*, (Pullivelli, 2005)

2.5. Relative Orientation

Relatif orientasi merupakan proses untuk menentukan nilai perputaran sudut rotasi dan pergeseran posisi antar foto yang diambil (*Wolf dan Dewitt, 2000*). Proses ini dilakukan dengan cara memberikan nilai posisi dan orientasi untuk foto pertama, kemudian dilakukan proses perhitungan nilai posisi dan orientasi pada foto kedua menggunakan parameter dari posisi kamera pertama dan koordinat foto dari kedua buah foto. Dalam proses relatif orientasi ini tidak menghasilkan nilai posisi dan orientasi dari foto yang sebenarnya, akan tetapi menghasilkan sebuah nilai relatif antara dua buah foto tersebut. Yaitu menetapkan beberapa parameter eksterior orientasi (EO) ω , φ , k , Y_L , Z_L dari foto kanan (foto kedua) dari pertemuan 5 berkas sinar dari koordinat obyek 3D (X_i , Y_i , Z_i) yang ada (*Wolf dan Dewitt, 2000*).

Dengan cara dijital, relatif orientasi dapat menggunakan syarat kesegaran (*colinearity condition*) atau syarat kesebidangan (*coplanarity condition*). Dimana kondisi kesegaran antar foto dapat dilukiskan seperti pada gambar dibawah ini :



Gambar 2.6 Relatif orientasi secara analitik

Proses dari penentuan relatif orientasi dapat dilukiskan dalam gambar (2.6), sinar-sinar yang berkaitan dengan enam titik dari A hingga F tampak memenuhi kondisi tersebut. Keenam buah titik tersebut pada dasarnya terletak pada bagian daerah yang sama pada kedua foto (Wolf dan Dewitt, 2000).

Sehingga dapat dituliskan persamaan-persamaan kebersamaan garis untuk kedua foto, dan minimal untuk lima buah titik objek. Persamaan dari kedua foto tersebut mengandung koordinat keruangan yang sama dan sistem persamaan kebersamaan garis yang dirumuskan terdapat lima buah parameter orientasi luar foto kanan (foto kedua) yaitu ω_2 , ϕ_2 , κ_2 , Y_{L2} dan Z_{L2} yang belum diketahui dan ditambah bentuk 3D koordinat objek yang belum diketahui (X_i , Y_i , Z_i) untuk masing-masing titik yang digunakan dalam pemecahan masalah sehingga parameter orientasi luar yang diperoleh nantinya akan dikoreksi pada relatif antara kedua foto (Wolf dan Dewitt, 2000).

Pada relatif orientasi analitik, biasanya parameter EO (ω , ϕ , κ , X_L , Y_L) dari foto kiri sama dengan nol. Dan juga untuk Z_L pada foto kiri (Z_{L1}) ditetapkan secara sembarang pada harga bulat dan sebagai alternatif yang

nyaman maka nilai dari Z_{L1} tepat pada angka nol, dan X_L pada foto kanan (X_{L2}) ditetapkan pada harga mendekati basis foto (jarak difoto pada kedua foto) yang mendekati nol dan harus ditentukan 5 parameter *unknown* pada foto kanan. Hal ini akan mempermudah dalam perhitungan koordinat objek X_i , Y_i , Z_i sehingga mendekati satuan koordinat foto yang terukur.

Metode yang digunakan sebagai solusi untuk mendapatkan parameter yang dicari adalah menggunakan teknik kuadrat terkecil (*Wolf dan Dewitt, 2000*).

$$b_{11}d\omega + b_{12}d\varphi + b_{13}dK - b_{14}dX_L - b_{15}dY_L - b_{16}dZ_L + b_{14}dX_A + b_{15}dY_A + b_{16}dZ_A = J + v_{xa}$$

$$b_{21}d\omega + b_{22}d\varphi + b_{23}dK - b_{24}dX_L - b_{25}dY_L - b_{26}dZ_L + b_{24}dX_A + b_{25}dY_A + b_{26}dZ_A = K + v_{ya} \dots \dots \dots (2.33)$$

Bentuk matriks A yaitu :

$$A = \begin{bmatrix} (b_{a11})_1 & (b_{a12})_1 & (b_{a13})_1 & (-b_{a15})_1 & (-b_{a16})_1 & (b_{a14})_1 & (b_{a15})_1 & (b_{a16})_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ (b_{a21})_1 & (b_{a22})_1 & (b_{a23})_1 & (-b_{a25})_1 & (-b_{a26})_1 & (b_{a24})_1 & (b_{a25})_1 & (b_{a26})_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & 0 & 0 & 0 & \ddots & \ddots & \ddots & 0 & 0 & 0 \\ (b_{n11})_1 & (b_{n12})_1 & (b_{n13})_1 & (-b_{n15})_1 & (-b_{n16})_1 & 0 & 0 & 0 & \ddots & \ddots & \ddots & (b_{n14})_1 & (b_{n15})_1 & (b_{n16})_1 \\ (b_{n21})_1 & (b_{n22})_1 & (b_{n23})_1 & (-b_{n25})_1 & (-b_{n26})_1 & 0 & 0 & 0 & 0 & 0 & 0 & (b_{n24})_1 & (b_{n25})_1 & (b_{n26})_1 \\ (b_{a11})_2 & (b_{a12})_2 & (b_{a13})_2 & (-b_{a15})_2 & (-b_{a16})_2 & (b_{a14})_2 & (b_{a15})_2 & (b_{a16})_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ (b_{a21})_2 & (b_{a22})_2 & (b_{a23})_2 & (-b_{a25})_2 & (-b_{a26})_2 & (b_{a24})_2 & (b_{a25})_2 & (b_{a26})_2 & \ddots & \ddots & \ddots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & 0 & 0 & 0 & \ddots & \ddots & \ddots & 0 & 0 & 0 \\ (b_{n11})_2 & (b_{n12})_2 & (b_{n13})_2 & (-b_{n15})_2 & (-b_{n16})_2 & 0 & 0 & 0 & 0 & 0 & 0 & (b_{n14})_2 & (b_{n15})_2 & (b_{n16})_2 \\ (b_{n21})_2 & (b_{n22})_2 & (b_{n23})_2 & (-b_{n25})_2 & (-b_{n26})_2 & 0 & 0 & 0 & \vdots & \vdots & \vdots & (b_{n24})_2 & (b_{n25})_2 & (b_{n26})_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Bentuk matriks X , L , dan V sebagai berikut :

$$X = \begin{bmatrix} d\omega_2 \\ d\phi_2 \\ d\kappa_2 \\ dY_2 \\ dZ_2 \\ dX_A \\ dY_A \\ dZ_A \\ \vdots \\ \vdots \\ dX_n \\ dY_n \\ dZ_n \end{bmatrix} \quad L = \begin{bmatrix} (J_a)_1 \\ (K_a)_1 \\ \vdots \\ \vdots \\ (J_n)_1 \\ (K_n)_1 \\ (J_a)_2 \\ (K_a)_2 \\ \vdots \\ \vdots \\ (J_n)_2 \\ (K_n)_2 \end{bmatrix} \quad V = \begin{bmatrix} (V_{xa})_1 \\ (V_{ya})_1 \\ \vdots \\ \vdots \\ (V_{xn})_1 \\ (V_{yn})_1 \\ (V_{xa})_2 \\ (V_{ya})_2 \\ \vdots \\ \vdots \\ (V_{xn})_2 \\ (V_{yn})_2 \end{bmatrix}$$

2.6. Stereo Kalibrasi

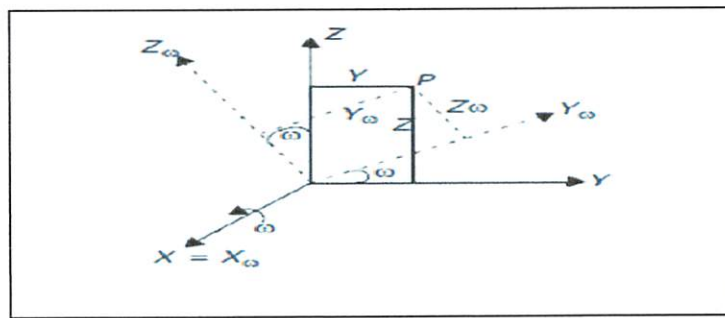
Stereo kalibrasi adalah proses menghitung hubungan geometris antara dua kamera (*Bradski & Keahler, 2008*). Geometri dari dua kamera ini berkaitan dengan nilai informasi parameter eksterinsik dan parameter instrinsik untuk tiap kamera. Parameter instrinsik biasanya dipakai sebagai parameter untuk mendefinisikan nilai geometri dari kamera, sedangkan parameter ekstrensik digunakan untuk menjelaskan hubungan secara geometri posisi kamera dan objek dalam suatu sistem koordinat referensi. Parameter ekstrensik terdiri dari rotasi dan translasi.

2.6.1. Rotasi

Parameter posisi kamera didefinisikan dalam ruang tiga-dimensi yang merupakan posisi dari *perspektif center* (pusat kamera) X_0, Y_0, Z_0 . Sedangkan parameter orientasi sudut berfungsi sebagai penghubung antara sistem koordinat kamera (x, y, z) dengan sistem koordinat referensi (X, Y, Z) . Sesuai dengan (*Mikhail et al, 2001; Wolf & Dewitt,*

2000) orientasi sudut dapat didefinisikan dalam sistem rotasi ω , φ , κ (ω , ϕ , κ).

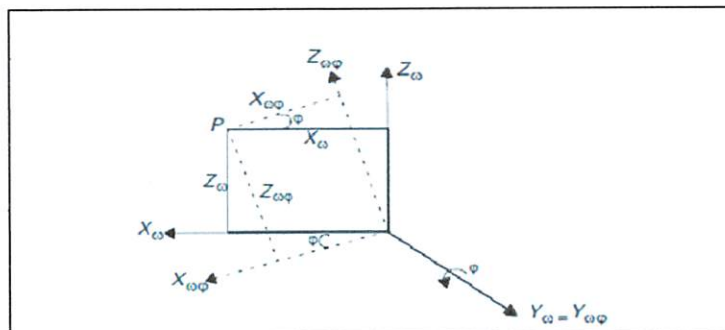
Dari masing-masing sistem rotasi, dapat dibangun sebuah matrik rotasi yang digunakan untuk menghubungkan kedua sistem koordinat. Sebagai contoh, akan dijelaskan proses penurunan sebuah persamaan pada matrik rotasi dalam sudut ω untuk rotasi x , ϕ untuk rotasi sumbu y , dan κ sebagai rotasi sumbu z . Dari ketiga sudut diatas didefinisikan dalam sistem rotasi tangan kanan, dimana apabila berlawanan dengan arah jarum jam akan bernilai positif dan sebaliknya.



Gambar 2.7 Rotasi pada sumbu x sebesar ω

Dari gambar diatas di dapat persamaan sebagai berikut :

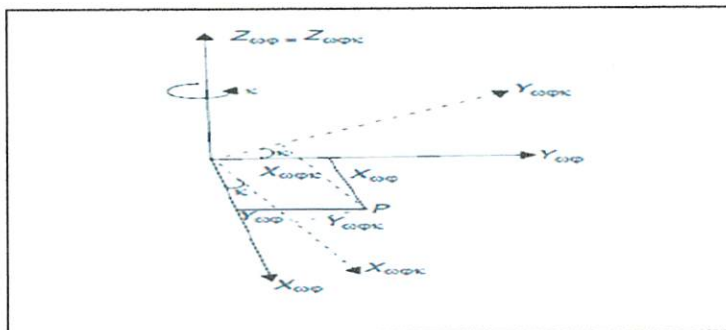
$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \omega & -\sin \omega \\ 0 & \sin \omega & \cos \omega \end{bmatrix} \begin{bmatrix} X_{\omega} \\ Y_{\omega} \\ Z_{\omega} \end{bmatrix} = M_{\omega} \begin{bmatrix} X_{\omega} \\ Y_{\omega} \\ Z_{\omega} \end{bmatrix} \dots\dots\dots (2.34)$$



Gambar 2.8 Rotasi pada sumbu y sebesar φ

Dari sistem rotasi kedua terhadap sumbu y didapat persamaan sebagai berikut :

$$\begin{bmatrix} X_{\omega} \\ Y_{\omega} \\ Z_{\omega} \end{bmatrix} = \begin{bmatrix} \cos \varphi & 0 & \sin \varphi \\ 0 & 1 & 0 \\ -\sin \varphi & 0 & \cos \varphi \end{bmatrix} \begin{bmatrix} X_{\omega\varphi} \\ Y_{\omega\varphi} \\ Z_{\omega\varphi} \end{bmatrix} = M_{\varphi} \begin{bmatrix} X_{\omega\varphi} \\ Y_{\omega\varphi} \\ Z_{\omega\varphi} \end{bmatrix} \dots\dots\dots (2.35)$$



Gambar 2.9 Rotasi pada sumbu z sebesar κ

Dari gambar diatas sistem rotasi terhadap sumbu z didapat persamaan sebagai berikut :

$$\begin{bmatrix} X_{\omega\varphi} \\ Y_{\omega\varphi} \\ Z_{\omega\varphi} \end{bmatrix} = \begin{bmatrix} \cos \kappa & -\sin \kappa & 0 \\ \sin \kappa & \cos \kappa & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_{\omega\varphi\kappa} \\ Y_{\omega\varphi\kappa} \\ Z_{\omega\varphi\kappa} \end{bmatrix} = M_{\kappa} \begin{bmatrix} X_{\omega\varphi\kappa} \\ Y_{\omega\varphi\kappa} \\ Z_{\omega\varphi\kappa} \end{bmatrix} \dots\dots\dots (2.36)$$

Dengan mengkombinasikan persamaan (2.34), (2.35), dan (2.36) akan didapat hubungan antara koordinat titik objek (P) relatif terhadap (X, Y, Z) dan ($X_{\omega\varphi\kappa}, Y_{\omega\varphi\kappa}, Z_{\omega\varphi\kappa}$).

$$P = M_{\omega} \times M_{\varphi} \times M_{\kappa} \times P_{\omega\varphi\kappa} \dots\dots\dots (2.37)$$

Persamaan diatas diganti menjadi :

$$M_{\omega} \times M_{\varphi} \times M_{\kappa} \dots\dots\dots (2.38)$$

Atau

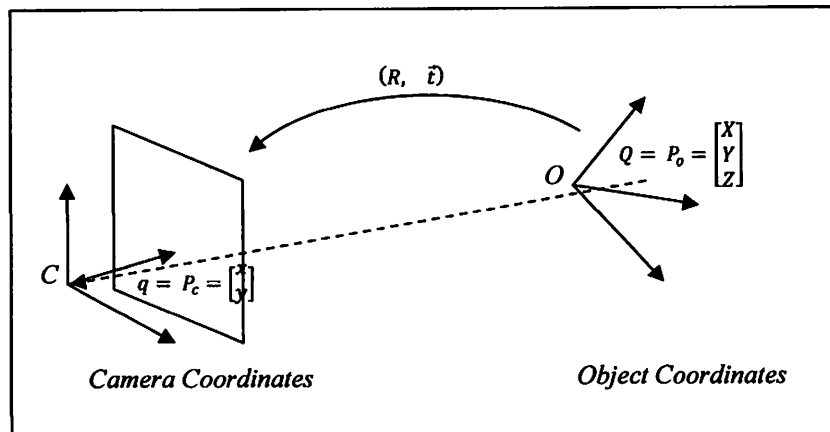
$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \dots\dots\dots (2.39)$$

Dimana M merupakan matrik 3 x 3, matriks rotasi yang persamaan matriksnya sama seperti persamaan (2.7).

2.6.2. Translasi

Vektor translasi adalah merupakan pergeseran dari suatu sistem koordinat ke suatu sistem koordinat sebenarnya yang dipindahkan ke lokasi lain dengan kata lain, vektor translasi hanya offset dari sistem koordinat sebenarnya yang pertama ke sistem koordinat sebenarnya yang kedua (*Bradski dan Keahler, 2008*). Jadi, untuk beralih dari pusat sistem koordinat objek ke suatu pusat kamera, vektor translasi yang tepat adalah $T = origin_{object} - origin_{camera}$. Dengan mengacu pada gambar (2.10) dimana terdapat sebuah titik q untuk sistem koordinat kamera (P_c) dan titik Q untuk sistem koordinat objek atau bumi (P_o) :

$$P_c = R (P_o - T) \dots\dots\dots (2.40)$$



Gambar 2.10 Konversi dari objek ke sistem koordinat kamera; titik P pada objek terlihat sebagai titik p pada bidang foto (*Bradski & Kaehler, 2008*)

2.7. Bundle Adjustment

Bundle adjustment merupakan proses penentuan atau perhitungan parameter *interior orientation* (IO), *exterior orientation* (EO), dan koordinat objek yang dihitung secara bersamaan dengan menggunakan teknik hitung kuadrat terkecil. Parameter-parameter yang digunakan dalam proses *bundle adjustment* adalah parameter IO (*interior orientation*), EO (*exterior orientation*), serta objek pendekatan dari objek yang diamati. Persamaan dasar dari *bundle adjustment* adalah persamaan kolinear, yang mendiskripsikan satuan dasar dari *photogrammetry*. Dengan menggunakan persamaan kolinear membuat bentuk persamaan dan proses penyelesaiannya akan lebih efisien (Mikhail et al., 2001).

$$V + B\dot{\delta} + \ddot{B}\ddot{\delta} = f \dots\dots\dots (2.41)$$

Dimana $\dot{\delta}$ merupakan koreksi untuk parameter orientasi luar dan $\ddot{\delta}$ merupakan koreksi untuk koordinat titik objek. V merupakan matrik residu, B merupakan matrik koefisien.

Jika kita sekarang membentuk persamaan normal yang berhubungan dengan sebuah foto untuk satu titik, maka akan diperoleh persamaan sebagai berikut (Mikhail et al., 2001) :

$$\begin{bmatrix} B_{1ij}^T W_{ij} B_{1ij} & B_{1ij}^T W_{ij} B_{2ij} \\ B_{2ij}^T W_{ij} B_{1ij} & B_{2ij}^T W_{ij} B_{2ij} \end{bmatrix} \begin{bmatrix} \dot{\delta}_{1i} \\ \ddot{\delta}_{2j} \end{bmatrix} = \begin{bmatrix} B_{1ij}^T W_{ij} f_{ij} \\ B_{2ij}^T W_{ij} f_{ij} \end{bmatrix} \dots\dots\dots (2.42)$$

Untuk W_{ij} merupakan matrik bobot 2x2 (*inverse* dari matrik *covarian* dari ketelitian pengukuran foto digital) yang berkaitan dengan koordinat doto dari titik j pada foto i . Persamaan tersebut juga dapat ditulis sebagai berikut :

$$\begin{bmatrix} \dot{N}_i & \bar{N}_{ij} \\ \bar{N}_{ij}^T & \ddot{N}_j \end{bmatrix} \begin{bmatrix} \dot{\delta}_i \\ \ddot{\delta}_j \end{bmatrix} = \begin{bmatrix} \dot{i}_i \\ \ddot{i}_j \end{bmatrix} \quad \text{atau } N\delta = t \dots\dots\dots (2.43)$$

Perlu diperhatikan bahwa $\dot{N}$ mengandung koefisien dari parameter foto, $\ddot{N}$ mengandung koefisien parameter titik (koordinat) dan $\bar{N}$ mengacu pada keduanya.

Bundle adjustment pada umumnya diterapkan dengan menggunakan gabungan kuadrat terkecil, yang tidak mengubah bentuk dasar algoritma secara signifikan. Matrik bobot dari observasi parameter untuk setiap foto dan titik W_i dan W_j ditambahkan pada masing-masing N_i dan N_j (Mikhail et al., 2001).

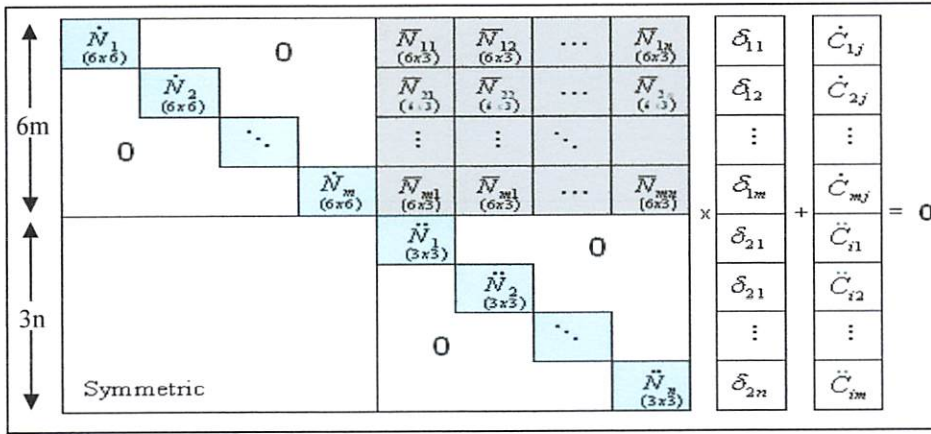
Persamaan ini merupakan pengembangan dari persamaan kolinier dan setiap elemen didalamnya didefinisikan sebagai :

$$\begin{aligned} \dot{N}_{ij} &= \sum_{j=1}^n B_{1ij}^T W_{ij} B_{1ij} ; \quad \dot{i}_i = \sum_{j=1}^n B_{1ij}^T W_{ij} f_{ij} \\ \ddot{N}_{ij} &= \sum_{i=1}^m B_{2ij}^T W_{ij} B_{2ij} ; \quad \ddot{i}_j = \sum_{i=1}^m B_{2ij}^T W_{ij} f_{ij} \dots\dots\dots (2.44) \\ \bar{N}_{ij} &= B_{1ij}^T W_{ij} B_{2ij} \end{aligned}$$

Dimana $\dot{N}$ dan $\ddot{N}$ adalah sub-matrik dari matrik blok-diagonal, dimana blok $\dot{N}$ mengacu pada parameter EO and $\ddot{N}$ mengacu pada koordinat titik-titik obyek serta $\bar{N}$ mengacu pada keduanya. Penyusunan matrik N dapat disajikan sebagai berikut dengan m banyak foto dan n banyak titik objek :

$$N_{(6m+3n) \times (6m+3n)} = \begin{bmatrix} \dot{N}_{1j} & 0 & \cdots & 0 & \bar{N}_{11} & \bar{N}_{12} & \cdots & \bar{N}_{1n} \\ 0 & \dot{N}_{2j} & \cdots & 0 & \bar{N}_{21} & \bar{N}_{22} & \cdots & \bar{N}_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \dot{N}_{mj} & \bar{N}_{m1} & \bar{N}_{m2} & \cdots & \bar{N}_{mn} \\ \hline \bar{N}_{11}^T & \bar{N}_{21}^T & \cdots & \bar{N}_{m1}^T & \ddot{N}_{i1} & 0 & \cdots & 0 \\ \bar{N}_{12}^T & \bar{N}_{22}^T & \cdots & \bar{N}_{m2}^T & 0 & \ddot{N}_{i1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{N}_{1n}^T & \bar{N}_{2n}^T & \cdots & \bar{N}_{mn}^T & 0 & 0 & \cdots & \ddot{N}_{il} \end{bmatrix} \dots\dots\dots (2.45)$$

Struktur persamaan normal *bundle adjustment* juga dapat disusun seperti Gambar (2.11) sebagai berikut :



Gambar 2.11 Struktur Persamaan Normal

Dan penyusunan untuk matrik δ dan t sebagai berikut :

$$\delta_{(6m+3n),1} = \begin{bmatrix} \delta_{11} \\ \delta_{12} \\ \vdots \\ \delta_{1m} \\ \delta_{21} \\ \vdots \\ \delta_{2n} \end{bmatrix} \quad C_{(6m+3n),1} = \begin{bmatrix} \sum_{j=1}^n \dot{C}_{1j} \\ \sum_{j=1}^n \dot{C}_{2j} \\ \vdots \\ \sum_{j=1}^n \dot{C}_{mj} \\ \sum_{i=1}^m \ddot{C}_{i1} \\ \sum_{i=1}^m \ddot{C}_{i2} \\ \vdots \\ \sum_{i=1}^m \ddot{C}_{im} \end{bmatrix} \dots\dots\dots (2.46)$$

Persamaan (2.44) adalah teknik *Bundle adjustment* untuk mendapatkan nilai parameter EO dan koordinat titik obyek didalam sistem kartesian 3D. Jika titik-titik obyek ini hendak dihitung dengan tingkat kekurasian yang lebih tinggi lagi, maka kesalahan sistematis didalam kamera harus dimodelkan.

2.7.1. *Self Calibrating Bundle Adjustment*

Didalam kamera dijital terdapat bidang sensor CCD/CMOS dan susunan lensa, dimana susunan lensa ini mentransformasikan (transformasi perspektif) gambaran suatu titik obyek di permukaan bumi ke bidang foto dijital. Karena adanya kesalahan sistematis pada bidang sensor dan susunan lensa ini, maka gambaran titik pada bidang foto mengalami distorsi. Karena koordinat titik-titik obyek ini menjadi fokus pembahasan, maka untuk mendapatkan nilai koordinat dengan akurasi tinggi, maka kesalahan sistematis ini harus dihilangkan atau diminimalkan pengaruhnya terhadap ketelitian titik-titik obyek. Teknik “*Self-Calibrating*” merupakan cara yang paling umum untuk meminimalkan kesalahan sistematis ini yang diperkenalkan oleh *Brown* di akhir tahun 1960 dan sudah dianggap teknik baku sejak tahun 1980an (*Gruen, 1985a*). Pada prinsipnya, teknik menggunakan parameter kalibrasi ke dalam model fungsi Δx dan Δy didalam sistem persamaan kolinier. *Forstner et al. (2004)* menyebutkan model fisik kesalahan sistematis ini dapat dituliskan seperti dalam persamaan (2.22) yang diubah dalam bentuk matriks menjadi



1. The first part of the report is a general introduction to the subject of the study. It includes a brief history of the subject and a statement of the purpose of the study.

