

SKRIPSI

DESAIN ALGORITMA *BUNDLE ADJUSTMENT*

DALAM CLOSE RANGE PHOTOGRAMMETRY

(Studi Kasus : Jembatan Fly Over-Arjosari Malang)



Diajukan untuk memenuhi persyaratan
dalam mencapai gelar sarjana S1 Teknik Geodesi

Disusun Oleh :



KADEK DODY ARIS PRAMANA

05.25.006

JURUSAN TEKNIK GEODESI

FAKULTAS TEKNIK SIPIL DAN PERENCANAAN

INSTITUT TEKNOLOGI NASIONAL

MALANG

2010

LEMBAR PENGESAHAN

**Desain Algoritma Bundle Adjustment Dalam Close Range Photogrammetry
(Studi Kasus : Jembatan Fly Over-Arjosari Malang)**

SKRIPSI

Dipertahankan di hadapan Majelis Penguji Sidang Skripsi
Jenjang Starata Satu (S-1)

Pada hari : Jumat

Tanggal : 20 Agustus 2010

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

Disusun oleh

Kadek Dody Aris Pramana

05.25.006

Panitia Ujian Tugas Akhir

Ketua



(Hery Purwanto, ST, M.Sc)

Sekretaris



Silvester Sari Sai, ST, MT

Anggota Penguji

Penguji I



(Ir. Leo Patimena, M.Sc)

Penguji II



Silvester Sari Sai, ST, MT

Penguji III



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

JURUSAN TEKNIK GEODESI

FAKULTAS TEKNIK SIPIL DAN PERENCANAAN

INSTITUT TEKNOLOGI NASIONAL

MALANG

2010

**LEMBAR PERSETUJUAN
SKRIPSI**

**Desain Algoritma Bundle Adjustment Dalam Close Range Photogrammetry
(Studi Kasus : Fly Over Arjosari Malang)**

Diajukan Sebagai Salah Satu Syarat Memperoleh Gelar Sarjana Teknik Geodesi

S-1

Institut Teknologi Nasional Malang

Disusun Oleh :

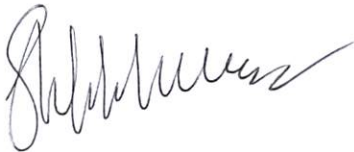
Kadek Dody Aris Pramana

05.25.006

Meyetujui,

Dosen Pembimbing I

Dosen Pembimbing II



Silvester Sari Sai, ST, MT



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

Mengetahui

Ketua Jurusan Teknik Geodesi S-1



Hery Purwanto, ST, M.Sc

ABSTARKSI

Kadek Dody Aris Pramana, 2010, “**Desain Algoritma Bundle Adjustment Dalam Close Range Photogrammetry (Studi Kasus : Fly Over Arjosari Malang)**”

Dosen Pembimbing I : Silvester Sari Sai, ST, MT.

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

Bundle adjustment merupakan salah satu bagian dari rangkaian proses *close range photogrammetry*. *Bundle adjustment* digunakan untuk mengevaluasi parameter *exterior orientation* (EO) dan koordinat titik objek yang dihasilkan pada tahap *relative orientation* dan *intersection*, sehingga diperoleh nilai akurasi dan presisi dari koordinat titik objek.

Algoritma perhitungan proses *bundle adjustment* dalam metode *close range photogrammetry* dapat didesain menggunakan perangkat lunak *Microsoft Excel*, *Console Application* pada bahasa Pemrograman C# dan MATLAB 2008.

Berdasarkan hasil analisis, perhitungan menggunakan beberapa aplikasi memiliki beberapa keunggulan dan kelemahan dalam proses perhitungan maupun efisiensi waktunya.

Kata Kunci : Algoritma, Bundle Adjustment, Close Range Photogrammetry

PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan dibawah ini.

Nama : Kadek Dody Aris Pramana

NIM : 0525006

Program Studi : Teknik Geodesi S-1

Fakultas : Fakultas Teknik Sipil Dan Perencanaan

Menyatakan dengan sesungguhnya bahwa Skripsi saya dengan judul :

**“Desain Algoritma Bundle Adjustment Dalam Close Range Photogrammetry
(Studi Kasus : Fly Over Arjosari Malang)”**

adalah hasil karya saya sendiri, bukan merupakan duplikat serta tidak mengutip atau menyadur dari hasil karya orang lain kecuali disebutkan sumbernya.

Malang, 29 September 2010

Yang membuat pernyataan

Kadek Dody Aris Pramana

0525006

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Ida Sang Hyang Widhi Wasa/ Tuhan Yang Maha Esa, karena berkat rahmat-Nya, penulis dapat menyelesaikan skripsi yang berjudul "**Desain Algoritma *Bundle Adjustment* Dalam *Close Range Photogrammetry* (Studi Kasus pada Fly Over Arjosari Malang)**", di mana penulisan skripsi ini disusun sebagai salah satu syarat untuk meraih gelar Sarjana Teknik pada Jurusan Teknik Geodesi Fakultas Teknik Sipil Dan Perencanaan Institut Teknologi Nasional Malang.

Penulisan ini tidak akan dapat terselesaikan tanpa bantuan dan dukungan berbagai pihak. Oleh karena itu, peneliti ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak Prof. Dr. Eng. Ir. Abraham Lomi, MSEE 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 Heri Purwanto, ST., M.Sc. selaku Ketua Jurusan Teknik Geodesi Institut Teknologi Nasional Malang.
4. Bapak Dr. Edwin Tjahjadi, ST., MGeom.Sc. selaku Dosen Pembimbing I dan Dosen Penguji.
5. Bapak Silvester Sari Sai, ST., MT. selaku Dosen Pembimbing II dan Dosen Penguji.
6. Bapak Leo Patimena, ST., M.Sc. selaku Dosen Penguji.

7. **Sege nap dosen, staff pengajar dan rekording Jurusan Teknik Geodesi Fakultas Teknik Sipil dan Perencanaan Institut Teknologi Nasional Malang.**
8. **Bapak, Ibu, Kakak, Luh Reni, Keluarga Singaraja dan Tabanan, yang selalu memberikan dukungan, semangat dan doa.**
9. **Team Deformasi dan Rapid mapping yang selalu memberika kerja sama dan dukungannya.**
10. **Semua pihak yang telah membantu peneliti yang tidak dapat disebutkan satu persatu.**

Semoga Ida Sang Hyang Widi Wasa memberikan karunia atas budi baik dari semua pihak yang terlibat dalam penelitian ini. Penulis menyadari bahwa penulisan laporan ini masih belum sempurna, baik dari segi materi, sistematika pembahasan, maupun susunan bahasanya Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan. Hasil penelitian ini dan dengan segala keterbatasannya dipersembahkan kepada dunia pendidikan, semoga ada manfaatnya untuk pengembangan sumber daya manusia di negara tercinta ini.

Malang, 29 September 2010

Penulis

DAFTAR ISI

Halaman Sampul Depan

Halaman Judul

Lembar Pengesahan

Lembar Persetujuan

Abstraksi

Pernyataan Keaslian Skripsi

Kata Pengantar

Daftar Isi

Daftar Gambar

Daftar Tabel

BAB 1 PENDAHULUAN

1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Tinjauan Pustaka.....	4

BAB II DASAR TEORI

2.1 <i>Bundle Adjustment</i> dan Kesalahan Observasi	6
--	---

2.2	<i>Metode Gabungan Kuadrat Terkecil (Unified Least Square Method)</i>	9
2.2.1	<i>Adjustment of Indirect Observations</i>	9
2.2.2	<i>Adjustment of Observations Only</i>	10
2.3	<i>Conventional Bundle Adjustment</i>	10
2.3.1	Persamaan Kolinear.....	11
2.3.1.1	Matrik Rotasi.....	11
2.3.1.2	Transformasi Perspektif Pusat (<i>Central Perspective Transformation</i>)	15
2.3.1.3	Kondisi Kolinear.....	17
2.3.2	Persamaan Observasi.....	18
2.3.3	Struktur Matrik Bobot	20
2.3.4	Formula Model Matematika	22
2.3.5	Linierisasi Persamaan Kolinier	23
2.3.6	Desain Matrik untuk <i>Bundle Adjustment</i>	24
2.3.7	Struktur dari Matrik Persamaan Normal	27
2.3.8	Penyelesaian Persamaan Normal <i>Bundle Adjustment</i>	30
2.3.9	Efisiensi Algoritma Perhitungan	30
2.4	<i>Separate Adjustment dalam Close Range Photogrammetry</i>	32
2.4.1	<i>Free Network Separate Adjustment</i>	32
2.4.1.1	Penyesuaian Titik Objek	33
2.4.1.2	Penyesuaian Parameter Kamera	35
2.4.2	Pengertian Datum	36
2.4.3	Estimasi Presisi	37
2.5	<i>Network Design</i>	38

2.5.1	Desain Target	39
2.5.2	Desain Pemotretan.....	40
2.5.3	Permasalahan Datum	42

BAB III METODOLOGI

3.1	Persiapan Penelitian	44
3.1.1	Materi Penelitian	44
3.1.2	Alat Penelitian	45
3.1.2.1	Perangkat Keras (<i>Hardware</i>).....	45
3.1.2.2	Perangkat Lunak (<i>Software</i>)	47
3.2	Diagram Alir Penelitian	49
3.3	Penjelasan Diagram Alir (<i>Flow Chart</i>) Penelitian.....	51
3.3.1	Mengumpulkan Data dan Menghitung Parameter Awal .	51
3.3.2	Menghitung Matrik Rotasi (R).....	51
3.3.3	Menghitung Parameter <i>qrs</i>	52
3.3.4	Menghitung Matrik B_1, B_2, f, w	52
3.3.5	Menghitung Matrik $\dot{N}_{ij}, \ddot{N}_{ij}, \bar{N}_{ij}, \dot{K}_{ij}, \ddot{K}_{ij}$	53
3.3.6	Menyusun Matrik N dan K	53
3.3.7	Menghitung matrik Koreksi ($\dot{\Delta}$ dan $\ddot{\Delta}$).....	54
3.3.8	Menghitung Ketelitian Varian-Kovarian.....	54
3.3.9	Menghitung Nilai Akhir	55

BAB IV HASIL DAN PEMBAHASAN

4.1	Hasil Perhitungan <i>Bundle Adjustment</i> Menggunakan <i>Microsoft Excel</i>	57
-----	---	----

4.1.1	Hasil Perhitungan Parameter Awal	57
4.1.2	Perhitungan Matrik Rotasi (R) setiap Foto	59
4.1.3	Koreksi Parameter $\hat{\Delta}$ dan $\hat{\Delta}$ pada Iterasi Terakhir	59
4.1.4	Parameter EO <i>Update</i> dan Koordinat <i>Update</i>	60
4.1.5	Presisi Koordinat Titik-titik Objek.....	61
4.2	<i>Source Code</i> Perhitungan <i>Bundle Adjustment</i> menggunakan <i>Console Application</i> pada Program C#.....	61
4.2.1	Pembuatan <i>Class</i> Utama Program.....	61
4.2.2	<i>Class</i> Photo Pair EO.....	62
4.2.3	<i>Class</i> Photo Pair Point.....	63
4.2.4	<i>Class</i> Proses <i>Bundle Adjustment</i>	63
4.3	Hasil Perhitungan <i>Bundle Adjustment</i> menggunakan <i>Console</i> <i>Application</i> pada Bahasa Pemrograman C#	65
4.3.1	Perhitungan Matrik Rotasi (R) setiap Foto	65
4.3.2	Perhitungan Koreksi δ_1 dan δ_2	65
4.3.3	Hasil Parameter EO <i>Update</i> dan Koordinat Titik Objek <i>Update</i>	66
4.3.4	Presisi Koordinat Titik-titik Objek.....	67
4.4	<i>Source Code</i> Perhitungan <i>Bundle Adjustment</i> menggunakan <i>MATLAB 2008</i>	67
4.4.1	Fungsi Untuk Matrik Rotasi	67
4.4.2	Fungsi Untuk Matrik B.....	68
4.4.3	Fungsi Untuk Elemen qrs	68
4.4.4	Fungsi Utama Program.....	68
4.5	<i>Prosedur Pengoperasian Program MATLAB dalam Menghitung</i> <i>Bundle Adjustment</i>	71

4.5.1	Tampilan Awal Program	71
4.5.2	Input Data	71
4.5.3	Proses Perhitungan <i>Bundle Adjustmen</i>	72
4.5.4	Penyimpanan Hasil Perhitungan <i>Bundle Adjustment</i>	72
4.6	Hasil Perhitungan <i>Bundle Adjustment</i> menggunakan <i>MATLAB 2008</i>	73
4.6.1	Perhitungan Matrik Rotasi (<i>R</i>) Tiap Foto	73
4.6.2	Perhitungan Koreksi Δ dan $\ddot{\Delta}$	73
4.6.3	Hasil Parameter EO <i>Update</i> dan Koordinat Titik Objek <i>Update</i>	74
4.6.4	Presisi Koordinat Titik-titik Objek.....	75
4.7	Analisis Hasil Penelitian <i>Bundle Adjustment</i>	75
4.7.1	Analisis Hasil Parameter EO <i>Update</i>	76
4.7.1.1	Hasil Parameter EO <i>Update</i> Menggunakan <i>Microsoft Excel</i>	76
4.7.1.2	Hasil Parameter EO <i>Update</i> Menggunakan <i>Con- sole Application</i> pada Bahasa Pemrograman C#. 76	
4.7.1.3	Hasil Parameter EO <i>Update</i> Menggunakan MATLAB 2008	77
4.7.2	Analisis Hasil Koordinat Titik-Titik Objek <i>Update</i>	78
4.7.3	Analisis Hasil Ketelitian Titik-Titik Objek	79
4.7.3.1	Hasil Ketelitian Titik-titik Objek Menggunakan <i>Microsoft Excel</i>	79
4.7.3.2	Hasil Ketelitian Titik-titik Objek Menggunakan <i>Console Application</i> pada Bahasa Pemrograman C#.....	79

4.7.3.3 Hasil Ketelitian Titik-titik Objek Menggunakan <i>MATLAB</i>	80
4.8 Evaluasi Masing-masing Metode Perhitungan <i>Bundle Adjustment</i>	81
4.8.1 Keunggulan dan Kelemahan Penggunaan <i>Microsoft Excel</i>	81
4.8.2 Keunggulan dan Kelemahan Penggunaan <i>Console Application</i> pada Bahasa Pemrograman <i>C#</i>	82
4.8.3 Keunggulan dan Kelemahan Penggunaan <i>MATLAB</i>	82

BAB V PENUTUP

5.1 Kesimpulan.....	84
5.2 Saran	86

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1	Proses <i>Bundle</i> Multi Foto	8
Gambar 2.2	Ilustrasi Proses <i>Bundle Adjustment</i>	8
Gambar 2.3	Rotasi Sudut Omega terhadap Sumbu $X\omega$	12
Gambar 2.4	Rotasi Sudut Phi terhadap Sumbu $Y\phi$	13
Gambar 2.5	Rotasi Sudut Kappa terhadap Sumbu $Z\kappa$	13
Gambar 2.6	Hubungan Keruangan antara Titik A Sistem Koordinat Sembarang dengan Titik a Sistem Koordinat Kamera.....	15
Gambar 2.7	Struktur Persamaan Normal	29
Gambar 2.8	Struktur Desain Matrik A (A_1 dan A_2)	33
Gambar 2.9	Struktur Desain Matrik A_{11} dan A_{22}	33
Gambar 2.10	Geometri Pemotretan	39
Gambar 2.11	Diagram Desain Target	40
Gambar 2.12	<i>Design Left-Right Measurements</i>	41
Gambar 2.13	<i>Design Front-Back Measurements</i>	41
Gambar 2.14	Desain Jaringan Empat Stasiun	42
Gambar 2.15	Ilustrasi Penempatan Stasiun Pemotretan	42
Gambar 3.1	Kamera SLR Nikon D60 Tampak Depan dan Belakang	45
Gambar 3.2	Aksesoris dari Kamera (Tipe Lensa, Baterai, Memori).....	45
Gambar 3.3	Stiker <i>Retro Reflective Target</i>	46
Gambar 3.4	Tampilan Awal Program C#.....	47
Gambar 3.5	Tampilan Awal Program MATLAB 2008	48
Gambar 3.6	Diagram Alir Penelitian (1).....	49
Gambar 3.6	Diagram Alir Penelitian (2).....	50

Gambar 3.7	Struktur Persamaan Normal	54
Gambar 4.1	Susunan Data Awal pada Notepad	62
Gambar 4.2	Susunan Data Input Parameter EO Awal pada Notepad.....	69
Gambar 4.3	Susunan Data Input Titik Koordinat Awal pada Notepad	70
Gambar 4.4	Susunan Data Input Ketelitian Titik untuk Koordinat Foto Awal pada Notepad.....	70
Gambar 4.5	Tampilan Awal Program	71
Gambar 4.6	Input Data Principle Point dan Panjang Fokus.....	72
Gambar 4.7	Hasil Urutan Iterasi dan Nilai Standar Deviasi	72
Gambar 4.8	Sebaran Titik-titik pada Foto.....	78
Gambar 4.6	Posisi Titik-titik Objek dan Posisi Kamera.....	79

DAFTAR TABEL

Tabel 4.1	Parameter IO (<i>Interior Orientation</i>).....	57
Tabel 4.2	Koordinat Foto dan Presisi setiap Titik pada Foto Satu	57
Tabel 4.3	Koordinat Foto dan Presisi setiap Titik pada Foto Dua.....	58
Tabel 4.4	Parameter EO Pendekatan Dua Buah Foto	58
Tabel 4.5	Koordinat Titik-titik Objek Pendekatan.....	58
Tabel 4.6	Koreksi Parameter EO Menggunakan <i>Microsoft Excel</i>	59
Tabel 4.7	Koreksi Koordinat Titik-titik Objek Menggunakan <i>Microsoft Excel</i>	60
Tabel 4.8	Parameter EO <i>Update</i> Menggunakan <i>Microsoft Excel</i>	60
Tabel 4.9	Titik-titik Objek <i>Update</i> Menggunakan <i>Microsoft Excel</i>	60
Tabel 4.10	Presisi Koordinat Titik-titik Objek Menggunakan <i>Microsoft Excel</i> ...	61
Tabel 4.11	Koreksi Parameter EO Menggunakan <i>Console Application</i> Pemrograman C#	65
Tabel 4.12	Koreksi Koordinat Titik Objek Menggunakan <i>Console Application</i> Pemrograman C#	66
Tabel 4.13	Parameter EO <i>Update</i> Menggunakan <i>Console Application</i> Pemrograman C#	66
Tabel 4.14	Koordinat Titik-titik <i>Update</i> Menggunakan <i>Console Application</i> Pemrograman C#	66
Tabel 4.15	Presisi Titik-titik Koordinat Menggunakan <i>Console Application</i> Pemrograman C#	67
Tabel 4.16	Koreksi Parameter EO Menggunakan MATLAB	73
Tabel 4.17	Koreksi Koordinat Titik-titik Objek Menggunakan MATLAB.....	74
Tabel 4.18	Parameter EO <i>Update</i> Menggunakan MATLAB	74
Tabel 4.19	Koordinat Titik-titik Objek <i>Update</i> Menggunakan MATLAB.....	74

Tabel 4.20	Presisi Titik-titik Koordinat Menggunakan MATLAB	75
Tabel 4.21	Hasil Selisih Parameter EO Awal dengan Parameter EO <i>Update</i> Menggunakan <i>Microsoft Excel</i>	76
Tabel 4.22	Hasil Selisih Parameter EO Awal dengan Parameter EO <i>Update</i> Menggunakan Console Aplication pada Bahasa Pemrograman C# .	77
Tabel 4.23	Hasil Selisih Parameter EO Awal dengan Parameter EO <i>Update</i> Menggunakan MATLAB 2008	77
Tabel 4.24	Ketelitian Koordinat Titik-titik Objek Menggunakan <i>Microsoft Excel</i>	80
Tabel 4.25	Ketelitian Koordinat Titik-titik Objek Menggunakan <i>Console Application</i> pada Bahasa Pemrograman C#	80
Tabel 4.26	Ketelitian Koordinat Titik-titik Objek Menggunakan MATLAB 2008	81

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam perkembangannya penggunaan metode *close range photogrammetry* dalam berbagai aplikasi teknik dan survei semakin meningkat (Linder, 2006). *Close range photogrammetry* digunakan untuk mendiskripsikan teknik fotogrametri saat jarak dari objek yang diukur kurang dari 100 meter dan kamera diletakan dekat dengan objek (Sahdan, 2006). Dengan menggunakan kamera resolusi tinggi yang digabungkan dengan beberapa persyaratan antara lain *network design*, *camera calibration* serta menggunakan proses perhitungan *bundle adjustment* akan menghasilkan tingkat pengukuran dengan akurasi tinggi (Shirkhani et al., 2006).

Perhitungan *bundle adjustment* dalam *close range photogrammetry* tidak terlepas dari proses *resection* dan *intersection*. Dimana proses *resection* merupakan proses penentuan posisi dan orientasi luar dalam tiap foto. Proses *Intersection* merupakan teknik untuk menentukan koordinat titik-titik objek pada dua buah foto atau lebih yang saling bertampalan sehingga dapat diketahui posisi secara 3D (Mikhail et al., 2001). Karena parameter-parameter dalam proses *resection* dan *intersection* akan digunakan dalam perhitungan *bundle adjustment*.

Bundle adjustment merupakan suatu bagian penting dalam *close range photogrammetry*. *Bundle adjustment* merupakan proses penentuan

atau perhitungan parameter *interior orientation* (IO), *exterior orientation* (EO yaitu $\omega, \varphi, \kappa, X_L, Y_L, Z_L$), dan koordinat objek yang dihitung secara bersamaan dengan menggunakan teknik hitung kuadrat terkecil (Tjahjadi, 2008a). *Bundle adjustment* atau sering disingkat dengan BA, dapat mengurangi kesalahan pada elemen kamera (posisi dan rotasi kamera) dan meningkatkan akurasi pada titik-titik objek 3D (Franck Jung, 2006; Fraundorfer et al., 2009). Sehingga keberadaan proses *bundle adjustment* sangat dibutuhkan untuk memperoleh hasil X_i, Y_i, Z_i yang akurat dan parameter orientasi kamera yang teliti. Melihat peranan penting dari *bundle adjustment* dalam mendukung proses *close range photogrammetry*, maka penulis bermaksud untuk menulis sebuah skripsi yang berjudul *Desain Algoritma Bundle Adjustment dalam Close Range Photogrammetry*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah disampaikan sebelumnya, adapun rumusan masalah yang akan dibahas dalam skripsi ini adalah bagaimana desain algoritma *bundle adjustment* dalam *close range photogrammetry*?

1.3 Batasan Masalah

Permasalahan yang akan dibahas dalam skripsi ini dibatasi pada beberapa komponen, yaitu :

1. Algoritma yang digunakan hanya algoritma yang berkaitan dengan perhitungan *bundle adjustment* dalam *close range photogrammetry*.

2. Parameter yang dihasilkan yaitu parameter orientasi luar (*eksterior orientation* yaitu $\hat{\omega}, \hat{\phi}, \hat{\kappa}, \hat{X}_L, \hat{Y}_L, \hat{Z}_L$), koordinat titik-titik objek ($\hat{X}_i, \hat{Y}_i, \hat{Z}_i$), akurasi dari titik-titik objek ($\delta x, \delta y, \delta z$) yang merupakan parameter akhir.
3. Penelitian dilakukan pada salah satu tiang di *fly over* Arjosari dengan dua data foto dan hanya menganalisis delapan titik untuk masing-masing foto.

1.4 Tujuan Penelitian

Sesuai dengan rumusan masalah yang akan dibahas dalam skripsi ini, maksud dan tujuan dari penulisan skripsi ini adalah

1. Mendesain perhitungan untuk proses *bundle adjustment* dalam metode *close range photogrammetry*.
2. Menyatakan pentingnya proses *bundle adjustment* pada *close range photogrammetry* untuk menilai keakurasian dan presisi dari parameter *exterior orientation* dan koordinat titik objek.

1.5 Manfaat Penelitian

Adapun manfaat yang diharapkan dalam penelitian ini yaitu :

1.5.1 Manfaat Praktis

Desain algoritma *bundle adjustment* dapat diaplikasikan dalam sebuah bahasa pemograman untuk memudahkan proses perhitungan.

1.5.2 Manfaat Teoritis

Hasil dari penelitian ini diharapkan dapat memberikan kontribusi teoritis terhadap perkembangan ilmu pengetahuan, khususnya yang berkaitan dengan *bundle adjustment close range photogrammetry*.

1.6 Tinjauan Pustaka

Beberapa tinjauan pustaka telah dilakukan dalam menyusun penelitian, guna mengumpulkan informasi mengenai proses *bundle adjustment* dalam *close range photogrammetry* yang didasarkan atas berbagai riset oleh para ilmuwan dalam bidang *close range photogrammetry* antara lain :

Menurut (Brown, 1974; Schut, 1980; Heindl, 1981) dan (Triggs et al., 2000), *Bundle adjustment* adalah proses penentuan atau perhitungan parameter IO, EO dan koordinat objek secara serempak bersamaan dengan menggunakan teknik hitung kuadrat terkecil (dalam Tjahjadi, 2008a).

Keakurasian (presisi dan *reliability*/keandalan) dari titik-titik objek, yaitu δx , δy , δz , dengan menggunakan kamera non metrik tergantung dari beberapa faktor, antara lain: Parameter kalibrasi kamera, parameter orientasi dalam, parameter orientasi luar, konfigurasi pemotretan, dan teknik perhitungan koordinat objek-objek dan parameter kamera (Tjahjadi, 2008b).

Menurut Mikhail dan Gracie (1981), Cooper (1987) : Dalam *surveying* dan *close range photogrammetry*, data ukuran lebih selalu dibutuhkan untuk presisi tinggi, realibilitas dan statistik. Ini berarti jumlah observasi harus lebih dari parameter *unknown* yang akan dicari untuk mendapatkan solusi yang unik (Wang dan Clarke, 1998).

Teknik *bundle adjustment* ini menjadi cara yang paling efisien untuk memproses data pemotretan hal ini dikuatkan oleh pernyataan Triggs et al (2000). Mengenai alasan pemilihan *bundle adjustment* dalam perhitungan parameter-parameter *close range photogrammetry*, karena *bundle adjustment* memiliki karakteristik :

1. Fleksibel : *Bundle adjustment* dapat dengan baik memecahkan berbagai masalah antara lain berbagai bentuk 3D dan tipe, berbagai tipe skema, sumber informasi (fitur 2D, tingkat intensitas, informasi 3D) dan error models. Tidak bermasalah dengan missing data.
2. Akurat : *Bundle Adjustment* memberikan hasil yang presisi dan mudah dalam interpretasinya karena menggunakan model kesalahan statistik secara akurat.
3. Efisien : Algoritma *bundle* yang telah matang dan relatif efisien meskipun pada permasalahan yang sangat besar

BAB II

DASAR TEORI

2.1 *Bundle Adjustment* dan Kesalahan Observasi

Dalam penulisan ini, akan dibahas secara khusus mengenai teknik perhitungan *bundle adjustment* dalam *photogrammetry*. *Photogrammetry* tidak terlepas dari dari proses *resection* dan *intersection*. Dimana proses *resection* merupakan proses penentuan posisi dan orientasi luar dalam tiap foto. Proses *intersection* merupakan teknik untuk menentukan koordinat titik-titik objek pada dua buah foto atau lebih yang saling bertampalan sehingga dapat diketahui posisi secara 3D (Mikhail et al, 2001). Dimana operasi ini akan dikombinasikan dalam proses triangulasi, atau *block adjustment*, dimana parameter orientasi foto dan koordinat objek akan dihitung secara serempak.

Penyesuaian dalam kuadrat terkecil membentuk dasar dari beberapa algoritma yang digunakan dalam proses perhitungan *close range photogrammetry*. *Bundle adjustment* merupakan aplikasi yang paling sering digunakan dalam teori penyesuaian kuadrat terkecil pada *close range photogrammetry*.

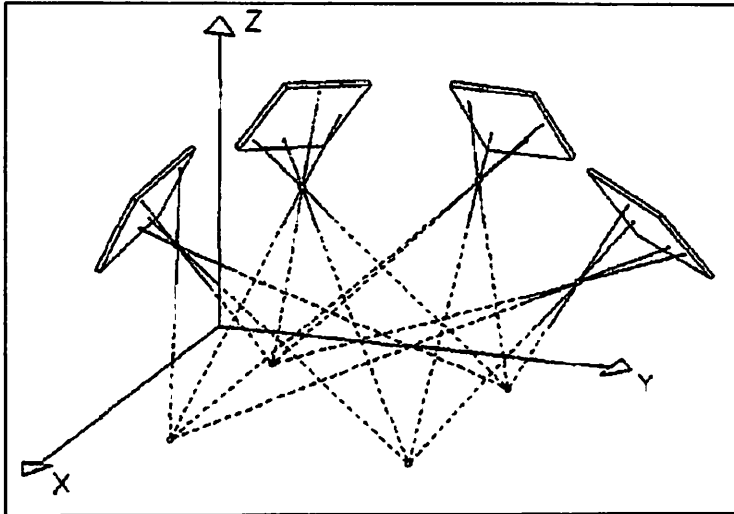
Bundle adjustment merupakan proses penentuan atau perhitungan parameter *interior orientation* (IO), *exterior orientation* (EO), dan koordinat objek yang dihitung secara bersamaan dengan menggunakan teknik hitung kuadrat terkecil (Tjahjadi, 2008a). Parameter-parameter yang digunakan dalam proses *bundle adjustmment* adalah parameter IO (*interior*

orientation), EO (*eksterior orientation*), serta koordinat objek pendekatan dari objek yang diamati.

Mengenai alasan pemilihan *bundle adjustment* dalam perhitungan parameter-parameter *close range photogrammetry*, karena menurut Triggs et al (2000), *bundle adjustment* memiliki karakteristik sebagai berikut :

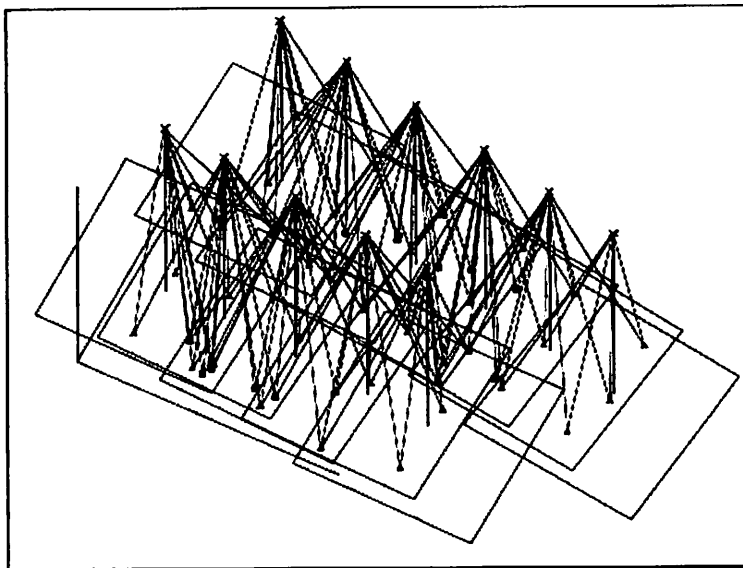
1. Fleksibel : *Bundle adjustment* dapat dengan baik memecahkan berbagai masalah antara lain berbagai bentuk 3D dan tipe kamera (titik, garis, kurva, surface, non-metrik, metrik, dsb), berbagai tipe skema (termasuk model dinamik dan artikulasi, skema constraints), sumber informasi (fitur 2D, tingkat intensitas, informasi 3D) dan *error models*.
2. Akurat : *Bundle adjustment* memberikan hasil yang presisi dan mudah dalam interpretasinya karena menggunakan model kesalahan statistik secara akurat.
3. Efisien : Algoritma *bundle* yang telah matang dan relatif efisien meskipun pada permasalahan yang sangat besar.

Bundle adjustment merupakan suatu bentuk penyesuaian yang ditujukan untuk melakukan estimasi secara simultan dalam penentuan nilai parameter orientasi dari koordinat objek yang diinginkan. Proses tersebut meliputi penempatan foto-foto secara bebas dalam *object space* membentuk jaringan *photogrammetry*, seperti ditunjukkan pada Gambar (2.1) (Uffenkam, 1998) :



Gambar 2.1 Proses Bundle Multi Foto

Secara grafis proses dari *bundle adjustment* dapat dilihat pada gambar berikut (Jacobsen, 2000) :



Gambar 2.2 Ilustrasi proses Bundle Adjustment

Dalam melakukan observasi maupun perhitungan, dapat terjadi tiga jenis kesalahan (*error*), yaitu *gross error*, *systematic error*, dan *random error*. Ukuran dari masing-masing kesalahan tersebut akan mempengaruhi keakuratan dan presisi dari parameter yang dihitung.

2.2 Metode Gabungan Kuadrat Terkecil (*Unified Least Square Method*)

Metode gabungan kuadrat terkecil (*Unified Least Square Method*) mengombinasikan dua jenis persamaan fungsional yang berbeda yang umum digunakan dalam penyesuaian kuadrat terkecil, yaitu persamaan observasi (*observation equation*) dan persamaan kondisi (*condition equation*). Kedua tipe persamaan tersebut telah digunakan untuk merumuskan metode pemisahan kuadrat terkecil (King, 1993).

Mikhail et al (2001) menyatakan bahwa terdapat dua metode dalam penyesuaian, yaitu *Adjustment of Indirect Observations* dan *Adjustment of Observations Only*

2.2.1 *Adjustment of Indirect Observations*

Persamaan dengan menggunakan metode ini mengandung observasi, konstanta dan parameter. Parameter merupakan sebuah variabel *stochastic* yang nilainya perlu diestimasi selama proses kuadrat terkecil. Masing-masing persamaan mengandung hanya satu observasi. Model fungsional penyesuaian dari observasi tidak langsung (*Adjustment of Indirect Observations*) sebagai berikut :

$$l + v + B\Delta = d \dots\dots\dots(2.1)$$

atau

$$v + B\Delta = -l + d = f \dots\dots\dots(2.2)$$

dimana l dan v merupakan vektor observasi dan residu, d merupakan vektor konstanta, B merupakan matrik parameter koefisien, dan Δ merupakan vektor parameter (King, 1993).

2.2.2 Adjustment of Observations Only

Dalam metode ini, kombinasi observasi dan residunya sama dengan *physical constant*-nya. Model fungsionalnya sebagai berikut:

$$A(l + v) = d \dots\dots\dots(2.3)$$

atau

$$Av = d - Al = f \dots\dots\dots(2.4)$$

Dimana A merupakan matrik koefisien observasi, l dan v adalah vektor observasi dan residunya, d adalah vektor nilai konstan (King, 1993).

Dengan pendekatan gabungan kuadrat terkecil (*unified least square*) diasumsikan bahwa semua variabel yang terkandung dalam fungsi matematika ini merupakan observasi. Maka persamaanya sebagai berikut :

$$AV + B\Delta = f \dots\dots\dots(2.5)$$

Jika dikombinasikan dengan matrik bobot (W) akan menghasilkan persamaan sebagai berikut :

$$\begin{aligned} \Delta &= (B^T W B)^{-1} B^T W f \dots\dots\dots(2.6) \\ &= N^{-1} f \end{aligned}$$

2.3 Conventional Bundle Adjustment

Conventional Bundle Adjustment, yang sering disebut dengan BA, merupakan bangun dasar dari *Least Square Bundle Adjustment* yang dikembangkan oleh Brown pada tahun 1958 (dalam King, 1993). Selain BA, terdapat beberapa model lain dalam *Bundle Adjustment*, seperti Model

DLT, yang dikembangkan oleh Abdel-Aziz dan Karara serta Model Penyesuaian Observasi Tidak Langsung yang dikembangkan oleh Mikhail. Menurut hasil penelitian yang dilakukan oleh Ruther (1989), 80% dari program komputer yang menyediakan solusi untuk *photogrammetry* mengacu pada *bundle adjustment* (King, 1993). Hal ini menunjukkan bahwa *bundle adjustment* menjadi model perhitungan yang dominan diterapkan dalam ilmu *photogrammetry* dibandingkan dengan model lainnya.

Pada umumnya, *bundle adjustment* dipergunakan untuk menentukan parameter *interior orientation* (IO), *exterior orientation* (EO), dan koordinat objek yang dihitung secara bersamaan dengan menggunakan teknik hitung kuadrat terkecil (Fraser, 1997).

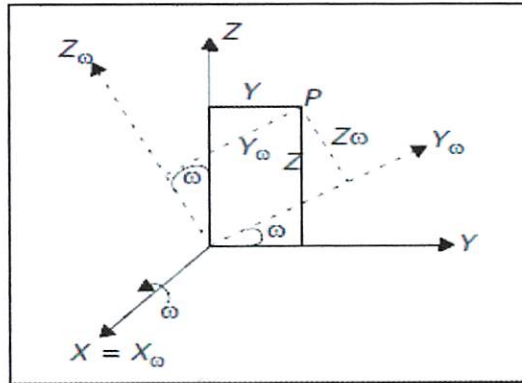
2.3.1 Persamaan Kolinear

Hubungan matematis antara titik konjugasi pada objek dengan titik konjugasi pada foto yang dijelaskan dengan transformasi projektif tiga dimensi disebut dengan persamaan kolinear. Disebut kolinear karena persamaan tersebut menunjukkan hubungan geometrik yang ideal antara sebuah titik objek, fotonya, dengan *imaging system perspective center*.

2.3.1.1 Matrik Rotasi

Matrik rotasi terdiri dari tiga parameter rotasi yaitu, omega (ω), phi (ϕ), dan kappa (κ). Dimana arah perputaran sumbu rotasi omega, phi, dan kappa beserta sistem kaedah dan persamaannya dapat ditunjukkan pada gambar berikut :

1. Rotasi pertama yaitu omega (ω) dilakukan terhadap sumbu X_ω , dengan menggunakan sistem kaedah tangan kanan perputaran bernilai positif jika berlawanan arah jarum jam, seperti Gambar (2.3) (Geosystem, 2006):

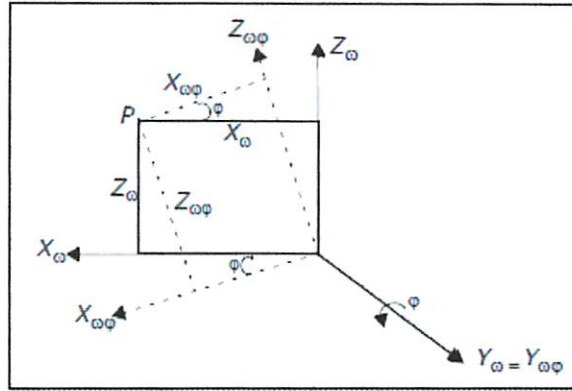


Gambar 2.3 Rotasi sudut omega terhadap sumbu X_ω

Dari Gambar (2.3) sistem rotasi ($X_\omega, Y_\omega, Z_\omega$) dapat diperoleh persamaan vektornya $[X_\omega \ Y_\omega \ Z_\omega]^T = R_\omega [X \ Y \ Z]^T$, dimana (Cooper dan Robson, 2001) :

$$R_\omega = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \omega & \sin \omega \\ 0 & -\sin \omega & \cos \omega \end{bmatrix} \dots\dots\dots(2.7)$$

2. Rotasi kedua yaitu rotasi phi (φ) dilakukan terhadap sumbu Y_φ , dengan sistem kaedah tangan kanan perputaran bernilai positif jika berlawanan arah jarum jam seperti Gambar (2.4) (Geosystem, 2006).

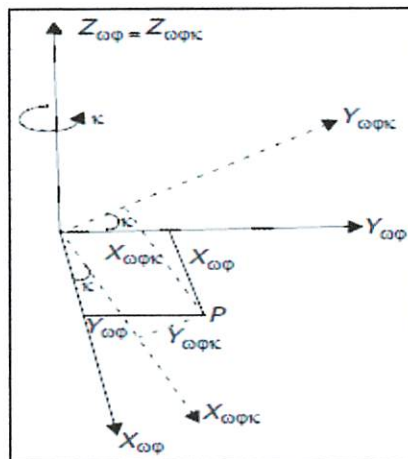


Gambar 2.4 Rotasi sudut phi terhadap sumbu Y_φ

Dari Gambar (2.4) sistem rotasi ($X_{\omega\phi}, Y_{\omega\phi}, Z_{\omega\phi}$) dapat diperoleh vektornya $[X_{\omega\phi} \ Y_{\omega\phi} \ Z_{\omega\phi}]^T = R_{\phi} [X \ Y \ Z]^T$, dimana (Cooper dan Robson, 2001) :

$$R_{\phi} = \begin{bmatrix} \cos \phi & 0 & -\sin \phi \\ 0 & 1 & 0 \\ \sin \phi & 0 & \cos \phi \end{bmatrix} \dots\dots\dots(2.8)$$

3. Rotasi ketiga yaitu rotasi kappa (κ) dilakukan terhadap sumbu $Z_{\omega\phi}$, dengan sistem kaedah tangan kanan perputaran bernilai positif jika berlawanan arah jarum jam seperti gambar di bawah ini (Geosystem, 2006).



Gambar 2.5 Rotasi sudut kappa terhadap sumbu Z_{ωφ}

Dari Gambar (2.5) sistem rotasi ($X_{\omega\varphi\kappa}$ $Y_{\omega\varphi\kappa}$ $Z_{\omega\varphi\kappa}$) dapat diperoleh persamaan $[X_{\omega\varphi\kappa} Y_{\omega\varphi\kappa} Z_{\omega\varphi\kappa}]^T = R_{\kappa} [X Y Z]^T$ dimana (Cooper dan Robson, 2001) :

$$R_{\kappa} = \begin{bmatrix} \cos \kappa & \sin \kappa & 0 \\ -\sin \kappa & \cos \kappa & 0 \\ 0 & 0 & 1 \end{bmatrix} \dots\dots\dots(2.9)$$

Dengan menggabungkan Persamaan (2.27, 2.28, 2.29) maka akan diperoleh persamaannya sebagai berikut (Mikhail et al., 2001):

$$R = R_{\omega} \times R_{\varphi} \times R_{\kappa} \dots\dots\dots(2.10)$$

dimana R merupakan matrik 3x3, dengan nilai elemen masing-masing matrik sebagai berikut :

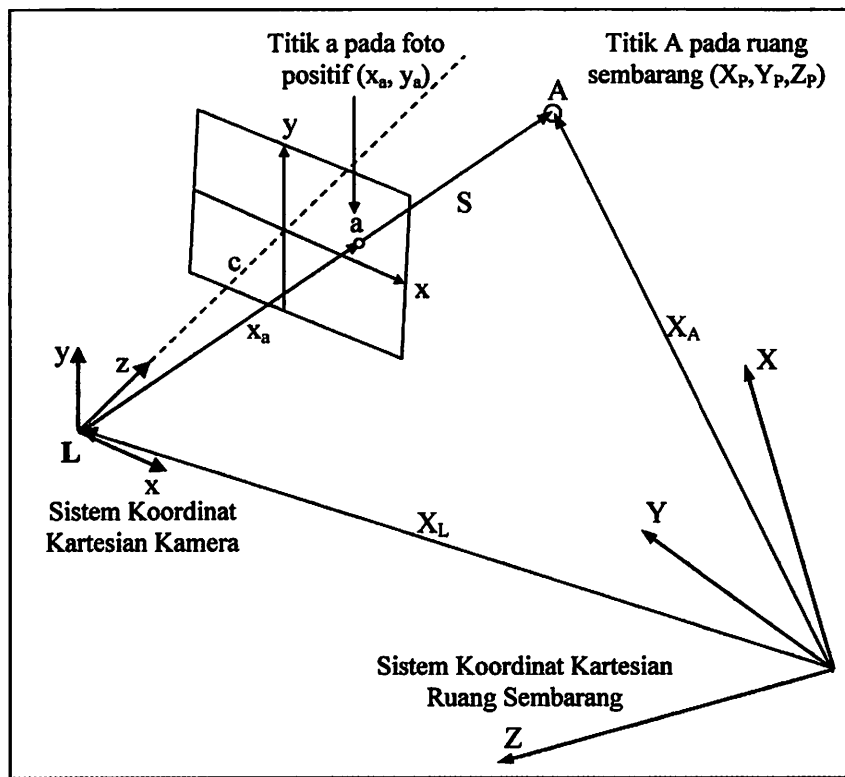
$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \dots\dots\dots(2.11)$$

dimana :

$$\begin{aligned} r_{11} &= \cos \varphi \cos \kappa \\ r_{12} &= \sin \omega \sin \varphi \cos \kappa + \cos \omega \sin \kappa \\ r_{13} &= -\cos \omega \sin \varphi \cos \kappa + \sin \omega \sin \kappa \\ r_{21} &= -\cos \varphi \sin \kappa \\ r_{22} &= -\sin \omega \sin \varphi \sin \kappa + \cos \omega \cos \kappa \dots\dots\dots(2.12) \\ r_{23} &= \cos \omega \sin \varphi \sin \kappa + \sin \omega \cos \kappa \\ r_{31} &= \sin \varphi \\ r_{32} &= -\sin \omega \cos \kappa \\ r_{33} &= \cos \omega \cos \varphi \end{aligned}$$

2.3.1.2 Transformasi Perspektif Pusat (*Central Perspective Transformation*)

Jika didalam kondisi yang ideal hubungan antara garis lurus Aa, sistem koordinat kamera dan sistem koordinat ruang dapat dijelaskan seperti Gambar (2.6), vektor S merupakan perpanjangan garis lurus Aa sebesar AL, dimana titik L berada pada bidang CCD/CMOS (Cooper dan Robson, 2001).



Gambar 2.6 Hubungan Keruangan antara titik A sistem koordinat sembarang dengan titik a sistem koordinat kamera

Berdasarkan hubungan antara vektor S, X_A dan X_L pada Gambar (2.6), vektor posisi titik A pada sistem koordinat ruang dapat ditulis sebagai berikut :

$$X_A = X_L + S \dots\dots\dots(2.13)$$

Didalam proses pemotretan berkas sinar dari A ke L merupakan garis lurus (Mikhail et al, 2001), maka vektor S pada sistem koordinat kamera haruslah dikonversikan ke dalam sistem koordinat ruang dan dikalikan faktor skala perbesaran (Cooper dan Robson, 2001) maka persamaannya menjadi :

$$X_A = X_L + \lambda R^T x_a \dots\dots\dots (2.14)$$

atau jika dinyatakan dengan notasi matrik akan setara dengan :

$$\begin{bmatrix} X_A \\ Y_A \\ Z_A \end{bmatrix} = \begin{bmatrix} X_L \\ Y_L \\ Z_L \end{bmatrix} + \lambda \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ -c \end{bmatrix} \dots\dots\dots (2.15)$$

dimana λ adalah faktor skala, x adalah vektor komponen titik a di dalam sistem koordinat kamera; dan X adalah vektor komponen titik A didalam sistem koordinat ruang (X_A, Y_A, Z_A) adalah koordinat titik A dan sistem ruang (X_L, Y_L, Z_L) merupakan koordinat titik L (pusat perspektif kamera) pada sistem ruang. Sedangkan R merupakan matrik rotasi dengan sudut perputaran omega (ω), phi (ϕ), dan kappa (κ) (Mikhail et al., 2001).

Dari Persamaan (2.14) dan (2.15) adalah sistem koordinat dari sistem koordinat kamera ke sistem koordinat ruang. Jika arah transformasi dibalik menjadi sistem koordinat ruang ke sistem koordinat kamera, maka Persamaan (2.14) menjadi :

$$x_a = \lambda^{-1} R(X_A - XL) \dots\dots\dots (2.16)$$

atau jika dinyatakan dengan notasi matrik untuk titik sembarang akan setara dengan :

$$\begin{bmatrix} x - x_o \\ y - y_o \\ -c \end{bmatrix} = \lambda^{-1} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_A - X_L \\ Y_A - Y_L \\ Z_A - Z_L \end{bmatrix} \dots\dots\dots(2.17)$$

Dimana $(x, y, -c)$ adalah nilai koordinat pada sistem kamera dan (X_A, Y_A, Z_A) adalah koordinat sistem ruang, serta (x_o, y_o) adalah koordinat titik tengah foto (*principal point*), dimana *principal point* merupakan nilai titik tengah *pixel* pada bidang CCD/CMOS yang tidak selalu berimpit dengan titik tengah foto (Wolf dan Dewitt, 2000).

2.3.1.3 Kondisi Kolinear

Dari Persamaan (2.17), karena faktor skala tidak dibutuhkan didalam pembahasan selanjutnya maka faktor skala ini dieliminir dengan membagi elemen baris pertama dan kedua dibagi dengan elemen baris ketiga pada Persamaan (2.15), maka persamaannya akan menjadi (Mikhail et al., 2001):

$$\begin{aligned} x_a = x_o - c \left[\frac{r_{11}(X_A - X_L) + r_{12}(Y_A - Y_L) + r_{13}(Z_A - Z_L)}{r_{31}(X_A - X_L) + r_{32}(Y_A - Y_L) + r_{33}(Z_A - Z_L)} \right] \\ y_a = y_o - c \left[\frac{r_{21}(X_A - X_L) + r_{22}(Y_A - Y_L) + r_{23}(Z_A - Z_L)}{r_{31}(X_A - X_L) + r_{32}(Y_A - Y_L) + r_{33}(Z_A - Z_L)} \right] \dots\dots(2.18) \end{aligned}$$

Persamaan ini merupakan persamaan garis lurus (persamaan kolinear) antara titik AaL seperti Gambar (2.6). Didalam Persamaan (2.18), jika titik-titik koordinat pada foto digital diukur (ditentukan nilainya) dan disimbolkan dengan \underline{x} , maka yang menjadi parameter (*unknowns*) adalah x_o dan y_o yang dikenal sebagai parameter orientasi dalam kamera; sedangkan

parameter orientasi kamera (\underline{O}) atau orientasi dan posisi kamera dikenal $\omega, \varphi, \kappa, X_L, Y_L, Z_L$; serta X, Y, Z yang merupakan titik-titik koordinat objek yang terekam pada saat pemotretan yang disimbolkan sebagai \underline{X} (Wolf dan Dewitt, 2000).

Jika penulisan Persamaan (2.18) disederhanakan, maka akan menjadi (Fraser, 1997) :

$$f(\underline{x}, \underline{O}, \underline{X})_{i,j} = 0 \dots\dots\dots(2.19)$$

Keterangan :

i, j = foto yang ke-i dan titik objek yang ke-j yang terekam oleh kamera.

\underline{x} = vektor yang berisikan nilai-nilai koordinat kamera untuk titik-titik objek pada foto ke-i.

\underline{O} = parameter orientasi luar kamera ($\omega_i, \varphi_i, \kappa_i, X_{L_i}, Y_{L_i}, Z_{L_i}$) untuk foto ke-i

\underline{X} = Koordinat ruang untuk titik yang ke-j (X_j, Y_j, Z_j)

2.3.2 Persamaan Observasi

Metode gabungan kuadrat terkecil mensyaratkan agar persamaan dituliskan untuk masing-masing observasi yang secara fisik dibuat dan untuk masing-masing parameter yang digunakan dalam model fungsional. Persamaan ini disebut dengan persamaan observasi. Persamaan observasi dikembangkan dalam tiga kelompok variabel, yaitu (King, 1993) :

1. *The plate coordinates*
2. The camera's exterior orientation parameters
3. The object point coordinates

Dari persamaan kolinier dapat diturunkan matrik untuk menghitung nilai koreksi setiap parameter sebagai berikut (Wong, 1980) :

$$\begin{bmatrix} Vx_{ij} \\ Vy_{ij} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & b_{16} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} & b_{26} \end{bmatrix} \begin{bmatrix} \Delta\omega_i \\ \Delta\phi_i \\ \Delta\kappa_i \\ \Delta X_i^c \\ \Delta Y_i^c \\ \Delta Z_i^c \end{bmatrix} \dots\dots (2.20)$$

$$+ \begin{bmatrix} b_{17} & b_{18} & b_{19} \\ b_{27} & b_{28} & b_{29} \end{bmatrix} \begin{bmatrix} \Delta X_j \\ \Delta Y_j \\ \Delta Z_j \end{bmatrix} = \begin{bmatrix} -f_x^o \\ -f_y^o \end{bmatrix}$$

atau secara sederhananya sebagai berikut :

$$V_{ij} + \dot{B}_{ij} \dot{\Delta}_i + \ddot{B}_{ij} \ddot{\Delta}_j = \varepsilon_{ij} \dots\dots\dots (2.21)$$

Untuk $\dot{\Delta}$ merupakan koreksi untk parameter orientasi luar dan $\ddot{\Delta}$ merupakan koreksi untk koordinat titik objek. V merupakan matrik residu, B merupakan matrik koefisien.

Berikut ini merupakan persamaan pengamatan untuk koordinat objek titik j yang diketahui (Wong, 1980).

$$\underset{(3,1)}{\ddot{V}_j} - \underset{(3,1)}{\ddot{\Delta}_j} = \underset{(3,1)}{\ddot{C}_j} \dots\dots\dots (2.22)$$

Berikut ini merupakan persamaan pengamatan akhir untuk parameter orientasi luar foto i (Wong, 1980).

$$\underset{(3,1)}{\dot{V}_i} - \underset{(3,1)}{\dot{\Delta}_i} = \underset{(3,1)}{\dot{C}_i} \dots\dots\dots (2.23)$$

2.3.3 Struktur Matrik Bobot

Salah satu keuntungan dari penerapan metode gabungan kuadrat terkecil adalah memungkinkan untuk menggabungkan parameter dan observasi yang digunakan dalam proses penyesuaian dengan mempertimbangkan bobot yang sesuai

Bobot dari sebuah observasi bernilai proporsional dengan varian observasi (σ^2). Varian utama untuk tiap bobot dilambangkan dengan (σ_0^2). Bobot dari observasi dinyatakan sebagai berikut

$$\text{(King, 1993) : } W = \sigma_0^2 / \sigma^2 \dots\dots\dots(2.24)$$

Dimana varian $\sigma_i^2 = \sigma_0^2$ akan mempunyai bobot 1 (satu)

Matrik bobot untuk *plate coordinates observation* dalam *i* titik pada foto *j* sebagai berikut (Wong, 1980) :

$$\sigma_{ij} = \begin{bmatrix} \sigma_{x_{ij}}^2 & \sigma_{y_i x_{ij}} \\ \sigma_{y_i x_{ij}} & \sigma_{y_{ij}}^2 \end{bmatrix} \dots\dots\dots(2.25)$$

Dimana $\sigma_{x_{ij}}^2$ dan $\sigma_{y_{ij}}^2$ merupakan varian dari pengukuran koordinat foto x_{ij} dan y_{ij} , dan $\sigma_{x_{ij}y_{ij}}$ adalah *covarian* dari kedua pengukuran tersebut. Acuan matrik bobot dapat digambarkan sebagai berikut :

$$W_{ij} = \sigma_0^2 \sigma_{ij}^{-1} \dots\dots\dots(2.26)$$

(2,2)

Untuk keseluruhan matrik bobot untuk *m* foto dan *n* titik sebagai berikut (Wong, 1980) :

$$W_{(2mn, 2mn)} = \begin{bmatrix} W_1 & & \\ & \ddots & \\ & & W_{mn} \end{bmatrix} \dots\dots\dots(2.27)$$

Pembahasan mengenai matrik bobot untuk kamera dan objek sebagai berikut (King, 1993).