2. The second part of the report is a detailed description of the methods used in the study. This includes a description of the subjects, the materials, and the procedures used.

3. The third part of the report is a presentation of the results of the study. This includes a description of the data collected and a discussion of the findings.

4. The fourth part of the report is a conclusion and a discussion of the implications of the study. This includes a summary of the findings and a discussion of the limitations of the study.

5. The fifth part of the report is a list of references. This includes a list of all the sources used in the study.

6. The sixth part of the report is a list of appendices. This includes a list of all the supplementary materials used in the study.

7. The seventh part of the report is a list of figures. This includes a list of all the figures used in the study.

8. The eighth part of the report is a list of tables. This includes a list of all the tables used in the study.

9. The ninth part of the report is a list of footnotes. This includes a list of all the footnotes used in the study.

10. The tenth part of the report is a list of glossary. This includes a list of all the terms used in the study.

11. The eleventh part of the report is a list of index. This includes a list of all the topics covered in the study.

12. The twelfth part of the report is a list of bibliography. This includes a list of all the sources used in the study.

13. The thirteenth part of the report is a list of appendix. This includes a list of all the supplementary materials used in the study.

14. The fourteenth part of the report is a list of figure. This includes a list of all the figures used in the study.

15. The fifteenth part of the report is a list of table. This includes a list of all the tables used in the study.

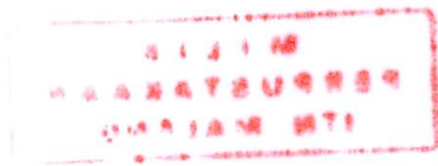
16. The sixteenth part of the report is a list of footnote. This includes a list of all the footnotes used in the study.

17. The seventeenth part of the report is a list of glossary. This includes a list of all the terms used in the study.

18. The eighteenth part of the report is a list of index. This includes a list of all the topics covered in the study.

19. The nineteenth part of the report is a list of bibliography. This includes a list of all the sources used in the study.

20. The twentieth part of the report is a list of appendix. This includes a list of all the supplementary materials used in the study.



$$A_3 \delta_3 = \begin{pmatrix} -1 & 0 & -\frac{\bar{x}}{c} & \frac{\bar{x}}{xr^2} & \frac{\bar{x}}{xr^4} & \frac{\bar{x}}{xr^6} & \frac{2\bar{x}^2 + r^2}{2xy} & \frac{\bar{x}}{2xy} & \frac{\bar{x}}{y} \\ 0 & -1 & -\frac{y}{c} & \frac{y}{yr^2} & \frac{y}{yr^4} & \frac{y}{yr^6} & \frac{2\bar{y}^2 + r^2}{2xy} & \frac{\bar{y}}{2xy} & 0 \end{pmatrix} \begin{pmatrix} \delta_0 \\ \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_1 \\ \delta_2 \\ \delta_1 \\ \delta_2 \end{pmatrix} \dots\dots\dots (2.47)$$

Dimana rumusan ini adalah elemen A_3 , δ_3 dan parameter-parameter ini dikenal sebagai parameter tambahan di dalam teknik *Bundle Adjustment* (Fraser, 1997).

$$\begin{pmatrix} A_1^T P A_1 & A_1^T P A_3 & A_1^T P A_2 \\ A_3^T P A_1 & A_3^T P A_3 & A_3^T P A_2 \\ A_2^T P A_1 & A_2^T P A_3 & A_2^T P A_2 \end{pmatrix} \begin{pmatrix} \delta_1 \\ \delta_3 \\ \delta_2 \end{pmatrix} + \begin{pmatrix} A_1^T P w \\ A_3^T P w \\ A_2^T P w \end{pmatrix} = 0 \dots\dots\dots (2.48)$$

Atau

$$\begin{pmatrix} \dot{N}_{ij} & \tilde{N}_i & \bar{N}_{ij} \\ \tilde{N}_i^T & \ddot{N}_p & \hat{N}_j \\ \bar{N}_{ij}^T & \hat{N}_j^T & \ddot{N}_{ij} \end{pmatrix} \begin{pmatrix} \delta_1 \\ \delta_3 \\ \delta_2 \end{pmatrix} + \begin{pmatrix} \dot{C}_i \\ \ddot{C}_p \\ \ddot{C}_j \end{pmatrix} = 0 \dots\dots\dots (2.49)$$

Atau

$$N\delta + C = 0 \dots\dots\dots (2.50)$$

Dimana :

$$\begin{aligned} \tilde{N}_i &= A_{1ij}^T P_{ij} A_3 & \ddot{N}_p &= \sum_{i=1}^m \sum_{j=1}^n A_3^T P_{ij} A_3 \\ \hat{N}_j &= A_3^T P_{ij} A_{2ij} & \ddot{C}_p &= \sum_{i=1}^m \sum_{j=1}^n A_3^T P_{ij} w_{ij} \end{aligned} \dots\dots\dots (2.51)$$

Solusi hitung kuadrat terkecil dapat ditentukan dengan menggunakan matriks *inverse Cayley N-1*. Tetapi matrik N memiliki

rank defect sebesar parameter penentuan datum (Cooper and Cross, 1988; 1991), yaitu 7 (Granshaw, 1980). Singularitas matrik N ini dapat diselesaikan dengan menambahkan parameter datum secara implisit untuk membentuk persamaan normal yang non-singular seperti:

$$\begin{pmatrix} \dot{N}_{ij} & \tilde{N}_i & \bar{N}_{ij} & 0 \\ \tilde{N}_i^T & \ddot{N}_p & \hat{N}_j & 0 \\ \bar{N}_{ij}^T & \hat{N}_j^T & \dot{N}_{ij} & G \\ 0 & 0 & G^T & 0 \end{pmatrix} \begin{pmatrix} \delta_1 \\ \delta_3 \\ \delta_2 \\ k \end{pmatrix} + \begin{pmatrix} \dot{C}_i \\ \ddot{C}_p \\ \dot{C}_j \\ 0 \end{pmatrix} = 0 \dots\dots\dots (2.52)$$

Dimana G adalah matrik transformasi *Helmert* dan k adalah faktor pengali *Lagrangian*. Matrik G didefinisikan sebagai (Fraser, 1997) :

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & Z_j & -Y_j & X_j \\ 0 & 1 & 0 & -Z_j & 0 & X_j & Y_j \\ 0 & 0 & 1 & Y_j & -X_j & 0 & Z_j \end{pmatrix} \dots\dots\dots (2.53)$$

Teknik hitungan yang dipakai untuk menyelesaikan Persamaan (2.52) ada beberapa macam, seperti misalnya teknik *minimum constraint*, *S-transformation*, *Pseudo Inverse*, dan *Free-Net Adjustment* (Tjahjadi, 2008b). Tetapi, teknik yang paling sesuai didalam kasus ini adalah teknik yang dapat mengoptimalkan tingkat keakurasian koordinat titik-titik obyek. Dengan kata lain, harus dipilih suatu teknik yang dapat meminimumkan matrik kovarian dari titik-titik objek.

2.7.2. Model Penilaian *Bundle Adjustment*

Kinerja dari *bundle adjustment* dinilai dengan beberapa kriteria, antara lain tingkat akurasi dan presisinya. Penilaian keakuratan *bundle adjustment* pada umumnya dilakukan dengan cara membandingkan

koordinat titik objek yang dihitung dari *bundle adjustment* dengan nilai yang diperoleh dari hasil pengamatan (King, 1993).

Kita dapat menilai ketepatan atau presisi suatu perhitungan dengan melihat nilai kovarian dari parameternya. Varian dari satu parameter dapat dikatakan sebagai sebaran dari nilai parameter itu. Semakin tinggi nilai varian dari parameter semakin tidak baik parameter tersebut, parameter yang sempurna akan mempunyai nilai varian nol. Sebuah persamaan kuadrat kecil akan menghasilkan varian dari parameternya. Varian ini sering disebut dengan standar deviasi, dimana standar deviasi merupakan varian yang berbentuk akar kuadrat dan memiliki satuan yang sama dengan parameternya (Mikhail et al, 2001).

Matriks kovarian merupakan hasil simpangan yang umumnya diperoleh dari *bundle adjustment* simultan secara langsung dan dalam waktu bersamaan saat perhitungan parameter yang tidak diketahui (Wang dan Clarke, 1998).

Dari separate adjustment, kofaktor matrik dari parameter kamera dapat dihitung secara pendekatan melalui persamaan sebagai berikut (Wang dan Clarke, 1998) :

$$Qx_1 = (A_1^T W_1 A_1)^{-1} = A_{11}^{-1} \dots\dots\dots (2.54)$$

Matrik kovarian untuk parameter kamera sebagai berikut (Wang dan Clarke, 1998) :

$$Cx_{1i} = \sigma_0^2 Qx_{1i} \dots\dots\dots (2.55)$$

Kofaktor matrik dari koordinat titik objek dapat dihitung secara pendekatan melalui persamaan berikut (Wang dan Clarke, 1998) :

$$Qx_2 = (A_2^T W_1 A_2)^{-1} = A_{22}^{-1} \dots\dots\dots (2.56)$$

Matrik kovarian untuk koordinat 3D sebagai berikut (*Wang dan Clarke, 1998*) :

$$Cx_{2j} = \sigma_0^2 Qx_{2j} \dots\dots\dots (2.57)$$

Persyaratan utama dari *bundle adjustment* adalah nilainya harus akurat. Dimana keakuratan dari proses *bundle adjustment* dapat dinilai dari nilai presisi yang sekecil mungkin (*Mikhail et al, 2001*).

BAB III

METODOLOGI PENELITIAN

3.1. Persiapan Penelitian

Sebelum melakukan penelitian, dilakukan persiapan terlebih dahulu agar proses penelitiannya dapat berjalan dengan lancar hingga pada hasil penyajian data. Oleh karena itu, persiapan yang dilakukan adalah sebagai berikut :

3.1.1. Materi Penelitian

Adapun materi yang digunakan sebagai bahan dalam penelitian ini meliputi data-data sebagai berikut :

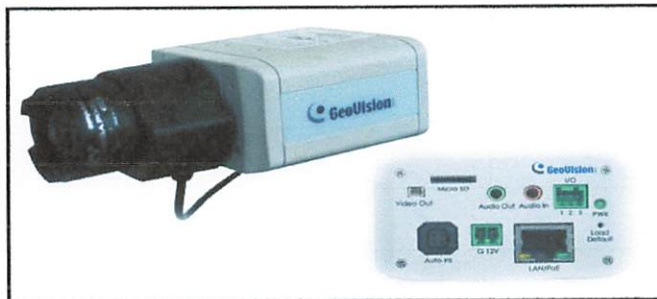
1. Data foto, merupakan data dasar yang dibutuhkan dalam proses kalibrasi kamera IP stereo.
2. Data parameter *Interior Orientation* (IO) pendekatan yaitu (x_0, y_0, f) dan koordinat foto (x, y) . Parameter interior mendefinisikan geometri internal sebuah kamera antara lain *perspektif center*, *principle point*, serta panjang fokus dari kamera.
3. Data parameter *Exterior Orientation* (EO) pendekatan yaitu $(X_L^0, Y_L^0, Z_L^0, \omega^0, \phi^0, \kappa^0)$ tiap foto dari proses *Relative Orientation* dan Proses *Resection*.
4. Data parameter *Object space* pendekatan yaitu (X_A, Y_A, Z_A) dari proses *Intersection*.

3.1.2. Alat Penelitian

Untuk mendukung pelaksanaan kegiatan penelitian ini, maka diperlukan beberapa perangkat keras (*hardware*) dan perangkat lunak (*software*) antara lain :

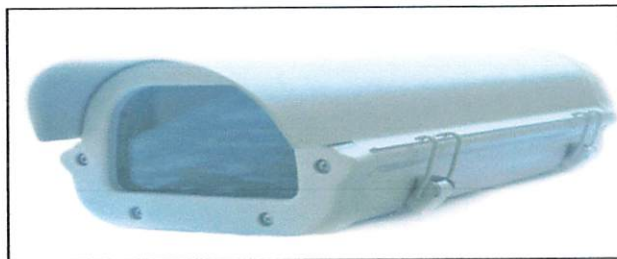
A. Perangkat keras (*hardware*)

- 1) Kamera IP GV-BX320D 3M H.264



Gambar 3.1 Kamera IP GV-BX320D 3M H.264

- 2) Dome kamera



Gambar 3.2 Dome kamera

- 3) HUB D-LINK 8 port



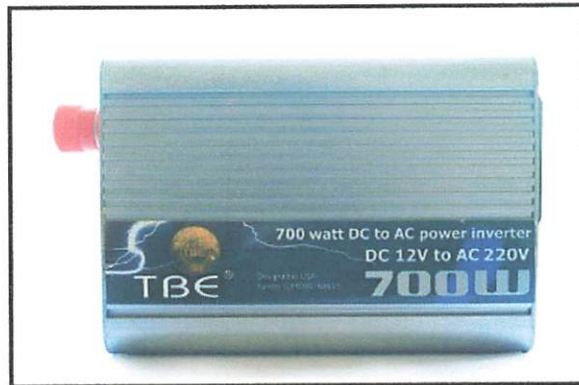
Gambar 3.3 HUB D-LINK 8 port

4) Kabel Lan



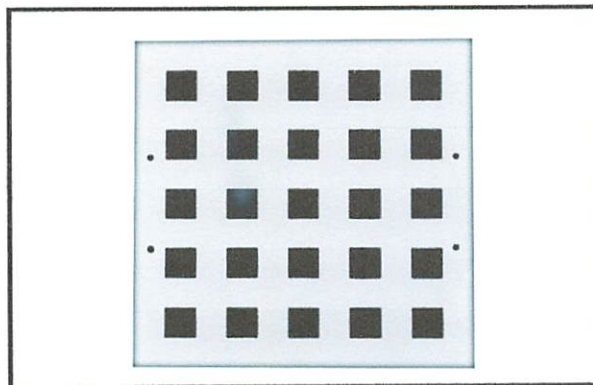
Gambar 3.4 Kabel Lan

5) Inverter 700 Watt



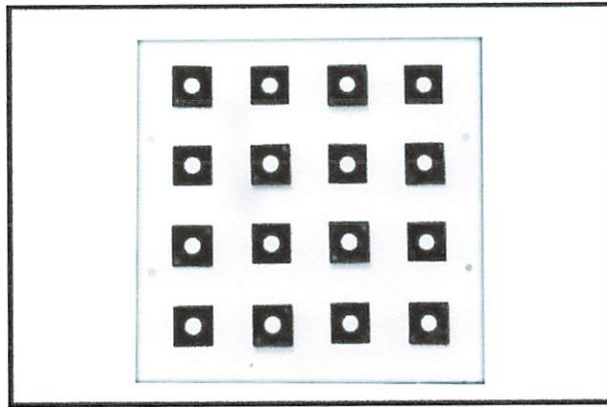
Gambar 3.5 Inverter 700 Watt

6) Papan/ plat kalibrasi 2D



Gambar 3.6 Papan/ plat kalibrasi 2D

7) Papan/ plat kalibrasi 3D



Gambar 3.7 Papan/ plat kalbrasi 3D

8) Kabel Roll

9) Aki

10) Laptop Toshiba

Spesifikasi sebagai berikut :

- *Processor Intel core i3 2.13 Ghz*
- *Ram 2 GB DDR III*
- *Hard disk 320 GB*
- *Windows 7 Ultimate*

11) *Roof Rack*



Gambar 3.8 *Roof Rack* untuk tempat kamera IP

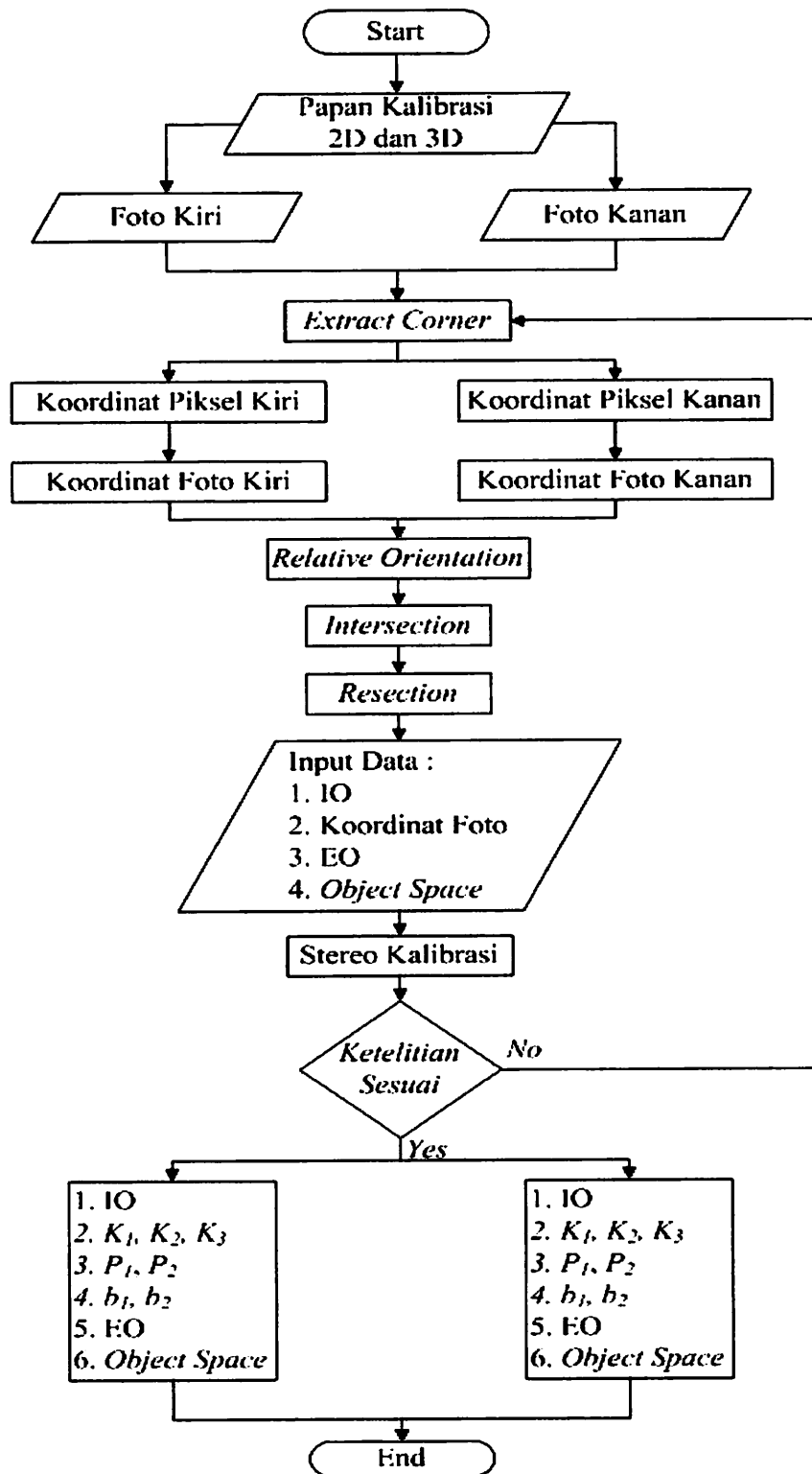
B. Perangkat lunak (*software*)

- 1) *GeoVision GV-NVR System* adalah sebuah perangkat lunak (*software*) bawaan dari kamera IP yang digunakan untuk mengkoneksikan kamera IP dengan laptop dan digunakan juga untuk menyimpan video dalam format Avi.
- 2) *Free Video to JPEG Converter Version 2.0.1.602* merupakan *software* yang digunakan untuk mengkonversi data video menjadi sebuah *image* agar dapat diproses.
- 3) *Australis Version 6.05* merupakan *software* yang digunakan untuk mendapatkan data parameter *Exterior Orientation* (EO) pendekatan yaitu $(X_L, Y_L, Z_L, \omega, \phi, \kappa)$ dan data parameter *Object space* pendekatan yaitu (X_i, Y_i, Z_i) serta data koordinat foto.
- 4) *Matlab R2011b Version 7.13.0.564* adalah sebuah perangkat lunak (*software*) yang digunakan untuk membuat program kalibrasi kamera IP stereo.
- 5) *Microsoft Office excel 2007* merupakan perangkat lunak (*software*) yang digunakan untuk melihat hasil dari proses perhitungan kalibrasi kamera IP stereo.

3.2. Metode Penelitian

Dalam proses penelitian haruslah dibuat suatu kerangka pekerjaan yang sistematis agar mudah dipahami dan mempermudah dalam penelitian. Adapun langkah atau alur penelitian yang akan dilakukan sebagai berikut :

3.2.1. Diagram Alir Penelitian



Gambar 3.9 Diagram alir penelitian

3.2.2. Penjelasan Diagram Alir Penelitian

1. Persiapan awal yaitu mendesain, membuat papan kalibrasi 2D dan 3D dan persiapan alat untuk pengambilan data foto stereo.
2. Pengambilan data foto stereo papan kalibrasi 2D dan 3D menggunakan kamera IP.
3. Dari dua buah foto stereo yang dihasilkan oleh kamera IP memiliki koordinat piksel.
4. Dimana proses untuk mendapatkan koordinat piksel tersebut adalah dengan proses *extract corner* yang bertujuan untuk mengidentifikasi titik-titik grid yang terdapat pada papan kalibrasi.
5. Setelah itu koordinat piksel ditransformasikan ke dalam koordinat foto.
6. Proses selanjutnya adalah proses *relative orientation* untuk mendapatkan parameter pendekatan EO ($X_L^0, Y_L^0, Z_L^0, \omega^0, \phi^0, \kappa^0$) dan koordinat *object space* pendekatan (X_A, Y_A, Z_A).
7. Proses *intersection* adalah proses untuk mendapatkan parameter koordinat *object space* pendekatan (X_A, Y_A, Z_A).
8. Dan proses *resection* adalah proses untuk mendapatkan parameter pendekatan EO ($X_L^0, Y_L^0, Z_L^0, \omega^0, \phi^0, \kappa^0$).
9. Jika sudah mendapatkan parameter IO, koordinat foto, EO, dan *object space* maka proses stereo kalibrasi kamera dapat langsung dilaksanakan.
10. Proses stereo kalibrasi kamera tersebut diproses dengan membuat program menggunakan *software Matlab 2011b*.

11. Apabila proses stereo kalibrasi kamera mendapatkan nilai ketelitian yang besar maka proses kalibrasi kamera diulang kembali ke proses *excrtaack corner*.
12. Dan apabila proses stereo kalibrasi kamera mendapatkan nilai ketelitian yang kecil maka proses kalibrasi kamera sesuai dan kamera terkalibrasi.
13. Stereo kalibrasi kamera akan menghasilkan sejumlah kostanta dari parameter-parameter kalibrasi kamera masing-masing kamera IP. Dimana parameter-parameter kalibrasi kamera tersebut antara lain : internal kamera (IO), K_1 , K_2 , K_3 , P_1 , P_2 , b_1 , b_2 , eksternal kamera (EO) terkoreksi dan *object space* terkoreksi.
14. Kalibrasi kamera selesai.