Matrik bobot diagonal untuk kamera i :

$$\dot{\sigma}_{(6,6)} = \begin{bmatrix} \sigma_{\omega}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{\phi}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{\kappa}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{XL}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{YL}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{ZL}^2 \end{bmatrix} \dots\dots\dots(2.28)$$

Acuan matrik bobot dapat digambarkan sebagai berikut :

$$\dot{W}_i = \sigma_0^2 \dot{\sigma}_i^{-1} \dots\dots\dots(2.29)$$

untuk keseluruhan m foto sebagai berikut :

$$\dot{W}_{(6m,6m)} = \begin{bmatrix} \dot{W}_1 & & \\ & \ddots & \\ & & \dot{W}_m \end{bmatrix} \dots\dots\dots(2.30)$$

Matrik diagonal untuk titik objek j sebagai berikut :

$$\sigma_j = \begin{bmatrix} \sigma_{X_j}^2 & 0 & 0 \\ 0 & \sigma_{Y_j}^2 & 0 \\ 0 & 0 & \sigma_{Z_j}^2 \end{bmatrix} \dots\dots\dots(2.31)$$

Acuan matrik bobot dapat digambarkan sebagai berikut :

$$\ddot{W}_j = \sigma_0^2 \ddot{\sigma}_j^{-1} \dots\dots\dots(2.32)$$

untuk keseluruhan n titik objek sebagai berikut :

$$\ddot{W}_{(3n,3n)} = \begin{bmatrix} \ddot{W}_1 & & \\ & \ddots & \\ & & \ddot{W}_n \end{bmatrix} \dots\dots\dots(2.33)$$

Total untuk matrik bobot dari gabungan persamaan bobot diatas sebagai berikut (Wong, 1980) :

$$\bar{W} = \begin{bmatrix} W & & \\ & \dot{W} & \\ & & \ddot{W} \end{bmatrix} \dots\dots\dots(2.34)$$

2.3.4 Formula Model Matematika

Diasumsi jika titik j berada ditiap m foto, dengan menggunakan persamaan kolinier, maka dapat disusun matriknya sebagai berikut (Wong, 1980) :

$$\begin{bmatrix} V_{1j} \\ \vdots \\ V_{mj} \end{bmatrix}_{(2m,1)} + \begin{bmatrix} \dot{B}_{1j} \\ \vdots \\ \dot{B}_{mj} \end{bmatrix}_{(2mn,6m)} \begin{bmatrix} \dot{\Delta}_1 \\ \vdots \\ \dot{\Delta}_m \end{bmatrix}_{(6m,1)} + \begin{bmatrix} \ddot{B}_1 & & \\ & \ddots & \\ & & \ddot{B}_n \end{bmatrix}_{(2mn,3n)} \begin{bmatrix} \ddot{\Delta}_1 \\ \vdots \\ \ddot{\Delta}_n \end{bmatrix}_{(3n,1)} = \begin{bmatrix} \varepsilon_{1j} \\ \vdots \\ \varepsilon_{mj} \end{bmatrix}_{(2mn,1)} \dots\dots(2.35)$$

Secara sederhana dapat dinyatakan sebagai berikut (Wong, 1980) :

$$V + \dot{B}\dot{\Delta} + \ddot{B}\ddot{\Delta} = f \dots\dots\dots(2.36)$$

Untuk n titik didalam m foto dengan mengacu pada persamaan observasi digabungkan untuk membentuk persamaan pengamatan sebagai berikut (King, 1993) :

$$V + B\Delta = f \dots\dots\dots(2.37)$$

Dimana Δ merupakan matrik koreksi parameter yang dicari, V adalah matrik residu, B merupakan matrik koefisien dan f yaitu matrik observasi. Dari Persamaan (2.37) akan dihasilkan persamaan normal sebagai berikut (Wong, 1980) :

$$(B^T W B)\Delta = (B^T W f) \dots\dots\dots(2.38)$$

Jika sekumpulan persamaan dikombinasikan dengan persamaan matrik bobotnya (W), solusi kuadrat terkecilnya sebagai berikut (King, 1993) :

$$\Delta = (B^T W B)^{-1} B^T W f \dots\dots\dots(2.39)$$

$$= N^{-1} C$$

2.3.5 Linierisasi Persamaan Kolinier

Persamaan dasar dari *bundle adjustment* adalah persamaan kolinier, yang mendiskripsikan satuan dasar dari *photogrammetry*. Dengan menggunakan persamaan kolinier membuat bentuk persamaan dan proses penyelesaiannya akan lebih efisien (Mikhail et al., 2001). Bentuk linier dari persamaan kolinier untuk foto i dan titik j . Karena persamaan pengamatan garis lurus pada Persamaan (2.19) tidak linier, maka untuk menghitung parameter didalamnya harus dilinierkan terlebih dahulu dengan menggunakan deret Taylor sebagai berikut (Stewart, 1999) :

$$\left(\frac{df}{d\underline{x}} \cdot d\underline{x}\right)_j + \left(\frac{df}{d\underline{O}} \cdot d\underline{O}\right)_j + \left(\frac{df}{d\underline{X}} \cdot d\underline{X}\right)_j + f(\underline{x}, \underline{O}^0, \underline{X}^0)_{ij} = 0$$

.....(2.40)

Bentuk linier dari persamaan pada Persamaan (2.40) ini setara dengan persamaan hitung kuadrat terkecil (*least square observation*) menurut Mikhail et al. (2001) yaitu :

$$\underset{(2,1)}{v} + \underset{(2,6)(6,1)}{\dot{B}} \underset{(2,3)(3,1)}{\dot{\Delta}} + \underset{(2,3)(3,1)}{\ddot{B}} \underset{(2,1)}{\ddot{\Delta}} = \underset{(2,1)}{f} \dots\dots\dots(2.41)$$

dengan :

$$O = O^0 + \dot{\Delta}, \quad X = X^0 + \ddot{\Delta} \dots\dots\dots(2.42)$$

Keterangan :

$$v = \begin{bmatrix} v_x & v_y \end{bmatrix}_{ij}^T = \text{Residu dari koordinat foto}$$

$$f = f(x, O^0, X^0)_{ij} = \text{Observasi}$$

$$\dot{\Delta} = [d\omega, d\phi, d\kappa, dXL, dYL, dZL]_{ij}^T = \text{Koreksi parameter EO (6m vektor)}$$

$$\ddot{\Delta} = [dX_j, dY_j, dZ_j]_{ij}^T = \text{Koreksi Koordinat koreksi XYZ (3n vektor)}$$

2.3.6 Desain Matrik untuk *Bundle Adjustment*

Menurut Mikhail et al. (2001) dan Wolf dan Dewitt (2000), jika sebuah titik objek (objek ke-j) terekam pada sebuah foto (foto ke-i), maka dimensi-dimensi matrik didalam Persamaan (2.37) adalah sebagai berikut :

$$B_{ij} = \begin{pmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} \end{pmatrix}_{ij} \dots\dots\dots (2.43)$$

$$v_{ij} = \begin{pmatrix} v_x \\ v_y \end{pmatrix}_{ij} \dots\dots\dots (2.44)$$

Susunan matrik B_i sebagai berikut (Fraser, 1997):

$$\dot{B}_{ij} = \begin{bmatrix} \left(\frac{\partial f_x}{\partial \omega} \right)_0 & \left(\frac{\partial f_x}{\partial \phi} \right)_0 & \left(\frac{\partial f_x}{\partial \kappa} \right)_0 & \left(\frac{\partial f_x}{\partial X_L} \right)_0 & \left(\frac{\partial f_x}{\partial Y_L} \right)_0 & \left(\frac{\partial f_x}{\partial Z_L} \right)_0 \\ \left(\frac{\partial f_y}{\partial \omega} \right)_0 & \left(\frac{\partial f_y}{\partial \phi} \right)_0 & \left(\frac{\partial f_y}{\partial \kappa} \right)_0 & \left(\frac{\partial f_y}{\partial X_L} \right)_0 & \left(\frac{\partial f_y}{\partial Y_L} \right)_0 & \left(\frac{\partial f_y}{\partial Z_L} \right)_0 \end{bmatrix}_{ij} \dots\dots\dots (2.45)$$

atau secara sederhana sebagai berikut (Wolf dan Dewitt, 2000).

$$\dot{B}_{ij} = \begin{bmatrix} (b_{11})_0 & (b_{12})_0 & (b_{13})_0 & (-b_{14})_0 & (-b_{15})_0 & (-b_{16})_0 \\ (b_{21})_0 & (a_{22})_0 & (b_{23})_0 & (-b_{24})_0 & (-b_{25})_0 & (-b_{26})_0 \end{bmatrix}_{ij} \quad (2.46)$$

Susunan matrik B_2 sebagai berikut (Fraser, 1997) :

$$\ddot{B}_{ij} = \begin{bmatrix} \left(\frac{\partial f_x}{\partial X_i}\right)_0 & \left(\frac{\partial f_x}{\partial X_i}\right)_0 & \left(\frac{\partial f_x}{\partial X_i}\right)_0 \\ \left(\frac{\partial f_x}{\partial X_i}\right)_0 & \left(\frac{\partial f_x}{\partial X_i}\right)_0 & \left(\frac{\partial f_x}{\partial X_i}\right)_0 \end{bmatrix}_{ij} \dots\dots\dots (2.47)$$

atau (Wolf dan Dewitt, 2000)

$$\ddot{B}_{ij} = \begin{bmatrix} (b_{14})_0 & (b_{15})_0 & (b_{16})_0 \\ (b_{24})_0 & (b_{25})_0 & (b_{26})_0 \end{bmatrix}_{ij} \dots\dots\dots (2.48)$$

\dot{B} mempunyai dimensi 2x6 sedangkan \ddot{B} mempunyai dimensi 2x3 dimana untuk tiap titik objek ke-j yang terekam pada foto ke-i.

Susunan matrik f sebagai berikut (Fraser, 1997) :

$$f_{ij} = \begin{bmatrix} J \\ K \end{bmatrix}_{ij} \dots\dots\dots (2.49)$$

Penjelasan elemen matrik B dan matrik f sebagai berikut (Wolf dan Dewitt, 2000) :

$$b_{11} = \frac{f}{q^2} [r(-m_{33}\Delta Y + m_{32}\Delta Z) - q(-m_{13}\Delta Y + m_{12}\Delta Z)]$$

$$b_{12} = \frac{f}{q^2} [r(\cos \phi \Delta X + \sin \omega \sin \phi \Delta Y - \cos \omega \sin \phi \Delta Z) - q(-\sin \phi \cos \kappa \Delta X + \sin \omega \cos \phi \cos \kappa \Delta Y - \cos \omega \cos \phi \cos \kappa \Delta Z)]$$

$$b_{13} = \frac{-f}{q} (m_{21}\Delta X + m_{22}\Delta Y + m_{23}\Delta Z)$$

$$b_{14} = \frac{f}{q^2}(rm_{31} - qm_{11})$$

$$b_{15} = \frac{f}{q^2}(rm_{32} - qm_{12})$$

$$b_{16} = \frac{f}{q^2}(rm_{33} - qm_{13})$$

$$b_{21} = \frac{f}{q^2}[s(-m_{33}\Delta Y + m_{32}\Delta Z) - q(-m_{23}\Delta Y + m_{22}\Delta Z)]$$

$$b_{22} = \frac{f}{q^2}[s(\cos \phi \Delta X + \sin \omega \sin \phi \Delta Y - \cos \omega \sin \phi \Delta Z) - q(-\sin \phi \cos \kappa \Delta X - \sin \omega \cos \phi \sin \kappa \Delta Y + \cos \omega \cos \phi \sin \kappa \Delta Z)]$$

$$b_{23} = \frac{f}{q}(m_{11}\Delta X + m_{12}\Delta Y + m_{13}\Delta Z)$$

$$b_{24} = \frac{f}{q^2}(sm_{31} - qm_{21})$$

$$b_{25} = \frac{f}{q^2}(sm_{32} - qm_{22})$$

$$b_{26} = \frac{f}{q^2}(sm_{33} - qm_{23})$$

$$J = x_a - x_o + f \frac{r}{q}$$

$$K = y_a - y_o + f \frac{s}{q}$$

..... (2.50)

dimana elemen *rsq* dapat dinyatakan sebagai berikut (Wolf, 1993) :

$$\begin{aligned} r &= r_{11}(X_A - X_L) + r_{12}(X_A - X_L) + r_{13}(X_A - X_L) \\ s &= r_{21}(X_A - X_L) + r_{22}(X_A - X_L) + r_{23}(X_A - X_L) \dots\dots\dots (2.51) \\ q &= r_{31}(X_A - X_L) + r_{32}(X_A - X_L) + r_{33}(X_A - X_L) \end{aligned}$$

Untuk matrik koreksi $\dot{\Delta}_i$ (Koreksi parameter orientasi luar kamera) dan $\ddot{\Delta}_j$ (koreksi koordinat titik objek 3D) struktur matriknya sebagai berikut (Mikhail et al. 2001) :

$$\dot{\Delta}_i = [\Delta\omega \ \Delta\phi \ \Delta\kappa \ \Delta X_L \ \Delta Y_L \ \Delta Z_L]_i^T \dots\dots\dots (2.52)$$

(6*m*,1)

$$\ddot{\Delta}_j = [\Delta X \ \Delta Y \ \Delta Z]_j^T \dots\dots\dots (2.53)$$

(3*n*,1)

Dimana $\dot{\Delta}$ besar dimensinya 6*m*x1 dan $\ddot{\Delta}$ besar dimensinya 3*n*x1

2.3.7 Struktur dari Matrik Persamaan Normal

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

$$\begin{bmatrix} \dot{B}_y^T W_y \dot{B}_y & \dot{B}_y^T W_y \ddot{B}_y \\ \ddot{B}_y^T W_y \dot{B}_y & \ddot{B}_y^T W_y \ddot{B}_y \end{bmatrix} \begin{bmatrix} \dot{\Delta}_i \\ \ddot{\Delta}_j \end{bmatrix} = \begin{bmatrix} \dot{B}_y^T W_y f_y \\ \ddot{B}_y^T W_y f_y \end{bmatrix} \dots\dots\dots (2.54)$$

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

$$\begin{bmatrix} \dot{N}_i & \bar{N}_y \\ \bar{N}_y^T & \dot{N}_j \end{bmatrix} \begin{bmatrix} \dot{\Delta}_i \\ \ddot{\Delta}_j \end{bmatrix} = \begin{bmatrix} \dot{K}_i \\ \ddot{K}_j \end{bmatrix} \dots\dots\dots (2.55)$$

atau,

$$\overset{N}{(6m+3n, 6m+3n)} \overset{\Delta}{(6m+3n, 1)} = \overset{K}{(6m+3n, 1)} \dots\dots\dots (2.56)$$

Perlu diperhatikan bahwa \dot{N} mengandung koefisien dari parameter foto, \dot{N} mengandung koefisien parameter titik (koordinat), \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).

Dari pengembangan persamaan kolinear pada Persamaan (2.40), maka akan dapat dibentuk persamaan yang elemen di dalamnya didefinisikan sebagai berikut :

$$\begin{aligned} \dot{N}_{ij} &= \sum_{j=1}^n B_{1ij}^T W_j B_{1ij} ; \dot{K}_i = \sum_{j=1}^n B_{1ij}^T W_j f_{ij} \\ \ddot{N}_{ij} &= \sum_{i=1}^m B_{2ij}^T W_j B_{2ij} ; \ddot{K}_j = \sum_{i=1}^m B_{2ij}^T W_j f_{ij} \dots\dots\dots(2.57) \\ \bar{N}_{ij} &= B_{1ij}^T W_j B_{2ij} \end{aligned}$$

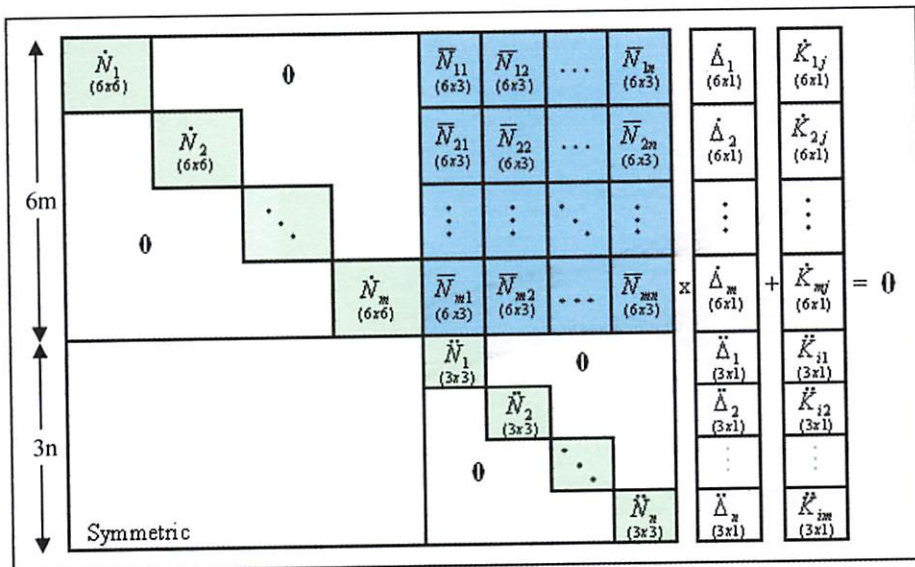
Dimana \dot{N} dan \ddot{N} adalah sub matrik dari matrik blok-diagonal, dimana blok \dot{N} mengacu pada parameter EO dan \ddot{N} mengacu pada koordinat titik-titik objek serta \bar{N} mengacu pada keduanya. Penyusunan matrik N dapat disajikan sebagai berikut dengan m banyak foto dan n banyak titik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

$$N_{(6m+3n) \times (6m+3n)} = \left[\begin{array}{cccc|cccc} \dot{N}_{1j} & 0 & \dots & 0 & \bar{N}_{11} & \bar{N}_{12} & \dots & \bar{N}_{1n} \\ 0 & \dot{N}_{2j} & \dots & 0 & \bar{N}_{21} & \bar{N}_{22} & \dots & \bar{N}_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \dot{N}_{mj} & \bar{N}_{m1} & \bar{N}_{m2} & \dots & \bar{N}_{mn} \\ \hline \bar{N}_{11}^T & \bar{N}_{21}^T & \dots & \bar{N}_{m1}^T & \ddot{N}_{i1} & 0 & \dots & 0 \\ \bar{N}_{12}^T & \bar{N}_{22}^T & \dots & \bar{N}_{m2}^T & 0 & \ddot{N}_{i1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{N}_{1n}^T & \bar{N}_{2n}^T & \dots & \bar{N}_{mn}^T & 0 & 0 & \dots & \ddot{N}_{in} \end{array} \right] \quad (2.58)$$

dan penyusunan untuk matrik Δ dan K sebagai berikut :

$$\Delta_{(6m+3n,1)} = \begin{bmatrix} \dot{\Delta}_1 \\ \dot{\Delta}_2 \\ \vdots \\ \dot{\Delta}_m \\ \ddot{\Delta}_1 \\ \ddot{\Delta}_1 \\ \vdots \\ \ddot{\Delta}_n \end{bmatrix}; \quad K_{(6m+3n,1)} = \begin{bmatrix} \sum_{j=1}^n \dot{K}_{1j} \\ \sum_{j=1}^n \dot{K}_{2j} \\ \vdots \\ \sum_{j=1}^n \dot{K}_{mj} \\ \sum_{i=1}^m \ddot{K}_{i1} \\ \sum_{i=1}^m \ddot{K}_{i2} \\ \vdots \\ \sum_{i=1}^m \ddot{K}_{im} \end{bmatrix} \dots\dots\dots (2.59) \text{ dan } (2.60)$$

Struktur persamaan normal *bundle adjustment* juga dapat disusun seperti Gambar (2.7) sebagai berikut :



Gambar 2.7 Struktur Persamaan Normal

Persamaan diatas adalah teknik *bundle adjustment normal equation* untuk mendapatkan nilai parameter dan koordinat titik objek didalam sistem kartesian 3D.

2.3.8 Penyelesaian Persamaan Normal *Bundle Adjustment*

Untuk penyelesaian persamaan normal diatas dapat diselesaikan sebagai berikut (Fraser, 1997) :

$$\begin{bmatrix} \dot{\Delta} \\ \ddot{\Delta} \end{bmatrix} = \begin{bmatrix} B_1^T W B_1 & B_1^T W B_2 \\ B_2^T W B_1 & B_2^T W B_2 \end{bmatrix}^{-1} \begin{bmatrix} B_1^T W f \\ B_2^T W f \end{bmatrix} \dots\dots\dots(2.61)$$

Penyelesaian dari teknik *bundle adjustment* bertujuan untuk mendapatkan nilai parameter EO (*eksterior orientation*) dan koordinat titik objek didalam sistem kartesian 3D.

Parameter orientasi luar (EO) yang terkoreksi dihitung menggunakan persamaan :

$$O = O^0 + \dot{\Delta} \dots\dots\dots(2.62)$$

Koordinat titik objek yang terkoreksi dihitung menggunakan persamaan berikut :

$$X = X^0 + \ddot{\Delta} \dots\dots\dots(2.63)$$

Dimana O^0 dan X^0 merupakan nilai pendekatan awal. O dan X merupakan nilai terkoreksi dari masing-masing parameter.

2.3.9 Efisiensi Algoritma Perhitungan

Dari persamaan normal seperti Gambar (2.7) dapat dibagi dua persamaan matrik sebagai berikut (Wong, 1980) :

$$\dot{N} \dot{\Delta} + \bar{N} \ddot{\Delta} = \dot{K} \dots\dots\dots(2.64)$$

$$\bar{N}^T \dot{\Delta} + \ddot{N} \ddot{\Delta} = \ddot{K} \dots\dots\dots(2.65)$$

Penyelesaian untuk $\ddot{\Delta}$ dalam Persamaan (2.65) dapat menghasilkan persamaan berikut (Wong, 1980) :

$$\ddot{\Delta} = \ddot{N}^{-1}(\dot{K} - \bar{N}^T \dot{\Delta}) \dots \dots \dots (2.66)$$

Substitusikan Persamaan (2.66) ke dalam Persamaan (2.64) sehingga diperoleh persamaan baru sebagai berikut (Wong, 1980) :

$$(\dot{N} - \bar{N} \ddot{N}^{-1} \bar{N}^T) \dot{\Delta} = (\dot{K} - \bar{N} \ddot{N}^{-1} \dot{K}) \dots \dots \dots (2.67)$$

Selanjutnya dari proses substitusi tersebut, maka Persamaan (2.67) secara sederhana dapat dibentuk sebagai berikut (Wong, 1980) :

$$\underset{(6m,6m)}{S} \underset{(6m,1)}{\dot{\Delta}} = \underset{(6m,1)}{E} \dots \dots \dots (2.68)$$

dimana :

$$S = \sum_{j=1}^n [\dot{N}_j - \bar{N}_j (\ddot{N}_j + \ddot{W}_j)^{-1} \bar{N}_j^T] + \dot{W}_j \dots \dots \dots (2.69)$$

dan

$$E = \sum_{j=1}^n [\dot{K}_j - \bar{N}_j (\ddot{N}_j + \ddot{W}_j)^{-1} (\dot{K}_j - \dot{W}_j)] - \dot{W} \dots \dots \dots (2.70)$$

Setelah diperoleh nilai $\dot{\Delta}$ (koreksi parameter eksterior luar kamera), maka dengan menggunakan Persamaan (2.66) untuk setiap titik objek akan dihitung $\ddot{\Delta}$ (koreksi titik objek), dengan persamaan sebagai berikut (Wong, 1980) :

$$\underset{(3,3)}{\ddot{\Delta}_j} = (\ddot{N}_j + \ddot{W}_j)^{-1} (\dot{K}_j - \dot{W}_j) - Q_j \dot{\Delta} \dots \dots \dots (2.71)$$

untuk $j = 1$ sampai ke- n (n merupakan banyaknya titik objek), dimana persamaan Q_j sebagai berikut (Wong, 1980) :

$$Q_j = (\dot{N}_j + \dot{W}_j)^{-1} \bar{N}_j^T \dots \dots \dots (2.72)$$

2.4 *Separate Adjustment dalam Close Range Photogrammetry*

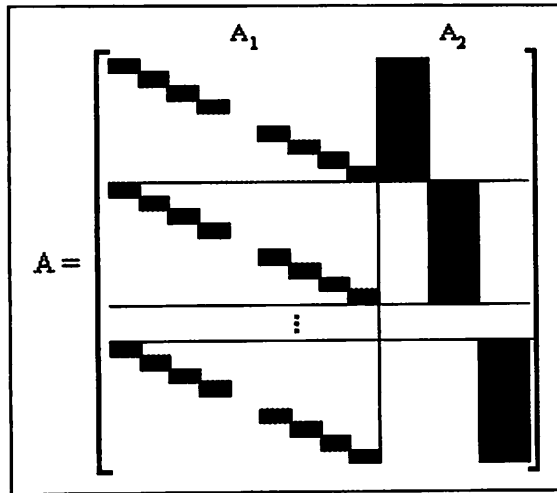
Dalam *Photogrammetry*, parameter yang tidak diketahui diklasifikasikan dalam dua kelompok yaitu parameter koordinat objek dan parameter orientasi luar kamera. Koordinat titik objek adalah persyaratan utama sedangkan output dari parameter kamera tidak terlalu dibutuhkan tetapi harus dimasukkan sebagai parameter yang tidak diketahui dalam persamaan observasi.

2.4.1 *Free Network Separate Adjustment*

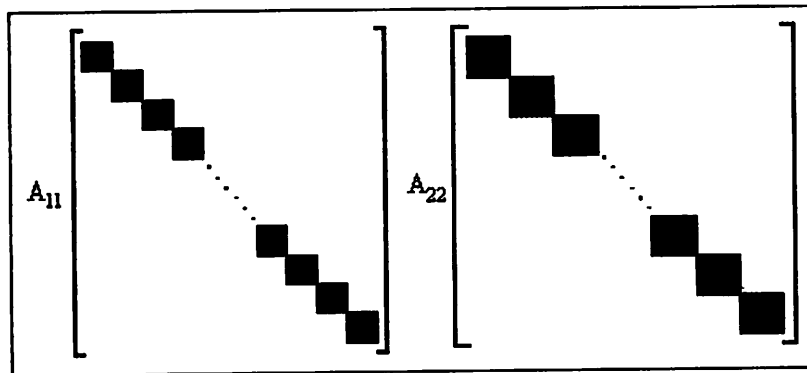
Separate adjustment dan *Bundle adjustment* pada dasarnya memiliki model fungsi yang sama dalam persamaan kolinier. Bentuk persamaan liniernya sebagai berikut (Wang dan Clarke, 1998) :

$$A_1\Delta x_1 + A_2\Delta x_2 = b \dots\dots\dots(2.73)$$

Dimana x_1 merupakan koordinat titik objek dan x_2 merupakan parameter eksterior luar kamera. A_1 dan A_2 adalah desain matrik dari turunan model fungsinya. Struktur matrik untuk A_{11} dan A_{22} sangat istimewa seperti diilustrasikan sebagai berikut (Wang dan Clarke, 1998) :



Gambar 2.8 Struktur Desain Matrik A (A_1 dan A_2)



Gambar 2.9 Struktur Desain Matrik A_{11} dan A_{22}

Pada intinya *separte adjustment* untuk mengolah parameter yang tidak diketahui secara terpisah. *Separate adjustment* mengiterasi diantara dua langkah dan hasilnya akan sama dengan hasil dari *bundle adjustment* yang simultan.

2.4.1.1 Penyesuaian Titik Objek

Saat menyesuaikan titik objek, parameter kamera dinyatakan sebagai konstanta. Sehingga $\Delta x_2 = 0$. Untuk itu persamaan observasi untuk mengestimasi koordinat titik objek sebagai berikut (Wang dan Clarke, 1998) :

$$A_1 \Delta x_1 = b \dots\dots\dots(2.74)$$

Dengan menggunakan metode kuadrat terkecil, koreksi untuk koordinat objek (3D) persamaannya sebagai berikut (Wang dan Clarke, 1998) :

$$\begin{aligned} \Delta x_i &= (A_i^T W_i A_i)^{-1} A_i^T W_i b \\ &= A_{ii}^{-1} A_i^T W_i b \end{aligned} \dots\dots\dots(2.75)$$

Seperti yang telah diilustrasikan pada Gambar (2.8) dan (2.9), besar ukuran submatrik A_i adalah 2×3 sedangkan A_{ii} adalah 3×3 . Untuk keseluruhan besar dimensi A_i adalah $2m \times 3n$ dan A_{ii} adalah $3n \times 3n$ (m adalah banyaknya foto dan n adalah jumlah titik objek). Koreksi koordinat objek 3D untuk titik objek ke- i dihitung dengan persamaan berikut (Wang dan Clarke, 1998) :

$$\Delta x_{ii} = A_{ii}^{-1} A_i^T W_{ii} b_i \dots\dots\dots(2.76)$$

Matrik A_{ii} bisa dihitung nilai keseluruhan sesuai dengan banyaknya kamera, sebagai berikut (Wang dan Clarke, 1998) :

$$\begin{aligned} A_{ii} &= A_{ii}^T W_{ii} A_{ii} \\ &= \sum_{j=1}^m A_{ij}^T W_{ij} A_{ij} \dots\dots\dots(2.77) \\ &= \sum_{j=1}^m A_{i1j} \end{aligned}$$

Dimana A_{ij} dimensi matriknya 2×3 dan W_{ij} besar dimensi matriknya 2×2 , ij merupakan jumlah titik ke- i dalam foto ke- j .

Sedangkan untuk $A_i^T W_{ii} b$ dapat juga dihitung sebagai berikut (Wang dan Clarke, 1998) :

$$A_i^T W_{ii} b_i = \sum_{j=1}^m A_{ij}^T W_{ij} b_{ij} \dots\dots\dots(2.78)$$

Dimana b_j besar dimensinya 2×1 untuk setiap titik i dalam foto j . Dalam kasus ini, besar dimensi untuk koreksi koordinat titik objek sebesar 3×3 .

2.4.1.2 Penyesuaian Parameter Kamera

Saat menyesuaikan parameter kamera, koordinat titik objek dinyatakan konstan. Sehingga $\Delta x_1 = 0$. Untuk itu persamaan observasi untuk mengestimasi parameter kamera sebagai berikut (Wang dan Clarke, 1998) :

$$A_2 \Delta x_2 = b \dots\dots\dots(2.79)$$

Seperti yang telah diilustrasikan pada Gambar (2.8) dan (2.9), besar ukuran submatrik A_2 adalah 2×3 sdangkan A_{22} adalah 3×3 . Untuk keseluruhan besar dimensi A_2 adalah $2m \times 6$ dan A_{22} adalah $6m \times 6m$ (m adalah banyaknya foto dan n adalah jumlah titik objek). Koreksi untuk parameter orientasi luar kamera persamaannya sebagai berikut (Wang dan Clarke, 1998) :

$$\begin{aligned} \Delta x_2 &= (A_2^T W_1 A_2)^{-1} A_2^T W_2 b \dots\dots\dots(2.80) \\ &= A_{22}^{-1} A_2^T W_1 b \end{aligned}$$

Koreksi parameter kamera untuk kamera ke- j dihitung dengan persamaan berikut :

$$\Delta x_{2j} = A_{22j}^{-1} A_{2j}^T W_j b_j \dots\dots\dots(2.81)$$

Matrik A_{22j} bisa dihitung nilai keseluruhan sesuai dengan banyaknya titik, sebagai berikut :

$$\begin{aligned}
A_{22j} &= A_{2j}^T W_{lj} A_{2j} \\
&= \sum_{i=1}^n A_{2ji}^T W_{lj} A_{2ji} \dots\dots\dots(2.82) \\
&= \sum_{i=1}^n A_{22ji}
\end{aligned}$$

Dimana A_{2ji} dimensi matriknya 2×6 dan W_{lj} besar dimensi matriknya 2×2 , ij merupakan jumlah titik ke- i dalam foto ke- j .

Sedangkan untuk $A_{2j}^T W_{lj} b_j$ sebagai berikut :

$$A_{2j}^T W_{lj} b_j = \sum_{i=1}^n A_{2ji}^T W_{lj} b_{ji} \dots\dots\dots(2.83)$$

Dimana b_{ji} besar dimensinya 2×1 untuk setiap titik i dalam foto j . Dalam kasus ini, besar dimensi untuk koreksi parameter eksterior luar kamera sebesar 6×6 .

2.4.2 Pengertian Datum

Sebuah datum wajib ada dalam *bundle adjustment* yang simultan untuk menghilangkan nilai yang salah dalam kolom pada desain matrik sehingga parameter yang tidak diketahui dapat diestimasi (Wang dan Clarke, 1998).

Karena nilai parameter awal bersifat tidak tentu, maka hasil dari *separate adjustment* merupakan sistem koordinat sembarang jika kontrol spasial tidak diterapkan. Hasil dari *separate adjustment* tidak dapat diidentifikasi secara numerik seperti halnya *bundle adjustment* pada umumnya dikarenakan adanya perbedaan pengertian datum (Wang dan Clarke, 1998).

2.4.3 Estimasi Presisi

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 terkecil akan menghasilkan varian dari parameternya. Varian ini sering disebut dengan standard deviasi, dimana standar deviasi merupakan varian yang berbentuk akar kuadrat dan memiliki satuan yang sama dengan parameternya (Mikhail et al, 2001).

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

Dari *separate adjustment*, kofaktor matrik dari koordinat titik objek dapat dihitung secara pendekatan melalui persamaan berikut (Wang dan Clarke, 1998) :

$$Qx_1 = (A_1^T W_1 A_1)^{-1} = A_{11}^{-1} \dots\dots\dots(2.84)$$

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

$$Cx_{1i} = \sigma_o^2 Qx_{1i} \dots\dots\dots(2.85)$$

Kofaktor matrik dari parameter kamera dapat dihitung secara pendekatan melalui persamaan berikut (Wang dan Clarke, 1998) :

$$Qx_2 = (A_2^T W_1 A_2)^{-1} = A_{22}^{-1} \dots\dots\dots(2.86)$$

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

$$Cx_{2j} = \sigma_o^2 Qx_{2j} \dots\dots\dots(2.87)$$

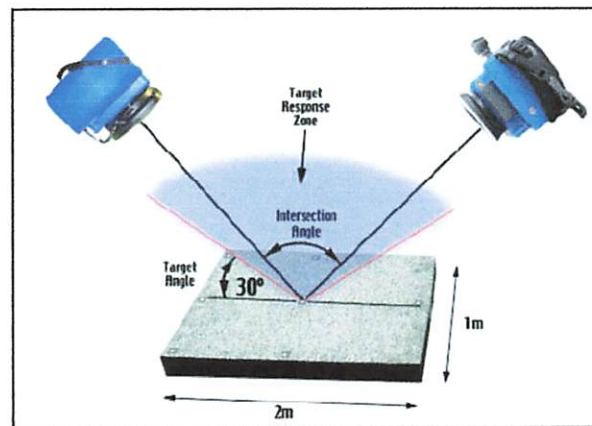
Persyaratan utama dari *bundle adjustment* adalah nilainya harus akurat. Dimana keakuratan dari proses *bundle adjustment* dapat dinilai dari nilai presisi yang sekecil mungkin (Mikhail et al, 2001).

2.5 Network Design

Pekerjaan *close range photogrammetry* tidak jauh dari proses *bundle adjustment*, dengan titik-titik objek dan posisi kamera. Solusi ini mencangkup parameter orientasi luar kamera baik itu parameter tambahan untuk memodelkan efek residual dari kesalahan sistematik.

Untuk akurasi maksimum, geometri pemotretan harus di desain secara hati-hati dan hasilnya diuji sesuai prosedur untuk menjamin hasilnya sesuai dengan spesifikasi yang diinginkan (Mikhail et al, 2001).

Pada umumnya tingkat presisi merupakan sebuah fungsi dari seberapa besar tingkat arah dari sinar yang dipancarkan bias diukur dan kekuatan geometri *intersection* dari sinar yang dipantulkan, dimana keadaan tersebut ditentukan oleh dua faktor antara lain jumlah titik pada tiap-tiap objek dan sudut antara titik-titik tersebut.



Gambar 2.10 Geometri Pemotretan

Sehingga dalam proses perencanaan ini akan melibatkan beberapa tahap yang harus dilaksanakan untuk memaksimalkan tingkat akurasi yang dihasilkan pada proses *close range photogrammetry* antara lain :

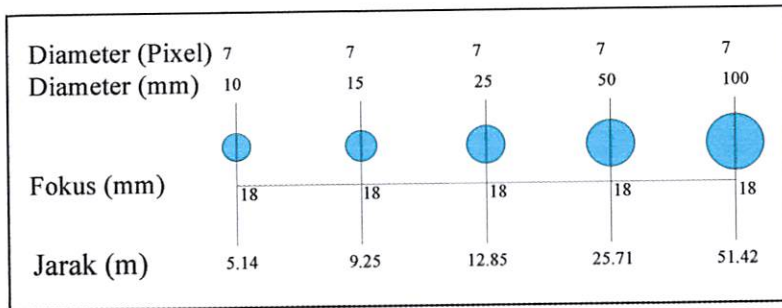
2.5.1 Desain Target

Pada proses pendesainan target yang perlu di perhatikan antara adalah panjang fokus yang akan di gunakan pada saat melakukan pemotretan, jarak objek terhadap posisi pengambilan gambar, dan jumlah *pixel* minimum yang akan terekam pada tiap-tiap target yang akan dipasang. Model persamaan yang digunakan dalam penentuan target ini merupakan persamaan perbandingan yang diberikan sebagai berikut (Manual Photomodeller):

$$r = \left(\frac{p \times fw \times d}{f \times pw} \right) \dots\dots\dots (2.88)$$

dimana f merupakan panjang fokus kamera, p merupakan jumlah *pixel* minimum target, d merupakan jarak kamera ke objek, fw lebar CCD kamera, pw merupakan jumlah horizontal *pixel* foto dan r merupakan diameter target.

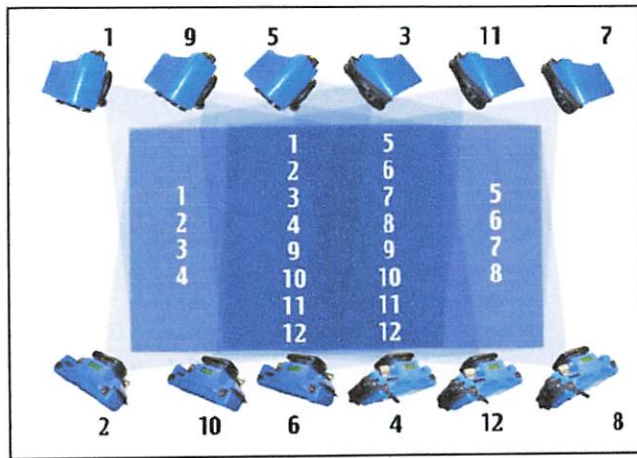
Untuk lebih jelasnya desain target akan diberikan sesuai dengan diagram sebagai berikut dengan jumlah minimum *pixel* sebesar 7 *pixel* dengan panjang fokus dari kamera Nikon D60 sebesar 18 mm (Manual Photomodeller).



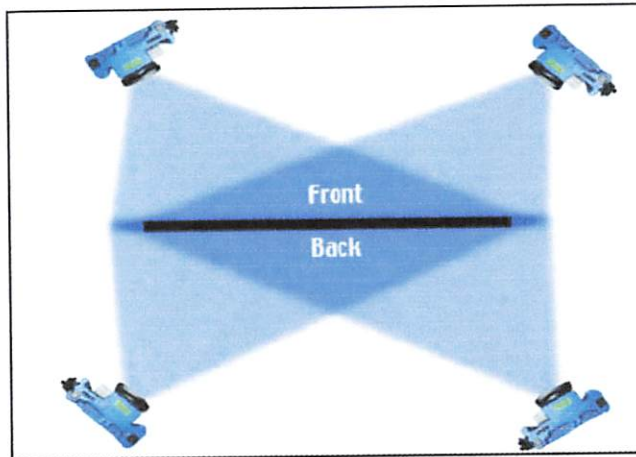
Gambar 2.11 Diagram Desain Target

2.5.2 Desain Pemotretan

Dalam mendesain pemotretan ada beberapa hal yang perlu diperhatikan antara lain adalah jumlah pertampalan masing-masing objek, dan sudut yang terbentuk antara posisi kamera. Sehingga dalam kasus mendesai pemotretan seperti *design left-right measurement* dan *design front-back measurements* dapat dilakukan seperti gambar di bawah ini (V-Stars, 2005) :

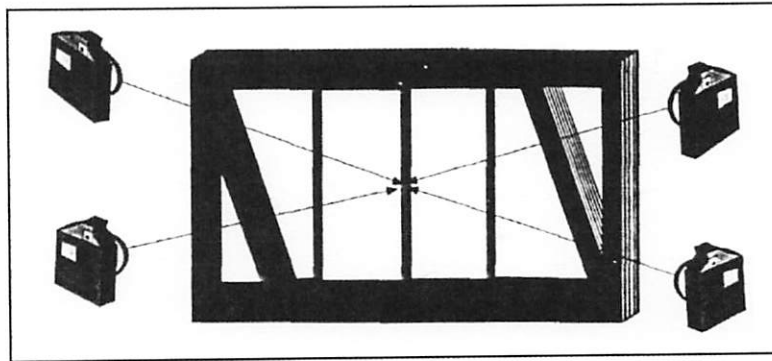


Gambar 2.12 Design Left-Right Measurements

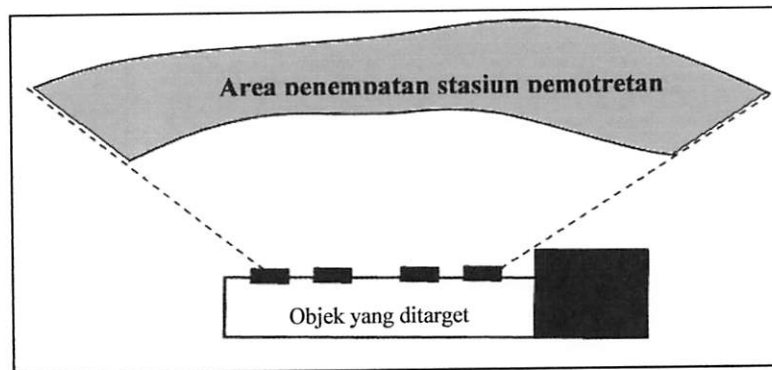


Gambar 2.13 Design Front-Back Measurements

Selain itu desain pemotretan dengan empat stasiun pemotretan mengacu pada objek yang sama yang dikemukakan oleh Fraser (1997) seperti Gambar (2.14). Sedangkan Gambar (2.15) menunjukkan area penempatan stasiun pemotretan yang baik terhadap suatu objek.



Gambar 2.14 Design Jaringan Empat Station



Gambar 2.15 Ilustrasi penempatan stasiun pemotretan (Fraser, 1997)

2.5.3 Permasalahan Datum

Parameter datum sering disebut dengan *zero-variance computational base*. Disini muncul permasalahan dimana diketahui bahwa nilai C_2 akan berbeda sesuai dengan koordinat kontrol minimal yang dipilih, tetapi tidak diketahui koordinat kontrol minimal mana yang dapat menghasilkan C_2 terbaik.

$$\bar{\sigma}_c^2 = \frac{\sigma_0^2}{3n} \text{trace} Cx_2 = \text{a minimum} \dots\dots\dots (2.89)$$

$$\begin{bmatrix} C_1 & \cdot & \cdot \\ \cdot & C_2 & \cdot \\ \cdot & \cdot & C_3 \end{bmatrix} = \sigma_0^2 \begin{bmatrix} B_1^T W B_1 & B_1^T W B_2 & 0 \\ B_2^T W B_1 & B_2^T W B_2 & G \\ 0 & G^T & 0 \end{bmatrix}^{-1} \dots\dots\dots (2.90)$$

dimana transformasi matrik G sebagai berikut (Fraser, 1997) :

$$G_j = \begin{bmatrix} 1 & 0 & 0 & 0 & Z_j & -Y_j & X_j \\ 0 & 1 & 0 & -Z_j & 0 & X_j & Y_j \\ 0 & 0 & 1 & Y_j & -X_j & 0 & Z_j \end{bmatrix} \dots\dots\dots(2.91)$$

dimana parameter matrik kovariannya sebagai berikut :

$$C_\delta = \sigma_0^2(A^TWA)^{-1} \text{ atau } \sigma_0^2(A^TWA + \bar{P})^{-1} \dots\dots\dots(2.92)$$

Matrik kovarian untuk koordinat titik objek sebagai berikut :

$$C_{xyz} = C_{\delta 2} = \begin{bmatrix} \sigma_{x_1^2} & \dots & \dots & \dots & \dots & \dots \\ \vdots & \sigma_{y_1^2} & & & & \vdots \\ \vdots & & \sigma_{z_1^2} & & & \vdots \\ \vdots & & & \ddots & & \vdots \\ \vdots & & & & \ddots & \vdots \\ \vdots & \dots & \dots & \dots & \dots & \sigma_{z_n^2} \end{bmatrix} \dots\dots\dots(2.93)$$

Presisi dari titik objek rata-rata dapat dicari menggunakan persamaan sebagai berikut :

$$\bar{\sigma}_c^2 = \text{trace } C_{xyz} / 3n \dots\dots\dots(2.94)$$

Matrik kovarian C_δ atau C_X diperoleh dari *bundle adjustment minimal constraint* tergantung dari pemilihan tujuh (7) parameter tetapnya. Dimana \bar{P} hanya berhubungan dengan tujuh (7) parameter datum.

BAB III

METODOLOGI

3.1 Persiapan Penelitian

Sebelum melakukan sebuah penelitian, diperlukan suatu persiapan yang matang guna memperlancar proses penelitian sampai penyajian hasil dari batasan penelitian yang dibahas. Untuk memperoleh hasil yang maksimal maka ada beberapa hal yang harus dipersiapkan terlebih dahulu, 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 *bundle adjustment close range photogrammetry* yang berupa data digital.
2. Data parameter *interior orientation* (IO) pendekatan yaitu (x_o, y_o, f) dan koordinat foto (x,y) serta ketelitian (σ_x, σ_y) tiap titik pada tiap foto dari proses *centroid*. Parameter interior mendefinisikan geometri internal sebuah kamera antara lain *perspektive center, principle point*, serta panjang fokus dari kamera.
3. Data parameter *exterior orientation* (EO) pendekatan yaitu $(\omega_0, \varphi_0, \kappa_0, XL_0, YL_0, ZL_0)$ tiap foto yang diperoleh dari proses *relative orientation* dan Proses *resection*.

4. Data parameter *object space* pendekatan yaitu (X_o, Y_o, Z_o) yang diperoleh dari proses *intersection*.

3.1.2 Alat Penelitian

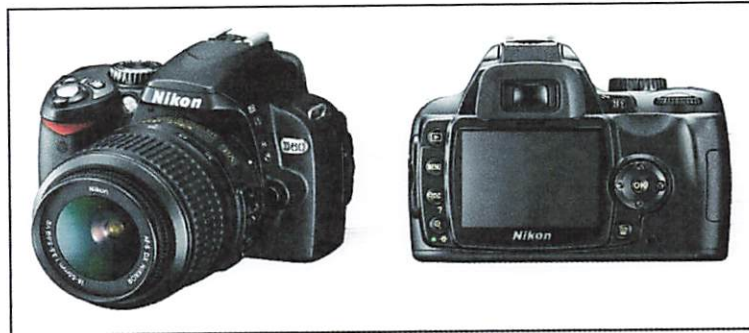
Untuk mendukung pelaksanaan kegiatan penelitian ini, maka diperlukan beberapa perangkat keras (*hardware*) dan perangkat lunak (*software*) antara lain :

3.1.2.1 Perangkat keras (*hardware*), terdiri dari :

a. Kamera SLR Nikon D60

Spesifikasi kamera SLR Nikon D60 sebagai berikut :

- Tipe Lensa : *Single-lens reflex digital camera*
- LCD screen : *2.5-inch*
- Image Sensor : *23.6 x 15.8 mm CCD sensor*



Gambar 3.1 Kamera SLR Nikon D 60 tampak depan dan belakang

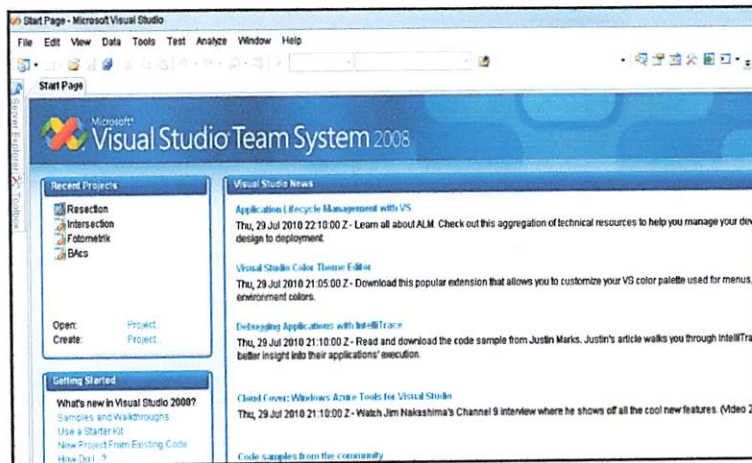


Gambar 3.2 Aksaesoris dari Kamera (Tipe Lensa, Baterai, Memori)

3.1.2.2 Perangkat lunak (*software*), terdiri dari :

a. *Microsoft Visual Studio 2008* (Bahasa Pemrograman C#)

Microsoft Visual Studio 2008 merupakan salah satu perangkat lunak yang berorientasi *object (object-oriented programming* disingkat OOP). Konsep pemrograman orientasi objek menekankan pada : *class, object, method, event* serta *action*.



Gambar 3.4 Tampilan awal Program C#

b. *Microsoft Excel 2003*

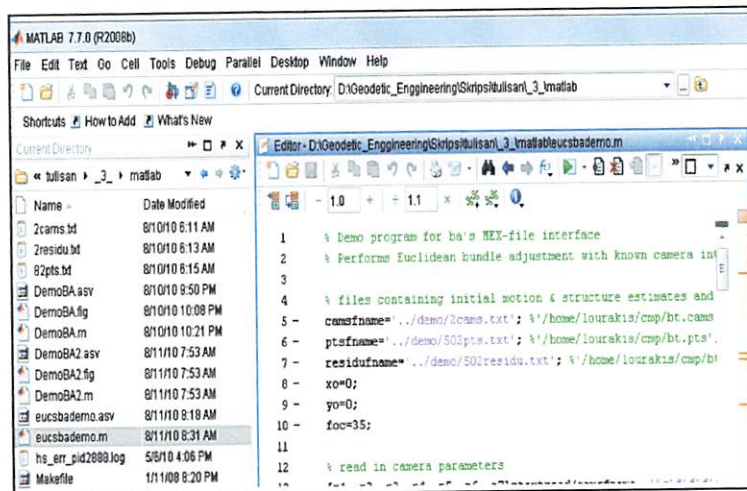
Microsoft Excel 2003 adalah sebuah perangkat lunak spreadsheet, yang berfungsi untuk melakukan proses perhitungan secara langsung tanpa harus menyiapkan bahasa pemrograman terlebih dahulu. Dimana *Microsoft Excel* dapat digunakan untuk untuk menghitung proses perhitungan *bundle adjustment* hingga menemukan nilai akhir.

c. *Microsoft Word 2003*

Microsoft Word 2003 dengan kemampuannya yang telah banyak dikenal dalam era komputerisasi digunakan sebagai media olah kata dalam penyusunan laporan penelitian.

d. *MATLAB 2008*

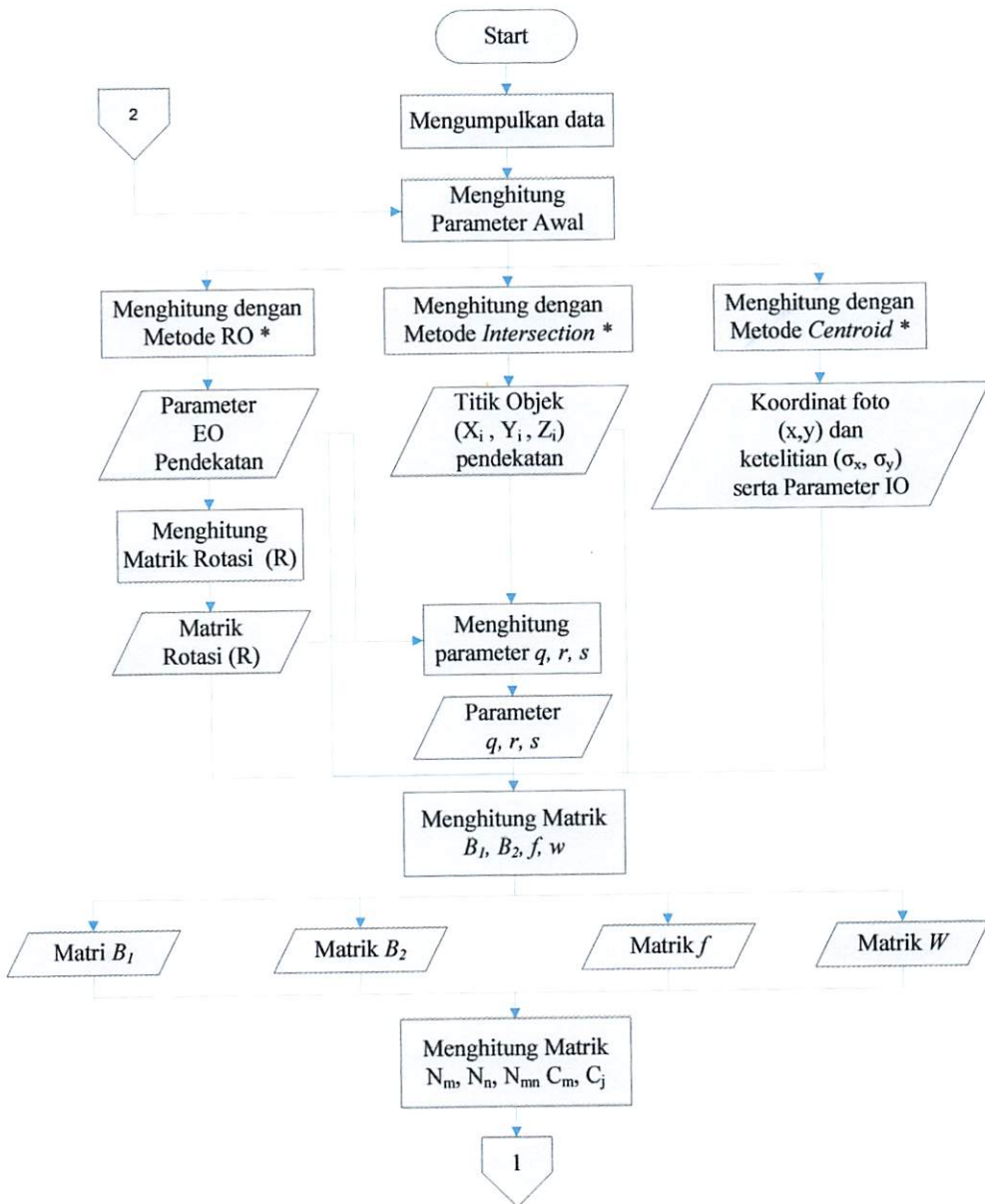
MATLAB adalah bahasa pemrograman level tinggi yang dikhususkan untuk komputasi teknik. Bahasa ini mengintegrasikan kemampuan komputasi, visualisasi dan pemrograman dalam sebuah lingkungan yang mudah digunakan. Matlab memberikan sistem interaktif yang menggunakan konsep matrik sebagai standar variabel elemennya tanpa membutuhkan pendeklarasian array seperti pada bahasa lainnya (Away, 2006).



Gambar 3.5 Tampilan awal Program Matlab 2008

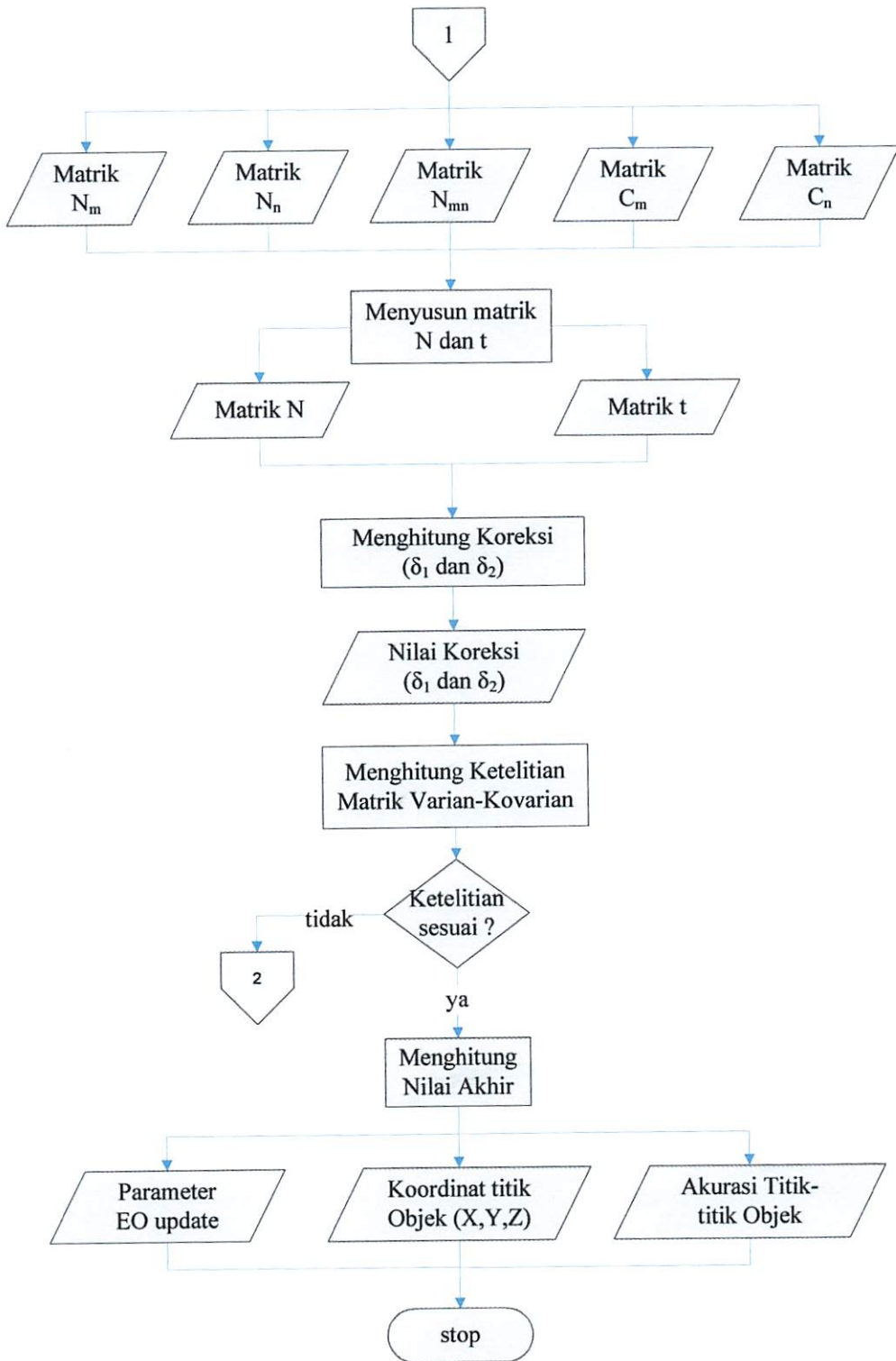
3.2 Diagram Alir Penelitian

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 :



ket : * Proses telah dilakukan oleh peneliti lain pada tahap sebelumnya.

Gambar 3.6 Diagram Alir Penelitian (1)



Gambar 3.7 Diagram Alir Penelitian (2)

3.3 Penjelasan Diagram Alir (*FlowChart*) Penelitian

Aktivitas penelitian ini dimulai dengan tahap persiapan alat, bahan dan perijinan guna untuk kelancaran dalam proses penelitian. Pengambilan data foto / akuisisi data foto sebagai data dasar dalam proses pendesainan perhitungan *bundle adjustment* dalam *close range photogrammetry*. Langkah selanjutnya, dilakukan sesuai dengan diagram alir penelitian ini.

3.3.1 Mengumpulkan Data dan Menghitung Parameter Awal

Mengumpulkan dan menghitung data parameter-parameter awal guna memperlancar proses *bundle adjustment*, dimana diantaranya adalah.

1. Koordinat foto (x,y) dan ketelitian (σ_x σ_y) serta Parameter *interior orientation* (IO). Proses ekstraksi data foto dilakukan untuk mendapatkan nilai-nilai parameter antara lain : koordinat foto (x,y) serta keakurasian titik-titik foto (σ_x σ_y) menggunakan metode centroid.
2. Parameter *exterior orientation* (EO) pendekatan yaitu omega (ω_o), phi (ϕ_o), kappa (κ_o), XL_o , YL_o , ZL_o . Diperoleh melalui proses *relative orientation* (RO).
3. Koordinat titik-titik objek (X_o, Y_o, Z_o)_i pendekatan. Diperoleh menggunakan proses *intersection*.