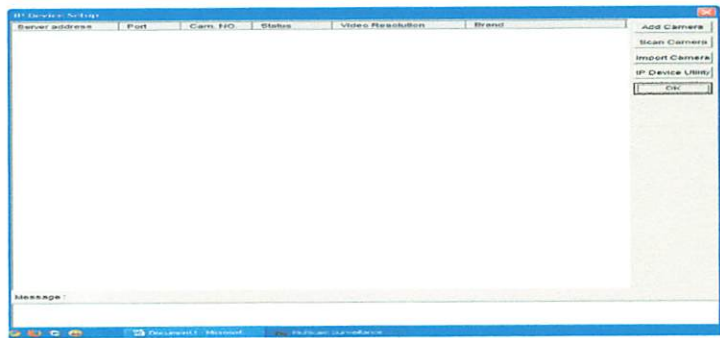
3.3. Pelaksanaan Penelitian

3.3.1. Pengambilan Data Foto Papan Kalibrasi

Dalam pengambilan data kamera IP terdapat langkah-langkah pelaksanaan pengambilan datanya. Adapun langkah-langkah yang harus dilakukan untuk pengambilan data kamera IP yaitu :

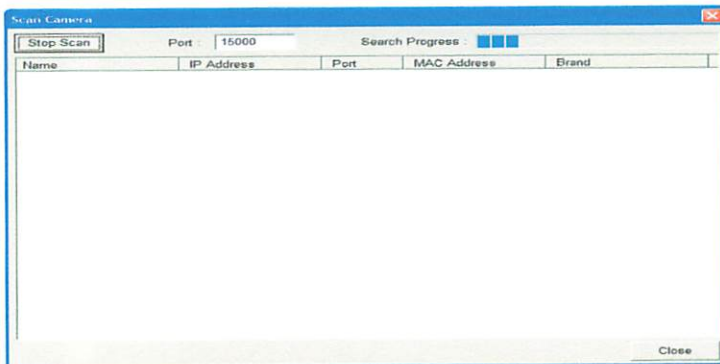
1. Persiapan, untuk melakukan pekerjaan pengambilan data, harus dipersiapkan secara maksimal semua peralatan yang dibutuhkan dengan maksud tujuan untuk meminimalisir kesalahan pada saat pengambilan data. Persiapan ini juga merupakan langkah melakukan rangkaian pada kamera IP, yaitu :

- a. Pasang *roof rack* pada mobil, lalu pasang dome di *Roof Rack* pada posisi yang telah diatur.
 - b. Masukkan kamera IP ke dalam dome, lalu dibaut.
 - c. Pasang kabel power dan kabel LAN ke kamera IP.
 - d. Sambungkan kabel power kamera IP ke kabel roll, maka kamera IP telah menyala.
 - e. 2 (dua) kabel LAN yang menyambung pada kamera IP disambungkan ke USB HUB D-LINK. Dan label LAN yang 1 (satu) disambungkan ke laptop.
 - f. Sambungkan kabel power USB HUB D-LINK ke kabel roll.
 - g. Untuk kabel roll disambungkan pada inverter 700 Watt, dan inverter disambungkan pada aki.
 - h. Maka selesai sudah persiapan untuk menggunakan kamera IP.
2. Integrasikan kamera IP pada laptop yang telah diinstal perangkat lunak *GV-NVR* untuk melakukan settingan pada kamera IP. Langkah untuk melakukan settingan pada kamera IP yaitu :
- a. Untuk memulai program *GV-NVR*, tekan *start GV-NVR GeoVision GV-NVR System*.
 - b. Untuk penggunaan pertama kali, akan diminta untuk memasukkan ID dan *Password*. Untuk memudahkan masukkan ID : admin dan *Password* : admin. Lalu tekan OK.
 - c. Akan muncul *window IP Device Setup* yang berfungsi untuk mencari kamera yang akan digunakan. Pilih *Scan Camera*.

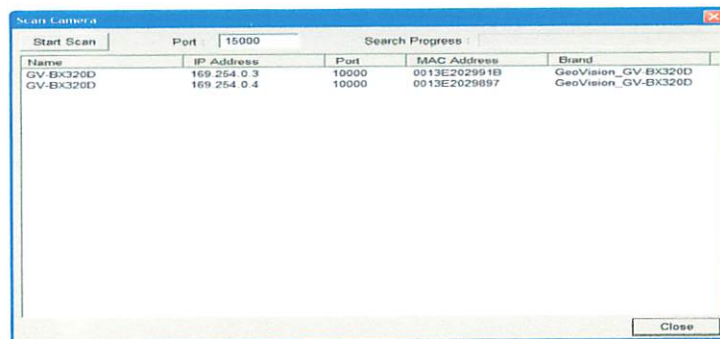


Gambar 3.10 *Window IP Device Setup*

- d. Lalu pada *window* selanjutnya tekan *Scan Camera* dan program ini akan secara otomatis mencari kamera yang sudah dihubungkan.



Gambar 3.11 *Window Scan Camera*



Gambar 3.12 Kamera terdeteksi

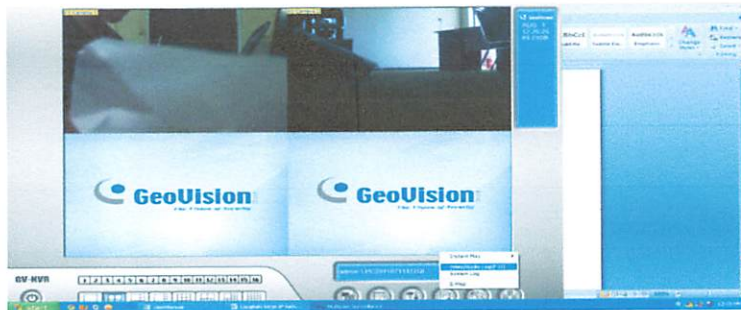
- e. Klik 2 kali pada tiap kamera yang terdeteksi, maka akan otomatis melakukan *Query* pada masing-masing kamera.

3. Untuk memulai *monitoring*, klik *Monitor* lalu pilih kamera yang akan melakukan *monitoring*. Jika sudah selesai, klik kembali *monitoring* dan pilih *stop*.



Gambar 3.16 Memulai *Monitoring* Kamera

4. Untuk melihat hasil *monitoring*, klik pada *icon View Log*, lalu pilih *video/audio log*.



Gambar 3.17 Melihat Hasil *Monitoring*

5. Maka muncul program bagian dari *GV-NVR system* untuk melihat dan menyimpan hasil *monitoring*.



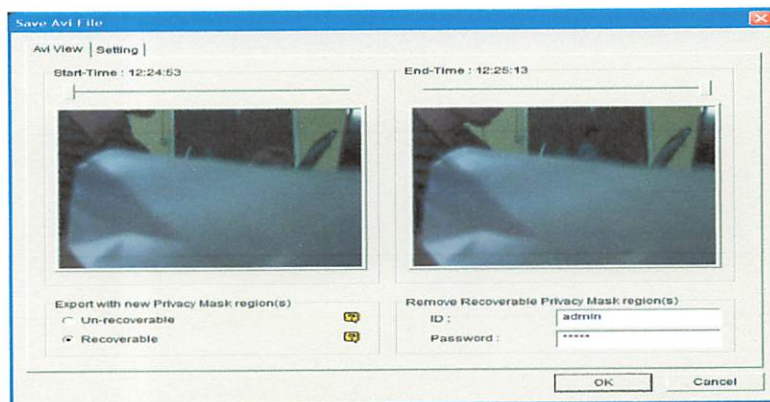
Gambar 3.18 *GV-NVR system* untuk melihat dan menyimpan hasil *monitoring*

6. Klik pada *icon Save Video*, pilih *Save as Avi*.

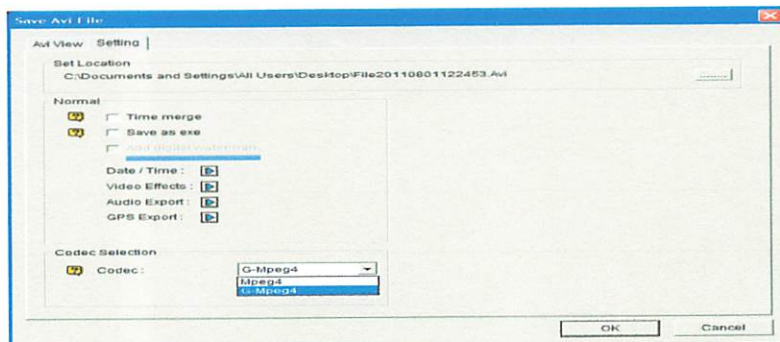


Gambar 3.19 *Save Video*

7. Pada *window Save avi File*, klik *tab setting* dan ganti *codec G-Mpeg4* menjadi *Mpeg4*. Ini bertujuan agar video Avi yang disimpan kompatibel dengan program ekstrasi yang digunakan pada penelitian kali ini.

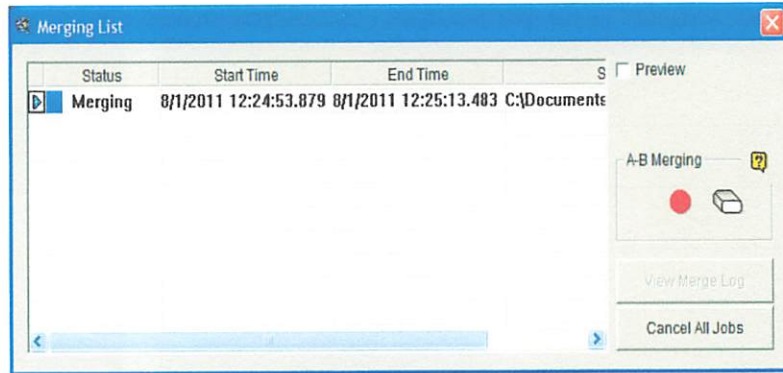


Gambar 3.20 *Window Save as Avi*



Gambar 3.21 *Window Save Avi File*

8. Klik OK jika sudah selesai maka akan muncul *window Merging List*, jika *Merging* sudah *Complete*, tutup *window* tersebut maka video hasil *monitoring* sudah berupa format *Avi*.

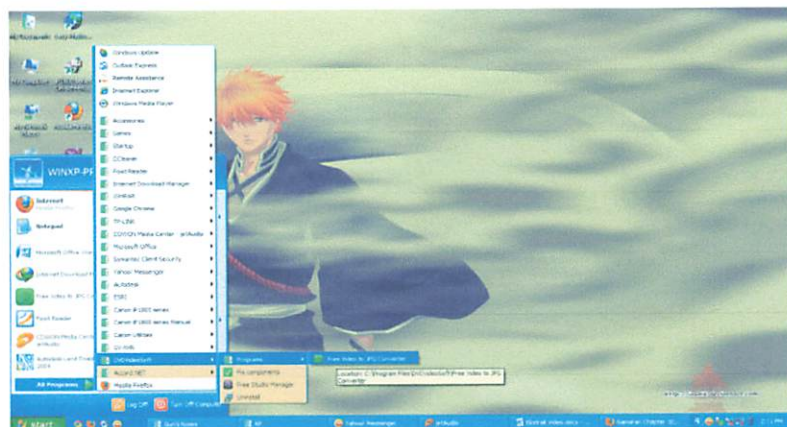


Gambar 3.22 *Window Merging List*

9. *Convert format Video to Jpg* (mengubah format video ke jpg)

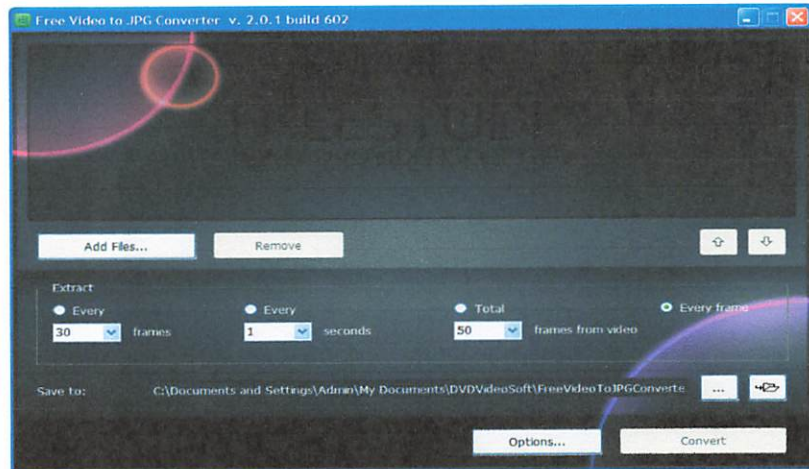
Untuk ekstraksi video, program yang digunakan adalah *Free Video to JPEG Converter Version 2.0.1.602*. Program ini bisa didapatkan dengan cara mendownload di berbagai situs di internet. *install* program ini pada computer atau laptop untuk menggunakannya. Langkah-langkah ekstraksi video sebagai berikut :

- a. Buka program *Free Video to JPEG Converter*.



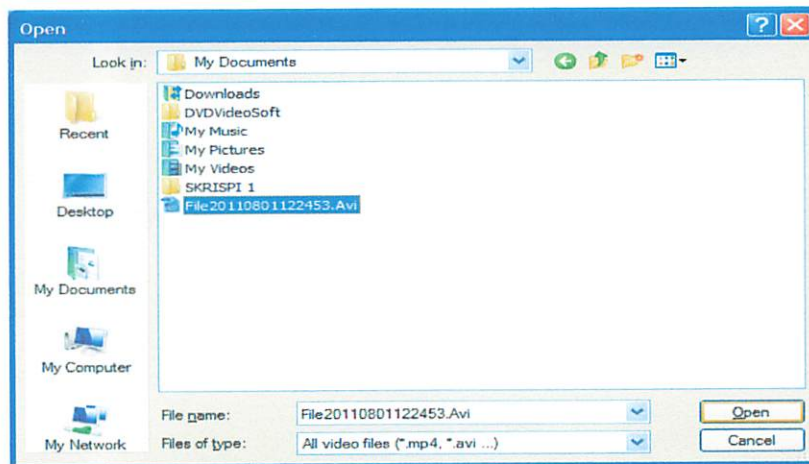
Gambar 3.23 Buka Program *Free Video to JPEG Converter*

- b. Maka program *Free Video to JPEG Converter* dapat digunakan.



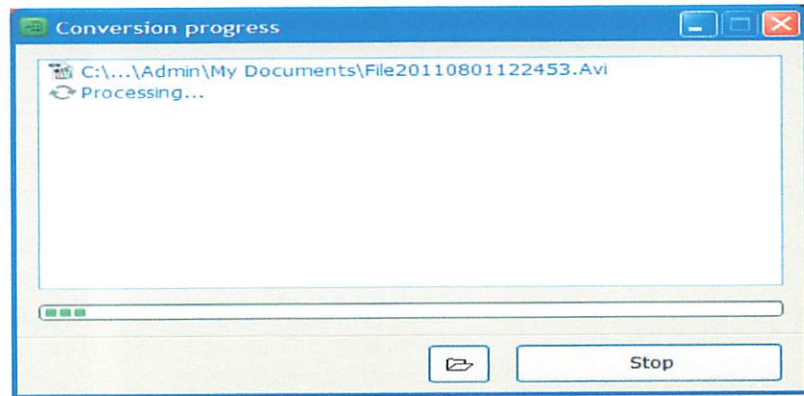
Gambar 3.24 Window *Free Video to JPEG Converter*

- c. Setelah itu klik *add files* untuk memasukkan file video yang ingin di ekstrak.



Gambar 3.25 Add Files Video

- d. pilih video yang ingin di ekstrak setelah itu klik *Open*.
- e. Setelah itu pilih hasil ekstraksi video yang diinginkan pada kolom *Extract*. Untuk kali ini dipilih hasil untuk *every frame*.
- f. Jika sudah selesai langsung klik tombol *convert* untuk memulai ekstraksi data foto.



Gambar 3.26 Window Conversion Progress

- g. Setelah *Conversion Progress* selesai, maka akan didapatkan hasil ekstraksi video berupa data foto (*.Jpg).



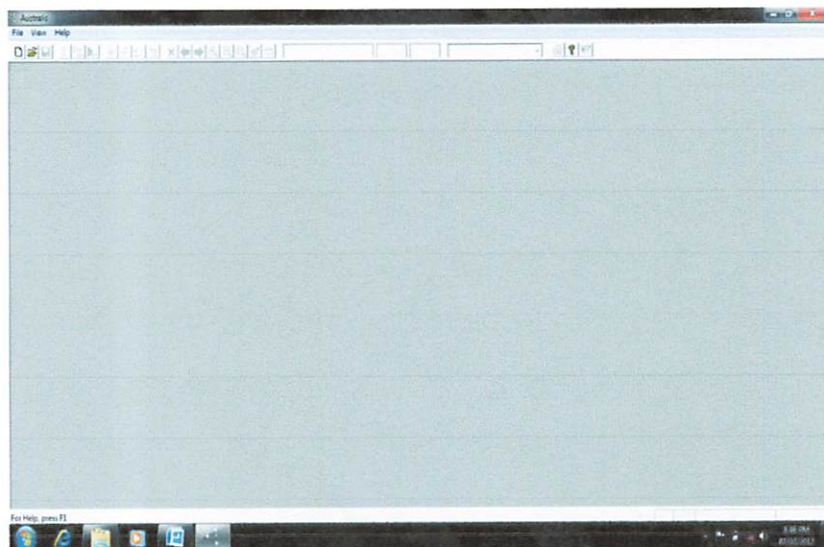
Gambar 3.27 Hasil Ekstraksi Video (*.Jpg)

3.3.2. Pengolahan Data Foto Papan Kalibrasi

Pengolahan data foto ini menggunakan proses fotogrametri yang bertujuan untuk mendapatkan parameter EO pendekatan ($X_L^0, Y_L^0, Z_L^0, \omega^0, \phi^0, \kappa^0$), parameter *object space* pendekatan (X_A, Y_A, Z_A), dan koordinat foto. Pengolahan data foto ini menggunakan perangkat lunak (*software*) yaitu *Australis 6.05*. Adapun proses pengolahan data foto

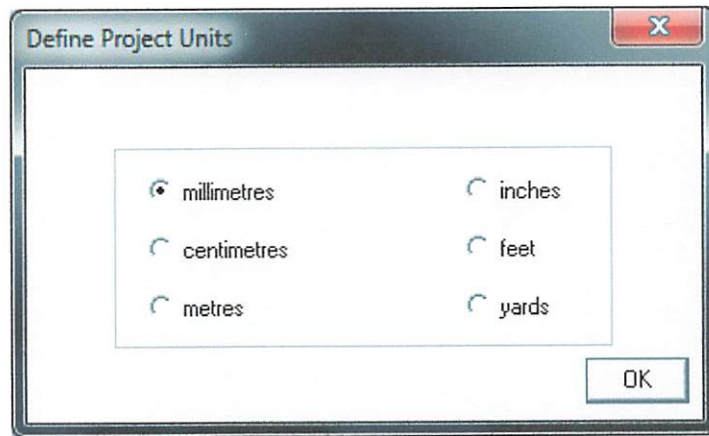
papan kalibrasi dengan menggunakan perangkat lunak *Australis 6.05* akan dijelaskan sebagai berikut :

1. Siapkan sebuah folder kosong, kemudian masukan data foto yang akan diolah sebagai data awal proses fotogrametri. Hal ini sangat penting dilakukan, karena pada aplikasi ini akan secara otomatis menyimpan seluruh hasil proses fotogrametri dalam folder data tersebut.
2. Buka aplikasi *Australis 6.05* dengan melakukan *double-click* pada *icon Australis 6.05* yang berada pada *desktop*. Sehingga, akan muncul sebuah tampilan awal aplikasi sebagai berikut.



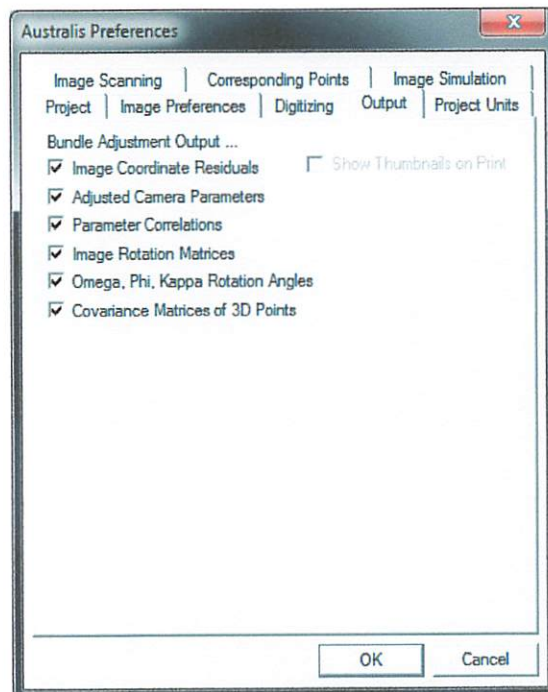
Gambar 3.28 Tampilan Australis 6.05

3. Untuk memulai *project* baru, *click*  atau melalui menu *File* → *New*. Kemudian akan muncul sebuah jendela *Define Project Units* → *Cawang salah satu pilihan* → *OK* untuk menentukan jenis satuan ukuran yang akan digunakan (meter, centimeter, milimeter, dan sebagainya).



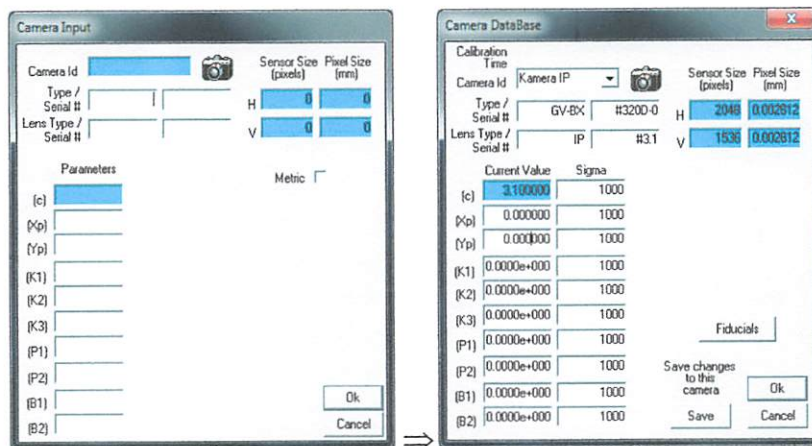
Gambar 3.29 *Define Project Units*

4. Sebelum melakukan pengolahan foto, terlebih dahulu melakukan *setting* terhadap nilai parameter keluaran yang dibutuhkan dengan cara *click* pada menu *Project* → *Preference* → Muncul jendela *Australis Preference* → *Tab Output*. Berikan tanda cawang pada tiap pilihan *Bundle Adjustment Output*, seperti pada gambar dibawah ini, kemudian *click* OK.



Gambar 3.30 *Australis Preferences*

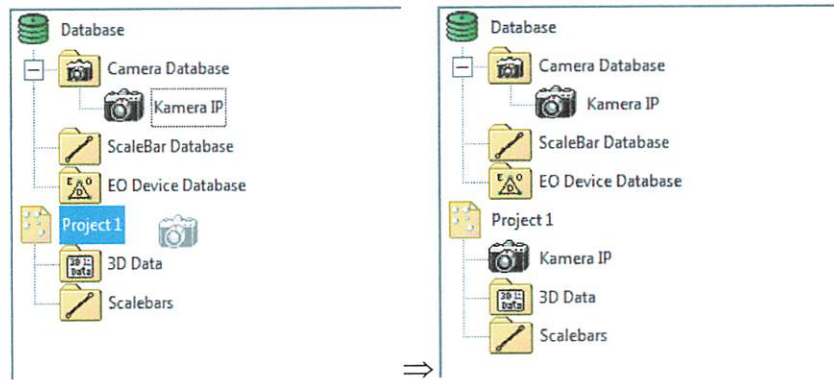
- Selanjutnya, *click* kanan pada *Camera Database* → *Add Camera Database*. Sehingga akan muncul sebuah jendela *Camera Input* seperti dibawah ini.



Gambar 3.31 *Camera Input*

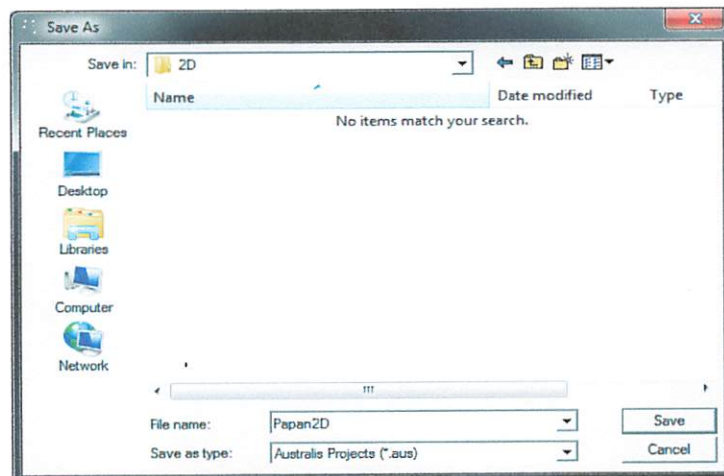
Isi tiap kolom parameter dengan data *Id camera*, ukuran sensor kamera, ukuran piksel, dan 10 parameter kalibrasi kamera berupa panjang fokus, *principle point* (x_0, y_0), 3 parameter distorsi radial, 2 parameter distorsi *decentring*, dan 2 parameter distorsi *affinity*. *Click* OK untuk menyimpan data parameter kamera. Di dalam *Camera Input*, parameter yang wajib diisi adalah data yang kolomnya berwarna biru yaitu *Id camera*, ukuran sensor kamera, ukuran piksel, dan panjang fokus.

- Untuk memulai proses pengolahan data foto pada *Project*, *click-tahan*  pada *Camera Database* → arahkan / letakkan pada *icon*  (nama *project*) → sehingga akan muncul *icon*  yang sama dengan *Id Camera* pada *Camera Database*.



Gambar 3.32 *Project Dengan Id Camera Yang Sama*

7. Simpan *project* untuk pertama kali, dan secara otomatis akan muncul jendela *Save As* seperti dibawah ini.

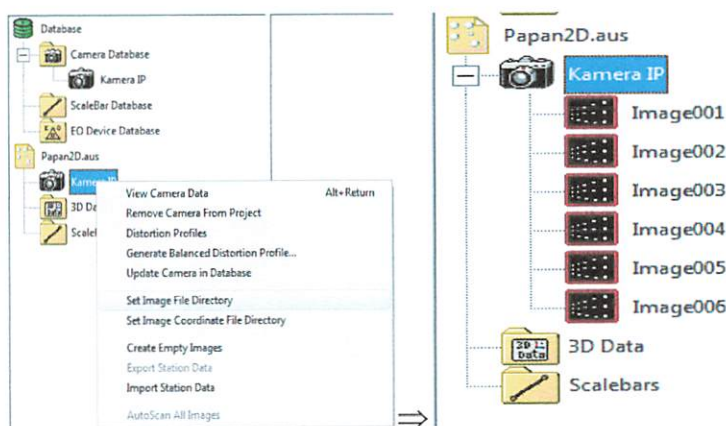


Gambar 3.33 Penyimpanan *Project*

Beri nama *project* sesuai keinginan kemudian *click Save*. Perlu di ingat *file project (*.aus)* harus disimpan satu folder bersama dengan *file foto* yang akan diolah.