3.3.2 Menghitung Matrik Rotasi (R)

Untuk menghitung matrik rotasi ini, peneliti menggunakan data omega (ω_o), phi (ϕ_o), kappa (κ_o) dan diproses menggunakan Persamaan (2.12) berikut :

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

hasil dari persamaan tersebut kemudian disusun dalam matrik R :

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

3.3.3 Menghitung Parameter qrs

Parameter qrs dihitung menggunakan data parameter matrik rotasi (R) dan koordinat titik objek pendekatan, dengan menggunakan Persamaan (2.51) sebagai berikut :

$$\begin{aligned}
r &= r_{11}(X_A - X_L) + r_{12}(X_A - X_L) + r_{13}(X_A - X_L) \\
s &= r_{21}(X_A - X_L) + r_{22}(X_A - X_L) + r_{23}(X_A - X_L) \\
q &= r_{31}(X_A - X_L) + r_{32}(X_A - X_L) + r_{33}(X_A - X_L)
\end{aligned}$$

3.3.4 Menghitung Matrik B_1, B_2, f, w

Data yang digunakan dalam menghitung matrik B_1, B_2, f, w adalah parameter EO , koordinat titik objek pendekatan, koordinat foto (x, y) dan ketelitian (σ_x, σ_y) serta parameter IO $(x_o, y_o, focus)$, matrik rotasi (R) dan parameter qrs . Dimana proses perhitungannya menggunakan Persamaan (2.46, 2.48, 2.49, 2.27) sebagai berikut :

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

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

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

$$W_{ij} = \sigma_0^2 \sigma_{ij}^{-1} \quad (2,2)$$

3.3.5 Menghitung Matrik $\dot{N}_y, \ddot{N}_y, \bar{N}_y, \dot{K}_y, \ddot{K}_y$

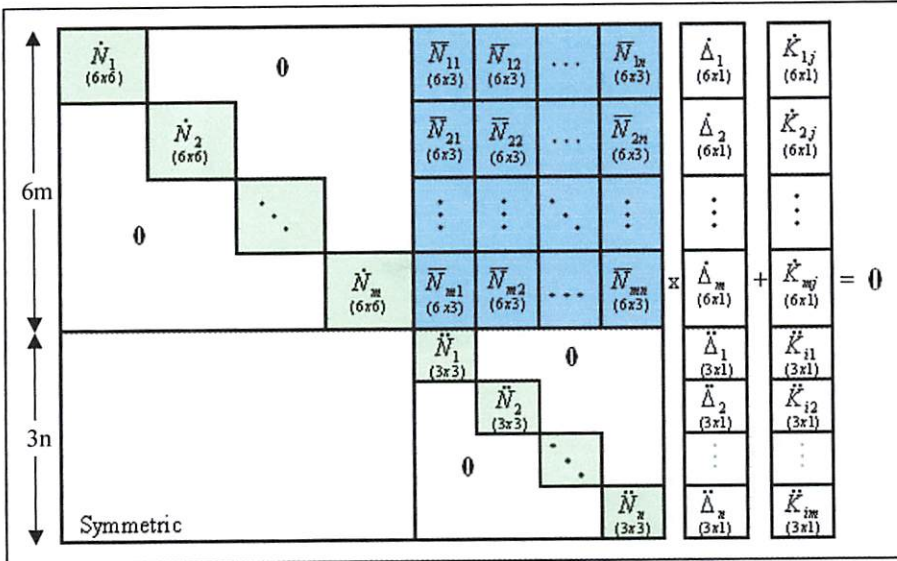
Data yang digunakan dalam proses ini adalah matrik $B_1, B_2,$

$f,$ dan w , yang akan diproses dengan Persamaan (2.57) berikut :

$$\begin{aligned} \dot{N}_y &= \sum_{j=1}^n B_{1ij}^T W_{ij} B_{1ij} & \dot{K}_i &= \sum_{j=1}^n B_{1ij}^T W_{ij} f_{ij} \\ \ddot{N}_y &= \sum_{i=1}^m B_{2ij}^T W_{ij} B_{2ij} & \ddot{K}_j &= \sum_{i=1}^m B_{2ij}^T W_{ij} f_{ij} \\ \bar{N}_y &= B_{1ij}^T W_{ij} B_{2ij} \end{aligned}$$

3.3.6 Menyusun Matrik N dan K

Menyusun Matrik N dan K untuk memudahkan dalam perhitungan. Matrik N dan t dapat disusun dengan menggabungkan data $\dot{N}_y, \ddot{N}_y, \bar{N}_y, \dot{K}_y, \ddot{K}_y$, Struktur persamaannya dapat dinyatakan sebagai berikut (Fraser, 1997) :



Gambar 3.7 Struktur Persamaan Normal

3.3.7 Menghitung matrik Koreksi ($\dot{\Delta}$ dan $\ddot{\Delta}$)

Setelah disusun matrik N dan K , maka nilai koreksi untuk tiap-tiap parameternya dapat dihitung. Dengan menggunakan Persamaan (2.61) sebagai berikut :

$$\begin{bmatrix} \dot{\Delta} \\ \ddot{\Delta} \end{bmatrix} = \begin{bmatrix} B_1^T W B_1 & B_1^T W B_2 \\ B_2^T W B_1 & B_2^T W B_2 \end{bmatrix}^{-1} \begin{bmatrix} B_1^T W f \\ B_2^T W f \end{bmatrix}$$

maka akan diperoleh nilai koreksi $\dot{\Delta}$ yaitu nilai koreksi untuk parameter EO tiap-tiap foto dan $\ddot{\Delta}$ yaitu koreksi untuk koordinat titik-titik objek.

3.3.8 Menghitung Ketelitian Varian-Kovarian

Menghitung nilai ketelitian varian-kovarian dari masing-masing parameter dapat dihitung menggunakan Persamaan (2.85) dan (2.87). Matrik kovarian untuk koordinat 3D sebagai berikut :

$$C x_{1i} = \sigma_o^2 Q x_{1i}$$

Matrik kovarian untuk parameter kamera sebagai berikut :

$$C_{x_{2j}} = \sigma_o^2 Q_{x_{2j}}$$

Jika nilai dari ketelitian varian-kovarian tidak sesuai, maka perlu dilakukan proses iterasi sampai mendapatkan nilai ketelitian 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 ketelitian yang diinginkan telah terpenuhi atau sesuai, maka proses iterasi dihentikan.

3.3.9 Menghitung Nilai Akhir

Menghitung nilai akhir, dimana proses *bundle adjustment* akan mendapatkan nilai parameter EO dan koordinat titik objek yang sudah terkoreksi. Dimana penyelesaiannya menggunakan Persamaan (2.62 dan 2.63) sebagai berikut (Mikhail et al, 2001).

$$O = O^0 + \dot{\Delta} \quad : \text{(Parameter } eksterior \text{ orientation yang terkoreksi)}$$

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

dan menghitung presisi dari tiap titik objek pengukuran, dapat dihitung menggunakan Persamaan (2.93) sebagai berikut :

$$C_{XYZ} = C_{\delta z} = \begin{bmatrix} \sigma_{x_1^2} & \dots & \dots & \dots & \dots & \dots \\ \vdots & \sigma_{y_1^2} & & & & \vdots \\ \vdots & & \sigma_{z_1^2} & & & \vdots \\ \vdots & & & \ddots & & \vdots \\ \vdots & & & & \ddots & \vdots \\ \vdots & \dots & \dots & \dots & \dots & \sigma_{z_n^2} \end{bmatrix}$$

Untuk presisi masing-masing unsur pada koordinat objek dapat dicari menggunakan persamaan berikut :

Untuk unsur X, Y, dan Z nilai presisinya sebagai berikut :

$$\sigma_{x_i} = \sqrt{\sigma_{x_i}}$$

$$\sigma_{y_i} = \sqrt{\sigma_{y_i}}$$

$$\sigma_{z_i} = \sqrt{\sigma_{z_i}}$$

untuk i merupakan titik objek 3D pada posisi titik ke-i.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Perhitungan *Bundle Adjustment* Menggunakan *Microsoft Excel*

Dengan menggunakan aplikasi *Microsoft Excel*, perhitungan *bundle adjustment* dapat menghasilkan nilai-nilai sebagai berikut.

4.1.1 Hasil Perhitungan Parameter Awal

1. Parameter IO (*interior orientation*) yaitu x_o , y_o , dan panjang fokus (f), koordinat foto (x , y) dan ketelitian (σ_x , σ_y).

Tabel 4.1 Parameter IO (*Interior Orientation*)

$x_o = 0$
$y_o = 0$
panjang fokus (mm) = 35

Tabel 4.2 Koordinat Foto dan Presisi setiap Titik pada Foto Satu

titik	Koordinat Foto 1		Ketelitian Titik	
	x (mm)	y (mm)	σ_x (mm)	σ_y (mm)
1	-3.601	1.848	0.00024	0.00024
2	4.738	5.617	0.00016	0.00016
3	-0.577	1.927	0.00020	0.00020
4	0.418	2.340	0.00020	0.00020
5	-0.570	-0.267	0.00017	0.00017
6	0.371	0.051	0.00017	0.00017
7	-3.760	-2.454	0.00014	0.00014
8	4.749	0.193	0.00014	0.00014

Tabel 4.3 Koordinat Foto dan Presisi setiap Titik pada Foto Dua

titik	Koordinat Foto 2		Ketelitian Titik	
	x (mm)	y (mm)	σ_x (mm)	σ_y (mm)
1	-3.909	2.785	0.00029	0.00029
2	3.374	0.637	0.00034	0.00035
3	-0.378	0.611	0.00031	0.00031
4	0.522	0.374	0.00030	0.00030
5	-0.202	-1.341	0.00021	0.00021
6	0.639	-1.527	0.00017	0.00017
7	-3.756	-1.718	0.00022	0.00023
8	3.616	-3.064	0.00014	0.00014

2. Parameter EO (*Exterior Orientation*) pendekatan

Tabel 4.4 Parameter EO Pendekatan Dua Buah Foto

Foto	ω_o	φ_o	κ_o	XL_o	YL_o	ZL_o
1	-0.03671	0.1918	0.05325	0.0089	0.0021	-0.0073
2	82.3534	-73.204	99.1724	-2.9687	-0.773	-2.4736

3. Koordinat titik-titik objek (X_o, Y_o, Z_o)_i pendekatan

Tabel 4.5 Koordinat Titik-titik Objek Pendekatan

Titik	Koordinat		
	Titik Objek Pendekatan		
	X_o	Y_o	Z_o
1	-0.30415	0.15554	-2.95033
2	0.31174	0.36989	-2.30529
3	-0.04365	0.14543	-2.63907
4	0.03041	0.17131	-2.55958
5	-0.04166	-0.01890	-2.57416
6	0.02677	0.00435	-2.50057
7	-0.30286	-0.19695	-2.82164
8	0.29736	0.01317	-2.18867

4.1.2 Perhitungan Matrik Rotasi (R) setiap Foto

Dari dua (2) buah data foto yang telah dihitung nilai omega (ω_o), phi (φ_o) dan kappa (κ_o), maka dapat dihitung matrik rotasi (R) untuk masing- masing foto dengan Persamaan (2.12).

$$\text{Rotasi 1} = \begin{bmatrix} 0.999994 & 0.000927 & -0.003350 \\ -0.000930 & 0.999999 & -0.000638 \\ 0.003349 & 0.000641 & 0.999994 \end{bmatrix}$$

$$\text{Rotasi 2} = \begin{bmatrix} -0.046062 & 0.282609 & 0.958129 \\ -0.285266 & 0.915485 & -0.28375 \\ -0.957341 & -0.286392 & 0.038450 \end{bmatrix}$$

4.1.3 Koreksi Parameter $\dot{\Delta}$ dan $\ddot{\Delta}$ pada Iterasi Terakhir

Untuk iterasi terakhir dari proses *bundle adjustment* dapat diperoleh nilai koreksi tiap-tiap parameter dan koordinat titik objeknya. Dimana $\dot{\Delta}$ merupakan koreksi untuk parameter EO (*Exterior Orientation*) pada dua foto seperti Tabel (4.6), sedangkan $\ddot{\Delta}$ merupakan koreksi tiap koordinat titik-titik objek untuk delapan titik objek seperti Tabel (4.7). Perhitungan koreksi $\dot{\Delta}$ dan $\ddot{\Delta}$ menggunakan Persamaan (2.61). Untuk data hitung yang lebih detail dapat dilihat pada Lampiran (A).

Tabel 4.6 Koreksi Parameter EO Menggunakan *Microsoft Excel*

Foto	Koreksi Parameter EO ($\dot{\Delta}$)					
	$\Delta\omega$	$\Delta\varphi$	$\Delta\kappa$	ΔXL	ΔYL	ΔZL
1	-5.85×10^{-6}	-1.83×10^{-4}	-1.04×10^{-3}	-8.98×10^{-4}	6.46×10^{-5}	2.68×10^{-3}
2	8.62×10^{-4}	8.64×10^{-5}	-3.85×10^{-5}	-1.55×10^{-3}	-5.61×10^{-3}	-1.08×10^{-3}

Tabel 4.7 Koreksi Koordinat Titik-titik Objek Menggunakan *Microsoft Excel*

Titik	Koreksi Koordinat Objek (\ddot{A})		
	ΔX	ΔY	ΔZ
1	-4.72×10^{-4}	6.54×10^{-4}	-6.41×10^{-4}
2	3.64×10^{-4}	2.44×10^{-4}	-1.63×10^{-4}
3	-2.61×10^{-4}	3.55×10^{-4}	-5.52×10^{-4}
4	-1.55×10^{-4}	3.05×10^{-4}	-4.98×10^{-4}
5	-4.58×10^{-4}	1.47×10^{-4}	-6.95×10^{-4}
6	-3.58×10^{-4}	9.72×10^{-5}	-6.52×10^{-4}
7	-9.32×10^{-4}	2.30×10^{-4}	-9.13×10^{-4}
8	-3.51×10^{-6}	-1.90×10^{-4}	-5.32×10^{-4}

4.1.4 Parameter EO Update dan Koordinat Update

Perhitungan parameter EO update atau EO terkoreksi dapat dilakukan menggunakan Persamaan (2.62) sebagai berikut :

Tabel 4.8 Parameter EO Update Menggunakan *Microsoft Excel*

Parameter EO Update						
Foto	$\hat{\omega}$	$\hat{\varphi}$	$\hat{\kappa}$	\hat{X}_L	\hat{Y}_L	\hat{Z}_L
1	-0.0368	0.1919	0.0546	0.0083	0.0017	-0.0075
2	82.3571	-73.2044	99.1786	-2.9695	-0.7747	-2.4770

Perhitungan koordinat titik-titik objek update atau titik-titik terkoreksi dapat dilakukan menggunakan Persamaan (2.63)

Tabel 4.9 Titik-titik Objek Update Menggunakan *Microsoft Excel*

Titik	Koordinat Titik-titik Objek Update		
	\hat{X}	\hat{Y}	\hat{Z}
1	-0.30413	0.15535	-2.95277
2	0.31216	0.36909	-2.30718
3	-0.04347	0.14498	-2.64125
4	0.03066	0.17080	-2.56169
5	-0.04168	-0.01952	-2.57646
6	0.02682	0.00369	-2.50287
7	-0.30337	-0.19764	-2.82425
8	0.29750	0.01230	-2.19093

4.1.5 Presisi Koordinat Titik-titik Objek

Perhitungan presisi koordinat titik-titik objek tiap sumbu koordinatnya dapat dihitung menggunakan Persamaan (2.93).

Tabel 4.10 Presisi Koordinat Titik-titik Objek Menggunakan *Microsoft Excel*

Titik	σ_x	σ_y	σ_z
1	1.930×10^{-5}	1.570×10^{-5}	2.340×10^{-5}
2	1.160×10^{-5}	1.130×10^{-5}	3.350×10^{-5}
3	1.480×10^{-5}	1.340×10^{-5}	2.700×10^{-5}
4	1.430×10^{-5}	1.310×10^{-5}	2.680×10^{-5}
5	1.230×10^{-5}	1.040×10^{-5}	1.840×10^{-5}
6	1.190×10^{-5}	9.570×10^{-6}	1.480×10^{-5}
7	1.160×10^{-5}	9.850×10^{-6}	1.730×10^{-5}
8	8.670×10^{-6}	7.260×10^{-6}	1.300×10^{-5}

4.2 Source Code Perhitungan *Bundle Adjustment* Menggunakan *Console Application* pada Program C#

4.2.1 Pembuatan *Class* Utama Program

```
namespace BAcs
{
    public class BA
    {
        private const string FILE_NAME_POINT = "8pts.txt";
        public static void Main(string[] args)
        {
            Bundle bundle = new Bundle(FILE_NAME_POINT, );
            Console.WriteLine(bundle.ToString());
        }
    }
}
```

Class ini berisikan input file (estensi .txt) data parameter, dimana file awalnya berisikan parameter-parameter yang sudah diproses pada proses sebelumnya seperti pada Gambar (4.1).

- Baris pertama : data jumlah foto, panjang fokus (f), x_o dan y_o .
- Baris kedua dan ketiga : *id* foto, *omega* (ω_o), *phi* (ϕ_o), *kappa* (κ_o), XL_o , YL_o , dan ZL_o untuk foto pertama. dan kedua.

- Baris keempat : jumlah titik yang akan diproses.
- Baris kelima sampai kedua belas : id titik, koordinat X_b , Y_b , Z_b , koordinat foto x_{1i} , y_{1i} foto pertama, x_{2i} , dan y_{2i} foto kedua.
- Baris ketigabelas sampai baris kedua puluh : id titik, ketelitian σx_i , σy_i foto pertama, dan ketelitian σx_i , σy_i foto kedua

File	Edit	Format	View	Help					
2		24	0	0					
1		0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2		-60.721	52.883	95.495	0.648	0.456		-0.610	
8									
1		0.029	0.046	-0.529	1.329	2.077		-9.043	5.066
2		0.111	0.102	-0.605	4.423	4.034		-6.633	3.750
3		0.148	0.128	-0.640	5.554	4.787		-5.270	3.050
4		0.218	0.179	-0.712	7.367	6.049		-2.019	1.444
5		0.126	0.221	-0.780	3.889	6.795		1.680	3.297
6		0.078	0.242	-0.814	2.291	7.122		3.219	4.110
7		-0.016	0.282	-0.880	-0.427	7.683		5.694	5.428
8		-0.101	0.222	-0.798	-3.048	6.685		2.183	6.499
1		0.025	0.025	0.040	0.040				
2		0.031	0.030	0.030	0.030				
3		0.027	0.027	0.028	0.027				
4		0.039	0.040	0.026	0.026				
5		0.036	0.036	0.028	0.028				
6		0.045	0.045	0.031	0.031				
7		0.048	0.047	0.037	0.036				
8		0.046	0.046	0.043	0.042				

Gambar 4.1 Susunan Data awal pada Notepad

4.2.2 Class Photo Pair EO

Dalam *Class* Photo Pair EO ini, menjelaskan cara pembuatan id supaya dalam proses di *class* proses lebih mudah.

```
namespace BAcs
{
    public struct PhotoPairCams
    {
        string _labelc; double _omega; double _phi;
        double _kappa; double _xL; double _yL;
        double _zL;
        public PhotoPairCams(string lblc, double om,
            double pi, double kp, double xL, double
            yL, double zL)
        {
            _labelc = lblc; _omega = om; _phi = pi;
            _kappa = kp; _xL = xL; _yL = yL;
            _zL = zL;
        }
        {
            string cams = String.Format("{0} {1} {2} {3}
                {4} {5} {6}", _labelc, _omega, _phi,
                _kappa, _xL, _yL, _zL);
            return cams; }
    }
}
```

4.2.3 Class Photo Pair Point

```
namespace BAcS
{
    public struct PhotoPairPoint
    {
        string _labelp; double _Xc; double _Yc;
        double _Zc; double _x1; double _y1;
        double _x2; double _y2;
        public PhotoPairPoint(string lblp, double xc,
            double yc, double zc, double x1, double
            y1, double x2, double y2)
        { _labelp = lblp; _Xc = xc; _Yc = yc;
            _Zc = zc; _x1 = x1; _x2 = x2; _y2 = y2; }
        public override string ToString()
        { string points = String.Format("{0} {1} {2}
            {3} {4} {5} {6} {7}", _labelp, _Xc,
            _Yc, _Zc, _x1, _y1, _x2, _y2);
            return points; }
    }
}
```

4.2.4 Class Proses Bundle Adjustment

1. Class Membaca File Data pada Notepad

```
private void ReadFile(ref string file)
{
    string[] readText = File.ReadAllLines(file);
    System.Diagnostics.Trace.WriteLine("readtext: " +
        readText.Count());
    for (int i = 0; i < readText.Count(); i++)
    { if (i == 0)
        { if (firstlineText.Length == 4)
            { ImageNumber = int.Parse(firstlineText[0]);
                Focal = Double.Parse(firstlineText[1]);
                Xo = Double.Parse(firstlineText[2]);
                Yo = Double.Parse(firstlineText[3]); }
            } else
        { if (textc.Length == 7)
            { PhotoPairCams pairc = new PhotoPairCams();
                PhotoPairListCams.Add(pairc); }
            } int im = _imageNumber + 1; if (i == im)
        { if (pointNumberline.Length == 1)
            { PointNumber = int.Parse(pointNumberline[0]); }
            } else //from the second line and beyond
        { if (textp.Length == 8)
            { PhotoPairPoint pairp = new PhotoPairPoint();
                PhotoPairListPoint.Add(pairp); }
            } int ip = _imageNumber + 1 + _pointNumber + 1;
            if (i == ip)
        { if (residuNumberline.Length == 1)
            { ResiduNumber = int.Parse(residuNumberline[0]); }
            } else
        { if (textr.Length == 5)
            { PhotoPairResidu pairr =new PhotoPairResidu(); }
            }
        }
    }
}
```

2. Algoritma Fungsi Rotasi (R)

```
private double[,] ProjectRotation(ref double Omega,
    ref double Phi, ref double Kappa)
{
    double[,] M = new double[3, 3];
    double a = Math.PI / 180;
    M[0, 0]; M[0, 1]; M[0, 2];
    M[1, 0]; M[1, 1]; M[1, 2];
    M[2, 0]; M[2, 1]; M[2, 2];
    return M;
}
```

3. Algoritma Fungsi q, r, s

```
private double[] ProjectUVW(ref double Omega, ref
    double Phi, ref double Kappa, ref double XL, ref
    double YL, ref double ZL, ref double Xc, ref double
    Yc, ref double Zc)
{
    double[] rsq = new double[3];
    double[,] M_rot = new double[3, 3];
    M_rot = ProjectRotation(ref Omega, ref Phi, ref
        Kappa);
    rsq[0]; rsq[1]; rsq[2];
    return rsq;
}
```

4. Algoritma Matrik B

```
private double[,] ProjectA1(ref double Omega, ref
    double Phi, ref double Kappa, ref double XL, ref
    double YL, ref double ZL, ref double Xc, ref double
    Yc, ref double Zc, ref double foc)
{
    double a = Math.PI / 180;
    double[,] B = new double[2, 6];
    double[,] M_rot = new double[3, 3];
    M_rot = ProjectRotation(ref Omega, ref Phi, ref
        Kappa);
    double[] Rsq = new double[3];
    Rsq = ProjectUVW(ref Omega, ref Phi, ref Kappa, ref
        XL, ref YL, ref ZL, ref Xc, ref Yc, ref Zc);
    B[0,0];B[0,1]; B[0,2]; B[0,3]; B[0,4]; B[0,5]
    B[1,0];B[1,1]; B[1,2]; B[1,3]; B[1,4]; B[1,5]
    return B;
}
```

Source code yang lebih detail dapat dilihat pada Lampiran (B).

4.3 Hasil Perhitungan *Bundle Adjustment* menggunakan *Console Application* pada Bahasa Pemrograman C#

Pada perhitungan *bundle adjustment* menggunakan *console application* pada bahasa pemrograman C# menampilkan hasil untuk iterasi terakhir yang diperoleh sebagai berikut.

4.3.1 Perhitungan Matrik Rotasi (*R*) setiap Foto

Dengan menggunakan *console application* pada bahasa pemrograman C# nilai rotasi setiap foto dapat diperoleh sebagai berikut :

$$\text{Rotasi 1} = \begin{bmatrix} 0.9999939 & 0.0009274 & -0.00335 \\ -0.000929 & 0.9999993 & -0.000638 \\ 0.0033492 & 0.0006407 & 0.9999994 \end{bmatrix}$$

$$\text{Rotasi 2} = \begin{bmatrix} -0.046062 & 0.2826080 & 0.958128 \\ -0.285266 & 0.9154840 & -0.283744 \\ -0.957341 & -0.286392 & 0.0384495 \end{bmatrix}$$

4.3.2 Perhitungan Koreksi $\dot{\Delta}$ dan $\ddot{\Delta}$

Nilai masing-masing koreksi $\dot{\Delta}$ dan $\ddot{\Delta}$ yang diperoleh dari perhitungan menggunakan *console application* dalam bahasa pemrograman C# sebagai berikut :

Tabel 4.11 Koreksi Parameter EO Menggunakan *Console Application* Pemrograman C#

Foto	Koreksi Parameter EO ($\dot{\Delta}$)					
	$\Delta\omega$	$\Delta\phi$	$\Delta\kappa$	ΔXL	ΔYL	ΔZL
1	6.02×10^{-4}	-1.05×10^{-3}	1.54×10^{-3}	-2.30×10^{-3}	-1.16×10^{-3}	-1.80×10^{-3}
2	-5.71×10^{-4}	-1.54×10^{-4}	8.56×10^{-4}	-1.65×10^{-3}	2.28×10^{-4}	2.91×10^{-17}

Tabel 4.12 Koreksi Koordinat Titik Objek Menggunakan *Console Application* Pemrograman C#

Titik	Koreksi Koordinat Titik Objek ($\ddot{\Delta}$)		
	ΔX	ΔY	ΔZ
1	9.48×10^{-4}	-8.39×10^{-5}	-7.74×10^{-4}
2	-4.91×10^{-4}	1.87×10^{-4}	-2.77×10^{-4}
3	5.68×10^{-4}	1.02×10^{-4}	-2.13×10^{-4}
4	4.00×10^{-4}	1.47×10^{-4}	-1.40×10^{-4}
5	7.35×10^{-4}	1.28×10^{-4}	3.88×10^{-5}
6	5.73×10^{-4}	1.93×10^{-4}	4.81×10^{-5}
7	1.33×10^{-3}	-1.88×10^{-4}	-2.14×10^{-4}
8	-1.13×10^{-4}	4.41×10^{-4}	1.90×10^{-4}

4.3.3 Hasil Parameter EO *Update* dan Koordinat Titik Objek *Update*

Dengan menggunakan *console application* dalam bahasa program C#, adapun hasil perhitungan parameter EO *update* dan koordinat titik objek *update* sebagai berikut :

Tabel 4.13 Parameter EO *Update* Menggunakan *Console Application* Pemrograman C#

Foto	$\hat{\omega}$	$\hat{\varphi}$	$\hat{\kappa}$	$\hat{X}L$	$\hat{Y}L$	$\hat{Z}L$
1	-0.0361	0.1908	0.0548	0.0067	0.0010	-0.0092
2	82.3529	-73.2044	99.1733	-2.9704	-0.7734	-2.4737

Tabel 4.14 Koordinat Titik Objek *Update* Menggunakan *Console Application* Pemrograman C#

Titik	\hat{X}	\hat{Y}	\hat{Z}
1	-0.30320	0.15545	-2.95111
2	0.31125	0.37007	-2.30557
3	-0.04308	0.14553	-2.63928
4	0.03081	0.17146	-2.55972
5	-0.04092	-0.01877	-2.57412
6	0.02734	0.00454	-2.50052
7	-0.30152	-0.19714	-2.82186
8	0.29725	0.01361	-2.18848

4.3.4 Presisi Koordinat Titik-titik Objek

Presisi tiap sumbu koordinat pada masing-masing titik diperoleh sebagai berikut :

Tabel 4.15 Presisi Koordinat Titik Objek Menggunakan *Console Application* Pemograman C#

Titik	σ_x	σ_y	σ_z
1	1.92576×10^{-5}	1.57373×10^{-5}	2.34201×10^{-5}
2	1.15829×10^{-5}	1.13385×10^{-5}	3.35382×10^{-5}
3	1.47707×10^{-5}	1.33721×10^{-5}	2.70241×10^{-5}
4	1.43445×10^{-5}	1.31105×10^{-5}	2.68299×10^{-5}
5	1.22889×10^{-5}	1.04356×10^{-5}	1.83936×10^{-5}
6	1.18886×10^{-5}	9.57446×10^{-6}	1.48004×10^{-5}
7	1.15862×10^{-5}	9.85003×10^{-6}	1.73434×10^{-5}
8	8.66908×10^{-6}	7.25695×10^{-6}	1.30006×10^{-5}

4.4 *Source Code* Perhitungan *Bundle Adjustment* Menggunakan *MATLAB 2008*

Membuat sebuah program untuk menjalankan proses *bundle adjustment* dengan bantuan *MATLAB*, terlebih dahulu harus menyiapkan serangkaian *source code* yang sesuai. Pada subbab ini akan ditampilkan beberapa *source code* utama yang dibutuhkan dalam proses *bundle adjustment* menggunakan *MATLAB*. *source code* yang lebih detail dapat dilihat pada Lampiran C.

4.4.1 Fungsi Untuk Matrik Rotasi

Source code untuk matrik rotasi (*R*) dapat dijalankan dengan membangun *source code* berikut ini :

```
function m=projectROTM(om,ph,kp)
m11; m12; m13;
m21; m22; m23;
m31; m32; m33;
m=[m11 m12 m13;m21 m22 m23;m31 m32 m33];
```


4.4.2 Fungsi Untuk Matrik B

Source code untuk Matrik B sebagai berikut :

```
function B=projectAR1(om,ph,kp,XL,YL,ZL,foc,X,Y,Z)
m=projectROTM(om,ph,kp);
uvw=projectUVW(om,ph,kp,XL,YL,ZL,X,Y,Z);
B(1); B(2); B(3); B(4); B(5); B(6);
B(7); B(8); B(9); B(10); B(11); B(12);
```

4.4.3 Fungsi Untuk Elemen qrs

Source code untuk menghitung elemen qrs sebagai berikut :

```
function uvw=projectUVW(om,ph,kp,XL,YL,ZL,X,Y,Z)
m=projectROTM(om,ph,kp);
uvw(1); uvw(2); uvw(3);
```

4.4.4 Fungsi Utama Program

Source code yang menampung keseluruhan *source code* lainnya dan menjadi media untuk menjalankan program disebut dengan fungsi utama. Berikut ini adalah *source code* yang dipergunakan untuk membangun fungsi utama dalam program ini :

```
camsfname='../demo/utara1_2.txt';
ptsfname='../demo/utara1_8.txt';
residufname='../demo/utara1_8re.txt';
xo=0; yo=0; foc=35;
for iter = 1:10
    iterasi =iter
    %Struktur N1=AT1PA1 (hitung nilai parameter EO)
    for i=1:ncams;
    for j=1:npts;
    N(1:12,1:(3*npts+6*ncams))=[reshape(Nki,12,12)
    reshape(Nkij',12,(3*npts))];

    N(13:(3*npts+6*ncams),1:(3*npts+6*ncams))=[reshape(Nki
    j,(3*npts),12) reshape(Nkj,(3*npts),(3*npts))];
    %membentuk matrik K
    C(1:12,1:1)=Cki;
    C(13:(3*npts+6*ncams),1:1)=Ckj;
    %hitung Matrik koreksi
    invP=pinv(N);
    xx=pinv(N)*C
    %hitung parameter EO update
    for i=1:ncams;
    for j=1:npts;
    sd=sqrt(trace(inv(Nkj))/(3*npts))
```



```

utara1_8.txt - Notepad
File Edit Format View Help
# X Y Z nframes frame x0 y0 frame x1 y1 ...
-0.076805 0.039414 -0.746605 2 0 -3.600514 1.847649 1 -3.908922 2.785188
0.0788770 0.093509 -0.582464 2 0 4.7381540 5.617089 1 3.374264 0.637134
-0.010994 0.036728 -0.667369 2 0 -0.576704 1.926520 1 -0.37780 0.611463
0.0077200 0.043258 -0.647140 2 0 0.4175890 2.340001 1 0.522467 0.374163
-0.010598 -0.00496 -0.650871 2 0 -0.569502 -0.26666 1 -0.20242 -1.340695
0.0067290 0.000917 -0.632160 2 0 0.3711090 0.050585 1 0.638662 -1.526967
-0.076698 -0.05006 -0.713884 2 0 -3.759614 -2.45396 1 -3.75609 -1.718058
0.0750260 0.003045 -0.552851 2 0 4.7493220 0.192740 1 3.616448 -3.063555

```

Gambar 4.3 Susunan Data input titik koordinat awal pada Notepad

```

utara1_8re.txt - Notepad
File Edit Format View Help
# nframe frame xres0 yres0 frame xres1 yres1 ...
2 0 2.36E-04 2.36E-04 1 2.893E-04 2.890E-04
2 0 1.64E-04 1.61E-04 1 3.441E-04 3.472E-04
2 0 1.99E-04 2.00E-04 1 3.081E-04 3.072E-04
2 0 1.99E-04 2.00E-04 1 2.981E-04 3.032E-04
2 0 1.69E-04 1.69E-04 1 2.130E-04 2.131E-04
2 0 1.69E-04 1.70E-04 1 1.676E-04 1.676E-04
2 0 1.45E-04 1.45E-04 1 2.240E-04 2.278E-04
2 0 1.39E-04 1.37E-04 1 1.380E-04 1.381E-04

```

Gambar 4.4 Susunan Data Input Ketelitian Titik untuk Koordinat Foto Awal pada Notepad

File data input parameter EO awal disusun seperti pada Gambar (4.2). File data input titik koordinat foto disusun seperti pada Gambar (4.3). Sedangkan untuk file data input ketelitian titik untuk koordinat foto disusun seperti pada Gambar (4.4).

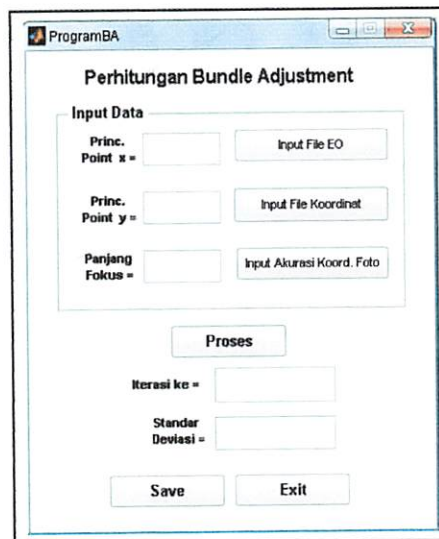
Penamaan untuk masing-masing file input dapat bervariasi sesuai dengan keinginan peneliti. Namun nama-nama tersebut harus diinputkan dalam *source code* fungsi utama program. Dalam penelitian ini, peneliti menggunakan nama `utara1_2` untuk data parameter EO awal, `utara1_8` untuk data koordinat titik, dan `utara1_8re` untuk data ketelitian titik untuk koordinat foto.

4.5 Prosedur Pengoperasian Program MATLAB dalam Menghitung *Bundle Adjustment*

Sesuai dengan *source code* yang telah disusun sebelumnya, pengoperasian program MATLAB untuk menghitung *bundle adjustment* melalui beberapa tahapan sebagai berikut :

4.5.1 Tampilan Awal Program

Tampilan awal saat pengoperasian program ini adalah seperti yang ditunjukkan pada Gambar (4.5). Pada form tampilan program tersebut, terdapat beberapa bagian yang harus dioperasikan pada tahapan berikutnya yaitu tahap input data, pemrosesan data dan penyimpanan data.



Gambar 4.5 Tampilan Awal Program

4.5.2 Input Data

Pada proses ini kelompok data pertama yang diinput adalah *principle point x*, *principle point y*, dan panjang fokus dengan nilai masing-masing 0, 0, dan 35 seperti ditunjukkan pada Gambar (4.6).

Gambar 4.6 Input Data Principle Point dan Panjang Fokus

Data selanjutnya yang harus diinput adalah parameter EO, titik-titik koordinat dan akurasi koordinat foto. Ketiga data ini diinput dengan menekan tombol input data masing-masing dan memilih file yang telah disiapkan sebelumnya dalam bentuk notepad.

4.5.3 Proses Perhitungan *Bundle Adjustment*

Setelah seluruh data diinputkan, langkah selanjutnya adalah menjalankan proses perhitungan *bundle adjustment* dengan menekan tombol proses. Maka akan menghasilkan urutan iterasi dan nilai standar deviasi seperti ditunjukkan Gambar (4.7).

Gambar 4.7 Hasil Urutan Iterasi dan Nilai Standar Deviasi

4.5.4 Penyimpanan Hasil Perhitungan *Bundle Adjustment*

Langkah terakhir dalam mengoperasikan perhitungan *bundle adjustment* dalam MATLAB adalah menyimpan nilai output yang dihasilkan oleh perhitungan pada tahap sebelumnya. Langkah ini dilakukan dengan menekan tombol *save* pada tampilan program.

4.6 Hasil Perhitungan *Bundle Adjustment* Menggunakan *MATLAB 2008*

Pada perhitungan *bundle adjustment* menggunakan bahasa pemrograman *MATLAB 2008* menampilkan hasil untuk iterasi terakhir yang diperoleh sebagai berikut.

4.6.1 Perhitungan Matrik Rotasi (*R*) tiap Foto

Implementasi dari Persamaan (2.12) dalam program *MATLAB* (*source code* untuk matrik rotasi) menghasilkan matrik rotasi untuk foto 1 dan foto 2 sebagai berikut :

$$\text{Rotasi 1} = \begin{bmatrix} 1.0000 & 0.0025 & -0.0025 \\ -0.003 & 1.0000 & -0.0002 \\ 0.0025 & 0.0002 & 1.0000 \end{bmatrix}$$

$$\text{Rotasi 2} = \begin{bmatrix} -0.0461 & 0.2841 & 0.9577 \\ -0.2852 & 0.9151 & -0.285 \\ -0.9574 & -0.286 & 0.0388 \end{bmatrix}$$

4.6.2 Perhitungan Koreksi $\dot{\Delta}$ dan $\ddot{\Delta}$

Implementasi dari Persamaan (2.61) dalam program *MATLAB* menghasilkan nilai untuk koreksi parameter EO ($\dot{\Delta}$) pada Tabel (4.16) dan koreksi koordinat titik objek ($\ddot{\Delta}$) pada Tabel (4.17) sebagai berikut.

Tabel 4.16 Koreksi Parameter *Exterior Orientation* Menggunakan *MATLAB*

Foto	Koreksi Parameter <i>Exterior Orientation</i>					
	$\Delta\omega$	$\Delta\phi$	$\Delta\kappa$	ΔXL	ΔYL	ΔZL
1	6.04×10^{-4}	-8.60×10^{-4}	8.91×10^{-4}	-3.40×10^{-6}	-3.44×10^{-5}	6.00×10^{-7}
2	-7.14×10^{-4}	-5.45×10^{-4}	3.97×10^{-4}	-1.69×10^{-5}	6.23×10^{-5}	-1.33×10^{-5}

Tabel 4.17 Koreksi Koordinat Titik-titik Objek Menggunakan MATLAB

Titik	Koreksi Koordinat Titik Objek		
	ΔX	ΔY	ΔZ
1	4.80E-06	7.90E-06	2.00E-06
2	8.60E-06	-1.37E-05	5.50E-06
3	4.10E-06	-1.70E-06	2.30E-06
4	4.60E-06	-4.30E-06	2.70E-06
5	3.00E-07	-2.60E-06	3.00E-07
6	7.00E-07	-5.00E-06	7.00E-07
7	-3.30E-06	6.30E-06	-2.20E-06
8	4.00E-07	-1.48E-05	1.30E-06

4.6.3 Hasil Parameter *Exterior Orientation Update* dan Koordinat Titik Objek *Update*

Implementasi dari persamaan (2.62 dan 2.63) dalam program MATLAB menghasilkan nilai *update* untuk parameter *Exterior Orientation* dan koordinat titik-titik objek *update* sebagai berikut :

Tabel 4.18 Parameter *Exterior Orientation Update* Menggunakan MATLAB

Foto	$\hat{\omega}$	$\hat{\phi}$	$\hat{\kappa}$	\hat{X}_L	\hat{Y}_L	\hat{Z}_L
1	-0.0311	0.1829	0.0649	0.0068	0.0006	-0.0090
2	82.3446	-73.2081	99.1765	-2.9706	-0.7722	-2.4733

Tabel 4.19 Koordinat Titik-titik Objek *Update* Menggunakan MATLAB

Titik	\hat{X}	\hat{Y}	\hat{Z}
1	-0.30316	0.15551	-2.95075
2	0.31140	0.36978	-2.30525
3	-0.04303	0.14544	-2.63896
4	0.03087	0.17132	-2.55940
5	-0.04095	-0.01887	-2.57381
6	0.02733	0.00441	-2.50022
7	-0.30164	-0.19709	-2.82154
8	0.29724	0.01333	-2.18820

4.6.4 Presisi Koordinat Titik-titik Objek

Implementasi dari persamaan (2.93) dalam program MATLAB menghasilkan nilai presisi titik-titik koordinat objek pada Tabel (4.20) sebagai berikut.

Tabel 4.20 Presisi Titik-titik Koordinat Menggunakan MATLAB

Titik	σ_x	σ_y	σ_z
1	0.1925×10^{-4}	0.1574×10^{-4}	0.2344×10^{-4}
2	0.1159×10^{-4}	0.1134×10^{-4}	0.3355×10^{-4}
3	0.1476×10^{-4}	0.1337×10^{-4}	0.2704×10^{-4}
4	0.1434×10^{-4}	0.1311×10^{-4}	0.2685×10^{-4}
5	0.1228×10^{-4}	0.1043×10^{-4}	0.1841×10^{-4}
6	0.1188×10^{-4}	0.0957×10^{-4}	0.1481×10^{-4}
7	0.1158×10^{-4}	0.0985×10^{-4}	0.1736×10^{-4}
8	0.0867×10^{-4}	0.0725×10^{-4}	0.1300×10^{-4}

4.7 Analisis Hasil Penelitian *Bundle Adjustment*

Bundle adjustment merupakan salah satu bagian dari serangkaian proses *close range photogrammetry* untuk analisis deformasi. *Bundle adjustment* dapat dikatakan sebagai salah satu tahap penting yang berperan dalam mengevaluasi nilai-nilai yang telah diperoleh dalam tahap sebelumnya dalam *close range photogrammetry*, yaitu menilai keakurasian presisi dari parameter *exterior orientation* ($\omega_o, \varphi_o, \kappa_o, XL_o, YL_o, ZL_o$) dan koordinat *object space* (X_o, Y_o, Z_o), yang dihasilkan pada tahap sebelumnya yaitu tahap *relative orientation, resection* dan *intersection*. Hasil parameter akhir dari tahap *bundle adjustment* merupakan hasil atau nilai terakurat yang dapat digunakan untuk proses selanjutnya dalam *close range photogrammetry*.

Dari hasil perhitungan *bundle adjustment close range photogrammetry* pada dua buah foto dengan delapan titik dari *fly over* Arjosari dapat dianalisis beberapa hal sebagai berikut :

4.7.1 Analisis Hasil Parameter EO Update

4.6.1.1 Hasil Parameter EO Update Menggunakan *Microsoft Excel*

Setelah dihitung menggunakan *Microsoft Excel* dapat dilihat nilai selisih parameter EO awal dengan parameter EO update pada Tabel (4.21).

Tabel 4.21 Hasil Selisih Parameter EO Awal dengan Parameter EO Update Menggunakan *Microsoft Excel*

Parameter EO Awal

Foto	ω_o	φ_o	κ_o	XL_o	YL_o	ZL_o
1	-0.03671	0.1918	0.05325	0.0089	0.0021	-0.007
2	82.3534	-73.204	99.1724	-2.9687	-0.773	-2.473

Parameter EO Update

Foto	$\hat{\omega}$	$\hat{\varphi}$	$\hat{\kappa}$	\hat{XL}	\hat{YL}	\hat{ZL}
1	-0.0368	0.1919	0.0546	0.0083	0.0017	-0.0075
2	82.3571	-73.2044	99.1786	-2.9695	-0.7747	-2.4770

Selisih Parameter EO awal dengan Parameter EO Update

Foto	$\Delta\omega$	$\Delta\varphi$	$\Delta\kappa$	ΔXL	ΔYL	ΔZL
1	0.00013	0.00001	-0.00131	0.00069	0.00040	0.00011
2	-0.00364	0.00014	-0.00620	0.00079	0.00099	0.00333

4.6.1.2 Hasil Parameter EO Update Menggunakan *Console Application* pada Bahasa Pemrograman C#

Setelah dihitung menggunakan *Console Application* pada bahasa pemrograman C# dapat dilihat nilai selisih parameter EO awal dengan parameter EO update pada Tabel (4.22).

Tabel 4.22 Hasil Selisih Parameter EO Awal dengan Parameter EO Update Menggunakan Console Application pada Bahasa Pemrograman C#

Parameter EO Awal

Foto	ω	φ	κ	X_L	Y_L	Z_L
1	-0.03671	0.1918	0.0532	0.0089	0.0021	-0.007
2	82.3534	-73.204	99.172	-2.968	-0.773	-2.473

Parameter EO Update

Foto	$\hat{\omega}$	$\hat{\varphi}$	$\hat{\kappa}$	\hat{X}_L	\hat{Y}_L	\hat{Z}_L
1	-0.0361	0.1908	0.0548	0.0067	0.0010	-0.0092
2	82.3529	-73.2044	99.1733	-2.9704	-0.7734	-2.4737

Selisih Parameter EO Awal dengan Parameter EO Update

Foto	$\Delta\omega$	$\Delta\varphi$	$\Delta\kappa$	ΔX_L	ΔY_L	ΔZ_L
1	-0.00060	0.00105	-0.00154	0.00230	0.00116	0.00180
2	0.00057	0.00015	-0.00086	0.00165	-0.00023	0.00000

4.6.1.3 Hasil Parameter EO Update Menggunakan MATLAB 2008

Setelah dihitung menggunakan MATLAB 2008 dapat dilihat nilai selisih parameter EO awal dengan parameter EO update pada Tabel (4.23).

Tabel 4.23 Hasil Selisih Parameter EO Awal dengan Parameter EO Update Menggunakan MATLAB 2008

Parameter EO Awal

Foto	ω	φ	κ	X_L	Y_L	Z_L
1	-0.03671	0.1918	0.05325	0.0089	0.0021	-0.007
2	82.3534	-73.204	99.1724	-2.9687	-0.773	-2.473

Parameter EO Update

Foto	$\hat{\omega}$	$\hat{\varphi}$	$\hat{\kappa}$	\hat{X}_L	\hat{Y}_L	\hat{Z}_L
1	-0.031	0.1829	0.0649	0.0068	0.0006	-0.0090
2	82.3446	-73.2081	99.1765	-2.9706	-0.7722	-2.4733

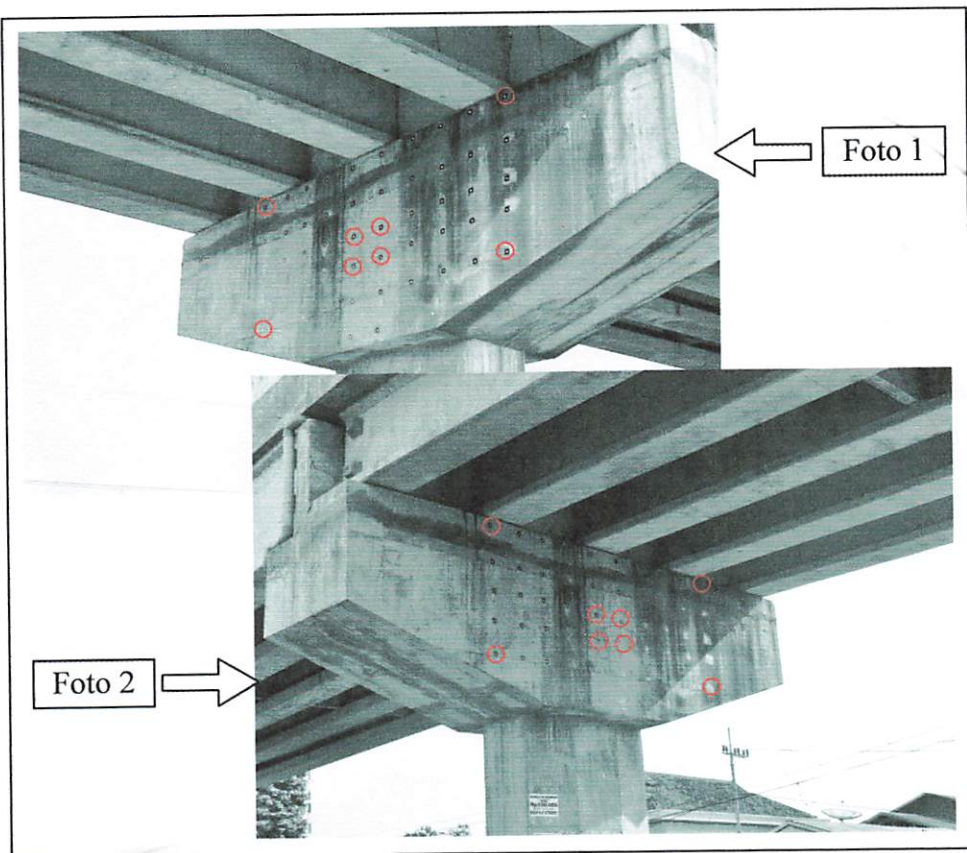
Selisih Parameter EO awal dengan Parameter EO Update

Foto	$\Delta\omega$	$\Delta\varphi$	$\Delta\kappa$	ΔX_L	ΔY_L	ΔZ_L
1	-0.00560	0.00898	-0.01168	0.00218	0.00150	0.00158
2	0.00888	0.00385	-0.00408	0.00192	-0.0015	-0.00039

Dari selisih yang diperoleh baik melalui perhitungan *Microsoft Excel*, *Console Application* pada bahasa pemrograman C#, maupun *MATLAB 2008*, dapat dianalisis bahwa nilai parameter EO sebelumnya sudah baik (akurat). Hal ini ditunjukkan oleh nilai selisih yang terjadi antara EO awal dengan EO *update* dari ketiga metode penghitungan menunjukkan rata-rata 0.0002.

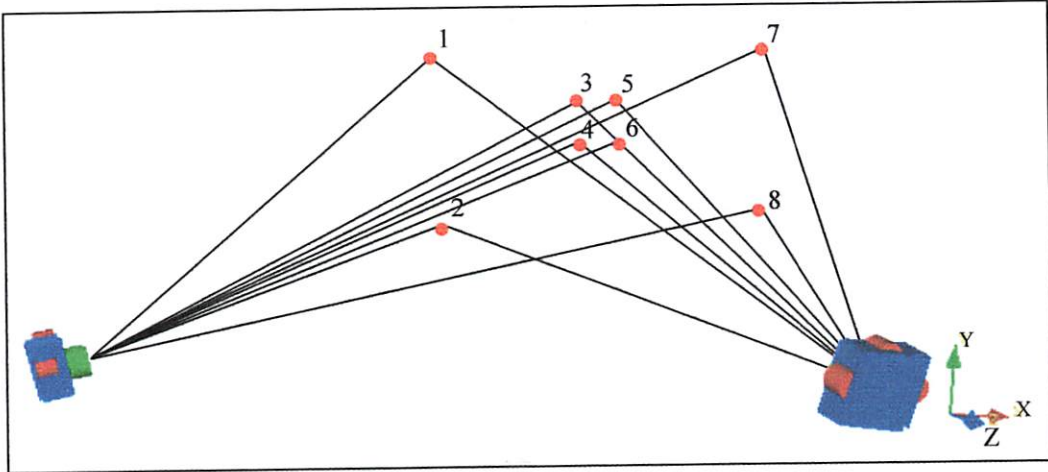
4.7.2 Analisis Hasil Koordinat Titik-titik Objek *Update*

Dari sebaran beberapa titik pada dua foto dengan objek yang sama (seperti Gambar 4.8), maka letak atau posisi koordinat titik-titik objek (delapan titik) hasil perhitungan *bundle adjustment*, dapat dihasilkan seperti Gambar (4.9).



Gambar 4.8 Sebaran Titik-titik Pada Foto

Gambar (4.9) menunjukkan posisi titik objek secara grafis dan posisi dua buah kamera. Untuk koordinat titik-titik objeknya dapat dilihat pada Tabel (4.19).



Gambar 4.9 Posisi Titik-titik Objek dan Posisi Kamera

4.7.3 Analisis Hasil Keakurasian Koordinat Titik-titik Objek

Dari perhitungan yang dilakukan pada dua buah foto dengan delapan titik, dapat dianalisis keakurasian koordinat titik-titik objeknya. Nilai rata-rata varian atau akurasi untuk setiap titiknya dihitung menggunakan Persamaan (2.58) sebagai berikut:

4.6.3.1 Hasil Keakurasian Koordinat Titik-titik Objek Menggunakan *Microsoft Excel*

Setelah dihitung menggunakan *Microsoft Excel* dapat dihasilkan nilai akurasi untuk masing-masing koordinat titik objek pada Tabel (4.24).

Tabel 4.24 Ketelitian Koordinat Titik-titik Objek Menggunakan *Microsoft Excel*

Titik	σ_x	σ_y	σ_z
1	1.930×10^{-5}	1.570×10^{-5}	2.340×10^{-5}
2	1.160×10^{-5}	1.130×10^{-5}	3.350×10^{-5}
3	1.480×10^{-5}	1.340×10^{-5}	2.700×10^{-5}
4	1.430×10^{-5}	1.310×10^{-5}	2.680×10^{-5}
5	1.230×10^{-5}	1.040×10^{-5}	1.840×10^{-5}
6	1.190×10^{-5}	9.570×10^{-5}	1.480×10^{-5}
7	1.160×10^{-5}	9.850×10^{-5}	1.730×10^{-5}
8	8.670×10^{-5}	7.260×10^{-5}	1.300×10^{-5}

4.6.3.2 Hasil Keakurasian Koordinat Titik-titik Objek Menggunakan *Console Application* pada Bahasa Pemrograman C#

Setelah dihitung menggunakan *Console Application* pada Bahasa Pemrograman C# dapat dihasilkan nilai akurasi untuk masing-masing koordinat titik objek pada Tabel (4.24).

Tabel 4.25 Ketelitian Koordinat Titik-titik Objek Menggunakan *Console Application* pada Bahasa Pemrograman C#

Titik	σ_x	σ_y	σ_z
1	1.925×10^{-5}	1.573×10^{-5}	2.342×10^{-5}
2	1.158×10^{-5}	1.133×10^{-5}	3.353×10^{-5}
3	1.477×10^{-5}	1.337×10^{-5}	2.702×10^{-5}
4	1.434×10^{-5}	1.311×10^{-5}	2.682×10^{-5}
5	1.229×10^{-5}	1.043×10^{-5}	1.839×10^{-5}
6	1.189×10^{-5}	9.574×10^{-6}	1.480×10^{-5}
7	1.158×10^{-5}	9.850×10^{-6}	1.734×10^{-5}
8	8.669×10^{-6}	7.256×10^{-6}	1.300×10^{-5}

4.6.3.3 Hasil Keakurasian Koordinat Titik-titik Objek Menggunakan MATLAB

Setelah dihitung menggunakan MATLAB dapat dihasilkan nilai akurasi rata-rata untuk koordinat titik-titik objek sebagai berikut.

Tabel 4.26 Ketelitian Koordinat Titik-titik Objek Menggunakan MATLAB 2008

Titik	σ_X	σ_Y	σ_Z
1	1.925×10^{-5}	1.574×10^{-5}	2.344×10^{-5}
2	1.159×10^{-5}	1.134×10^{-5}	3.355×10^{-5}
3	1.476×10^{-5}	1.337×10^{-5}	2.704×10^{-5}
4	1.434×10^{-5}	1.311×10^{-5}	2.685×10^{-5}
5	1.228×10^{-5}	1.043×10^{-5}	1.841×10^{-5}
6	1.188×10^{-5}	9.570×10^{-6}	1.481×10^{-5}
7	1.158×10^{-5}	9.850×10^{-6}	1.736×10^{-5}
8	8.670×10^{-6}	7.250×10^{-6}	1.300×10^{-5}

Dilihat dari hasil perhitungan *Microsoft Excel*, *Console Application* pada Bahasa Pemrograman C#, MATLAB dari algoritma yang telah dibuat dapat dinilai bahwa hasil presisi yang dihasilkan oleh ketiga perhitungan menyatakan tidak adanya perbedaan yang signifikan. Dengan demikian, dapat dikatakan bahwa algoritma yang dibuat dalam ketiga program tersebut tepat digunakan dalam proses perhitungan *bundle adjustment*.

4.8 Evaluasi Masing-masing Metode Perhitungan *Bundle Adjustment*

Ketiga metode yang digunakan dalam penelitian ini memiliki keunggulan dan kelemahan masing-masing.

4.8.1 Keunggulan dan Kelemahan Penggunaan *Microsoft Excel*

Dibandingkan dengan bahasa C# dan *MATLAB*, *Microsoft Excel* memiliki keunggulan, yaitu mudah dipersiapkan karena penggunaan menggunakan *Microsoft Excel* untuk menghitung *bundle adjustment* tidak menggunakan *source code* dan *form* khusus untuk menjalankannya.

Sedangkan penggunaan *Microsoft Excel* ini memiliki kelemahan, yaitu

1. Tidak efisien waktu karena proses pekerjaan secara manual.
2. Keterbatasan dalam melakukan perhitungan matrik dengan dimensi yang besar.

4.8.2 Keunggulan dan Kelemahan Penggunaan *Console Application* pada Bahasa Pemrograman C#

Dibandingkan dengan bahasa *Excel* dan *MATLAB*, *Console application* bahasa C# memiliki keunggulan, yaitu penggunaan waktunya setingkat lebih efisien dibandingkan dengan *Microsoft Excel*. Karena sebagian perhitungan sudah dilakukan secara otomatis tanpa dihitung secara manual.

Sedangkan kelemahan dari penggunaan *Console Application* pada bahasa pemrograman C# ini, yaitu masih sulit membangun *sourcecode* perhitungan yang fleksibel untuk banyak jumlah data.

4.8.3 Keunggulan dan Kelemahan Penggunaan *MATLAB 2008*

Dibandingkan dengan bahasa *Microsoft Excel* dan *Console application C#*, *MATLAB* memiliki keunggulan, yaitu

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 input dan iterasi yang akan terjadi.
2. Bisa menghitung berapapun besar dimensi matriknya. karena matlab memang khusus digunakan untuk menghitung matrik.

3. Penggunaan waktu yang lebih efisien saat proses penghitungan karena keseluruhan tahap dapat dihitung secara otomatis setelah *source code* disusun secara benar.