8. Setelah seluruh parameter kamera berada pada struktur *Project.aus*, langkah selanjutnya ialah melakukan *input* data foto dengan cara *click* kanan pada  yang berada didalam *Project.aus* → *Set*

Image File Directory → *Open*. Seluruh file foto akan secara otomatis masuk kedalam struktur *Project.aus*.



Gambar 3.34 Penginputan Data Foto

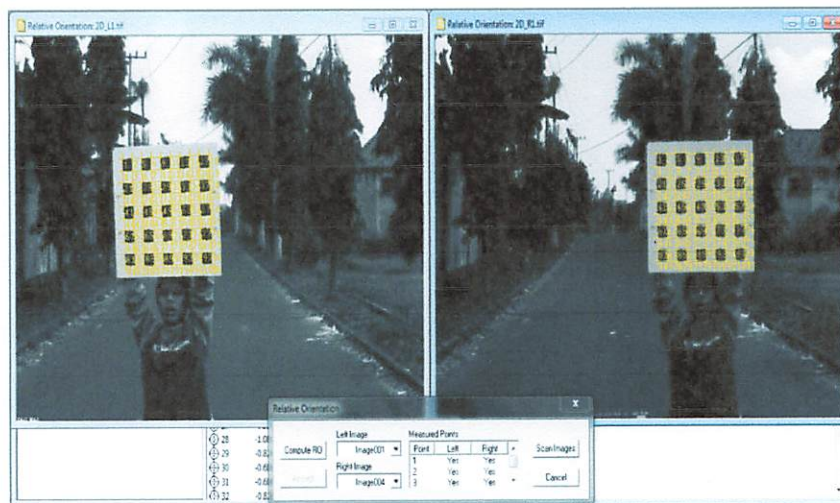
9. Apabila seluruh data foto yang akan diolah telah masuk ke dalam *project* yang telah dibuat, proses selanjutnya ialah melakukan ekstraksi data koordinat foto menggunakan metode *centroid* dengan cara *double-click* pada  *Image001* → muncul sebuah jendela yang berisikan sebuah foto → aktifkan icon  pada *toolbar* → kemudian arahkan *pointer* ke target yang ingin diekstraksi data fotonya → *click target* → muncul *tanda silang dan nomor id* titik yang menandakan target telah ditandai.



Gambar 3.35 Proses Ekstraksi Data Koordinat Foto

Lakukan langkah diatas pada seluruh titik target yang terlihat pada foto tersebut.

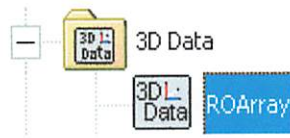
10. Selanjutnya ialah melakukan proses *Relative Orientation* untuk mengorientasikan data foto lain terhadap data foto image001. Proses ini dapat dilakukan dengan cara click *Adjust* → *Relative Orientation* sehingga akan muncul jendela relative orientasi sebagai berikut :



Gambar 3.36 Proses *Relative Orientation*

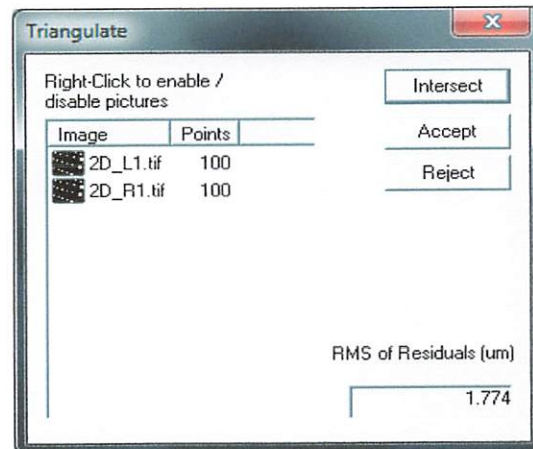
Pada jendela *Relative Orientation* → set *Left Image* dengan *image001* yang telah diproses pada *point 9* → set *Right Image* dengan data foto yang memiliki basis pemotretan yang lebih besar terhadap *image001* → ekstraksi data koordinat foto untuk data foto *Right Image* dengan cara yang sama sesuai dengan *point 9* → *Compute RO* → apabila nilai *Measured Points* sudah sesuai maka click *Accept*. Setelah proses ini dilakukan, maka akan secara otomatis muncul sebuah file *ROArray* pada folder *3D Data*.





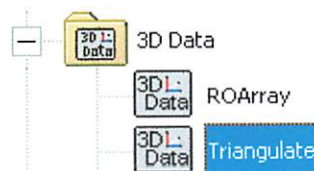
Gambar 3.37 File Output Relative Orientation

11. Setelah proses *Relative Orientasi* sukses dilakukan, proses selanjutnya ialah proses *Intersection*. Adapun prosedur pelaksanaan proses intersection dapat dilakukan dengan cara *click Adjust* → *Triangulate* → *Intersect* → apabila nilai *RMS of Residual* sudah sesuai maka *click Accept*.



Gambar 3.38 Proses Intersection

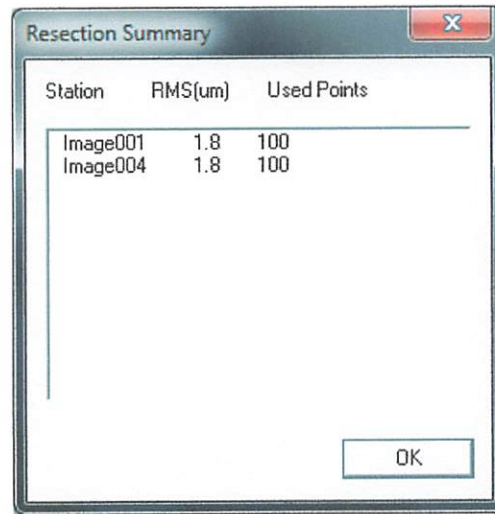
Setelah proses ini dilakukan, maka akan secara otomatis muncul sebuah file *Triangulate* pada folder *3D Data*.



Gambar 3.39 File Output Intersection

12. Proses fotogrametri selanjutnya ialah proses *Resection*. Proses ini dilakukan untuk tiap data foto yang belum terorientasi. Prosedur

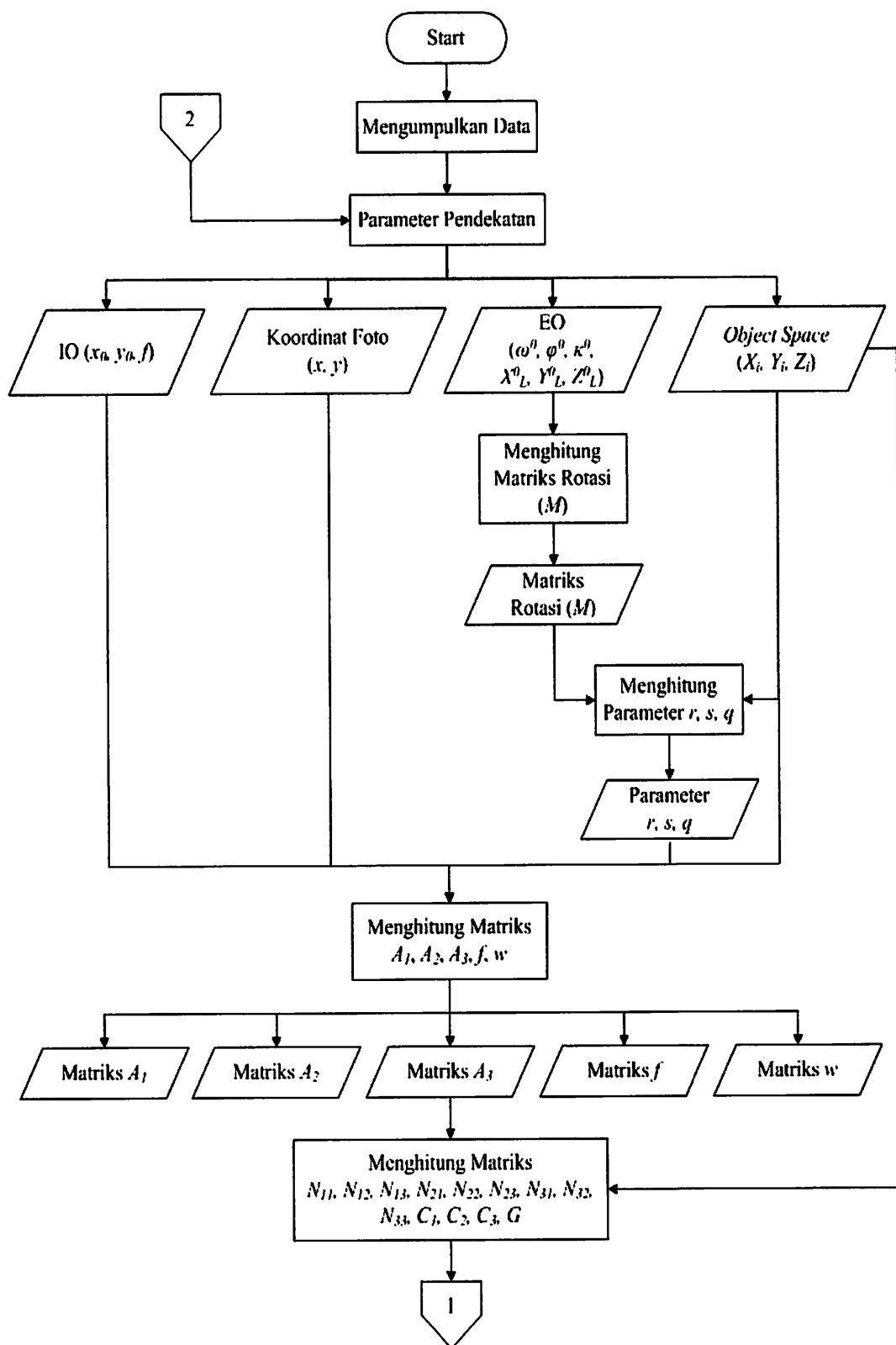
pelaksanaannya ialah terlebih dahulu melakukan proses ekstraksi data koordinat foto sesuai dengan *point 9* untuk seluruh titik yang terekam dalam foto. Selanjutnya apabila seluruh proses ekstraksi data foto telah selesai dilakukan untuk semua foto *click Adjust* → *Resect All Project Images* → *OK*.



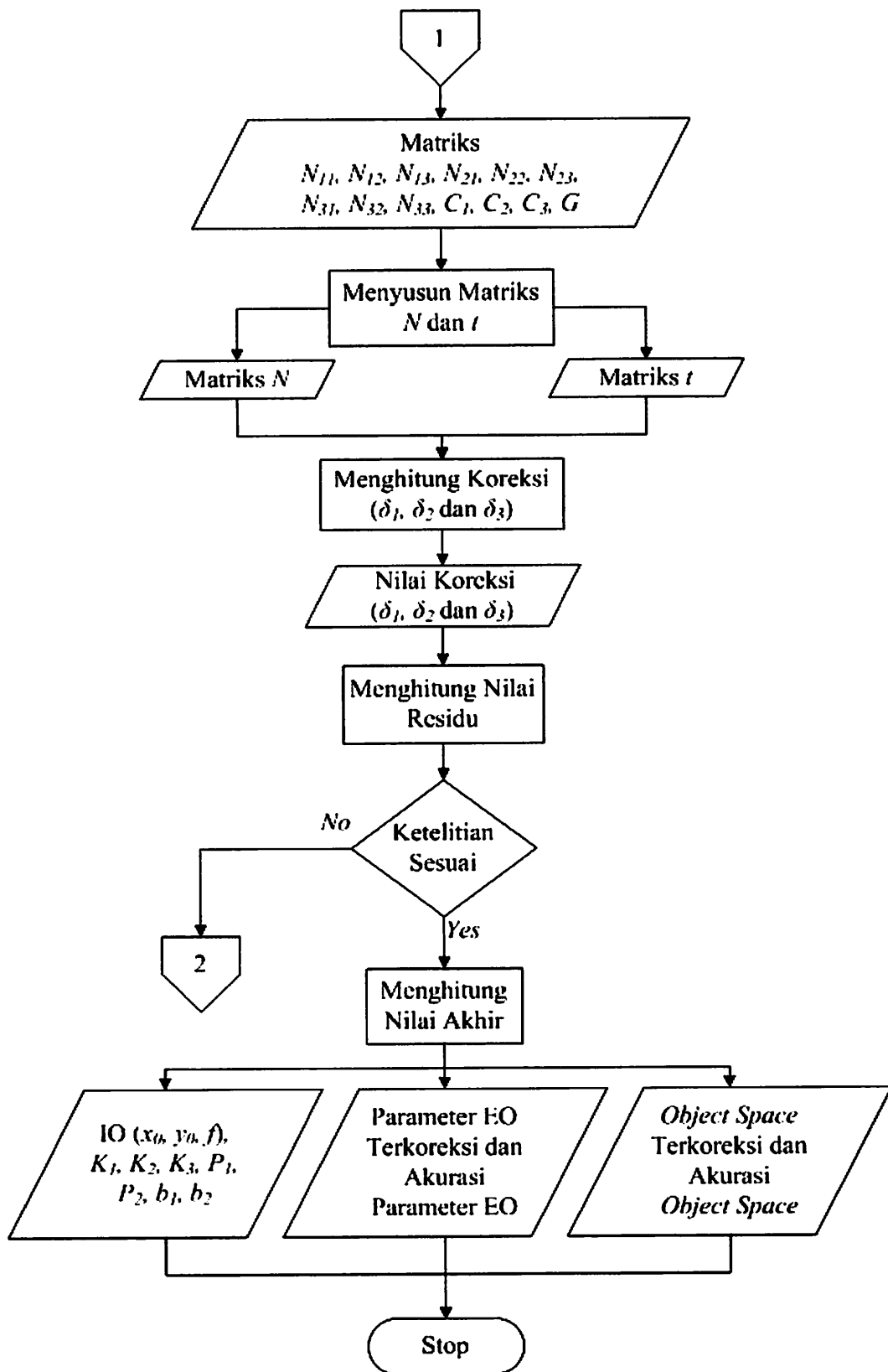
Gambar 3.40 Proses *Resection*

3.3.3. Pembuatan Program Stereo Kalibrasi

Pembuatan program stereo kalibrasi ini menggunakan perangkat lunak (*software*) *Matlab R2011b Version 7.13.0.564* dengan menggunakan prinsip kolineariti untuk algoritmanya. Dimana dalam proses pembuatan program ini haruslah dibuat suatu kerangka pekerjaan yang sistematis agar mudah dipahami dan mempermudah dalam pembuatannya. Adapun langkah atau alur pembuatan program stereo kalibrasi ini yang akan dilakukan sebagai berikut :



Gambar 3.41 Diagram Alir Program Stereo Kalibrasi (1)



Gambar 3.42 Diagram Alir Program Stereo Kalibrasi (2)

Menurut diagram alir diatas maka langkah pembuatan program stereo kalibrasi adalah sebagai berikut :

1. Mengumpulkan Data dan Menghitung Parameter Pendekatan

Mengumpulkan dan menghitung data parameter-parameter pendekatan guna memperlancar proses kalibrasi kamera IP stereo yang didapat dari proses pengolahan data menggunakan *software Australis 6.05* dimana diantaranya akan menghasilkan parameter sebagai berikut :

- a. Data parameter *Interior Orientation* (IO) pendekatan yaitu (x_0, y_0, f) dan koordinat foto (x, y) .
- b. Data parameter *Exterior Orientation* (EO) pendekatan yaitu $(X_L^0, Y_L^0, Z_L^0, \omega^0, \varphi^0, \kappa^0)$ tiap foto dari proses *Relative Orientation* dan Proses *Resection*.
- c. Data parameter *Object space* pendekatan yaitu (X_A, Y_A, Z_A) dari proses *Intersection*.

2. Menghitung Matrik Rotasi (M)

Untuk menghitung matrik rotasi ini, peneliti menggunakan data *omega, phi* dan *kappa* (ω, φ , dan κ) dan diproses menggunakan persamaan :

$$m_{11} = \cos \varphi \cos \kappa$$

$$m_{12} = \sin \omega \sin \varphi \cos \kappa + \cos \omega \sin \kappa$$

$$m_{13} = -\cos \omega \sin \varphi \cos \kappa + \sin \omega \sin \kappa$$

$$m_{21} = -\cos \varphi \sin \kappa$$

$$m_{22} = -\sin \omega \sin \varphi \sin \kappa + \cos \omega \cos \kappa$$

$$m_{23} = \cos \omega \sin \varphi \sin \kappa + \sin \omega \cos \kappa$$

$$m_{31} = \sin \varphi$$

$$m_{32} = -\sin \omega \cos \varphi$$

$$m_{33} = \cos \omega \cos \varphi$$

hasil dari persamaan tersebut kemudian disusun dalam matrik M :

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$$

3. Menghitung Parameter rsq

Parameter rsq dihitung menggunakan data parameter matrik rotasi (M) dan koordinat titik objek/object space pendekatan, dengan menggunakan persamaan :

$$q = m_{31}(X_A - X_L) + m_{32}(Y_A - Y_L) + m_{33}(Z_A - Z_L)$$

$$r = m_{11}(X_A - X_L) + m_{12}(Y_A - Y_L) + m_{13}(Z_A - Z_L)$$

$$s = m_{21}(X_A - X_L) + m_{22}(Y_A - Y_L) + m_{23}(Z_A - Z_L)$$

4. Menghitung Matriks A_1, A_2, A_3, f, W

Data yang digunakan dalam menghitung matrik A_1, A_2, A_3, f, W adalah parameter EO, koordinat titik objek pendekatan, koordinat foto (x, y) dan parameter IO ($x_0, y_0, focus$), matriks rotasi (M) dan parameter rsq . Dimana proses perhitungannya menggunakan persamaan berikut dengan penjelasan penjabaran dari matriks A dan f dapat dilihat pada bab II dasar teori.

$$A_{1ij}^{(2,6)} = \begin{bmatrix} (b_{11})_0 & (b_{12})_0 & (b_{13})_0 & (-b_{14})_0 & (-b_{15})_0 & (-b_{16})_0 \\ (b_{21})_0 & (b_{22})_0 & (b_{23})_0 & (-b_{24})_0 & (-b_{25})_0 & (-b_{26})_0 \end{bmatrix}_{ij}$$

$$A_{2ij}^{(2,3)} = \begin{bmatrix} (b_{14})_0 & (b_{15})_0 & (b_{16})_0 \\ (b_{24})_0 & (b_{25})_0 & (b_{26})_0 \end{bmatrix}_{ij}$$

$$A_{3ij}^{(2,10)} = \begin{bmatrix} -1 & 0 & -\frac{\bar{x}}{c} & \bar{x}r^2 & \bar{x}r^4 & \bar{x}r^6 & 2\bar{x}^2 + r^2 & 2\bar{x}y & \bar{x} & \bar{y} \\ 0 & -1 & -\frac{y}{c} & \bar{y}r^2 & \bar{y}r^4 & \bar{y}r^6 & 2\bar{x}y & 2\bar{y}^2 + r^2 & 0 & 0 \end{bmatrix}_{ij}$$

$$f_{ij} = \begin{bmatrix} J \\ K \end{bmatrix}_{ij}$$

$$W_{ij} = \begin{pmatrix} 1/\sigma_x^2 & 0 \\ 0 & 1/\sigma_y^2 \end{pmatrix}_{ij} \quad W_{ij} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}_{ij}$$

5. Menghitung Matriks N_{1l} , N_{12} , N_{13} , N_{2l} , N_{22} , N_{23} , N_{3l} , N_{32} , N_{33} , C_l , C_2 ,

C_3 , G

Data yang digunakan dalam proses ini adalah matrik A_l , A_2 , A_3 , f , W dan koordinat titik objek pendekatan yang akan diproses dengan persamaan berikut :

$$\begin{aligned} N_{11} &= \dot{N}_{ij} = A_{1ij}^T W_{ij} A_{1ij} \\ N_{13} &= \tilde{N}_i = A_{1ij}^T W_{ij} A_{3ij} \\ N_{12} &= \bar{N}_{ij} = A_{1ij}^T W_{ij} A_{2ij} \\ N_{31} &= \tilde{N}_i^T = A_{3ij}^T W_{ij} A_{1ij} \\ N_{33} &= \ddot{N}_p = A_{3ij}^T W_{ij} A_{3ij} \\ N_{32} &= \hat{N}_j = A_{3ij}^T W_{ij} A_{2ij} \\ N_{21} &= \bar{N}_{ij}^T = A_{2ij}^T W_{ij} A_{1ij} \\ N_{23} &= \hat{N}_j = A_{3ij}^T W_{ij} A_{2ij} \\ N_{22} &= \ddot{N}_{ij} = A_{2ij}^T W_{ij} A_{2ij} \\ C_1 &= \dot{C}_i = A_{1ij}^T W_{ij} f_{ij} \\ C_3 &= \ddot{C}_p = A_{3ij}^T W_{ij} f_{ij} \\ C_2 &= \ddot{C}_j = A_{2ij}^T W_{ij} f_{ij} \end{aligned}$$

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & Z_j & -Y_j & X_j \\ 0 & 1 & 0 & -Z_j & 0 & X_j & Y_j \\ 0 & 0 & 1 & Y_j & -X_j & 0 & Z_j \end{pmatrix}$$

6. Menyusun Matriks N dan t

Menyusun Matriks N dan t untuk memudahkan dalam perhitungan.

Matriks N dan t dapat disusun dengan menggabungkan data N_{11} , N_{12} , N_{13} , N_{21} , N_{22} , N_{23} , N_{31} , N_{32} , N_{33} , C_1 , C_2 , C_3 , G Struktur persamaannya dapat dinyatakan sebagai berikut :

$$\begin{pmatrix} \dot{N}_{ij} & \tilde{N}_i & \bar{N}_{ij} & 0 \\ \tilde{N}_i^T & \ddot{N}_p & \hat{N}_j & 0 \\ \bar{N}_{ij}^T & \hat{N}_j^T & \ddot{N}_{ij} & G \\ 0 & 0 & G^T & 0 \end{pmatrix} \begin{pmatrix} \delta_1 \\ \delta_3 \\ \delta_2 \\ k \end{pmatrix} + \begin{pmatrix} \dot{C}_i \\ \ddot{C}_p \\ \ddot{C}_j \\ 0 \end{pmatrix} = 0$$

7. Menghitung Matriks Koreksi (δ_1 , δ_2 dan δ_3)

Setelah disusun matriks N dan t , maka nilai koreksi untuk tiap-tiap parameternya dapat dihitung. Dengan menggunakan persamaan berikut :

$$\delta = N^{-1}t$$



maka akan diperoleh nilai koreksi δ_1 yaitu nilai koreksi untuk parameter EO, δ_2 yaitu koreksi untuk titik-titik objek dan δ_3 yaitu untuk parameter kalibrasi kamera.

8. Menghitung Nilai Residu (v)

Hitung nilai residu (v) dari menggunakan data matriks N , δ dan t . Proses perhitungannya dapat menggunakan persamaan berikut :

$$v = N\delta - t$$

Jika nilai residu tidak sesuai, maka perlu dilakukan proses iterasi sampai mendapatkan nilai residual yang sekecil-kecilnya. Proses iterasi ini memerlukan pengulangan proses yang sama sesuai dengan diagram alir dimulai kembali dari tahap menghitung semua parameter awal. Jika nilai residual yang diinginkan telah terpenuhi atau sesuai, maka proses iterasi dihentikan.

9. Menghitung Nilai Akhir

Menghitung nilai akhir, dimana proses *self calibrating bundle adjustment* akan mendapatkan nilai parameter EO terkoreksi, koordinat titik objek terkoreksi dan parameter kalibrasi kamera. Dimana penyelesaiannya menggunakan persamaan berikut :

$$O = O^0 + \delta_1 \quad : \text{(Parameter orientasi luar (EO) yang terkoreksi)}$$

$$X = X^0 + \delta_2 \quad : \text{(Koordinat titik objek yang terkoreksi)}$$

$$\delta_3 \quad : \text{(Parameter kalibrasi kamera)}$$

10. Menghitung Ketelitian Varian-Kovarian

Hasil dari proses *bundle adjustment* dengan nilai residual yang kecil berupa koordinat *object space point* final beserta nilai ketelitiannya dan parameter eksterior orientasi.

BAB IV

HASIL DAN PEMBAHASAN

Untuk mengetahui tingkat keberhasilan dari penelitian, perlu dilakukan proses analisa atau pembahasan dari sebuah hasil yang telah dicapai selama proses pelaksanaan penelitian. Parameter keberhasilan dapat diukur dengan membandingkan tujuan dari penelitian dan hasil yang dicapai. Adapun beberapa parameter yang telah dihasilkan selama pelaksanaan penelitian akan disajikan secara jelas dalam bab ini.

4.1. Hasil Penelitian

Dari hasil pelaksanaan penelitian yang telah dilakukan baik dilapangan maupun dilaboratorium dan dengan menggunakan berbagai jenis data yang telah dikumpulkan. Didapat beberapa parameter berupa parameter kalibrasi kamera, parameter *Exterior Orientation* (EO) yaitu $(X_L, Y_L, Z_L, \omega, \phi, \kappa)$ dan parameter *Object space* yaitu (X_A, Y_A, Z_A) . Selain parameter-parameter tersebut dihasilkan pula suatu fungsi matematika yang ditulis dalam sebuah bahasa pemrograman *Matlab* untuk dapat dipergunakan secara fleksibel dengan berbagai kondisi data. Keseluruhan data yang dihasilkan, akan ditampilkan pada tiap sub-bab dibawah ini

4.1.1. Aplikasi dan Listing Kode Pemrograman

Dari setiap tahapan proses penelitian dan perhitungan yang dijelaskan pada *bab-3*, dapat diaplikasikan dalam sebuah bahasa

pemrograman. Adapun perangkat lunak yang digunakan dalam pembuatan aplikasi *windows-form* sederhana ini ialah *Matlab R2011b*. Untuk memudahkan dalam proses pembangunan aplikasi perhitungan, maka setiap tahapan dibagi dalam beberapa fungsi. Beberapa contoh fungsi tersebut disajikan dalam bentuk tabel seperti yang terlihat dibawah ini.

Tabel 4.1 Nama Fungsi, Input Data dan Output Data yang Telah Dibuat

No.	Nama Fungsi	Inputan Data/Keterangan	Output Data
1	Stereo_Calibration.fig	Desain <i>windows-form</i> aplikasi perhitungan	-
2	Stereo_Calibration.m	Fungsi utama dari aplikasi perhitungan untuk menghubungkan tiap fungsi yang ada	-
3	Objectspk.m	Fungsi untuk menentukan selisih antara koordinat <i>objcet space point</i> dan koordinat posisi kamera	$\Delta X, \Delta Y, \Delta Z$
4	MatriksM.m	Fungsi untuk menyusun matriks rotasi	Matriks rotasi (M)
5	Rsqform.m	Fungsi untuk perataan koordinat <i>object space point</i>	r, s, q
6	MatriksA1.m	Fungsi untuk menyusun matriks <i>resection</i> (mencari parameter eksterior orientasi)	Matriks A_1
7	MatriksA2.m	Fungsi untuk menyusun matriks <i>intersection</i> (mencari parameter koordinat <i>object space point</i>)	Matriks A_2
8	MatriksA3.m	Fungsi untuk menyusun matriks parameter kalibrasi kamera	Matriks A_3

9	MatriksL.m	Fungsi untuk menyusun matriks observasi	Matriks $L(f)$
10	MatriksP.m	Fungsi untuk menyusun matriks bobot	Matriks $P(w)$
11	MatriksG.m	Fungsi untuk menyusun matriks <i>transformasi helmert</i>	Matriks G
12	Matriks_Normal.m	Fungsi untuk menyusun matriks normal dan matriks observasi	Matriks N dan Matriks t

Keseluruhan fungsi yang telah dibuat dapat dilihat pula pada lampiran. Adapun isi masing-masing fungsi tersebut merupakan suatu kode logika dalam sebuah bahasa pemrograman. Sebagai contoh, dibawah ini akan ditampilkan sebuah kode dari salah satu fungsi *Rsqform.m*.

```
function rsq = Rsqform(pts3D,eks)

ospk = Objectspk(pts3D,eks);
M = MatriksM(eks);
m = size(pts3D,1);

for i=1:m
    % Camera 1
    rsq(i,1)=M(1,1)*(ospk(i,1))+M(1,2)*(ospk(i,2))+M(1,3)*
    (ospk(i,3));
    rsq(i,2)=M(2,1)*(ospk(i,1))+M(2,2)*(ospk(i,2))+M(2,3)*
    (ospk(i,3));
    rsq(i,3)=M(3,1)*(ospk(i,1))+M(3,2)*(ospk(i,2))+M(3,3)*
    (ospk(i,3));

    % Camera 2
    rsq(i,4)=M(1,4)*(ospk(i,4))+M(1,5)*(ospk(i,5))+M(1,6)*
    (ospk(i,6));
    rsq(i,5)=M(2,4)*(ospk(i,4))+M(2,5)*(ospk(i,5))+M(2,6)*
    (ospk(i,6));
    rsq(i,6)=M(3,4)*(ospk(i,4))+M(3,5)*(ospk(i,5))+M(3,6)*
    (ospk(i,6));
end
```

Dari keseluruhan fungsi tersebut disatukan dalam sebuah bahasa utama (*Main fuction*) yang menjadi penggerak utama aplikasi atau program yang dibuat. Adapun contoh tampilan dari aplikasi perhitungan kalibrasi kamera IP stereo dapat dilihat pada gambar (4.1) dibawah ini.

The screenshot shows a software window titled "Stereo_Calibration". Inside, there are two main sections for "Camera 1" and "Camera 2". Each section has input fields for "Focal Lenght", "xo", and "yo". Below these, there are three buttons: "Parameter EO", "Object Space", and "Image Coordinates". Further down, there are two sections for "Output Camera 1" and "Output Camera 2", each with input fields for "Focal Lenght", "xo", "yo", "K1", "K2", "K3", "P1", "P2", "b1", and "b2". At the bottom, there is a "Standar Deviasi" field with the value "0.00" and three buttons: "Process", "Save", and "Exit".

Gambar 4.1 Tampilan aplikasi perhitungan kalibrasi kamera IP stereo

4.1.2. Prosedur Pengoperasian Program *Matlab Stereo Calibration*

Sesuai dengan listing kode pemrograman yang telah dibuat dan disusun menjadi sebuah program, maka untuk pengoperasian program *matlab* untuk menentukan kalibrasi kamera ip stereo tersebut melalui tahapan sebagai berikut :

1. Tampilan awal program

Tampilan awal saat pengoperasian program ini dapat dilihat pada gambar (4.1). Pada form tampilan program tersebut, terdapat beberapa bagian yang harus dioperasikan pada tahapan berikutnya yaitu tahapan input data, pemrosesan, keluaran data, dan penyimpanan data.

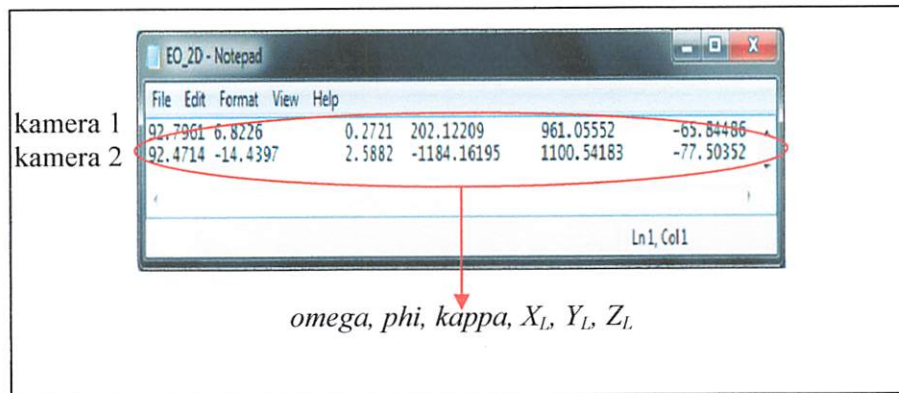
2. Inputan data

Pada proses ini data yang harus diinput adalah parameter IO yaitu panjang fokus, *principle point x*, dan *principle point y* untuk masing-masing kamera yang ditunjukkan seperti gambar (4.2) dibawah ini.

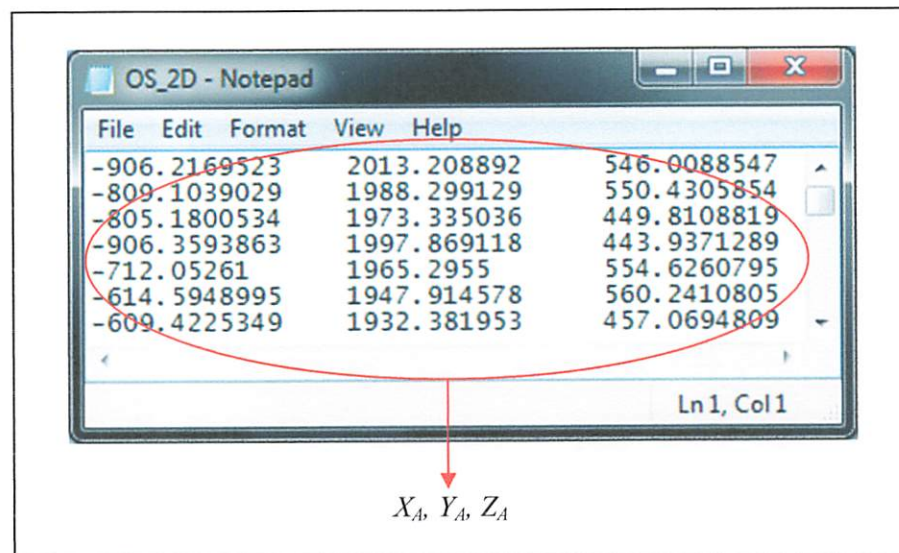
The image shows a software window titled "Input Data". Inside, there is a section labeled "Parameter IO". Under this, there are two sub-sections: "Camera 1" and "Camera 2". Each camera section contains three input fields: "Focal Lenght" (with a value of 3.1), "xo" (with a value of 0), and "yo" (with a value of 0). At the bottom of the window, there are three buttons: "Parameter EO", "Object Space", and "Image Coordinates".

Gambar 4.2 Proses penginputan data

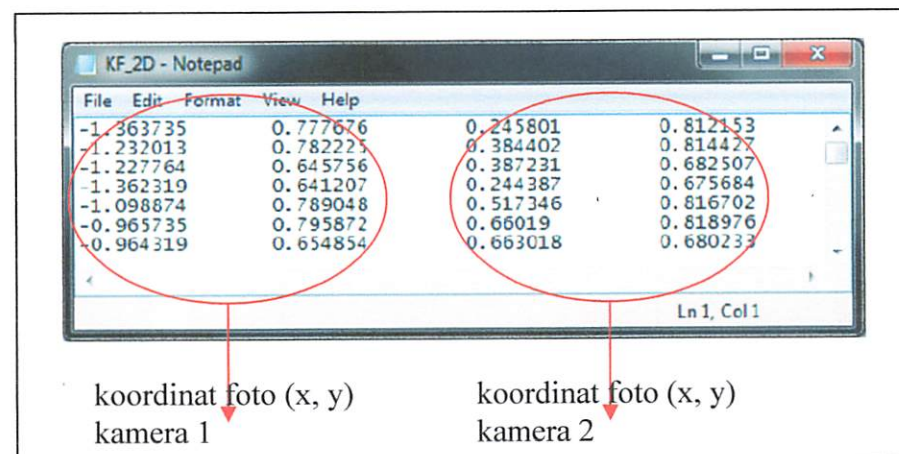
Data selanjutnya yang harus diinputkan adalah parameter EO pendekatan, *object space* pendekatan, dan titik-titik koordinat foto. Ketiga data ini diinputkan dengan menekan tombol input data masing-masing dan memilih *file* yang telah disiapkan sebelumnya dalam bentuk notepad (*file* dalam bentuk format *.txt*).



Gambar 4.3 Susunan data input parameter EO awal



Gambar 4.4 Susunan data input parameter koordinat *object space* awal



Gambar 4.5 Susunan data input parameter koordinat foto masing-masing

kamera

3. Proses perhitungan kalibrasi kamera IP stereo

Setelah seluruh data diinputkan, langkah selanjutnya adalah menjalankan proses perhitungan kalibrasi kamera IP stereo dengan menekan tombol *process*. Maka akan menghasilkan keluaran parameter kalibrasi kamera untuk masing-masing kamera dan akan menghasilkan nilai standar deviasi seperti yang ditunjukkan gambar (4.6) dibawah ini.

Output Camera 1		Output Camera 2	
Focal Lenght	4.79593	Focal Lenght	2.95109
x0	2.13042	x0	-2.91396
y0	-1.36878	y0	-0.652215
K1	1.01127	K1	0.837667
K2	-0.196231	K2	-0.0909101
K3	0.0192289	K3	0.000637234
P1	-0.0441488	P1	0.0601526
P2	-0.0166177	P2	0.00258969
b1	-0.35728	b1	-0.412659
b2	-0.0428446	b2	0.189774

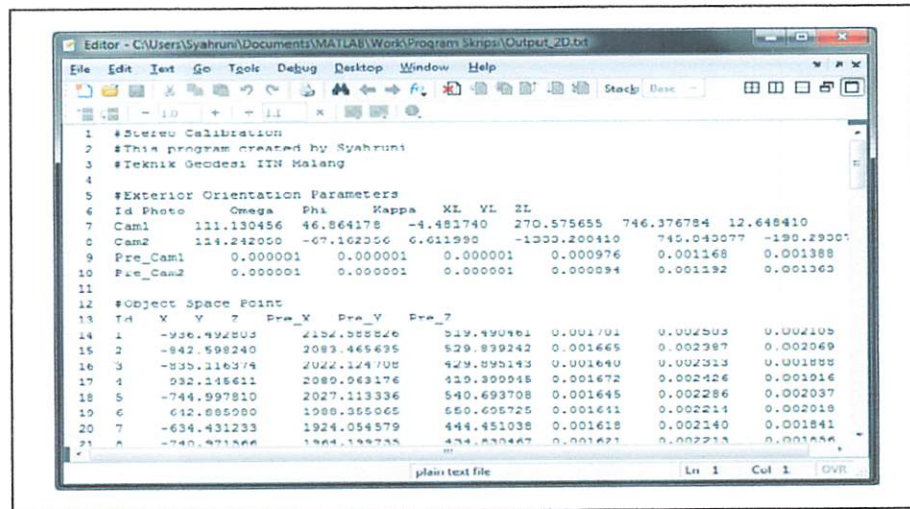
Standar Deviasi 6.24929e-006

Process Save Exit

Gambar 4.6 Hasil keluaran dari proses perhitungan kalibrasi kamera IP stereo

4. Penyimpanan hasil perhitungan kalibrasi kamera IP stereo

Langkah terakhir dalam mengoperasikan perhitungan kalibrasi kamera IP stereo adalah menyimpan nilai proses perhitungan kalibrasi kamera IP stereo tersebut dengan cara menekan tombol *save* seperti pada gambar (4.6). Data yang disimpan adalah parameter EO terkoreksi, akurasi EO, parameter koordinat *object space point* terkoreksi dan akurasi *object space point*.



Gambar 4.7 Hasil penyimpanan perhitungan kalibrasi kamera IP stereo

4.1.3. Hasil Perhitungan Kalibrasi Kamera IP Stereo

Pada perhitungan kalibrasi kamera IP stereo menggunakan bahasa pemrograman *Matlab 2011b* menampilkan hasil yang diperoleh sebagai berikut.

A. Papan kalibrasi 2D

1. Parameter kalibrasi kamera

Tabel 4.2 Parameter Kalibrasi Kamera (Papan 2D)

Parameter Kalibrasi Kamera		
Parameter	Camera 1 (mm)	Camera 2 (mm)
Fokus	3.99777	3.02067
x_0	1.28390	-1.51496
y_0	-0.06913	-0.25913
K_1	0.56521	0.38071
K_2	-0.29234	-0.20960
K_3	0.04181	0.02461
P_1	-0.44836	0.54214
P_2	-0.05250	-0.06692
b_1	-0.89863	-1.29360
b_2	-0.28534	0.20905

2. Parameter *exterior orientation* (EO) dan akurasi parameter *exterior orientation* (EO)

Tabel 4.3 Parameter *exterior orientation* (Papan 2D)

Parameter <i>Exterior Orientation</i>						
Id Camera	ω (Derajat)	ϕ (Derajat)	κ (Derajat)	X_L (mm)	Y_L (mm)	Z_L (mm)
Camera 1	102.15115	33.44075	2.83274	481.90942	895.70255	-260.01619
Camera 2	105.81076	-43.17605	2.98297	-1627.46538	1059.36275	-297.24324

Tabel 4.4 Akurasi Parameter *exterior orientation* (Papan 2D)

Akurasi Parameter <i>Exterior Orientation</i>						
Id Camera	ω (Derajat)	ϕ (Derajat)	κ (Derajat)	X_L (mm)	Y_L (mm)	Z_L (mm)
Camera 1	0.000002	0.000002	0.000002	0.001766	0.002419	0.002611
Camera 2	0.000002	0.000002	0.000002	0.001674	0.002284	0.002428

3. Parameter koordinat *object space point* dan akurasi koordinat *object space point*

Tabel 4.5 Parameter Koordinat *Object Space Point* (Papan 2D)

Parameter Koordinat <i>Object Space Point</i>			
Id	X (mm)	Y (mm)	Z (mm)
1	-1149.88835	2262.68604	797.06535
2	-992.01884	2175.74237	775.35304
3	-983.03445	2118.88670	603.70361
4	-1132.43375	2197.94652	615.39676
5	-850.98386	2112.51643	765.41113
6	-706.58543	2066.70987	759.65421
7	-706.16292	2007.83737	592.03975
8	-844.89404	2059.02475	596.14096
9	-567.81753	2035.81969	760.09448
10	-428.29426	2014.20765	762.49685
11	-426.44492	1960.99266	598.60751
12	-571.16314	1978.32310	592.89610
13	-290.27719	2011.15793	771.93963
14	-144.94267	2026.83942	788.87067
15	-145.19761	1973.12587	620.15849

Tabel 4.6 Akurasi Koordinat *Object Space Point* (Papan 2D)

Akurasi Koordinat <i>Object Space Point</i>			
Id	X (mm)	Y (mm)	Z (mm)
1	0.00615	0.00614	0.00873
2	0.00616	0.00566	0.00838
3	0.00600	0.00537	0.00750
4	0.00595	0.00581	0.00775
5	0.00622	0.00532	0.00814
6	0.00629	0.00509	0.00797
7	0.00615	0.00478	0.00711
8	0.00608	0.00505	0.00729
9	0.00634	0.00495	0.00785
10	0.00634	0.00484	0.00776
11	0.00622	0.00458	0.00697
12	0.00620	0.00465	0.00702
13	0.00634	0.00484	0.00776
14	0.00633	0.00493	0.00786
15	0.00619	0.00467	0.00704

B. Papan kalibrasi 3D

1. Parameter kalibrasi kamera

Tabel 4.7 Parameter Kalibrasi Kamera (Papan 3D)

Parameter Kalibrasi Kamera		
Parameter	Camera 1 (mm)	Camera 2 (mm)
Fokus	2.27236	2.26591
x_0	0.00750	-0.01457
y_0	-0.18961	-0.08434
K_1	0.01569	-0.02840
K_2	-0.08158	0.01258
K_3	0.08376	-0.00263
P_1	0.00041	0.01122
P_2	0.00238	0.00032
b_1	0.02434	-0.00525
b_2	0.00304	0.00326





2. Parameter *exterior orientation* (EO) dan akurasi parameter *exterior orientation* (EO)

Tabel 4.8 Parameter *exterior orientation* (Papan 3D)

Parameter <i>Exterior Orientation</i>						
Id Camera	ω (Derajat)	ϕ (Derajat)	κ (Derajat)	X_L (mm)	Y_L (mm)	Z_L (mm)
Camera 1	93.23414	-0.10838	-0.19922	-15.00356	-868.64563	12.13169
Camera 2	89.82541	-1.75112	0.68916	-1021.67420	-880.84406	129.64735

Tabel 4.9 Akurasi Parameter *exterior orientation* (Papan 3D)

Akurasi Parameter <i>Exterior Orientation</i>						
Id Camera	ω (Derajat)	ϕ (Derajat)	κ (Derajat)	X_L (mm)	Y_L (mm)	Z_L (mm)
Camera 1	0.00009	0.00009	0.00001	0.27619	0.03826	0.27062
Camera 2	0.00005	0.00005	0.00001	0.15027	0.06685	0.16060

3. Parameter koordinat *object space point* dan akurasi koordinat *object space point*

Tabel 4.10 Parameter Koordinat *Object Space Point* (Papan 3D)

Parameter Koordinat <i>Object Space Point</i>			
Id	X (mm)	Y (mm)	Z (mm)
1	-398.09306	2262.82779	385.88740
2	-233.54396	2257.81120	393.06010
3	-71.29442	2250.99230	400.78470
4	91.71072	2251.66263	406.92945
5	-389.16589	2258.50794	223.07695
6	-226.47208	2247.03970	231.46218
7	-64.50504	2248.90682	236.55560
8	98.99462	2242.35122	245.98080
9	-382.59669	2251.42833	59.67869
10	-219.01728	2245.59753	69.46032
11	-57.53996	2235.35195	78.02915
12	104.14663	2238.87938	83.95861
13	-374.42862	2248.53569	-99.42686
14	-211.52922	2234.60879	-90.81429
15	-50.54777	2234.79100	-84.02758

16	111.02011	2230.10824	-77.84540
----	-----------	------------	-----------

Tabel 4.11 Akurasi Koordinat *Object Space Point* (Papan 3D)

Akurasi Koordinat <i>Object Space Point</i>			
Id	X (mm)	Y (mm)	Z (mm)
1	0.00516	0.03106	0.00590
2	0.00575	0.03098	0.00593
3	0.00665	0.03075	0.00595
4	0.00778	0.03072	0.00598
5	0.00517	0.03091	0.00522
6	0.00575	0.03070	0.00522
7	0.00667	0.03065	0.00523
8	0.00779	0.03045	0.00525
9	0.00517	0.03074	0.00498
10	0.00577	0.03064	0.00497
11	0.00669	0.03040	0.00495
12	0.00781	0.03036	0.00495
13	0.00518	0.03070	0.00525
14	0.00579	0.03047	0.00520
15	0.00673	0.03042	0.00517
16	0.00784	0.03024	0.00514

4.2. Pembahasan

Kalibrasi kamera IP stereo ini menggunakan prinsip kolineariti yang lebih tepatnya menggunakan metode *self calibrating bundle adjustment*. Dengan metode ini akan menghasilkan parameter kalibrasi kamera, parameter EO, dan parameter koordinat *object space point*. Metode ini dibuat kedalam bahasa pemrograman *Matlab 2011b* dan akan menghasilkan parameter-parameter yang telah terkoreksi. Pemrograman dengan menggunakan *software Matlab 2011b* ini bertujuan untuk mempermudah dalam proses perhitungan kalibrasi kamera IP stereo.

4.2.1. Analisis Hasil Perhitungan Kalibrasi Kamera IP Stereo

Dalam proses perhitungan kalibrasi kamera IP stereo yang dihasilkan dengan menggunakan bahasa pemrograman *Matlab 2011b*. Dimana program tersebut akan menghasilkan parameter kalibrasi kamera, parameter EO terkoreksi dan parameter koordinat *object space point* terkoreksi. Parameter tersebut perlu dilakukan analisis untuk mengetahui selisih antara parameter pendekatan awal dengan parameter yang telah dikoreksi.

A. Papan Kalibrasi 2D

1. Analisis hasil parameter EO terkoreksi

Setelah dihitung menggunakan *Matlab 2011b* dapat dilihat nilai selisih parameter EO awal dengan parameter EO terkoreksi.

Tabel 4.12 Hasil Selisih Parameter EO Awal dengan Parameter EO terkoreksi untuk Papan 2D

Id Camera	$\Delta\omega$ (Derajat)	$\Delta\phi$ (Derajat)	$\Delta\kappa$ (Derajat)	ΔX_L (mm)	ΔY_L (mm)	ΔZ_L (mm)
Camera 1	6.58845	25.96385	3.25384	104.16660	-355.92662	-56.74164
Camera 2	9.04326	-25.79715	-0.83604	-193.04464	-397.84460	-48.02987

2. Analisis hasil parameter koordinat *object space point* terkoreksi

Setelah dihitung menggunakan *Matlab 2011b* dapat dilihat nilai selisih parameter koordinat *object space point* awal dengan parameter koordinat *object space point* terkoreksi

Tabel 4.13 Hasil Selisih Parameter Koordinat *Object Space Point* Awal dengan Parameter Koordinat *Object Space Point* Terkoreksi untuk Papan 2D

Id	ΔX (mm)	ΔY (mm)	ΔZ (mm)
1	-44.06735	126.70014	58.79815
2	-28.66084	79.56347	33.83114
3	-16.70275	41.44460	-0.10659
4	-28.57725	83.44752	19.67196
5	-18.03566	47.40713	17.03963
6	-9.10213	26.60597	4.53171
7	-6.63152	-10.71043	-23.77985
8	-10.34344	11.61355	-14.48864
9	-3.68763	15.80649	-0.00732
10	1.84694	12.74425	0.57355
11	-0.09582	-21.75915	-27.83749
12	-3.09254	-21.02530	-28.26590
13	10.49481	20.51663	8.02653
14	21.15673	39.92342	21.24517
15	12.03119	5.60817	-10.61571

B. Papan Kalibrasi 3D

1. Analisis hasil parameter EO terkoreksi

Setelah dihitung menggunakan *Matlab 2011b* dapat dilihat nilai selisih parameter EO awal dengan parameter EO terkoreksi.