Sedangkan kelemahan dari penggunaan *MATLAB* ini, yaitu inefisiensi waktu saat menyusun *source code* dan formnya. Untuk menghasilkan program yang dapat melakukan penghitungan secara efisien memerlukan persiapan yang lebih matang sehingga akan memerlukan waktu yang lebih banyak dalam melakukan persiapan tersebut.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil studi penelitian dengan judul “Desain Algoritma *Bundle Adjustment* dalam *Close Range Photogrammetry*”, Studi Kasus : *Fly Over Arjosari Malang*, dapat diambil beberapa kesimpulan sebagai berikut :

- 1 *Bundle adjustment* merupakan salah satu bagian dari rangkaian proses *close range photogrammetry*. *Bundle adjustment* digunakan untuk mengevaluasi parameter *exterior orientation* (EO) dan koordinat titik objek yang dihasilkan pada tahap *relative orientation* dan *intersection*, sehingga diperoleh nilai akurasi dan presisi dari koordinat titik objek.
- 2 Algoritma perhitungan proses *bundle adjustment* dalam metode *close range photogrammetry* dapat didesain menggunakan perangkat lunak *Microsoft Excel*, *Console Application* pada bahasa Pemrograman C# dan MATLAB 2008.
- 3 Hasil solusi parameter dan ukuran ketelitian dari tahap *bundle adjustment* yaitu parameter *exterior orientation* (EO) *update* dan koordinat titik objek *update* merupakan hasil atau nilai terakurat yang dapat digunakan untuk proses selanjutnya dalam *close range photogrammetry*.

- 4 Ketiga metode yang diterapkan memiliki keunggulan dan kelemahan masing-masing.
- 5 Hasil akhir dari perhitungan *bundle adjustment* dari ketiga metode tersebut menunjukkan nilai yang hampir sama, sehingga dapat dikatakan layak dipergunakan untuk menghitung *bundle adjustment*.
- 6 Nilai pendekatan parameter EO diperoleh dari proses *relative orientation*, dan nilai pendekatan titik-titik objek diperoleh dari proses *intersection* pada tahap sebelumnya sudah mencerminkan nilai yang akurat.
- 7 Hasil akhir dari *bundle adjustment* dapat digunakan untuk analisis deformasi, dengan ketentuan pemotretan dilakukan pada epok yang berbeda.

5.2 Saran

Saran yang dapat diberikan peneliti berdasarkan hasil penelitian “Desain Algoritma *Bundle Adjustment* dalam *Close Range Photogrammetry*”, Studi Kasus : Fly Over Arjosari Malang, antara lain :

1. Melakukan pemotretan di malam hari supaya stiker *retro reflective target* dapat memantulkan cahaya lebih terang sehingga kualitas data foto yang diperoleh dapat lebih baik seperti yang diharapkan.
2. Untuk tujuan memperoleh ketelitian yang relatif lebih tinggi, hendaknya bagi peneliti yang berminat melanjutkan penelitian ini dapat mengembangkan lagi algoritma ini, sehingga mendapatkan nilai presisi yang minimal.

Daftar Pustaka

- Away, Gunaidi Abdia. 2006. *MATLAB Programming*. Bandung : Informatika
- Brown, D. C. 1974. *Evolution, Application and Potential of The Bundle Method of Photogrammetric Triangulation*. Melbourne, Florida, Geodetic Services, Inc.: 95.
- Cooper, M.A.R. and Robson, S. 2001. *Theory Of Close Range Photogrammetry*. Wittles Publishing, London.
- Franck Jung, D. B. 2006. *Bundle Adjustment and Incidence of Linear Features on the Accuracy of External Calibration Parameters*. Matis Laboratory. Saint Mandé Cedex.
- Fraser, C. S. 1997. *Photogrammetric Orientation : Transformation from Image to Object Space*. Photogrammetry 451-447. Melbourne, The University of Melbourne.
- Fraundorfer, F., D. Scaramuzza, et al. 2009. *A Constricted Bundle Adjustment Parameterization for Relative Scale Estimation in Visual Odometry*. IEEE International Conference on Robotics and Automation.
- Geosystem, L. 2006. *Leica Photogrammetry Suite Project Manager, Leica Geosystems Geospatial Imaging*, United States of America.
- Heindl, F. J. 1981. *Direct Editing of Normal Equations of the Banded-Bordered Form*. Photogrammetric Engineering And Remote Sensing 47(4): 489-493.
- Jacobsen, K. 2000. *Combined Bundle Block adjustment Versus Direct Sensor Orientation*. Institute for Photogrammetry and Engineering Surveys. Hannover, University of Hannover.
- 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.
- Linder, W. 2006. *Digital Photogrammetry A Practical Course*. Berlin, Springer
- Manual of Photomodeller*
- Mikhail, J. S. Bethel, et al. 2001. *Introduction To Modern Photogrammetry*. New York, John Wiley & Sons, Inc.

- Sahdan, A. Y. b. 2006. *Penggunaan Penyepadanan Imej Berdasarkan kawasan Dalam Fotogrametri Jarak Dekat Bagi Pengukuran Permukaan Struktur*. Fakulti Kejuruteraan Awam. Johor, Universiti Teknologi Malaysia.
- Schut, G., H. 1980. *Block Adjustment of Bundles*. The Canadian Surveyor 34(2): 139-152.
- Shirkhani, A., D. M. Varshosaz, et al. 2006. *3D Coordinate Measurement of Dam by Close Range Photogrammetry*. Department of Photogrammetry and Remote Sensing. Tehran, K.N. Toosi University of Technology.
- Photogrammetry and Remote Sensing. Tehran, K.N. Toosi University of Technology.*
- Ruther, H. 1989. An Overview of Software in Non-topographic Photogrammetry. In : Non-topographic Photogrammetry, 2nd Edition. *American Society for Photogrammetry and Remote Sensing. Falls Church, Virginia, USA., pp129-145.*
- Tjahjadi, E. 2008a. *Precision Feature Extraction From Unmanned Aerialplatforms*. Jurusan Teknik Geodesi. Malang, Institut Teknologi Nasional.
- Tjahjadi, E. 2008b. *Ketelitian Pengukuran Dari Kamera Non Metrik*. Jurusan Teknik Geodesi. Malang, Institut Teknologi Nasional.
- Triggs, B., P. McLauchlan, F., et al. 2000. *Bundle Adjustment—A Modern Synthesis*. Lecture Notes in Computer Science 1883: 298-372.
- Uffenkamp, V. 1998. *State Of The Art Of High Precision Industrial Photogrammetry*. Institut für Photogrammetrie und Bildverarbeitung. Berlin, Technische Universität Braunschweig
- V-Stars. 2005. *Basics of Photogrammetry*.
- Wang, X. and T. A. Clarke. 1998. Separate Adjustment of Close Range Photogrammetric Measurements. *ISPRS XXXII*.
- Wolf, P. R. and B. A. Dewitt. 2000. *Element Of Photogrammetry With Applications in GIS*. Madison, Thomas Casson
- Wolf, P. R. 1993. *Element of Photogrammetry With Air Photo Interpretation and Remote Sensing, Second Edition*. Madison, The University of Wisconsin.
- Wong, 1980. *Manual Of Photogrammetry*.

Lampiran A

Perhitungan *Bundle Adjustment* Pada

Microsoft Excel

Data Parameter Awal

parameter kamera	
xo	0
yo	0
foc	35

parameter EO						
photo	w	j	k	XL	YL	ZL
1	-0.037	0.192	0.056	0.009	0.002	-0.010
2	82.356	-73.204	99.179	-2.968	-0.774	-2.476

Point	Foto 1				Foto 2			
	Koordinat Photo		akurasi		Koordinat Photo		akurasi	
	x	y	X	Y	x	y	X	Y
1	-3.60	1.85	0.00024	0.00024	-3.91	2.79	0.00029	0.00029
10	4.74	5.62	0.00016	0.00016	3.37	0.64	0.00034	0.00035
15	-0.58	1.93	0.00020	0.00020	-0.38	0.61	0.00031	0.00031
16	0.42	2.34	0.00020	0.00020	0.52	0.37	0.00030	0.00030
35	-0.57	-0.27	0.00017	0.00017	-0.20	-1.34	0.00021	0.00021
36	0.37	0.05	0.00017	0.00017	0.64	-1.53	0.00017	0.00017
41	-3.76	-2.45	0.00014	0.00014	-3.76	-1.72	0.00022	0.00023
50	4.75	0.19	0.00014	0.00014	3.62	-3.06	0.00014	0.00014

Point	Koordinat		
	Object Space Point		
	X	Y	Z
1	-0.30366	0.15469	-2.95213
10	0.31179	0.36885	-2.30701
15	-0.04320	0.14462	-2.64070
16	0.03082	0.17050	-2.56119
35	-0.04122	-0.01967	-2.57576
36	0.02718	0.00360	-2.50221
41	-0.30244	-0.19787	-2.82334
50	0.29751	0.01249	-2.19040

Proses Perhitungan Bundle Adjustment

1. Parameter Matrik Rotasi

$$R_1 = \begin{vmatrix} 0.99999 & 0.00097 & -0.00335 \\ -0.00097 & 1.00000 & -0.00064 \\ 0.00335 & 0.00064 & 0.99999 \end{vmatrix} \quad R_2 = \begin{vmatrix} -0.04609 & 0.28266 & 0.95811 \\ -0.28526 & 0.91547 & -0.28380 \\ -0.95734 & -0.28639 & 0.03844 \end{vmatrix}$$

2. Sin, Cos [ω , ϕ , κ]

	sin_w	sin_j	sin_k	cos_w	cos_j	cos_k
1	0.00	0.00	0.00	1.00	1.00	1.00
2	0.99	-0.96	0.99	0.13	0.29	-0.16

3. delta X, Y, Z

Point	1			2		
	delta X	delta Y	delta Z	delta X	delta Y	delta Z
1	-0.313	0.153	-2.942	2.664	0.929	-0.476
10	0.303	0.367	-2.297	3.280	1.143	0.169
15	-0.052	0.143	-2.631	2.925	0.919	-0.165
16	0.022	0.169	-2.551	2.999	0.945	-0.085
35	-0.050	-0.021	-2.566	2.927	0.754	-0.100
36	0.018	0.002	-2.492	2.995	0.778	-0.026
41	-0.312	-0.200	-2.813	2.666	0.576	-0.347
50	0.288	0.011	-2.180	3.265	0.787	0.286

4. Parameter r, s, q

Titik	Foto 1			Foto 2		
	r	s	q	r	s	q
1	-0.303	0.155	-2.943	-0.317	0.225	-2.835
10	0.311	0.368	-2.296	0.334	0.063	-3.461
15	-0.043	0.145	-2.631	-0.033	0.054	-3.069
16	0.030	0.170	-2.551	0.047	0.034	-3.145
35	-0.042	-0.020	-2.566	-0.017	-0.116	-3.022
36	0.026	0.004	-2.492	0.057	-0.135	-3.091
41	-0.302	-0.197	-2.814	-0.293	-0.134	-2.730
50	0.296	0.012	-2.179	0.345	-0.292	-3.340

5. Menghitung Nilai B1, B2, f dan w

Foto 1

Titik

1	B ₁	0.16	35.37	1.85	-11.89	-0.01	1.26	B ₂	11.89	0.01	-1.26
	B ₁	-35.08	-0.22	3.60	0.01	-11.89	-0.62	B ₂	-0.01	11.89	0.62
2	B ₁	-0.78	35.64	5.62	-15.25	-0.02	-2.01	B ₂	15.25	0.02	2.01
	B ₁	-35.92	0.73	-4.74	0.01	-15.25	-2.44	B ₂	-0.01	15.25	2.44
3	B ₁	0.00	35.01	1.93	-13.30	-0.01	0.26	B ₂	13.30	0.01	-0.26
	B ₁	-35.10	-0.07	0.58	0.01	-13.31	-0.72	B ₂	-0.01	13.31	0.72
4	B ₁	-0.05	35.00	2.34	-13.72	-0.01	-0.12	B ₂	13.72	0.01	0.12
	B ₁	-35.16	-0.01	-0.42	0.01	-13.72	-0.91	B ₂	-0.01	13.72	0.91
5	B ₁	-0.04	35.01	-0.27	-13.64	-0.01	0.27	B ₂	13.64	0.01	-0.27
	B ₁	-35.00	-0.03	0.57	0.01	-13.64	0.11	B ₂	-0.01	13.64	-0.11
6	B ₁	-0.03	35.00	0.05	-14.05	-0.01	-0.10	B ₂	14.05	0.01	0.10
	B ₁	-35.00	-0.03	-0.37	0.01	-14.05	-0.01	B ₂	-0.01	14.05	0.01
7	B ₁	-0.31	35.40	-2.46	-12.43	-0.01	1.38	B ₂	12.43	0.01	-1.38
	B ₁	-35.16	0.23	3.76	0.01	-12.44	0.88	B ₂	-0.01	12.44	-0.88
8	B ₁	-0.06	35.64	0.19	-16.07	-0.02	-2.13	B ₂	16.07	0.02	2.13
	B ₁	-35.02	-0.01	-4.75	0.02	-16.06	-0.08	B ₂	-0.02	16.06	0.08

1	f	0.001		W	1.8E+07	0	
	f	0.002		W	0	1.8E+07	
2	f	0.001		W	3.7E+07	0	
	f	0.001		W	0	3.9E+07	
3	f	0.001		W	2.5E+07	0	
	f	0.001		W	0	2.5E+07	
4	f	0.001		W	2.5E+07	0	
	f	0.001		W	0	2.5E+07	
5	f	0.001		W	3.5E+07	0	
	f	0.001		W	0	3.5E+07	
6	f	0.001		W	3.5E+07	0	
	f	0.001		W	0	3.5E+07	
7	f	0.001		W	4.8E+07	0	
	f	0.001		W	0	4.8E+07	
8	f	0.001		W	5.2E+07	0	
	f	0.001		W	0	5.3E+07	

5. Menghitung Nilai B1, B2, f dan w

Foto 2

Titik

1	B ₁	-12.79	-5.35	2.78	-0.75	-3.88	-11.78	B ₂	0.75	3.88	11.78
		-2.03	-34.72	3.91	4.46	-11.02	3.47		-4.46	11.02	-3.47
2	B ₁	-10.68	-5.70	0.64	1.40	-2.58	-9.73	B ₂	-1.40	2.58	9.73
		4.83	-34.57	-3.38	3.06	-9.21	2.86		-3.06	9.21	-2.86
3	B ₁	-10.57	-5.58	0.61	0.41	-3.26	-10.92	B ₂	-0.41	3.26	10.92
		1.26	-34.56	0.38	3.44	-10.38	3.23		-3.44	10.38	-3.23
4	B ₁	-10.34	-5.59	0.37	0.67	-3.10	-10.67	B ₂	-0.67	3.10	10.67
		2.11	-34.56	-0.52	3.29	-10.16	3.15		-3.29	10.16	-3.15
5	B ₁	-8.70	-5.59	-1.34	0.47	-3.29	-11.09	B ₂	-0.47	3.29	11.09
		1.42	-34.60	0.20	2.88	-10.73	3.30		-2.88	10.73	-3.30
6	B ₁	-8.53	-5.56	-1.53	0.72	-3.14	-10.86	B ₂	-0.72	3.14	10.86
		2.24	-34.61	-0.64	2.76	-10.51	3.23		-2.76	10.51	-3.23
7	B ₁	-8.44	-5.83	-1.72	-0.73	-4.02	-12.23	B ₂	0.73	4.02	12.23
		-2.03	-34.66	3.75	3.05	-11.92	3.66		-3.05	11.92	-3.66
8	B ₁	-7.17	-5.33	-3.06	1.52	-2.65	-10.08	B ₂	-1.52	2.65	10.08
		5.18	-34.77	-3.62	2.11	-9.85	3.01		-2.11	9.85	-3.01

1	f	-0.001	W	1.2E+07	0
		0.002		0	1.2E+07
2	f	-0.001	W	8.4E+06	0
		0.002		0	8.3E+06
3	f	-0.001	W	1.1E+07	0
		0.001		0	1.1E+07
4	f	-0.002	W	1.1E+07	0
		0.001		0	1.1E+07
5	f	-0.002	W	2.2E+07	0
		0.001		0	2.2E+07
6	f	-0.002	W	3.6E+07	0
		0.001		0	3.6E+07
7	f	-0.002	W	2.0E+07	0
		0.003		0	1.9E+07
8	f	-0.002	W	5.2E+07	0
		0.000		0	5.2E+07

6. Menyusun Matrik N

Titik

FOTO 1

1	A1 ^T WA1	2.2E+10	2.4E+08	-2.3E+09	-4.0E+07	7.5E+09	3.9E+08	A1 ^T WA2	4.0E+07	-7.5E+09	-3.9E+08	Ci=A1 ^T Wf	-9.9E+05	
		2.4E+08	2.2E+10	1.2E+09	-7.5E+09	4.1E+07	8.0E+08		7.5E+09	-4.1E+07	-8.0E+08		5.1E+05	
		-2.3E+09	1.2E+09	2.9E+08	-3.9E+08	-7.7E+08	1.6E+06		3.9E+08	7.7E+08	-1.6E+06		1.3E+05	
	A2 ^T WA1	-4.0E+07	-7.5E+09	-3.9E+08	2.5E+09	2.6E+05	-2.7E+08	A2 ^T WA2	-2.5E+09	-2.6E+05	2.7E+08	Ci=A2 ^T Wf	-1.7E+05	
		7.5E+09	4.1E+07	-7.7E+08	2.6E+05	2.5E+09	1.3E+08		-2.6E+05	-2.5E+09	-1.3E+08		-3.4E+05	
		3.9E+08	8.0E+08	1.6E+06	-2.7E+08	1.3E+08	3.5E+07		2.7E+08	-1.3E+08	-3.5E+07		1.0E+03	
2	A1 ^T WA1	4.0E+07	7.5E+09	3.9E+08	-2.5E+09	-2.6E+05	2.7E+08	A1 ^T WA2	2.5E+09	2.6E+05	-2.7E+08	Ci=A1 ^T Wf	1.7E+05	
		-7.5E+09	-4.1E+07	7.7E+08	-2.6E+05	-2.5E+09	-1.3E+08		A2 ^T WA2	2.6E+05	2.5E+09		1.3E+08	3.4E+05
		-3.9E+08	-8.0E+08	-1.6E+06	2.7E+08	-1.3E+08	-3.5E+07			-2.7E+08	1.3E+08		3.5E+07	-1.0E+03
	5.0E+10	-2.0E+09	6.4E+09	4.3E+08	2.1E+10	3.4E+09	A1 ^T WA2	-4.3E+08		-2.1E+10	-3.4E+09	Ci=A1 ^T Wf	-9.3E+05	
	-2.0E+09	4.7E+10	7.3E+09	-2.0E+10	-4.5E+08	-2.7E+09		2.0E+10	4.5E+08	2.7E+09	1.7E+06			
	6.4E+09	7.3E+09	2.0E+09	-3.2E+09	2.8E+09	2.6E+07		3.2E+09	-2.8E+09	-2.6E+07	1.4E+05			
A2 ^T WA1	4.3E+08	-2.0E+10	-3.2E+09	8.6E+09	5.2E+06	1.1E+09	A2 ^T WA2	-8.6E+09	-5.2E+06	-1.1E+09	Ci=A2 ^T Wf	-7.1E+05		
	2.1E+10	-4.5E+08	2.8E+09	5.2E+06	9.0E+09	1.4E+09		-5.2E+06	-9.0E+09	-1.4E+09		-3.8E+05		
	3.4E+09	-2.7E+09	2.6E+07	1.1E+09	1.4E+09	3.8E+08		-1.1E+09	-1.4E+09	-3.8E+08		-1.5E+05		
3	A1 ^T WA1	-4.3E+08	2.0E+10	3.2E+09	-8.6E+09	-5.2E+06	-1.1E+09	A1 ^T WA2	8.6E+09	5.2E+06	1.1E+09	Ci=A1 ^T Wf	7.1E+05	
		-2.1E+10	4.5E+08	-2.8E+09	-5.2E+06	-9.0E+09	-1.4E+09		A2 ^T WA2	5.2E+06	9.0E+09		1.4E+09	3.8E+05
		-3.4E+09	2.7E+09	-2.6E+07	-1.1E+09	-1.4E+09	-3.8E+08			1.1E+09	1.4E+09		3.8E+08	1.5E+05
	3.1E+10	6.2E+07	-5.1E+08	-1.1E+07	1.2E+10	6.3E+08	A1 ^T WA2	1.1E+07		-1.2E+10	-6.3E+08	Ci=A1 ^T Wf	-1.1E+06	
	6.2E+07	3.1E+10	1.7E+09	-1.2E+10	1.1E+07	2.4E+08		1.2E+10	-1.1E+07	-2.4E+08	7.6E+05			
	-5.1E+08	1.7E+09	1.0E+08	-6.5E+08	-1.9E+08	2.5E+06		6.5E+08	1.9E+08	-2.5E+06	6.0E+04			
A2 ^T WA1	-1.1E+07	-1.2E+10	-6.5E+08	4.5E+09	8.2E+05	-8.9E+07	A2 ^T WA2	-4.5E+09	-8.2E+05	8.9E+07	Ci=A2 ^T Wf	-2.9E+05		
	1.2E+10	1.1E+07	-1.9E+08	8.2E+05	4.4E+09	2.4E+08		-8.2E+05	-4.4E+09	-2.4E+08		-4.3E+05		
	6.3E+08	2.4E+08	2.5E+06	-8.9E+07	2.4E+08	1.5E+07		8.9E+07	-2.4E+08	-1.5E+07		-1.7E+04		
A2 ^T WA1	1.1E+07	1.2E+10	6.5E+08	-4.5E+09	-8.2E+05	8.9E+07	A2 ^T WA2	4.5E+09	8.2E+05	-8.9E+07	Ci=A2 ^T Wf	2.9E+05		
	-1.2E+10	-1.1E+07	1.9E+08	-8.2E+05	-4.4E+09	-2.4E+08		8.2E+05	4.4E+09	2.4E+08		4.3E+05		
	-6.3E+08	-2.4E+08	-2.5E+06	8.9E+07	-2.4E+08	-1.5E+07		-8.9E+07	2.4E+08	1.5E+07		1.7E+04		

4	A1 ^T WA1	3.1E+10	-4.2E+07	3.6E+08	9.8E+06	1.2E+10	8.0E+08	A1 ^T WA2	-9.8E+06	-1.2E+10	-8.0E+08	Ci=A1 ^T Wf	-1.1E+06
		-4.2E+07	3.1E+10	2.1E+09	-1.2E+10	-9.7E+06	-1.0E+08		1.2E+10	9.7E+06	1.0E+08		8.0E+05
		3.6E+08	2.1E+09	1.4E+08	-8.1E+08	1.4E+08	2.5E+06		8.1E+08	-1.4E+08	-2.5E+06		4.1E+04
		9.8E+06	-1.2E+10	-8.1E+08	4.8E+09	1.1E+06	4.1E+07		-4.8E+09	-1.1E+06	-4.1E+07		-3.1E+05
	A2 ^T WA1	1.2E+10	-9.7E+06	1.4E+08	1.1E+06	4.7E+09	3.1E+08	A2 ^T WA2	-1.1E+06	-4.7E+09	-3.1E+08	Ci=A2 ^T Wf	-4.1E+05
		8.0E+08	-1.0E+08	2.5E+06	4.1E+07	3.1E+08	2.1E+07		-4.1E+07	-3.1E+08	-2.1E+07		-3.0E+04
		-9.8E+06	1.2E+10	8.1E+08	-4.8E+09	-1.1E+06	-4.1E+07		4.8E+09	1.1E+06	4.1E+07		3.1E+05
		-1.2E+10	9.7E+06	-1.4E+08	-1.1E+06	-4.7E+09	-3.1E+08		1.1E+06	4.7E+09	3.1E+08		4.1E+05
		-8.0E+08	1.0E+08	-2.5E+06	-4.1E+07	-3.1E+08	-2.1E+07		4.1E+07	3.1E+08		3.0E+04	

5	A1 ^T WA1	4.3E+10	-1.2E+07	-7.0E+08	2.1E+06	1.7E+10	-1.4E+08	A1 ^T WA2	-2.1E+06	-1.7E+10	1.4E+08	Ci=A1 ^T Wf	-1.4E+06
		-1.2E+07	4.3E+10	-3.3E+08	-1.7E+10	-1.9E+06	3.3E+08		1.7E+10	1.9E+06	-3.3E+08		9.1E+05
		-7.0E+08	-3.3E+08	1.4E+07	1.3E+08	-2.7E+08	-2.6E+05		-1.3E+08	2.7E+08	2.6E+05		1.6E+04
		2.1E+06	-1.7E+10	1.3E+08	6.5E+09	-2.4E+05	-1.3E+08		-6.5E+09	2.4E+05	1.3E+08		-3.5E+05
	A2 ^T WA1	1.7E+10	-1.9E+06	-2.7E+08	-2.4E+05	6.5E+09	-5.4E+07	A2 ^T WA2	2.4E+05	-6.5E+09	5.4E+07	Ci=A2 ^T Wf	-5.5E+05
		-1.4E+08	3.3E+08	-2.6E+05	-1.3E+08	-5.4E+07	3.0E+06		1.3E+08	5.4E+07	-3.0E+06		1.2E+04
		-2.1E+06	1.7E+10	-1.3E+08	-6.5E+09	2.4E+05	1.3E+08		6.5E+09	-2.4E+05	-1.3E+08		3.5E+05
		-1.7E+10	1.9E+06	2.7E+08	2.4E+05	-6.5E+09	5.4E+07		-2.4E+05	6.5E+09	-5.4E+07		5.5E+05
		1.4E+08	-3.3E+08	2.6E+05	1.3E+08	5.4E+07	-3.0E+06		-1.3E+08	-5.4E+07		-1.2E+04	

6	A1 ^T WA1	4.2E+10	-1.7E+06	4.5E+08	4.9E+05	1.7E+10	1.3E+07	A1 ^T WA2	-4.9E+05	-1.7E+10	-1.3E+07	Ci=A1 ^T Wf	-1.4E+06
		-1.7E+06	4.3E+10	6.1E+07	-1.7E+10	-5.7E+05	-1.2E+08		1.7E+10	5.7E+05	1.2E+08		9.2E+05
		4.5E+08	6.1E+07	4.8E+06	-2.4E+07	1.8E+08	-3.7E+04		2.4E+07	-1.8E+08	3.7E+04		-1.3E+04
		4.9E+05	-1.7E+10	-2.4E+07	6.9E+09	1.6E+05	5.0E+07		-6.9E+09	-1.6E+05	-5.0E+07		-3.7E+05
	A2 ^T WA1	1.7E+10	-5.7E+05	1.8E+08	1.6E+05	6.8E+09	5.3E+06	A2 ^T WA2	-1.6E+05	-6.8E+09	-5.3E+06	Ci=A2 ^T Wf	-5.6E+05
		1.3E+07	-1.2E+08	-3.7E+04	5.0E+07	5.3E+06	3.6E+05		-5.0E+07	-5.3E+06	-3.6E+05		-3.1E+03
		-4.9E+05	1.7E+10	2.4E+07	-6.9E+09	-1.6E+05	-5.0E+07		6.9E+09	1.6E+05	5.0E+07		3.7E+05
		-1.7E+10	5.7E+05	-1.8E+08	-1.6E+05	-6.8E+09	-5.3E+06		1.6E+05	6.8E+09	5.3E+06		5.6E+05
		-1.3E+07	1.2E+08	3.7E+04	-5.0E+07	-5.3E+06	-3.6E+05		5.0E+07	5.3E+06		3.1E+03	

7	A1 ^T WA1	5.9E+10	-9.0E+08	-6.3E+09	1.6E+08	2.1E+10	-1.5E+09	A1 ^T WA2	-1.6E+08	-2.1E+10	1.5E+09	Ci=A1 ^T Wf	-2.1E+06
		-9.0E+08	6.0E+10	-4.1E+09	-2.1E+10	-1.5E+08	2.3E+09		2.1E+10	1.5E+08	-2.3E+09		1.1E+06
		-6.3E+09	-4.1E+09	9.6E+08	1.5E+09	-2.2E+09	-3.6E+06		-1.5E+09	2.2E+09	3.6E+06		1.4E+05
		1.6E+08	-2.1E+10	1.5E+09	7.4E+09	-2.3E+06	-8.2E+08		-7.4E+09	2.3E+06	8.2E+08		-3.8E+05
	A2 ^T WA1	2.1E+10	-1.5E+08	-2.2E+09	-2.3E+06	7.4E+09	-5.2E+08	A2 ^T WA2	2.3E+06	-7.4E+09	5.2E+08	Ci=A2 ^T Wf	-7.2E+05
		-1.5E+09	2.3E+09	-3.6E+06	-8.2E+08	-5.2E+08	1.3E+08		8.2E+08	5.2E+08	-1.3E+08		9.3E+04
		-1.6E+08	2.1E+10	-1.5E+09	-7.4E+09	2.3E+06	8.2E+08		7.4E+09	-2.3E+06	-8.2E+08		3.8E+05
		-2.1E+10	1.5E+08	2.2E+09	2.3E+06	-7.4E+09	5.2E+08		-2.3E+06	7.4E+09	-5.2E+08		7.2E+05
		1.5E+09	-2.3E+09	3.6E+06	8.2E+08	5.2E+08	-1.3E+08		-8.2E+08	-5.2E+08	1.3E+08		-9.3E+04
8	A1 ^T WA1	6.5E+10	-9.6E+07	8.8E+09	2.1E+07	3.0E+10	1.5E+08	A1 ^T WA2	-2.1E+07	-3.0E+10	-1.5E+08	Ci=A1 ^T Wf	-1.6E+06
		-9.6E+07	6.6E+10	3.6E+08	-3.0E+10	-2.4E+07	-3.9E+09		3.0E+10	2.4E+07	3.9E+09		1.4E+06
		8.8E+09	3.6E+08	1.2E+09	-1.6E+08	4.0E+09	-1.5E+06		1.6E+08	-4.0E+09	1.5E+06		-2.0E+05
		2.1E+07	-3.0E+10	-1.6E+08	1.3E+10	1.0E+06	1.8E+09		-1.3E+10	-1.0E+06	-1.8E+09		-6.4E+05
	A2 ^T WA1	3.0E+10	-2.4E+07	4.0E+09	1.0E+06	1.4E+10	6.8E+07	A2 ^T WA2	-1.0E+06	-1.4E+10	-6.8E+07	Ci=A2 ^T Wf	-7.1E+05
		1.5E+08	-3.9E+09	-1.5E+06	1.8E+09	6.8E+07	2.3E+08		-1.8E+09	-6.8E+07	-2.3E+08		-8.9E+04
		-2.1E+07	3.0E+10	1.6E+08	-1.3E+10	-1.0E+06	-1.8E+09		1.3E+10	1.0E+06	1.8E+09		6.4E+05
		-3.0E+10	2.4E+07	-4.0E+09	-1.0E+06	-1.4E+10	-6.8E+07		1.0E+06	1.4E+10	6.8E+07		7.1E+05
		-1.5E+08	3.9E+09	1.5E+06	-1.8E+09	-6.8E+07	-2.3E+08		1.8E+09	6.8E+07	2.3E+08		8.9E+04