Tabel 4.14 Hasil Selisih Parameter EO Awal dengan Parameter EO terkoreksi untuk Papan 3D

Id Camera	$\Delta \omega$ (Derajat)	$\Delta \phi$ (Derajat)	$\Delta \kappa$ (Derajat)	ΔX_L (mm)	ΔY_L (mm)	ΔZ_L (mm)
Camera 1	1.89494	-1.93028	-0.06272	-0.01730	-0.02142	0.00791
Camera 2	0.25071	-2.18082	0.01715	0.01926	-0.00990	-0.01297

2. Analisis hasil parameter koordinat *object space point* terkoreksi

Setelah dihitung menggunakan *Matlab 2011b* dapat dilihat nilai selisih parameter koordinat *object space point* awal dengan parameter koordinat *object space point*.

Tabel 4.15 Hasil Selisih Parameter Koordinat *Object Space Point* Awal dengan Parameter Koordinat *Object Space Point* Terkoreksi untuk Papan 3D

Id	ΔX (mm)	ΔY (mm)	ΔZ (mm)
1	-0.54555	-0.25011	0.15170
2	0.06444	-3.25070	-0.15530
3	-0.44632	-1.05120	0.00120
4	0.17192	-0.94027	0.06805
5	0.40931	2.25203	-0.19735
6	0.10072	-0.71170	-0.17413
7	0.48746	1.50092	-0.06840
8	-0.05138	2.56422	0.10240
9	0.16211	2.73443	0.08719
10	0.54982	-0.26257	-0.07318
11	-0.60796	-0.27555	0.42915
12	0.26123	3.16598	0.05821
13	-0.04232	1.02769	-0.07476
14	-0.41592	-3.13591	0.32421
15	-0.09137	-2.83050	-0.07788
16	-0.00619	-0.53676	-0.40110

4.2.2. Keunggulan dan Kelemahan Penggunaan *Matlab 2011b*

Keunggulan perhitungan menggunakan matlab adalah sebagai berikut :

1. Mudah untuk melakukan perhitungan matematis karena telah tersedia fitur otomatis untuk menghitung, seperti dapat disusun program untuk melakukan perhitungan secara fleksibel berapapun jumlah data yang diinput.
2. Bisa melakukan berapapun besar dimensi matriksnya, karena matlab memang khusus digunakan untuk menghitung matriks.

3. Penggunaan waktu yang lebih efisien saat proses perhitungan karena keseluruhan tahap dapat dihitung secara otomatis setelah listing kode disusun secara benar.

Sedangkan kelemahan dari penggunaan matlab ini yaitu inefisiensi waktu saat menyusun listing kode dan formnya. Untuk menghasilkan program yang dapat melakukan perhitungan secara efisien memerlukan persiapan yang lebih matang sehingga akan memerlukan waktu yang lebih banyak dalam melakukan persiapan tersebut.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Kesimpulan yang dapat diambil selama proses pelaksanaan penelitian mengenai “Pemrograman kalibrasi kamera IP stereo dengan menggunakan prinsip kolineariti untuk papan kalibrasi 2D dan 3D” ialah sebagai berikut :

1. Kamera IP merupakan kamera yang dapat melakukan pengambilan data dengan cepat, karena kamera IP menghasilkan data dalam bentuk format video (.avi).
2. Kamera IP mempunyai kekurangan yaitu memiliki resolusi yang kurang baik, kerana dipengaruhi oleh sensor (CMOS) yang kecil dan jarak antara panjang fokus dengan sensor (CMOS) pendek sehingga menghasilkan objek yang kurang baik.
3. Program kalibrasi kamera IP stereo membutuhkan inputan awal parameter IO masing-masing kamera, parameter EO masing-masing kamera, parameter koordinat *object space point*, dan koordinat foto masing-masing kamera.
4. Program kalibrasi kamera IP stereo mengeluarkan parameter kalibrasi kamera, parameter EO terkoreksi, akurasi parameter EO, parameter koordinat *object space point* terkoreksi, dan akurasi parameter koordinat *object space point*.
5. Dalam proses perhitungan dengan menggunakan program yang dibuat, parameter-parameter yang dihasilkan sangat baik.

6. Parameter-parameter yang dihasilkan sangat baik dapat dilihat dari standard deviasi dan akurasi parameter dari proses perhitungan tersebut.

6.2. Saran

Dalam proses penelitian ini terdapat beberapa hal yang belum dapat dipenuhi secara keseluruhan. Sehingga, berbagai kekurangan tersebut akan diakomodir dalam beberapa saran sebagai pertimbangan dalam proses pengembangan selanjutnya. Adapun saran-saran tersebut antara lain :

1. Pada aplikasi yang telah dibuat, inputan data masih dilakukan secara manual (menggunakan notepad). Sehingga, perlu ditambahkan sebuah *interface* yang digunakan untuk mengekstrak titik koordinat foto secara langsung dari foto, terdapat proses perhitungan relatif orientasi, *intersection*, dan *resection* sehingga didapat parameter pendekatan yang digunakan untuk proses kalibrasi kamera IP stereo.
2. Aplikasi yang telah dibuat digunakan untuk kamera stereo atau foto stereo, sehingga perlu dikembangkan lagi bagaimana untuk aplikasi kamera stereo dengan menggunakan banyak foto (multi foto).
3. Dalam proses perhitungan program kalibrasi kamera ip didapat akurasi yang kurang bagus pada papan kalibrasi 2D karena dalam proses ekstrak kornor dilakukan secara manual, hal ini disebabkan karena papan kalibrasi 2D tidak menggunakan *centroid* seperti halnya papan kalibrasi 3D.

DAFTAR PUSTAKA

- Ariff, M. F. B. M. 2005. *Pembangunan Sistem Perolehan Imej Fotogrametri Digital Jarak Dekat Bagi Pengukuran Craniofacial*. Kejuruteraan Geomatik, Fakulti Kejuruteraan Dan Sains Geoinformasi, Universiti Teknologi Malaysia.
- Atkinson, K. B. 1987. *Developments in Close Range Photogrammetry-1*. Applied Science Publishers. London.
- Atkinson, K. B. 2001. *Close Range Photogrammetry and Mechine Vision*. Whittles Publishing. Scotland, UK.
- Australis. 2004. *Users Manual*. Photometrix, Australia.
- Bradski, G. and A. Keahler. 2008. *Learning OpenCV*. Gravenstein Highway North, Sebastopol.
- Dorstel, C., Jacobsen, K. and Stallmann, D. 2004. *DMC – Photogrammetric Accuracy – Calibration Aspect And Generation Of Synthetic DMC Images*. University of Hannover, Germany.
- Fraser, C. S., M. R. Shortis and G. Ganci. 1995. *Multi-sensor system self-calibration*. Department of Geomatics, University of Melbourne, Parkville 3052, Australia.
- Fraser, C. S., 1997. *Digital Camera Self-Calibration*. ISPRS Journal Of Photogrammetry and Remote Sensing, 52(4) : 149-159.
- Fraser, C. S., Kenneth L. E. 2000. *Design and Implementation of a Computational Processing System for Off-line Digital Close Range Photogrammetry*. ISPRS Journal of Photogrammetry and Remote Sensing, 55(2): 94-104.
- Mikhail, J. S. Bethel, et al. 2001. *Introduction To Modern Photogrammetry*. New York, John Wiley and Sons, Inc.
- Pullivelli, Anoop. 2005. *Low-Cost Digital Cameras: Calibration, Stability Analysis, and Applications*. Tesis Magister Departemen of Geomatics Engineering. University of Calgary.
- Hanifa, N. R. 2007. *Studi Penggunaan Kamera Digital Low-Cost Non Metric Auto-Focus Untuk Pemantauan Deformasi*. Program Studi Teknik Geodesi

dan Geoinformatika, Fakultas Teknik Sipil dan Lingkungan, Institut Teknologi Bandung.

Hartono, DEA DESS. 2004. *Aplikasi Penginderaan Jauh dan Interpretasi Citra*. UGM, Yogyakarta.

Shortis, M. R. Clarke, T. A., and Short, T. 1994. *A Comparison Of Some Techniques For The Subpixel Location Of Discrete Target Images*. Videometrics III, SPIE Vol. 2350 239-250.

Tjahjadi, M. E. 2008. *Pemetaan Cepat Profil Jalan Dengan Mengintegrasikan Kamera CCTV dan DGPS*. ITN, Malang.

Tjahjadi, M. E. 2008b. *Precesion Feature Extraction Form Unmanned Aerialplatforms*. ITN, Malang.

King, B. A. 1993. *Methods For The Photogrammetric adjustment of Bundles of Constrained Streospairs*. Departement of Civil Engineering And Surveying. New South Wales, The University of Newcastle.

Wang, X. and T. A. Clarke. 1998. *Separate Adjustment of Close Range Photogrammetric Measurements*. ISPRS XXXII.

Wolf, P. R. 1993. *Element of Photogrammetry, Dengan Interpretasi Foto Udara dan Penginderaan Jauh*. UGM, Yogyakarta.

Wolf, P. R. and Dewitt, B. A. 2000. *Element Of Photogrammetry with Application in GIS*. Mc Graw Hill, New York.



Institut Teknologi Nasional Malang
Jl. Bendungan Sigura-gura No.2
Malang

**LEMBAR REVISI
SEMINAR HASIL SKRIPSI
PROGRAM STUDI TEKNIK GEODESI
FAKULTAS TEKNIK SIPIL & PERENCANAAN**

NAMA : Syahni
NIM : 09.25.026
HARI, TGL : Rabu, 8-8-2012

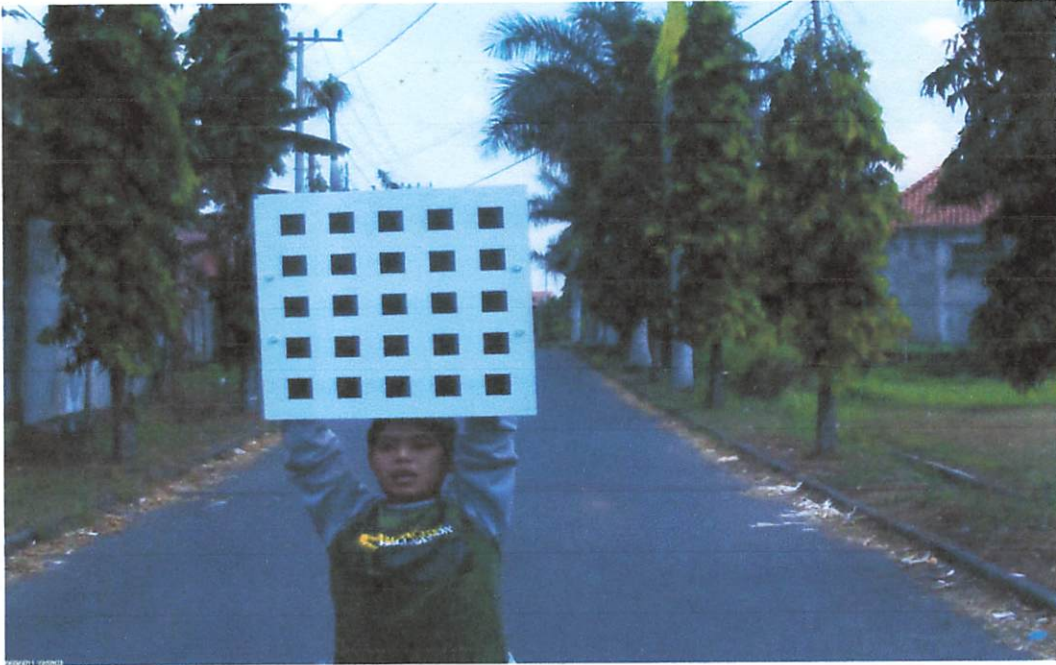
NO	MATERI REVISI
1	Selam tatel dimasukan
2	Kelengkapan dan kelengkapan kamern ip dimasukan
Ace Rulki 15/8 '12	

DOSEN PENGUJI

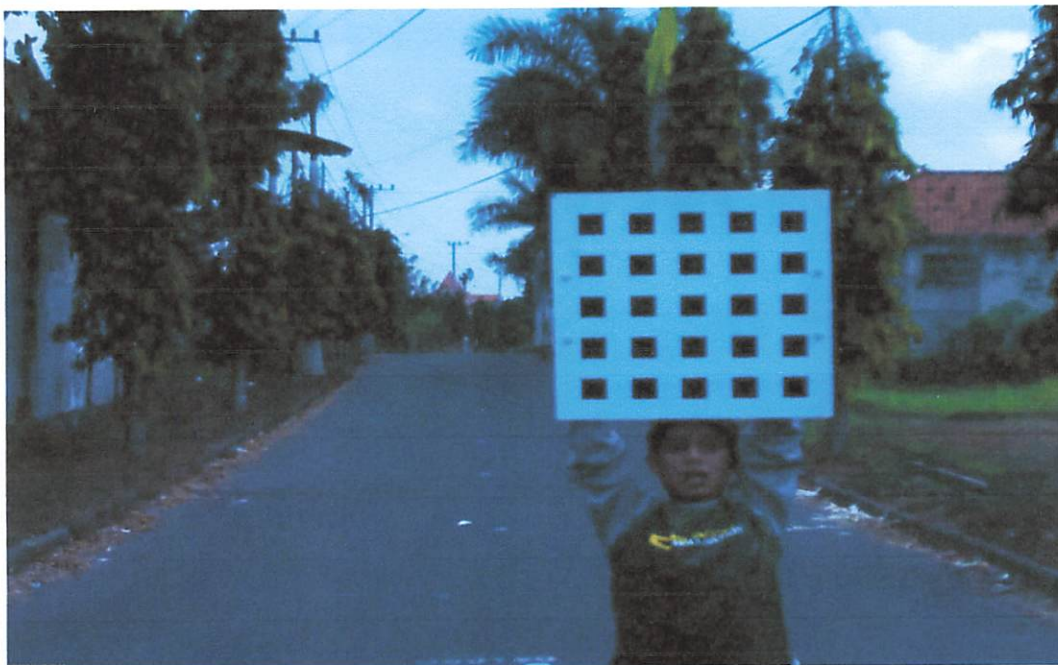
(.....)

LAMPIRAN A
DATA FOTO STEREO PAPAN KALIBRASI 2D
DAN 3D

A1. Data Foto Papan Kalibrasi 2D

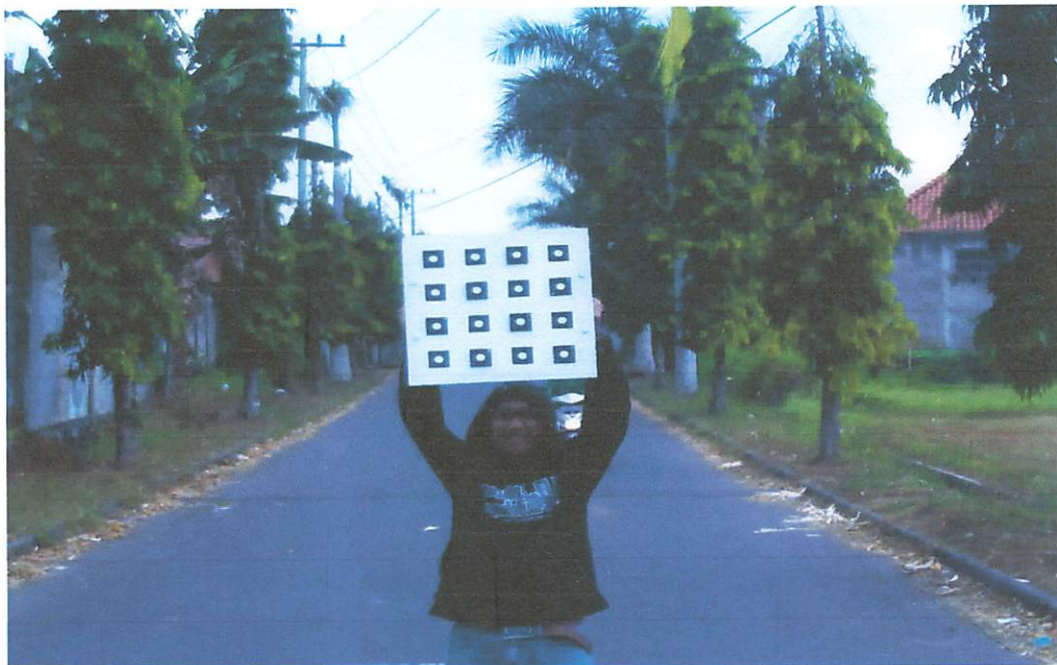


Gambar 1A. Data foto papan kalibrasi 2D (foto kiri)



Gambar 2A. Data foto papan kalibrasi 2D (foto kanan)

A2. Data Foto Papan Kalibrasi 3D



Gambar 3A. Data foto papan kalibrasi 3D (foto kiri)



Gambar 4A. Data foto papan kalibrasi 3D (foto kanan)

LAMPIRAN B
DATA INPUTAN PAPAN KALIBRASI 2D DAN 3D



B1. Data Inputan Papan Kalibrasi 2D

1. Parameter *Exterior Orientation*

Parameter <i>Exterior Orientation</i>						
Id Camera	ω	ϕ	κ	X_L	Y_L	Z_L
Camera 1	95.5627	7.4769	-0.4211	377.7428	1251.6292	-203.2746
Camera 2	96.7675	-17.3789	3.8190	-1434.4207	1457.2073	-249.2134

2. Parameter Koordinat *Object Space Point*

Parameter Koordinat <i>Object Space Point</i>			
Id	X	Y	Z
1	-1105.8210	2135.9859	738.2672
2	-963.3580	2096.1789	741.5219
3	-966.3317	2077.4421	603.8102
4	-1103.8565	2114.4990	595.7248
5	-832.9482	2065.1093	748.3715
6	-697.4833	2040.1039	755.1225
7	-699.5314	2018.5478	615.8196
8	-834.5506	2047.4112	610.6296
9	-564.1299	2020.0132	760.1018
10	-430.1412	2001.4634	761.9233
11	-426.3491	1982.7518	626.4450
12	-568.0706	1999.3484	621.1620
13	-300.7720	1990.6413	763.9131
14	-166.0994	1986.9160	767.6255
15	-157.2288	1967.5177	630.7742
16	-294.9119	1973.6913	629.4169
17	-37.2602	1985.2109	766.0368
18	96.2626	1991.6150	765.9235
19	107.0034	1971.2222	632.0717
20	-29.5348	1966.4960	630.2709
21	-1102.5095	2100.7123	456.2503
22	-965.9373	2059.8454	463.4566
23	-968.0787	2052.1325	328.3699
24	-1103.2377	2087.5704	320.4893
25	-833.1093	2030.7824	472.9189
26	-697.8038	2005.4110	478.9767
27	-693.6112	1995.2646	340.2005
28	-834.0487	2021.5807	333.3250
29	-562.1797	1984.3640	483.5651

30	-424.1457	1968.4537	487.1906
31	-419.8225	1958.7888	346.9245
32	-560.7056	1975.8126	345.6149
33	-290.5331	1957.9091	491.3387
34	-150.7824	1952.3095	495.5017
35	-144.6738	1942.5769	356.7144
36	-286.2983	1948.4151	349.9830
37	-17.5777	1953.5023	498.0549
38	124.9898	1960.7706	498.4372
39	130.7559	1948.6531	357.7155
40	-13.1551	1942.2262	358.1757
41	-1099.2174	2082.6550	182.4985
42	-966.4171	2047.7916	188.2706
43	-958.1362	2046.7225	50.8937
44	-1092.0859	2082.5040	42.1847
45	-830.0308	2017.6488	195.0854
46	-692.4271	1991.1559	198.1393
47	-685.9908	1989.7940	59.0094
48	-826.8440	2017.6335	53.6929
49	-555.9923	1970.0062	205.2916
50	-417.6738	1954.3069	208.4185
51	-412.1370	1954.5035	66.0153
52	-551.4768	1970.3457	62.5605
53	-282.2534	1943.6417	212.6301
54	-141.0872	1937.7023	216.2181
55	-139.6238	1937.4023	75.5382
56	-278.7623	1942.9880	69.4318
57	-5.5162	1937.6100	219.0026
58	132.2576	1943.8631	220.6625
59	127.4460	1942.5023	82.7780
60	-4.4943	1937.7319	77.6826
61	-1088.2476	2086.5362	-90.3328
62	-952.8119	2052.2597	-86.3656
63	-944.1802	2064.5443	-224.9118
64	-1079.7182	2100.2878	-230.7870
65	-819.4646	2022.1349	-83.0574
66	-681.5372	1995.9870	-77.7189
67	-672.7358	2006.4555	-219.8683
68	-810.9316	2034.1661	-224.6576
69	-546.4485	1976.0417	-74.8508

70	-405.1369	1959.4245	-70.2575
71	-400.5347	1969.5685	-210.4853
72	-538.9064	1985.5698	-215.8757
73	-272.0586	1948.5542	-64.6131
74	-131.0964	1942.7187	-59.9579
75	-127.2744	1953.2702	-201.6986
76	-267.3139	1959.2705	-207.8648
77	-1.1077	1942.4979	-57.4857
78	131.7743	1947.0269	-52.3621
79	135.9735	1957.1812	-188.9208
80	0.2683	1952.5392	-196.0969
81	-1066.2081	2111.6551	-352.7371
82	-937.7241	2079.7142	-358.8713
83	-922.5832	2093.8606	-484.6529
84	-1055.2201	2128.3541	-486.2184
85	-804.5453	2049.4193	-357.3031
86	-663.7244	2023.6626	-354.3379
87	-657.9564	2041.7126	-481.2492
88	-793.0824	2067.6706	-485.3925
89	-533.0307	2003.3624	-348.4538
90	-395.4359	1987.5326	-345.0859
91	-387.5774	2005.5447	-472.8881
92	-523.5947	2021.9689	-480.9923
93	-262.4608	1973.9602	-339.0993
94	-126.1954	1967.5392	-332.7472
95	-122.6715	1986.2894	-460.6270
96	-254.6636	1996.8337	-479.4969
97	5.3129	1968.2461	-325.7046
98	135.9584	1972.0694	-320.5222
99	138.1712	1992.6965	-452.2019
100	8.4388	1988.9210	-460.9034

3. Parameter Koordinat Foto

Id	Camera 1		Camera 2	
	x	y	x	y
1	-1.3663	0.7865	0.2441	0.8114
2	-1.2347	0.7843	0.3872	0.8159
3	-1.2305	0.6501	0.3886	0.6795
4	-1.3620	0.6432	0.2484	0.6772
5	-1.0990	0.7934	0.5203	0.8182

6	-0.9660	0.7956	0.6620	0.8182
7	-0.9632	0.6569	0.6605	0.6840
8	-1.0990	0.6523	0.5217	0.6817
9	-0.8317	0.7979	0.7909	0.8228
10	-0.6987	0.8002	0.9325	0.8273
11	-0.6945	0.6660	0.9339	0.6863
12	-0.8288	0.6614	0.7909	0.6817
13	-0.5644	0.8070	1.0685	0.8273
14	-0.4272	0.8115	1.2087	0.8296
15	-0.4215	0.6751	1.2115	0.6863
16	-0.5615	0.6728	1.0685	0.6795
17	-0.2900	0.8184	1.3447	0.8296
18	-0.1528	0.8206	1.4835	0.8319
19	-0.1471	0.6842	1.4849	0.6908
20	-0.2843	0.6819	1.3461	0.6840
21	-1.3578	0.5068	0.2470	0.5430
22	-1.2277	0.5136	0.3872	0.5430
23	-1.2220	0.3839	0.3900	0.4156
24	-1.3535	0.3771	0.2470	0.4134
25	-1.0891	0.5204	0.5217	0.5475
26	-0.9590	0.5204	0.6605	0.5521
27	-0.9547	0.3930	0.6634	0.4156
28	-1.0877	0.3862	0.5232	0.4111
29	-0.8260	0.5318	0.7951	0.5498
30	-0.6888	0.5295	0.9382	0.5544
31	-0.6860	0.3930	0.9396	0.4179
32	-0.8218	0.3953	0.7979	0.4134
33	-0.5545	0.5386	1.0713	0.5498
34	-0.4159	0.5386	1.2143	0.5544
35	-0.4130	0.4067	1.2143	0.4247
36	-0.5531	0.3999	1.0755	0.4134
37	-0.2801	0.5432	1.3489	0.5544
38	-0.1429	0.5500	1.4877	0.5566
39	-0.1387	0.4135	1.4891	0.4224
40	-0.2759	0.4067	1.3489	0.4179
41	-1.3479	0.2452	0.2498	0.2723
42	-1.2178	0.2452	0.3914	0.2769
43	-1.2135	0.1110	0.3929	0.1450
44	-1.3436	0.1065	0.2512	0.1359
45	-1.0834	0.2498	0.5260	0.2769
46	-0.9505	0.2498	0.6676	0.2746
47	-0.9462	0.1178	0.6690	0.1450

48	-1.0806	0.1156	0.5274	0.1427
49	-0.8161	0.2589	0.8008	0.2792
50	-0.6803	0.2611	0.9410	0.2792
51	-0.6775	0.1224	0.9438	0.1427
52	-0.8105	0.1201	0.8022	0.1404
53	-0.5446	0.2657	1.0741	0.2792
54	-0.4060	0.2702	1.2143	0.2905
55	-0.4031	0.1338	1.2200	0.1495
56	-0.5446	0.1315	1.0812	0.1381
57	-0.2716	0.2748	1.3517	0.2883
58	-0.1358	0.2793	1.4891	0.2860
59	-0.1288	0.1383	1.4905	0.1495
60	-0.2674	0.1338	1.3546	0.1427
61	-1.3394	-0.0255	0.2555	0.0062
62	-1.2079	-0.0209	0.3943	0.0108
63	-1.2022	-0.1551	0.3985	-0.1280
64	-1.3352	-0.1619	0.2555	-0.1348
65	-1.0749	-0.0141	0.5274	0.0039
66	-0.9392	-0.0141	0.6662	0.0108
67	-0.9363	-0.1506	0.6733	-0.1302
68	-1.0707	-0.1528	0.5331	-0.1393
69	-0.8062	-0.0118	0.8008	0.0108
70	-0.6690	-0.0118	0.9424	0.0153
71	-0.6676	-0.1437	0.9452	-0.1280
72	-0.8020	-0.1460	0.8050	-0.1325
73	-0.5361	-0.0027	1.0784	0.0130
74	-0.3975	-0.0004	1.2214	0.0176
75	-0.3932	-0.1392	1.2257	-0.1257
76	-0.5333	-0.1460	1.0784	-0.1257
77	-0.2631	-0.0004	1.3532	0.0130
78	-0.1245	0.0018	1.4920	0.0153
79	-0.1203	-0.1301	1.4934	-0.1234
80	-0.2589	-0.1346	1.3546	-0.1280
81	-1.3281	-0.2870	0.2540	-0.2576
82	-1.1994	-0.2870	0.3957	-0.2576
83	-1.1923	-0.4212	0.4028	-0.3941
84	-1.3238	-0.4258	0.2583	-0.3986
85	-1.0636	-0.2870	0.5345	-0.2599
86	-0.9293	-0.2893	0.6719	-0.2553
87	-0.9264	-0.4167	0.6733	-0.3941
88	-1.0594	-0.4212	0.5317	-0.3964
89	-0.7977	-0.2802	0.8022	-0.2599

90	-0.6634	-0.2802	0.9438	-0.2576
91	-0.6577	-0.4144	0.9467	-0.3895
92	-0.7921	-0.4167	0.8064	-0.4032
93	-0.5262	-0.2802	1.0798	-0.2553
94	-0.3890	-0.2757	1.2214	-0.2553
95	-0.3862	-0.4098	1.2243	-0.3895
96	-0.5234	-0.4167	1.0826	-0.4009
97	-0.2532	-0.2688	1.3546	-0.2553
98	-0.1174	-0.2711	1.4905	-0.2508
99	-0.1118	-0.4030	1.4962	-0.3895
100	-0.2476	-0.4076	1.3560	-0.3941

B2. Data Inputan Papan Kalibrasi 3D

1. Parameter *Exterior Orientation*

Parameter <i>Exterior Orientation</i>						
Id Camera	ω	ϕ	κ	X_L	Y_L	Z_L
Camera 1	91.3392	1.8219	-0.1365	-14.9863	-868.6242	12.1238
Camera 2	89.5747	0.4297	0.6720	-1021.6935	-880.8342	129.6603

2. Parameter Koordinat *Object Space Point*

Parameter Koordinat <i>Object Space Point</i>			
Id	X	Y	Z
1	-397.5475	2263.0779	385.7357
2	-233.6084	2261.0619	393.2154
3	-70.8481	2252.0435	400.7835
4	91.5388	2252.6029	406.8614
5	-389.5752	2256.2559	223.2743
6	-226.5728	2247.7514	231.6363
7	-64.9925	2247.4059	236.6240
8	99.0460	2239.7870	245.8784
9	-382.7588	2248.6939	59.5915
10	-219.5671	2245.8601	69.5335
11	-56.9320	2235.6275	77.6000
12	103.8854	2235.7134	83.9004
13	-374.3863	2247.5080	-99.3521
14	-211.1133	2237.7447	-91.1385
15	-50.4564	2237.6215	-83.9497
16	111.0263	2230.6450	-77.4443

3. Parameter Koordinat Foto

Id	Camera 1		Camera 2	
	x	y	x	y
1	-0.5491	0.5229	0.7403	0.4643
2	-0.3217	0.5337	0.9666	0.4719
3	-0.0963	0.5464	1.1926	0.4802
4	0.1313	0.5551	1.4157	0.4850
5	-0.5390	0.3045	0.7497	0.2463
6	-0.3144	0.3178	0.9757	0.2551
7	-0.0873	0.3250	1.1984	0.2595
8	0.1422	0.3388	1.4270	0.2700
9	-0.5327	0.0842	0.7566	0.0255
10	-0.3048	0.0978	0.9824	0.0360
11	-0.0781	0.1097	1.2096	0.0453
12	0.1496	0.1185	1.4318	0.0510
13	-0.5226	-0.1320	0.7647	-0.1898
14	-0.2957	-0.1210	0.9925	-0.1818
15	-0.0685	-0.1119	1.2155	-0.1741
16	0.1600	-0.1025	1.4413	-0.1687

LAMPIRAN C
DATA OUTPUT PAPAN KALIBRASI 2D DAN 3D

C1. Data Output Papan Kalibrasi 2D

1. Parameter Kalibrasi Kamera

Parameter Kalibrasi Kamera		
Parameter	Camera 1 (mm)	Camera 2 (mm)
Fokus	3.99777	3.02067
x_0	1.28390	-1.51496
y_0	-0.06913	-0.25913
K_1	0.56521	0.38071
K_2	-0.29234	-0.20960
K_3	0.04181	0.02461
P_1	-0.44836	0.54214
P_2	-0.05250	-0.06692
b_1	-0.89863	-1.29360
b_2	-0.28534	0.20905

2. Parameter *Exterior Orientation* dan Akurasi Parameter *Exterior Orientation*

Parameter <i>Exterior Orientation</i>						
Id Camera	ω (Derajat)	ϕ (Derajat)	κ (Derajat)	X_L (mm)	Y_L (mm)	Z_L (mm)
Camera 1	102.15115	33.44075	2.83274	481.90942	895.70255	-260.01619
Camera 2	105.81076	-43.17605	2.98297	-1627.46538	1059.36275	-297.24324

Akurasi Parameter <i>Exterior Orientation</i>						
Id Camera	ω (Derajat)	ϕ (Derajat)	κ (Derajat)	X_L (mm)	Y_L (mm)	Z_L (mm)
Camera 1	0.000002	0.000002	0.000002	0.001766	0.002419	0.002611
Camera 2	0.000002	0.000002	0.000002	0.001674	0.002284	0.002428

3. Parameter Koordinat *Object Space Point*

Parameter Koordinat <i>Object Space Point</i>			
Id	X	Y	Z
1	-1149.88835	2262.68604	797.06535
2	-992.01884	2175.74237	775.35304
3	-983.03445	2118.88670	603.70361
4	-1132.43375	2197.94652	615.39676
5	-850.98386	2112.51643	765.41113
6	-706.58543	2066.70987	759.65421
7	-706.16292	2007.83737	592.03975
8	-844.89404	2059.02475	596.14096

9	-567.81753	2035.81969	760.09448
10	-428.29426	2014.20765	762.49685
11	-426.44492	1960.99266	598.60751
12	-571.16314	1978.32310	592.89610
13	-290.27719	2011.15793	771.93963
14	-144.94267	2026.83942	788.87067
15	-145.19761	1973.12587	620.15849
16	-289.53936	1960.61284	606.98749
17	1.34802	2051.60548	808.28195
18	155.85288	2098.40155	835.41164
19	149.45757	2041.33102	660.99707
20	-3.56569	1998.96514	637.91386
21	-1120.18138	2155.65787	455.65517
22	-975.55329	2071.70999	446.00720
23	-972.51385	2046.51348	304.32459
24	-1114.42681	2120.35186	310.49036
25	-839.36869	2013.79449	443.18696
26	-702.63397	1969.51213	441.33200
27	-696.01382	1942.41375	300.03339
28	-836.80784	1987.17173	299.04616
29	-564.74982	1938.45496	443.04530
30	-427.56984	1922.18207	446.66128
31	-424.99462	1895.86641	303.60017
32	-564.20773	1913.44356	301.95233
33	-289.86664	1919.81581	456.02429
34	-146.23054	1932.96881	468.77536
35	-144.48617	1907.07887	324.99121
36	-290.04358	1894.46533	311.06927
37	-3.79701	1962.25602	485.00784
38	151.60747	2008.44995	503.21087
39	148.93042	1977.39941	351.87846
40	-5.25405	1933.18381	337.61072
41	-1103.93162	2104.37741	171.47567
42	-967.27812	2032.28178	166.06267
43	-957.87428	2030.57449	37.85659
44	-1094.28974	2103.24246	38.56690
45	-831.10219	1974.84780	164.54773
46	-695.02183	1929.49569	162.09482
47	-688.78161	1928.03598	34.02332
48	-827.59595	1974.83243	33.54415

49	-560.43884	1898.93650	166.91102
50	-423.74268	1883.56863	170.53813
51	-419.04247	1884.63369	38.92583
52	-556.63344	1899.71796	35.21447
53	-287.26835	1881.54275	178.08670
54	-142.95016	1894.35444	187.66968
55	-143.31429	1895.26851	56.39479
56	-285.28717	1882.10719	45.82656
57	-1.50839	1921.32852	199.82988
58	145.07379	1964.55570	214.35355
59	140.17237	1963.84472	81.87092
60	-2.51955	1922.52859	66.37191
61	-1089.21974	2112.47005	-82.95456
62	-952.39606	2043.70055	-86.47966
63	-943.56531	2073.17847	-208.62371
64	-1080.50010	2145.11416	-207.67975
65	-820.49083	1987.18116	-89.64278
66	-684.93630	1942.76786	-88.52723
67	-676.23975	1971.04958	-212.92588
68	-812.27240	2017.26775	-213.78248
69	-551.71247	1914.36474	-87.68297
70	-411.46191	1897.75081	-83.66205
71	-406.48742	1925.32067	-206.35563
72	-544.21672	1941.05207	-211.23806
73	-277.83172	1895.86607	-75.79029
74	-134.53263	1908.88233	-67.19343
75	-130.01183	1937.54780	-192.81392
76	-272.17046	1924.01252	-202.68088
77	1.39928	1934.93938	-58.34618
78	143.87973	1976.04262	-44.70990
79	149.44759	2002.64146	-167.97607
80	4.54536	1962.09968	-182.13900
81	-1070.87743	2177.81605	-315.74086
82	-939.20058	2112.57636	-325.21568
83	-927.35067	2154.79590	-435.87764
84	-1063.87238	2224.44769	-433.73764
85	-807.56415	2056.04398	-328.33273
86	-667.25638	2013.70375	-329.13814
87	-662.87672	2062.55133	-438.87780
88	-797.89665	2105.80193	-439.88811

89	-538.15372	1984.50037	-325.83698
90	-400.90647	1969.45904	-323.06349
91	-391.54771	2018.44013	-433.97835
92	-528.90084	2035.60045	-440.36844
93	-265.58491	1963.00102	-317.70272
94	-125.81445	1974.61913	-309.07070
95	-119.76895	2024.83431	-421.33057
96	-256.84827	2021.71965	-438.60671
97	11.93204	2002.14837	-297.68597
98	152.67615	2040.90776	-288.05311
99	159.32254	2096.53126	-405.73503
100	19.02406	2056.55758	-417.55756

C2. Data Output Papan Kalibrasi 3D

1. Parameter Kalibrasi Kamera

Parameter Kalibrasi Kamera		
Parameter	Camera 1 (mm)	Camera 2 (mm)
Fokus	2.27236	2.26591
x_0	0.00750	-0.01457
y_0	-0.18961	-0.08434
K_1	0.01569	-0.02840
K_2	-0.08158	0.01258
K_3	0.08376	-0.00263
P_1	0.00041	0.01122
P_2	0.00238	0.00032
b_1	0.02434	-0.00525
b_2	0.00304	0.00326

2. Parameter *Exterior Orientation* dan Akurasi Parameter *Exterior Orientation*

Parameter <i>Exterior Orientation</i>						
Id Camera	ω (Derajat)	ϕ (Derajat)	κ (Derajat)	X_L (mm)	Y_L (mm)	Z_L (mm)
Camera 1	93.23414	-0.10838	-0.19922	-15.00356	-868.64563	12.13169
Camera 2	89.82541	-1.75112	0.68916	-1021.67420	-880.84406	129.64735



Akurasi Parameter <i>Exterior Orientation</i>						
Id Camera	ω (Derajat)	ϕ (Derajat)	κ (Derajat)	X_L (mm)	Y_L (mm)	Z_L (mm)
Camera 1	0.00009	0.00009	0.00001	0.27619	0.03826	0.27062
Camera 2	0.00005	0.00005	0.00001	0.15027	0.06685	0.16060

3. Parameter Koordinat *Object Space Point*

Parameter Koordinat <i>Object Space Point</i>			
Id	X (mm)	Y (mm)	Z (mm)
1	-398.09306	2262.82779	385.88740
2	-233.54396	2257.81120	393.06010
3	-71.29442	2250.99230	400.78470
4	91.71072	2251.66263	406.92945
5	-389.16589	2258.50794	223.07695
6	-226.47208	2247.03970	231.46218
7	-64.50504	2248.90682	236.55560
8	98.99462	2242.35122	245.98080
9	-382.59669	2251.42833	59.67869
10	-219.01728	2245.59753	69.46032
11	-57.53996	2235.35195	78.02915
12	104.14663	2238.87938	83.95861
13	-374.42862	2248.53569	-99.42686
14	-211.52922	2234.60879	-90.81429
15	-50.54777	2234.79100	-84.02758
16	111.02011	2230.10824	-77.84540

LAMPIRAN D
LIST CODE PROGRAM APLIKASI

D1. Listing Fungsi Utama (*Main Function*)

```
function varargout = Stereo_Calibration(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @Stereo_Calibration_OpeningFcn, ...
                  'gui_OutputFcn',    @Stereo_Calibration_OutputFcn, ...
                  'gui_LayoutFcn',    [] , ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before Stereo_Calibration is made visible.
function Stereo_Calibration_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Stereo_Calibration (see
VARARGIN)

% Choose default command line output for Stereo_Calibration
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Stereo_Calibration wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = Stereo_Calibration_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
```

```

varargout{1} = handles.output;

% --- Executes during object creation, after setting all
properties.
function Focal_Lenght1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all
properties.
function X_Nol1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all
properties.
function Y_Nol1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all
properties.
function Focal_Lenght2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all
properties.
function X_Nol2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all
properties.
function Y_Nol2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all
properties.
function Focal_Lenght3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```
% --- Executes during object creation, after setting all
properties.
```

```
function X_Nol3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes during object creation, after setting all
properties.
```

```
function Y_Nol3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes during object creation, after setting all
properties.
```

```
function Ksatu_1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes during object creation, after setting all
properties.
```

```
function Kdua_1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes during object creation, after setting all
properties.
```

```
function Ktiga_1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes during object creation, after setting all
properties.
```

```
function Psatu_1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes during object creation, after setting all
properties.
```

```
function Pdua_1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```

% --- Executes during object creation, after setting all
properties.
function bsatu_1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all
properties.
function bdua_1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all
properties.
function Focal_Lenght4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all
properties.
function X_Nol4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all
properties.
function Y_Nol4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all
properties.
function Ksatu_2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all
properties.
function Kdua_2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes during object creation, after setting all
properties.
function Ktiga_2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all
properties.
function Psatu_2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all
properties.
function Pdua_2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all
properties.
function bsatu_2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all
properties.
function bdua_2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all
properties.
function Akurasi_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% Input Data Parameter IO Camera 1
function Focal_Lenght1_Callback(hObject, eventdata, handles)
edit_foc1 = guidata(gcbo);
foc_lenght1 = str2double(get(edit_foc1.Focal_Lenght1,'String'));
set(edit_foc1.Focal_Lenght1,'Userdata',foc_lenght1);

function X_Noll_Callback(hObject, eventdata, handles)
edit_xo1 = guidata(gcbo);
xo1 = str2double(get(edit_xo1.X_Noll,'String'));

```

```

set(edit_xo1.X_Nol1,'Userdata',xo1);

function Y_Nol1_Callback(hObject, eventdata, handles)
edit_yo1 = guidata(gcbo);
yo1 = str2double(get(edit_yo1.Y_Nol1,'String'));
set(edit_yo1.Y_Nol1,'Userdata',yo1);

% Input Data Parameter IO Camera 2
function Focal_Lenght2_Callback(hObject, eventdata, handles)
edit_foc2 = guidata(gcbo);
foc_lenght2 = str2double(get(edit_foc2.Focal_Lenght2,'String'));
set(edit_foc2.Focal_Lenght2,'Userdata',foc_lenght2);

function X_Nol2_Callback(hObject, eventdata, handles)
edit_xo2 = guidata(gcbo);
xo2 = str2double(get(edit_xo2.X_Nol2,'String'));
set(edit_xo2.X_Nol2,'Userdata',xo2);

function Y_Nol2_Callback(hObject, eventdata, handles)
edit_yo2 = guidata(gcbo);
yo2 = str2double(get(edit_yo2.Y_Nol2,'String'));
set(edit_yo2.Y_Nol2,'Userdata',yo2);

% Input Data Parameter_EO.
function Parameter_EO_Callback(hObject, eventdata, handles)
ParEO = guidata(gcbo);
[fileName,directory] = uigetfile('*.txt','Open Data File');
if isequal([fileName,directory],[0,0])
    return
else
eval([' cd '' directory '' ;']);
eval(['EO = load('' fileName '')']);
end
set(ParEO.Parameter_EO,'Userdata',EO);

% Input Data Object_Space.
function Object_Space_Callback(hObject, eventdata, handles)
ObjectS = guidata(gcbo);
[fileName,directory] = uigetfile('*.txt','Open Data File');
if isequal([fileName,directory],[0,0])
    return
else
eval([' cd '' directory '' ;']);
eval(['OS = load('' fileName '')']);
end
set(ObjectS.Object_Space,'Userdata',OS);

% Input Data Image_Coordinates.
function Image_Coordinates_Callback(hObject, eventdata, handles)
ImageCoor = guidata(gcbo);
[fileName,directory] = uigetfile('*.txt','Open Data File');
if isequal([fileName,directory],[0,0])
    return
else
eval([' cd '' directory '' ;']);

```

```

eval(['KF = load('' fileName '')']);
end
set(ImageCoor.Image_Coordinates, 'Userdata', KF);

% --- Executes on button press in Proses.
function Proses_Callback(hObject, eventdata, handles)
hitung = guidata(gcbo);
% Panggil Parameter IO Camera 1
foc1 = get(hitung.Focal_Lenght1, 'Userdata');
xo_1 = get(hitung.X_Nol1, 'Userdata');
yo_1 = get(hitung.Y_Nol1, 'Userdata');

% Panggil Parameter IO Camera 2
foc2 = get(hitung.Focal_Lenght2, 'Userdata');
xo_2 = get(hitung.X_Nol2, 'Userdata');
yo_2 = get(hitung.Y_Nol2, 'Userdata');

% Panggil Paramater EO
Eks_Or = get(hitung.Parameter_EO, 'Userdata');
ncam = size(Eks_Or, 1);
npareo = length(Eks_Or);
eks = reshape(Eks_Or, ncam, npareo);

% Panggil Parameter Object Space
Obj_S = get(hitung.Object_Space, 'Userdata');
npoint = length(Obj_S);
nos = size(Obj_S, 2);
pts3D = reshape(Obj_S, npoint, nos);

% Panggil Parameter Image Coordinates
Image_Cor = get(hitung.Image_Coordinates, 'Userdata');
nico = size(Image_Cor, 2);
pts2D = reshape(Image_Cor, npoint, nico);

M = MatriksM(eks);
rsq = Rsqform(pts3D, eks);
assignin('base', 'rsq', rsq);
assignin('base', 'M', M);

% Proses Bundle Adjustment
convergen = 5;
for iter=1:50
    [N, T] =
Matriks_Normal(foc1, foc2, xo_1, yo_1, xo_2, yo_2, pts2D, pts3D, eks);
    xx = pinv(N)*T;
    assignin('base', 'xx', xx);

    % Proses koreksi parameter eo, koordinat foto dan koordinat
    tiga-dimensi obyek
    ncam = size(eks, 1);
    np2 = size(pts2D, 1);
    np3 = size(pts3D, 1);
    cam_par = 6;
    p2_par = 10;
    p3_par = 3;

```

```

camcor = xx(1:ncam*cam_par);
p2cor = xx(ncam*cam_par+1:ncam*cam_par+2*p2_par);
p3cor =
xx(ncam*cam_par+2*p2_par+1:ncam*cam_par+2*p2_par+np3*p3_par);

delcam = reshape(camcor, cam_par, ncam)';
delp2 = reshape(p2cor, p2_par, ncam)';
delp3 = reshape(p3cor, p3_par, np3)';

% Proses koreksi nilai parameter eo
for i=1:ncam
    for j=1:3
        eks_cor(i,j) = eks(i,j)+(delcam(i,j)*180/pi);
    end
    for s=4:6
        eks_cor(i,s) = eks(i,s)+delcam(i,s);
    end
end
assignin('base','eks_cor',eks_cor);

% Proses koreksi nilai parameter koordinat foto
for k=1:np2
    rsq = Rsqform(pts3D, eks);
    x1 = pts2D(k,1);
    y1 = pts2D(k,2);
    x2 = pts2D(k,3);
    y2 = pts2D(k,4);

    % Kalibrasi camera 1
    xo_1cor = xo_1+delp2(1,1);
    yo_1cor = yo_1+delp2(1,2);
    foc1cor = foc1+delp2(1,3);
    K1_1 = delp2(1,4);
    K2_1 = delp2(1,5);
    K3_1 = delp2(1,6);
    P1_1 = delp2(1,7);
    P2_1 = delp2(1,8);
    b1_1 = delp2(1,9);
    b2_1 = delp2(1,10);

    sls(k,1) = x1-xo_1;
    sls(k,2) = y1-yo_1;
    r(k,1) = sqrt((sls(k,1))^2+(sls(k,2))^2);

    fcq(k,1) = foc1cor*(rsq(k,1)/rsq(k,3));
    fcq(k,2) = foc1cor*(rsq(k,2)/rsq(k,3));

    delta(k,1) = -xo_1cor-
    (foc1cor*(sls(k,1)/foc1))+sls(k,1)*r(k,1)^2*K1_1+sls(k,1)*r(k,1)^4
    *K2_1+sls(k,1)*r(k,1)^6*K3_1+(r(k,1)^2+2*sls(k,1)^2)*P1_1+2*P2_1*sls
    (k,1)*sls(k,2)+b1_1*sls(k,1)+b2_1*sls(k,2);
    delta(k,2) = -yo_1cor-
    (foc1cor*(sls(k,2)/foc1))+sls(k,2)*r(k,1)^2*K1_1+sls(k,2)*r(k,1)^4
    *K2_1+sls(k,2)*r(k,1)^6*K3_1+2*P1_1*sls(k,1)*sls(k,2)+(r(k,1)^2+2*
    sls(k,2)^2)*P2_1;

    % Kalibrasi camera 2

```

```

xo_2cor = xo_2+delp2(2,1);
yo_2cor = yo_2+delp2(2,2);
foc2cor = foc2+delp2(2,3);
K1_2 = delp2(2,4);
K2_2 = delp2(2,5);
K3_2 = delp2(2,6);
P1_2 = delp2(2,7);
P2_2 = delp2(2,8);
b1_2 = delp2(2,9);
b2_2 = delp2(2,10);

sls(k,3) = x2-xo_2;
sls(k,4) = y2-yo_2;
r(k,2) = sqrt((sls(k,3))^2+(sls(k,4))^2);

fcq(k,3) = foc2*(rsq(k,4)/rsq(k,6));
fcq(k,4) = foc2*(rsq(k,5)/rsq(k,6));

delta(k,3) = -xo_2cor-
(foc2cor*(sls(k,3)/foc2))+sls(k,3)*r(k,2)^2*K1_2+sls(k,3)*r(k,2)^4
*K2_2+sls(k,3)*r(k,2)^6*K3_2+(r(k,2)^2+2*sls(k,3)^2)*P1_2+2*P2_2*s
ls(k,3)*sls(k,4)+b1_2*sls(k,3)+b2_2*sls(k,4);
delta(k,4) = -yo_2cor-
(foc2cor*(sls(k,4)/foc2))+sls(k,4)*r(k,2)^2*K1_2+sls(k,4)*r(k,2)^4
*K2_2+sls(k,4)*r(k,2)^6*K3_2+2*P1_2*sls(k,3)*sls(k,4)+(r(k,2)^2+2*
sls(k,4)^2)*P2_2;

%Koreksi koordinat foto
pts2D_cor(k,1) = sls(k,1)-delta(k,1)+fcq(k,1);
pts2D_cor(k,2) = sls(k,2)-delta(k,2)+fcq(k,2);
pts2D_cor(k,3) = sls(k,3)-delta(k,3)+fcq(k,3);
pts2D_cor(k,4) = sls(k,4)-delta(k,4)+fcq(k,4);
assignin('base','pts2D_cor',pts2D_cor);
end

% Proses koreksi nilai parameter koordinat tiga-dimensi obyek
for l=1:np3
X = pts3D(l,1);
Y = pts3D(l,2);
Z = pts3D(l,3);

pts3D_cor(l,1) = X+delp3(l,1);
pts3D_cor(l,2) = Y+delp3(l,2);
pts3D_cor(l,3) = Z+delp3(l,3);
assignin('base','pts3D_cor',pts3D_cor);
end
cor = sum(abs(xx));
if(cor<0.00001)
break;
else
if(cor<convergen)
convergen = cor;
else
break;
end
end
end
A1 = MatriksA1(foc1,foc2,pts3D,eks);

```

```

A2 = MatriksA2(foc1,foc2,pts3D,eks);
P = MatriksP(eks,pts3D);
V = N*xx-T;
std = sqrt((V'*V)/((2*ncam*np3)-(cam_par*ncam+p3_par*np3)));
Na = pinv(A1'*P*A1);
Nb = pinv(A2'*P*A2);
assignin('base','V',V);

% Menambahkan data nilai presisi kedalam data parameter eo
nph = cam_par*ncam;
for ii=1:nph
    for jj=1:nph
        if ii==jj
            preeks(ii,1) = std*sqrt(Na(ii,jj));
        end
    end
end
for ip=1:ncam
    pre = (ip-1)*6;
    for ipr=1:6
        eks_cor(2+ip,ipr)=preeks(pre+ipr,1);
    end
end
Eksterior=eks_cor;
assignin('base','Eksterior',Eksterior);

% Menambahkan data nilai presisi kedalam data parameter koordinat
obyek
np3D = p3_par*np3;
for kk=1:np3D
    for ll=1:np3D
        if kk==ll
            pre3D(ll,1) = std*sqrt(Nb(ll,kk));
        end
    end
end
for ip3=1:np3
    pr3 = (ip3-1)*3;
    for ppr=1:3
        pts3D_cor(ip3,3+ppr) = pre3D(pr3+ppr,1);
    end
end
Point3D = pts3D_cor;
assignin('base','Point3D',Point3D);

% Simpan Parameter Eksterior dan Koordinat Obyek
result = [Eksterior;Point3D];
set(hitung.Proses,'Userdata',result);
set(hitung.Focal_Lenght3,'String',foc1cor);
set(hitung.X_Nol3,'String',xo_1cor);
set(hitung.Y_Nol3,'String',yo_1cor);
set(hitung.Ksatu_1,'String',K1_1);
set(hitung.Kdua_1,'String',K2_1);
set(hitung.Ktiga_1,'String',K3_1);
set(hitung.Psatu_1,'String',P1_1);
set(hitung.Pdua_1,'String',P2_1);
set(hitung.bsatu_1,'String',b1_1);
set(hitung.bdua_1,'String',b2_1);

```

```

set(hitung.Focal_Lenght4, 'String', foc2cor);
set(hitung.X_Nol4, 'String', xo_2cor);
set(hitung.Y_Nol4, 'String', yo_2cor);
set(hitung.Ksatu_2, 'String', K1_2);
set(hitung.Kdua_2, 'String', K2_2);
set(hitung.Ktiga_2, 'String', K3_2);
set(hitung.Psatu_2, 'String', P1_2);
set(hitung.Pdua_2, 'String', P2_2);
set(hitung.bsatu_2, 'String', b1_2);
set(hitung.bdua_2, 'String', b2_2);
set(hitung.Akurasi, 'String', std);

% --- Executes on button press in Simpan.
function Simpan_Callback(hObject, eventdata, handles)
saving = guidata(gcbo);
[fileName, directory]=uiputfile('*.txt','Save File');
if isequal([fileName,directory],[0,0])
    return
else
eval(['cd '' directory '';']);
fileName = [fileName];
fout = fopen(fileName,'w');
Par = get(saving.Proses, 'Userdata');
end

fprintf(fout, '#Stereo Calibration\n');
fprintf(fout, '#This program created by Syahruni\n');
fprintf(fout, '#Teknik Geodesi ITN Malang\n');
fprintf(fout, '\n');
fprintf(fout, '#Exterior Orientation Parameters\n');
fprintf(fout, '%s\t %s\t %s\t %s\t %s\t %s\t %s\n', 'Id
Camera', 'Omega', 'Phi', 'Kappa', 'XL', 'YL', 'ZL');
fprintf(fout, '%s\t %f\t %f\t %f\t %f\t %f\t
%f\n', 'Cam1', Par(1,1), Par(1,2), Par(1,3), Par(1,4), Par(1,5), Par(1,6)
);
fprintf(fout, '%s\t %f\t %f\t %f\t %f\t %f\t
%f\n', 'Cam2', Par(2,1), Par(2,2), Par(2,3), Par(2,4), Par(2,5), Par(2,6)
);
fprintf(fout, '%s\t %f\t %f\t %f\t %f\t %f\t
%f\n', 'Pre_Cam1', Par(3,1), Par(3,2), Par(3,3), Par(3,4), Par(3,5), Par(
3,6));
fprintf(fout, '%s\t %f\t %f\t %f\t %f\t %f\t
%f\n', 'Pre_Cam2', Par(4,1), Par(4,2), Par(4,3), Par(4,4), Par(4,5), Par(
4,6));
fprintf(fout, '\n');
fprintf(fout, '#Object Space Point\n');
fprintf(fout, '%s\t %s\t %s\t %s\t %s\t %s\t
%s\n', 'Id', 'X', 'Y', 'Z', 'Pre_X', 'Pre_Y', 'Pre_Z');
for p=5:size(Par,1)
    r = (p-4);
    fprintf(fout, '%i\t %f\t %f\t %f\t %f\t %f\t
%f\n', r, Par(p,1), Par(p,2), Par(p,3), Par(p,4), Par(p,5), Par(p,6));
end
fprintf(fout, '\n');
fprintf(fout, '#Thanks for using this program\n');
fclose(fout);

% --- Executes on button press in Keluar.

```

```

function Keluar_Callback(hObject, eventdata, handles)
close;

function Focal_Lenght3_Callback(hObject, eventdata, handles)

function X_Nol3_Callback(hObject, eventdata, handles)

function Y_Nol3_Callback(hObject, eventdata, handles)

function Ksatu_1_Callback(hObject, eventdata, handles)

function Kdua_1_Callback(hObject, eventdata, handles)

function Ktiga_1_Callback(hObject, eventdata, handles)

function Psatu_1_Callback(hObject, eventdata, handles)

function Pdua_1_Callback(hObject, eventdata, handles)

function bsatu_1_Callback(hObject, eventdata, handles)

function bdua_1_Callback(hObject, eventdata, handles)

function Focal_Lenght4_Callback(hObject, eventdata, handles)

function X_Nol4_Callback(hObject, eventdata, handles)

function Y_Nol4_Callback(hObject, eventdata, handles)

function Ksatu_2_Callback(hObject, eventdata, handles)

function Kdua_2_Callback(hObject, eventdata, handles)

function Ktiga_2_Callback(hObject, eventdata, handles)

function Psatu_2_Callback(hObject, eventdata, handles)

function Pdua_2_Callback(hObject, eventdata, handles)

function bsatu_2_Callback(hObject, eventdata, handles)

function bdua_2_Callback(hObject, eventdata, handles)

function Akurasi_Callback(hObject, eventdata, handles)

```

D2. Fungsi Objectspk.m

```

function ospk = Objectspk(pts3D,eks)

m = size(pts3D,1);

for i=1:m
    X = pts3D(i,1);

```

```

Y = pts3D(i,2);
Z = pts3D(i,3);
XL1 = eks(1,4);
YL1 = eks(1,5);
ZL1 = eks(1,6);
XL2 = eks(2,4);
YL2 = eks(2,5);
ZL2 = eks(2,6);

% Camera 1
ospk(i,1) = X-XL1;
ospk(i,2) = Y-YL1;
ospk(i,3) = Z-ZL1;

% Camera 2
ospk(i,4) = X-XL2;
ospk(i,5) = Y-YL2;
ospk(i,6) = Z-ZL2;

end

```

D3. Fungsi MatriksM.m

```

function M = MatriksM(eks)
om1 = eks(1,1)*pi/180;
ph1 = eks(1,2)*pi/180;
kp1 = eks(1,3)*pi/180;
om2 = eks(2,1)*pi/180;
ph2 = eks(2,2)*pi/180;
kp2 = eks(2,3)*pi/180;

% Camera 1
M(1,1) = cos(ph1)*cos(kp1);
M(1,2) = sin(om1)*sin(ph1)*cos(kp1)+cos(om1)*sin(kp1);
M(1,3) = -cos(om1)*sin(ph1)*cos(kp1)+sin(om1)*sin(kp1);
M(2,1) = -cos(ph1)*sin(kp1);
M(2,2) = -sin(om1)*sin(ph1)*sin(kp1)+cos(om1)*cos(kp1);
M(2,3) = cos(om1)*sin(ph1)*sin(kp1)+sin(om1)*cos(kp1);
M(3,1) = sin(ph1);
M(3,2) = -sin(om1)*cos(ph1);
M(3,3) = cos(om1)*cos(ph1);

% Camera 2
M(1,4) = cos(ph2)*cos(kp2);
M(1,5) = sin(om2)*sin(ph2)*cos(kp2)+cos(om2)*sin(kp2);
M(1,6) = -cos(om2)*sin(ph2)*cos(kp2)+sin(om2)*sin(kp2);
M(2,4) = -cos(ph2)*sin(kp2);
M(2,5) = -sin(om2)*sin(ph2)*sin(kp2)+cos(om2)*cos(kp2);
M(2,6) = cos(om2)*sin(ph2)*sin(kp2)+sin(om2)*cos(kp2);
M(3,4) = sin(ph2);
M(3,5) = -sin(om2)*cos(ph2);
M(3,6) = cos(om2)*cos(ph2);

end

```

D4. Fungsi Rsqform.m

```
function rsq = Rsqform(pts3D, eks)

ospk = Objectspk(pts3D, eks);
M = MatriksM(eks);
m = size(pts3D, 1);

for i=1:m
    % Camera 1
    rsq(i, 1) =
M(1,1)*(ospk(i,1))+M(1,2)*(ospk(i,2))+M(1,3)*(ospk(i,3));
    rsq(i, 2) =
M(2,1)*(ospk(i,1))+M(2,2)*(ospk(i,2))+M(2,3)*(ospk(i,3));
    rsq(i, 3) =
M(3,1)*(ospk(i,1))+M(3,2)*(ospk(i,2))+M(3,3)*(ospk(i,3));

    % Camera 2
    rsq(i, 4) =
M(1,4)*(ospk(i,4))+M(1,5)*(ospk(i,5))+M(1,6)*(ospk(i,6));
    rsq(i, 5) =
M(2,4)*(ospk(i,4))+M(2,5)*(ospk(i,5))+M(2,6)*(ospk(i,6));
    rsq(i, 6) =
M(3,4)*(ospk(i,4))+M(3,5)*(ospk(i,5))+M(3,6)*(ospk(i,6));

end
```

D5. Fungsi MatriksA1.m

```
function A1 = MatriksA1(foc1, foc2, pts3D, eks)

ospk = Objectspk(pts3D, eks);
M = MatriksM(eks);
om1 = eks(1,1)*pi/180;
ph1 = eks(1,2)*pi/180;
kp1 = eks(1,3)*pi/180;
om2 = eks(2,1)*pi/180;
ph2 = eks(2,2)*pi/180;
kp2 = eks(2,3)*pi/180;
rsq = Rsqform(pts3D, eks);
m = size(pts3D, 1);

for i=1:m
    row=(i-1)*2+1;

    % Camera 1
    A1(row,1)=foc1/rsq(i,3)^2*(rsq(i,1)*(-
M(3,3)*ospk(i,2)+M(3,2)*ospk(i,3))-rsq(i,3)*(-
M(1,3)*ospk(i,2)+M(1,2)*ospk(i,3)));

    A1(row,2)=foc1/rsq(i,3)^2*(rsq(i,1)*(cos(ph1)*ospk(i,1)+sin(om1)*s
in(ph1)*ospk(i,2)-cos(om1)*sin(ph1)*ospk(i,3))-rsq(i,3)*(-
sin(ph1)*cos(kp1)*ospk(i,1)+sin(om1)*cos(ph1)*cos(kp1)*ospk(i,2)-
cos(om1)*cos(ph1)*cos(kp1)*ospk(i,3)));
    A1(row,3)=-
foc1/rsq(i,3)*(M(2,1)*ospk(i,1)+M(2,2)*ospk(i,2)+M(2,3)*ospk(i,3))
;
    A1(row,4)=-foc1/rsq(i,3)^2*(rsq(i,1)*M(3,1)-rsq(i,3)*M(1,1));
```

```

A1(row,5)=-foc1/rsq(i,3)^2*(rsq(i,1)*M(3,2)-rsq(i,3)*M(1,2));
A1(row,6)=-foc1/rsq(i,3)^2*(rsq(i,1)*M(3,3)-rsq(i,3)*M(1,3));

A1(row+1,1)=foc1/rsq(i,3)^2*(rsq(i,2)*(-
M(3,3)*ospk(i,2)+M(3,2)*ospk(i,3))-rsq(i,3)*(-
M(2,3)*ospk(i,2)+M(2,2)*ospk(i,3)));

A1(row+1,2)=foc1/rsq(i,3)^2*(rsq(i,2)*(cos(ph1)*ospk(i,1)+sin(om1)
*sin(ph1)*ospk(i,2)-cos(om1)*sin(ph1)*ospk(i,3))-
rsq(i,3)*(sin(ph1)*sin(kp1)*ospk(i,1)-
sin(om1)*cos(ph1)*sin(kp1)*ospk(i,2)+cos(om1)*cos(ph1)*sin(kp1)*os
pk(i,3)));

A1(row+1,3)=foc1/rsq(i,3)*(M(1,1)*ospk(i,1)+M(1,2)*ospk(i,2)+M(1,3)
)*ospk(i,3));
A1(row+1,4)=-foc1/rsq(i,3)^2*(rsq(i,2)*M(3,1)-
rsq(i,3)*M(2,1));
A1(row+1,5)=-foc1/rsq(i,3)^2*(rsq(i,2)*M(3,2)-
rsq(i,3)*M(2,2));
A1(row+1,6)=-foc1/rsq(i,3)^2*(rsq(i,2)*M(3,3)-
rsq(i,3)*M(2,3));

% Camera 2
A1(row+m*2,7)=foc2/rsq(i,6)^2*(rsq(i,4)*(-
M(3,6)*ospk(i,5)+M(3,5)*ospk(i,6))-rsq(i,6)*(-
M(1,6)*ospk(i,5)+M(1,5)*ospk(i,6)));

A1(row+m*2,8)=foc2/rsq(i,6)^2*(rsq(i,4)*(cos(ph2)*ospk(i,4)+sin(om
2)*sin(ph2)*ospk(i,5)-cos(om2)*sin(ph2)*ospk(i,6))-rsq(i,6)*(-
sin(ph2)*cos(kp2)*ospk(i,4)+sin(om2)*cos(ph2)*cos(kp2)*ospk(i,5)-
cos(om2)*cos(ph2)*cos(kp2)*ospk(i,6)));
A1(row+m*2,9)=-
foc2/rsq(i,6)*(M(2,4)*ospk(i,4)+M(2,5)*ospk(i,5)+M(2,6)*ospk(i,6))
;
A1(row+m*2,10)=-foc2/rsq(i,6)^2*(rsq(i,4)*M(3,4)-
rsq(i,6)*M(1,4));
A1(row+m*2,11)=-foc2/rsq(i,6)^2*(rsq(i,4)*M(3,5)-
rsq(i,6)*M(1,5));
A1(row+m*2,12)=-foc2/rsq(i,6)^2*(rsq(i,4)*M(3,6)-
rsq(i,6)*M(1,6));

A1(row+m*2+1,7)=foc2/rsq(i,6)^2*(rsq(i,5)*(-
M(3,6)*ospk(i,5)+M(3,5)*ospk(i,6))-rsq(i,6)*(-
M(2,6)*ospk(i,5)+M(2,5)*ospk(i,6)));

A1(row+m*2+1,8)=foc2/rsq(i,6)^2*(rsq(i,5)*(cos(ph2)*ospk(i,4)+sin(
om2)*sin(ph2)*ospk(i,5)-cos(om2)*sin(ph2)*ospk(i,6))-
rsq(i,6)*(sin(ph2)*sin(kp2)*ospk(i,4)-
sin(om2)*cos(ph2)*sin(kp2)*ospk(i,5)+cos(om2)*cos(ph2)*sin(kp2)*os
pk(i,6)));

A1(row+m*2+1,9)=foc2/rsq(i,6)*(M(1,4)*ospk(i,4)+M(1,5)*ospk(i,5)+M
(1,6)*ospk(i,6));
A1(row+m*2+1,10)=-foc2/rsq(i,6)^2*(rsq(i,5)*M(3,4)-
rsq(i,6)*M(2,4));
A1(row+m*2+1,11)=-foc2/rsq(i,6)^2*(rsq(i,5)*M(3,5)-
rsq(i,6)*M(2,5));

```

```

    A1(row+m*2+1,12)=-foc2/rsq(i,6)^2*(rsq(i,5)*M(3,6)-
rsq(i,6)*M(2,6));

```

```

end

```

D6. Fungsi MatriksA2.m

```

function A2 = MatriksA2(foc1,foc2,pts3D,eks)

```

```

M = MatriksM(eks);
rsq = Rsqform(pts3D,eks);
m = size(pts3D,1);

```

```

for i=1:m
    row = (i-1)*2+1;
    col = (i-1)*3+1;

    % Camera 1
    A2(row,col) = foc1/rsq(i,3)^2*(rsq(i,1)*M(3,1)-
rsq(i,3)*M(1,1));
    A2(row,col+1) = foc1/rsq(i,3)^2*(rsq(i,1)*M(3,2)-
rsq(i,3)*M(1,2));
    A2(row,col+2) = foc1/rsq(i,3)^2*(rsq(i,1)*M(3,3)-
rsq(i,3)*M(1,3));

    A2(row+1,col) = foc1/rsq(i,3)^2*(rsq(i,2)*M(3,1)-
rsq(i,3)*M(2,1));
    A2(row+1,col+1) = foc1/rsq(i,3)^2*(rsq(i,2)*M(3,2)-
rsq(i,3)*M(2,2));
    A2(row+1,col+2) = foc1/rsq(i,3)^2*(rsq(i,2)*M(3,3)-
rsq(i,3)*M(2,3));

```

```

    % Camera 2
    A2(row+m*2,col) = foc2/rsq(i,6)^2*(rsq(i,4)*M(3,4)-
rsq(i,6)*M(1,4));
    A2(row+m*2,col+1) = foc2/rsq(i,6)^2*(rsq(i,4)*M(3,5)-
rsq(i,6)*M(1,5));
    A2(row+m*2,col+2) = foc2/rsq(i,6)^2*(rsq(i,4)*M(3,6)-
rsq(i,6)*M(1,6));

    A2(row+m*2+1,col) = foc2/rsq(i,6)^2*(rsq(i,5)*M(3,4)-
rsq(i,6)*M(2,4));
    A2(row+m*2+1,col+1) = foc2/rsq(i,6)^2*(rsq(i,5)*M(3,5)-
rsq(i,6)*M(2,5));
    A2(row+m*2+1,col+2) = foc2/rsq(i,6)^2*(rsq(i,5)*M(3,6)-
rsq(i,6)*M(2,6));

```

```

end

```

D7. Fungsi MatriksA3.m

```

function A3 = MatriksA3(foc1,foc2,xo_1,yo_1,xo_2,yo_2,pts2D)

```

```

m = size(pts2D,1);

```

```

for i=1:m
    x1 = pts2D(i,1);

```

```

y1 = pts2D(i,2);
x2 = pts2D(i,3);
y2 = pts2D(i,4);
row = (i-1)*2+1;

% Camera 1
sls(i,1) = x1-xo_1;
sls(i,2) = y1-yo_1;
r(i,1) = sqrt((sls(i,1))^2+(sls(i,2))^2);

A3(row,1) = -1;
A3(row,3) = -sls(i,1)/foc1;
A3(row,4) = sls(i,1)*r(i,1)^2;
A3(row,5) = sls(i,1)*r(i,1)^4;
A3(row,6) = sls(i,1)*r(i,1)^6;
A3(row,7) = 2*(sls(i,1)^2)+r(i,1)^2;
A3(row,8) = 2*sls(i,1)*sls(i,2);
A3(row,9) = sls(i,1);
A3(row,10) = sls(i,2);

A3(row+1,2) = -1;
A3(row+1,3) = -sls(i,2)/foc1;
A3(row+1,4) = sls(i,2)*r(i,1)^2;
A3(row+1,5) = sls(i,2)*r(i,1)^4;
A3(row+1,6) = sls(i,2)*r(i,1)^6;
A3(row+1,7) = 2*sls(i,1)*sls(i,2);
A3(row+1,8) = 2*(sls(i,2)^2)+r(i,1)^2;

% Camera 2
sls(i,3) = x2-xo_2;
sls(i,4) = y2-yo_2;
r(i,2) = sqrt((sls(i,3))^2+(sls(i,4))^2);

A3(row+m*2,11) = -1;
A3(row+m*2,13) = -sls(i,3)/foc2;
A3(row+m*2,14) = sls(i,3)*r(i,2)^2;
A3(row+m*2,15) = sls(i,3)*r(i,2)^4;
A3(row+m*2,16) = sls(i,3)*r(i,2)^6;
A3(row+m*2,17) = 2*(sls(i,3)^2)+r(i,2)^2;
A3(row+m*2,18) = 2*sls(i,3)*sls(i,4);
A3(row+m*2,19) = sls(i,3);
A3(row+m*2,20) = sls(i,4);

A3(row+m*2+1,12) = -1;
A3(row+m*2+1,13) = -sls(i,4)/foc2;
A3(row+m*2+1,14) = sls(i,4)*r(i,2)^2;
A3(row+m*2+1,15) = sls(i,4)*r(i,2)^4;
A3(row+m*2+1,16) = sls(i,4)*r(i,2)^6;
A3(row+m*2+1,17) = 2*sls(i,3)*sls(i,4);
A3(row+m*2+1,18) = 2*(sls(i,4)^2)+r(i,2)^2;

```

end

D8. Fungsi MatriksL.m

```
function L =  
MatriksL(foc1,foc2,xo_1,yo_1,xo_2,yo_2,pts2D,pts3D,eks)  
  
rsq = Rsqform(pts3D,eks);  
m = size(pts3D,1);  
  
for i=1:m  
    x1 = pts2D(i,1);  
    y1 = pts2D(i,2);  
    x2 = pts2D(i,3);  
    y2 = pts2D(i,4);  
    row = (i-1)*2+1;  
  
    % Camera 1  
    L(row,1) = (x1-xo_1)+(foc1*(rsq(i,1)/rsq(i,3)));  
    L(row+1,1) = (y1-yo_1)+(foc1*(rsq(i,2)/rsq(i,3)));  
  
    % Camera 2  
    L(row+m*2,1) = (x2-xo_2)+(foc2*(rsq(i,4)/rsq(i,6)));  
    L(row+m*2+1,1) = (y2-yo_2)+(foc2*(rsq(i,5)/rsq(i,6)));  
  
end
```

D9. Fungsi MatriksP.m

```
function P = MatriksP(eks,pts3D)  
  
npts = size(pts3D,1);  
neks = size(eks,1);  
m = 2*neks*npts;  
P = eye (m);  
  
end
```

D10. Fungsi MatriksG.m

```
function G = MatriksG(pts3D)  
  
m = size(pts3D,1);  
  
for i=1:m  
    row = (i-1)*3+1;  
    G(row,1) = 1;  
    G(row,5) = pts3D(i,3);  
    G(row,6) = -pts3D(i,2);  
    G(row,7) = pts3D(i,1);  
    G(row+1,2) = 1;  
    G(row+1,4) = -pts3D(i,3);  
    G(row+1,6) = pts3D(i,1);  
    G(row+1,7) = pts3D(i,2);  
    G(row+2,3) = 1;  
    G(row+2,4) = pts3D(i,2);  
    G(row+2,5) = -pts3D(i,1);  
    G(row+2,7) = pts3D(i,3);  
  
end
```

end

D11. Fungsi Matriks_Normal.m

```
function [N,T] =  
Matriks_Normal(foc1,foc2,xo_1,yo_1,xo_2,yo_2,pts2D,pts3D,eks)  
  
ncamer = size(eks,1);  
  
A1 = MatriksA1(foc1,foc2,pts3D,eks);  
A2 = MatriksA2(foc1,foc2,pts3D,eks);  
A3 = MatriksA3(foc1,foc2,xo_1,yo_1,xo_2,yo_2,pts2D);  
L = MatriksL(foc1,foc2,xo_1,yo_1,xo_2,yo_2,pts2D,pts3D,eks);  
G = MatriksG(pts3D);  
P = MatriksP(eks,pts3D);  
  
N1 = [A1'*P*A1,A1'*P*A3,A1'*P*A2,zeros(6*ncamer,7)];  
N2 = [A3'*P*A1,A3'*P*A3,A3'*P*A2,zeros(10*ncamer,7)];  
N3 = [A2'*P*A1,A2'*P*A3,A2'*P*A2,G];  
N4 = [zeros(7,6*ncamer),zeros(7,10*ncamer),G',zeros(7,7)];  
  
N = [N1;N2;N3;N4];  
T = [A1'*P*L;A3'*P*L;A2'*P*L;zeros(7,1)];  
  
end
```