Titik

FOTO 2

1	A1 ^T WA1	2.0E+09	1.7E+09	-5.2E+08	6.3E+06	8.6E+08	1.7E+09	A1 ^T WA2	-6.3E+06	-8.6E+08	-1.7E+09	Ci=A1 ^T Wf	1.0E+05
		1.7E+09	1.5E+10	-1.8E+09	-1.8E+09	4.8E+09	-6.9E+08		1.8E+09	-4.8E+09	6.9E+08		-7.5E+05
		-5.2E+08	-1.8E+09	2.8E+08	1.8E+08	-6.4E+08	-2.3E+08		-1.8E+08	6.4E+08	2.3E+08		6.0E+04
		6.3E+06	-1.8E+09	1.8E+08	2.5E+08	-5.5E+08	2.9E+08		-2.5E+08	5.5E+08	-2.9E+08		1.1E+05
	A2 ^T WA1	8.6E+08	4.8E+09	-6.4E+08	-5.5E+08	1.6E+09	8.9E+07	A2 ^T WA2	5.5E+08	-1.6E+09	-8.9E+07	Ci=A2 ^T Wf	-2.1E+05
		1.7E+09	-6.9E+08	-2.3E+08	2.9E+08	8.9E+07	1.8E+09		-2.9E+08	-8.9E+07	-1.8E+09		2.2E+05
		-6.3E+06	1.8E+09	-1.8E+08	-2.5E+08	5.5E+08	-2.9E+08		2.5E+08	-5.5E+08	2.9E+08		-1.1E+05
		-8.6E+08	-4.8E+09	6.4E+08	5.5E+08	-1.6E+09	-8.9E+07		-5.5E+08	1.6E+09	8.9E+07		2.1E+05
		-1.7E+09	6.9E+08	2.3E+08	-2.9E+08	-8.9E+07	-1.8E+09		2.9E+08	8.9E+07	1.8E+09		-2.2E+05

$A1^T W A1$	1.7E+09	-1.1E+07	2.6E+08	-1.2E+05	3.0E+08	2.2E+09	1.2E+05	-3.0E+08	-2.2E+09	4.0E+05
	-1.1E+07	2.7E+10	1.3E+07	-2.3E+09	8.6E+09	-1.1E+09	2.3E+09	-8.6E+09	1.1E+09	-7.7E+05
	2.6E+08	1.3E+07	4.1E+07	-1.2E+06	5.0E+07	3.4E+08	1.2E+06	-5.0E+07	-3.4E+08	6.1E+04
	-1.2E+05	-2.3E+09	-1.2E+06	1.9E+08	-7.1E+08	9.4E+07	-1.9E+08	7.1E+08	-9.4E+07	6.4E+04
	3.0E+08	8.6E+09	5.0E+07	7.1E+08	2.8E+09	2.5E+07	7.1E+08	-2.8E+09	-2.5E+07	-1.7E+05
	2.2E+09	-1.1E+09	3.4E+08	9.4E+07	2.5E+07	3.0E+09	-9.4E+07	-2.5E+07	-3.0E+09	5.5E+05
	1.2E+05	2.3E+09	1.2E+06	-1.9E+08	7.1E+08	-9.4E+07	1.9E+08	-7.1E+08	9.4E+07	-6.4E+04
$A2^T W A1$	-3.0E+08	-8.6E+09	-5.0E+07	7.1E+08	-2.8E+09	-2.5E+07	-7.1E+08	2.8E+09	2.5E+07	1.7E+05
	-2.2E+09	1.1E+09	-3.4E+08	-9.4E+07	-2.5E+07	-3.0E+09	9.4E+07	2.5E+07	-3.0E+09	-5.5E+05
	2.8E+09	-1.1E+09	4.1E+08	9.3E+05	1.2E+08	3.6E+09	-9.3E+05	-1.2E+08	-3.6E+09	6.6E+05
	-1.1E+09	4.4E+10	1.1E+09	-3.5E+09	1.4E+10	-1.8E+09	3.5E+09	-1.4E+10	1.8E+09	-8.8E+05
	4.1E+08	1.1E+09	9.8E+07	-1.0E+08	4.1E+08	5.2E+08	1.0E+08	-4.1E+08	-5.2E+08	8.1E+04
$A1^T W A1$	9.3E+05	-3.5E+09	-1.0E+08	2.9E+08	-1.1E+09	3.9E+07	-2.9E+08	1.1E+09	-3.9E+07	5.1E+04
	1.2E+08	1.4E+10	4.1E+08	-1.1E+09	4.3E+09	5.1E+06	1.1E+09	-4.3E+09	-5.1E+06	-1.7E+05
	3.6E+09	-1.8E+09	5.2E+08	3.9E+07	5.1E+06	4.6E+09	-3.9E+07	-5.1E+06	-4.6E+09	8.6E+05
	-9.3E+05	3.5E+09	1.0E+08	-2.9E+08	1.1E+09	-3.9E+07	2.9E+08	-1.1E+09	3.9E+07	-5.1E+04
$A2^T W A1$	-1.2E+08	-1.4E+10	-4.1E+08	1.1E+09	-4.3E+09	-5.1E+06	-1.1E+09	4.3E+09	5.1E+06	1.7E+05
	-3.6E+09	1.8E+09	-5.2E+08	-3.9E+07	-5.1E+06	-4.6E+09	3.9E+07	5.1E+06	-4.6E+09	-8.6E+05

$A1^T W A1$	1.5E+09	2.3E+09	1.4E+08	2.6E+06	1.1E+09	1.9E+09	-2.6E+06	-1.1E+09	-1.9E+09	2.1E+05
	2.3E+09	2.4E+10	-2.3E+09	-2.0E+09	8.4E+09	-1.0E+09	2.0E+09	-8.4E+09	1.0E+09	-1.5E+06
	1.4E+08	-2.3E+09	3.3E+08	2.5E+08	-7.2E+08	6.8E+08	-2.5E+08	7.2E+08	-6.8E+08	2.5E+05
	2.6E+06	-2.0E+09	2.5E+08	1.9E+08	-6.4E+08	3.9E+08	-1.9E+08	6.4E+08	-3.9E+08	1.8E+05
	1.1E+09	8.4E+09	-7.2E+08	-6.4E+08	3.1E+09	1.4E+08	6.4E+08	-3.1E+09	-1.4E+08	-4.5E+05
	1.9E+09	-1.0E+09	6.8E+08	3.9E+08	1.4E+08	3.2E+09	-3.9E+08	-1.4E+08	3.2E+09	6.4E+05
	-2.6E+06	2.0E+09	-2.5E+08	-1.9E+08	6.4E+08	-3.9E+08	1.9E+08	-6.4E+08	3.9E+08	-1.8E+05
$A2^T W A1$	-1.1E+09	-8.4E+09	7.2E+08	6.4E+08	-3.1E+09	-1.4E+08	-6.4E+08	3.1E+09	1.4E+08	4.5E+05
	-1.9E+09	1.0E+09	-6.8E+08	-3.9E+08	-1.4E+08	-3.2E+09	3.9E+08	-1.4E+08	-3.2E+09	-6.4E+05

$A1^T W A1$	4.1E+09	-7.4E+09	1.7E+08	1.0E+06	-1.7E+09	4.6E+09	-1.0E+06	-1.7E+09	-4.6E+09	9.6E+05
	-7.4E+09	6.5E+10	7.5E+09	-4.3E+09	1.9E+10	-2.7E+09	4.3E+09	-1.9E+10	2.7E+09	1.3E+05
	1.7E+08	7.5E+09	1.2E+09	-6.4E+08	2.3E+09	1.1E+09	-6.4E+08	-2.3E+09	-1.1E+09	3.2E+05
	1.0E+06	-4.3E+09	-6.4E+08	3.5E+08	-1.3E+09	-4.7E+08	-3.5E+08	1.3E+09	4.7E+08	-1.5E+05
	-1.7E+09	1.9E+10	2.3E+09	-1.3E+09	5.5E+08	-1.5E+08	1.3E+09	-5.5E+08	1.5E+08	1.8E+05
	4.6E+09	-2.7E+09	1.1E+09	-4.7E+08	-1.5E+08	5.8E+09	4.7E+08	-1.5E+08	-5.8E+09	1.3E+06
	-1.0E+06	4.3E+09	6.4E+08	-3.5E+08	1.3E+09	4.7E+08	3.5E+08	-1.3E+09	-4.7E+08	1.5E+05
$A2^T W A1$	1.7E+09	-1.9E+10	-2.3E+09	1.3E+09	-5.5E+08	1.5E+08	-1.3E+09	5.5E+08	-1.5E+08	-1.8E+05
	-4.6E+09	2.7E+09	-1.1E+09	4.7E+08	1.5E+08	-5.8E+09	-4.7E+08	-1.5E+08	5.8E+09	-1.3E+06

N

3.42E+11	-2.77E+09	6.283E+09	567774455	1.366E+11	3.788E+09	0	0	0	0	0	0	0	40360252
-2.8E+09	3.425E+11	8.196E+09	-1.36E+11	-5.87E+08	-3.16E+09	0	0	0	0	0	0	0	7.52E+09
6.28E+09	8.196E+09	4.746E+09	-3.63E+09	3.681E+09	27645583	0	0	0	0	0	0	0	391897636
5.68E+08	-1.36E+11	-3.63E+09	5.449E+10	6142080.3	1.687E+09	0	0	0	0	0	0	0	-2.53E+09
1.37E+11	-5.87E+08	3.681E+09	6142080.3	5.496E+10	1.614E+09	0	0	0	0	0	0	0	-262379.1
3.79E+09	-3.16E+09	27645583	1.687E+09	1.614E+09	813830521	0	0	0	0	0	0	0	268772377
0	0	0	0	0	0	0	1.569E+10	-5.37E+09	158315552	4702078	953583325	1.759E+10	-6314194
0	0	0	0	0	0	0	-5.37E+09	2.108E+11	5.383E+09	-1.73E+10	6.488E+10	-8.77E+09	1.807E+09
0	0	0	0	0	0	0	158315552	5.383E+09	2.031E+09	-3.96E+08	1.613E+09	2.112E+09	-1.84E+08
0	0	0	0	0	0	0	4702078	-1.73E+10	-3.96E+08	1.611E+09	-5.37E+09	405647291	-2.45E+08
0	0	0	0	0	0	0	953583325	6.488E+10	1.613E+09	-5.37E+09	2.045E+10	142012721	553869581
0	0	0	0	0	0	0	1.759E+10	-8.77E+09	2.112E+09	405647291	142012721	2.199E+10	-2.91E+08
40360252	7.52E+09	391897636	-2.53E+09	-262379.1	268772377	-6314194	1.807E+09	-1.84E+08	-2.45E+08	553869581	-2.91E+08	2.773E+09	
-7.5E+09	-41127773	770654394	-262379.1	-2.54E+09	-1.32E+08	-8.61E+08	-4.83E+09	644851166	553869581	-1.63E+09	-89289668	-5.54E+08	
-3.9E+08	-8.02E+08	-1582576	268772377	-1.32E+08	-35461447	-1.72E+09	688551275	229492491	-2.91E+08	-89289668	-1.8E+09	22007509	
-4.3E+08	2.011E+10	3.17E+09	-8.61E+09	-5205762	-1.13E+09	3699592.5	945359378	78202204	-94283560	264297845	42292934	0	
-2.1E+10	447095556	-2.78E+09	-5205762	-8.95E+09	-1.43E+09	136023583	-2.76E+09	-2.44E+08	264297845	-7.59E+08	6790044.1	0	
-3.4E+09	2.721E+09	-26467779	-1.13E+09	-1.43E+09	-3.78E+08	-9.92E+08	353430161	132398780	42292934	6790044.1	-8.67E+08	0	
10608654	1.181E+10	649171900	-4.49E+09	-819619.6	89283996	-343669.6	1.285E+09	-16354313	-1.27E+08	392772218	-70823224	0	
-1.2E+10	-10585567	192687742	-819619.6	-4.42E+09	-2.4E+08	-2.25E+08	-3.99E+09	62350316	392772218	-1.25E+09	-19685719	0	
-6.3E+08	-2.36E+08	-2451190	89283996	-2.4E+08	-14846439	-1.26E+09	540687652	57353683	-70823224	-19685719	-1.37E+09	0	
-9775430	1.218E+10	813657501	-4.77E+09	-1145651	-40587338	2683269	1.278E+09	15916883	-1.23E+08	386688224	-32069167	0	
-1.2E+10	9731892.3	-1.42E+08	-1145651	-4.71E+09	-3.11E+08	-1.27E+08	-4.01E+09	-44860048	386688224	-1.23E+09	-23627870	0	
-8E+08	103992416	-2499734	-40587338	-3.11E+08	-20959724	-1.31E+09	514364761	62803242	-32069167	-23627870	-1.39E+09	0	
-2091947	1.667E+10	-1.28E+08	-6.5E+09	244896.53	127569823	118209.4	2.251E+09	1203373.8	-1.87E+08	714163272	-94363721	0	
-1.7E+10	1880016.3	271285491	244896.53	-6.49E+09	53956866	-2.96E+08	-8.58E+09	-50040374	714163272	-2.77E+09	-24888789	0	
1.38E+08	-3.27E+08	255403.32	127569823	53956866	-2953781	-2.23E+09	1.15E+09	-3.43E+08	-94363721	-24888789	-2.95E+09	0	
-493649	1.721E+10	24478281	-6.91E+09	-159170.8	-49912837	-925461.1	3.538E+09	102022995	-2.89E+08	1.111E+09	-38778953	0	
-1.7E+10	571123.77	-1.79E+08	-159170.8	-6.81E+09	-5309605	-1.17E+08	-1.36E+10	-4.1E+08	1.111E+09	-4.28E+09	-5127772	0	
-1.3E+07	124391078	36961.922	-49912837	-5309605	-364898	-3.55E+09	1.834E+09	-5.17E+08	-38778953	-5127772	-4.57E+09	0	
-1.6E+08	2.097E+10	-1.46E+09	-7.36E+09	2252333	815512647	-2607561	1.956E+09	-2.46E+08	-1.9E+08	643344621	-3.92E+08	0	
-2.1E+10	154821501	2.225E+09	2252333	-7.36E+09	521929204	-1.14E+09	-8.43E+09	724630472	643344621	-3.06E+09	-1.38E+08	0	
1.49E+09	-2.33E+09	3585556.5	815512647	521929204	-1.27E+08	-1.91E+09	1.027E+09	-6.84E+08	-3.92E+08	-1.38E+08	-3.24E+09	0	
-2.1E+07	2.956E+10	163005381	-1.33E+10	-1046727	-1.76E+09	-1012263	4.272E+09	644852225	-3.55E+08	1.302E+09	471281628	0	
-3E+10	24439852	-4.04E+09	-1046727	-1.37E+10	-68058187	1.678E+09	-1.87E+10	-2.3E+09	1.302E+09	-5.46E+09	151612400	0	
-1.5E+08	3.91E+09	1477774.9	-1.76E+09	-68058187	-2.33E+08	-4.61E+09	2.664E+09	-1.05E+09	471281628	151612400	-5.81E+09	0	

-6.1E-10	5.0E-10	-4.2E-10	2.2E-09	-4.3E-11	-6.6E-09	-2.8E-09	-6.7E-10	-2.0E-09	9.5E-09	3.0E-09	3.9E-09	1.9E-09
5.0E-10	-8.6E-09	-3.3E-08	-3.3E-08	4.9E-09	1.1E-07	2.3E-08	4.3E-09	-1.1E-08	-7.1E-08	-2.4E-08	-2.9E-08	-1.6E-08
-4.2E-10	-3.3E-08	-1.6E-07	-1.5E-07	6.6E-09	4.7E-07	1.2E-07	1.4E-08	-3.8E-08	-3.0E-07	-1.1E-07	-1.5E-07	-8.5E-08
2.2E-09	-3.3E-08	-1.5E-07	-1.4E-07	2.2E-08	4.8E-07	9.5E-08	2.0E-08	-6.0E-08	-3.3E-07	-1.1E-07	-1.3E-07	-7.8E-08
-4.3E-11	4.9E-09	6.6E-09	2.2E-08	-9.4E-09	-6.6E-08	-1.5E-06	-4.6E-09	-1.3E-08	7.4E-08	2.4E-08	3.3E-08	1.6E-08
-6.6E-09	1.1E-07	4.7E-07	4.8E-07	-6.6E-08	-1.5E-06	-2.8E-07	-6.5E-08	2.0E-07	1.0E-06	3.4E-07	3.8E-07	2.8E-07
2.8E-09	2.3E-08	1.2E-07	9.5E-08	-2.4E-08	-2.8E-07	-1.4E-07	-1.6E-08	-6.3E-09	2.0E-07	7.3E-08	2.3E-07	5.2E-08
-6.7E-10	4.3E-09	4.6E-09	4.6E-09	-4.6E-09	-6.5E-08	-1.6E-08	-1.9E-09	-1.2E-09	6.0E-08	1.4E-08	2.1E-08	1.4E-08
2.0E-09	-1.1E-08	-3.8E-08	-6.0E-08	-1.3E-08	2.0E-07	-6.3E-09	-1.2E-09	-3.4E-08	-1.3E-07	-3.8E-08	5.9E-08	-3.8E-08
9.5E-09	-7.1E-08	-3.0E-07	-3.3E-07	7.4E-08	1.0E-06	2.0E-07	6.0E-08	-1.3E-07	-6.9E-07	-2.4E-07	-2.5E-07	-2.0E-07
3.0E-09	-2.4E-08	-1.1E-07	-1.1E-07	2.4E-08	3.4E-07	7.3E-07	1.4E-08	-3.8E-08	-2.4E-07	-6.5E-08	-9.3E-08	-6.5E-08
3.9E-09	-2.9E-08	-1.5E-07	-1.3E-07	3.3E-08	3.8E-07	2.3E-07	2.1E-08	5.9E-08	-2.5E-07	-9.3E-08	-3.0E-07	-7.0E-08
1.9E-09	-1.6E-08	-8.5E-08	-7.8E-08	1.6E-08	2.8E-07	5.2E-08	1.4E-08	-3.8E-08	-2.0E-07	-6.5E-08	-7.0E-08	-5.2E-08
-2.2E-09	2.3E-08	8.2E-08	1.0E-07	-1.6E-08	-3.2E-07	-9.1E-08	-1.5E-08	5.9E-08	2.6E-07	8.6E-08	1.2E-07	6.5E-08
3.2E-09	-1.9E-08	-7.8E-08	-8.6E-08	2.5E-08	2.6E-07	1.3E-07	1.5E-08	4.1E-08	-1.6E-07	-5.9E-08	-1.4E-07	-5.0E-08
2.4E-10	1.3E-08	4.7E-08	5.7E-08	-1.3E-09	-1.3E-07	-4.9E-08	-3.7E-09	-2.4E-09	8.4E-08	3.0E-08	5.3E-08	2.7E-08
-2.8E-09	1.3E-08	3.1E-08	5.9E-08	-1.9E-08	-1.8E-07	-5.0E-08	-1.2E-08	-7.3E-09	1.7E-07	5.4E-08	6.5E-08	3.9E-08
7.5E-10	1.5E-09	8.2E-09	4.9E-09	5.2E-09	-1.4E-08	6.1E-08	4.9E-11	6.4E-08	2.2E-08	4.9E-09	-4.2E-08	2.0E-09
1.2E-09	-8.2E-09	-5.0E-08	-4.1E-08	9.7E-09	1.7E-07	2.5E-08	8.1E-09	-2.8E-08	-1.2E-07	-3.9E-08	-3.6E-08	-3.0E-08
-2.1E-09	1.4E-08	4.0E-08	6.3E-08	-1.4E-08	-1.9E-07	-5.8E-08	-1.1E-08	-8.8E-09	1.7E-07	5.7E-08	7.6E-08	4.2E-08
2.2E-09	-1.3E-08	-5.7E-08	-6.1E-08	1.7E-08	1.9E-07	1.2E-07	1.0E-08	4.9E-08	-1.2E-07	-4.2E-08	-1.2E-07	-3.6E-08
9.6E-10	-4.5E-09	-3.3E-08	-2.4E-08	7.6E-09	1.1E-07	1.2E-08	5.9E-09	-2.4E-08	-8.3E-08	-2.7E-08	-2.1E-08	-2.0E-08
-2.2E-09	1.3E-08	3.4E-08	5.8E-08	-1.5E-08	-1.8E-07	-5.3E-08	-1.1E-08	-9.8E-09	1.6E-07	5.3E-08	7.0E-08	3.9E-08
1.9E-09	-1.1E-08	-4.7E-08	-5.1E-08	1.5E-08	1.6E-07	1.1E-07	8.2E-09	5.2E-08	-9.4E-08	-3.5E-08	-1.1E-07	-3.0E-08
1.1E-09	-1.4E-08	-7.8E-08	-6.7E-08	1.1E-08	2.5E-07	4.6E-08	1.1E-08	-3.5E-08	-1.7E-07	-5.8E-08	-6.3E-08	-4.5E-08
-1.5E-09	6.4E-09	7.2E-09	2.9E-08	-9.1E-09	-8.7E-08	-3.2E-08	-6.2E-09	-1.8E-08	1.0E-07	3.3E-08	4.4E-08	2.1E-08
2.1E-09	-1.7E-08	-7.5E-08	-7.8E-08	1.7E-08	2.4E-07	1.3E-07	1.1E-08	4.7E-08	-1.5E-07	-5.4E-08	-1.4E-07	-4.6E-08
9.4E-10	-1.1E-08	-6.2E-08	-5.1E-08	9.1E-09	2.0E-07	3.4E-08	8.6E-09	-3.1E-08	-1.4E-07	-4.6E-08	-4.8E-08	-3.5E-08
-1.6E-09	5.2E-09	1.0E-09	2.3E-08	-9.4E-09	-6.9E-08	-2.8E-08	-5.9E-09	-1.9E-08	9.0E-08	2.9E-08	3.8E-08	1.8E-08
1.9E-09	-1.5E-08	-6.7E-08	-6.9E-08	1.5E-08	2.1E-07	1.3E-07	9.5E-09	4.9E-08	-1.3E-07	-4.8E-08	-1.3E-07	-4.1E-08
1.8E-09	-3.0E-08	-1.5E-07	-1.4E-07	1.9E-08	4.7E-07	1.0E-07	2.0E-08	-5.4E-08	-3.2E-07	-1.1E-07	-1.3E-07	-8.8E-08
-9.8E-10	7.2E-09	1.4E-08	3.2E-08	-5.3E-09	-9.9E-08	-3.8E-08	-5.0E-09	-1.9E-08	1.1E-07	3.5E-08	5.1E-08	2.3E-08
3.2E-09	-2.6E-08	-1.1E-07	-1.2E-07	2.5E-08	3.6E-07	1.6E-07	1.7E-08	3.8E-08	-2.3E-07	-8.2E-08	-1.8E-07	-6.9E-08
1.7E-10	1.8E-09	-6.4E-09	6.3E-09	1.2E-09	2.4E-08	-9.0E-09	1.1E-09	-1.6E-08	-1.7E-08	-5.4E-09	4.6E-09	-1.1E-09
-1.5E-09	-3.2E-09	-3.9E-08	-1.5E-08	-8.0E-09	4.9E-08	3.4E-09	-2.3E-09	-2.8E-08	1.3E-08	1.6E-09	-6.1E-12	-3.1E-09
7.5E-10	-8.6E-09	-4.1E-08	-4.1E-08	6.8E-09	1.3E-07	1.0E-07	3.7E-09	5.7E-08	-7.8E-08	-2.8E-08	-9.6E-08	-2.4E-08

-2.2E-09	3.2E-09	2.4E-10	-2.8E-09	7.5E-10	1.2E-09	-2.1E-09	2.2E-09	9.6E-10	-2.2E-09	1.9E-09	1.1E-09	-1.5E-09	2.1E-09
2.3E-08	-1.9E-08	1.3E-08	1.3E-08	1.5E-09	-8.2E-09	1.4E-08	-1.3E-08	-4.5E-09	1.3E-08	-1.1E-08	-1.4E-08	6.4E-09	-1.7E-08
8.2E-08	-7.8E-08	4.7E-08	3.1E-08	8.2E-09	-5.0E-08	4.0E-08	-5.7E-08	-3.3E-08	3.4E-08	-4.7E-08	-7.8E-08	7.2E-09	-7.5E-08
1.0E-07	-8.6E-08	5.7E-08	5.9E-08	4.9E-09	-4.1E-08	6.3E-08	-6.1E-08	-2.4E-08	5.8E-08	-5.1E-08	-6.7E-08	2.9E-08	-7.8E-08
-1.6E-08	2.5E-08	-1.3E-09	-1.9E-08	5.2E-09	9.7E-09	-1.4E-08	1.7E-08	7.6E-09	-1.5E-08	1.5E-08	1.1E-08	-9.1E-09	1.7E-08
-3.2E-07	2.6E-07	-1.3E-07	-1.8E-07	-1.4E-08	1.7E-07	-1.9E-07	1.9E-07	1.1E-07	-1.8E-07	1.6E-07	2.5E-07	-8.7E-08	2.4E-07
-9.1E-08	1.3E-07	-4.9E-08	-5.0E-08	6.1E-08	2.5E-08	-5.8E-08	1.2E-07	1.2E-08	-5.3E-08	1.1E-07	4.6E-08	-3.2E-08	1.3E-07
-1.5E-08	1.5E-08	-3.7E-09	-1.2E-08	4.9E-11	8.1E-09	-1.1E-08	1.0E-08	5.9E-09	-1.1E-08	8.2E-09	1.1E-08	-6.2E-09	1.1E-08
5.9E-10	4.1E-08	-2.4E-09	-7.3E-09	6.4E-08	-2.8E-08	-8.8E-09	4.9E-08	-2.4E-08	-9.8E-09	5.2E-08	-3.5E-08	-1.8E-08	4.7E-08
2.6E-07	-1.6E-07	8.4E-08	1.7E-07	2.2E-08	-1.2E-07	1.7E-07	-1.2E-07	-8.3E-08	1.6E-07	-9.4E-08	-1.7E-07	1.0E-07	-1.5E-07
8.6E-08	-5.9E-08	3.0E-08	5.4E-08	4.9E-09	-3.9E-08	5.7E-08	-4.2E-08	-2.7E-08	5.3E-08	-3.5E-08	-5.8E-08	3.3E-08	-5.4E-08
1.2E-07	-1.4E-07	5.3E-08	6.5E-08	-4.2E-08	-3.6E-08	7.6E-08	-1.2E-07	-2.1E-08	7.0E-08	-1.1E-07	-6.3E-08	4.4E-08	-1.4E-07
6.5E-08	-5.0E-08	2.7E-08	3.9E-08	2.0E-09	-3.0E-08	4.2E-08	-3.6E-08	-2.0E-08	3.9E-08	-3.0E-08	-4.5E-08	2.1E-08	-4.6E-08
-6.8E-08	7.4E-08	-2.3E-08	-4.9E-08	3.2E-09	3.9E-08	-4.6E-08	5.2E-08	2.8E-08	-4.3E-08	4.4E-08	5.4E-08	-2.2E-08	6.1E-08
7.4E-08	-4.7E-08	2.0E-08	4.9E-08	1.2E-08	-3.0E-08	5.2E-08	-3.2E-08	-2.1E-08	4.9E-08	-2.5E-08	-4.4E-08	3.4E-08	-4.3E-08
-2.3E-08	2.0E-08	-7.7E-09	-7.3E-09	-5.9E-09	1.8E-08	-1.1E-08	1.4E-08	1.4E-08	-8.8E-09	1.1E-08	2.7E-08	-1.3E-09	2.0E-08
-4.9E-08	4.9E-08	-7.3E-09	-4.6E-08	4.4E-09	2.4E-08	-3.9E-08	3.3E-08	1.8E-08	-3.9E-08	2.8E-08	3.0E-08	-2.6E-08	3.6E-08
3.2E-09	1.2E-08	-5.9E-09	4.4E-09	1.8E-08	3.0E-10	5.4E-09	1.3E-08	-5.9E-10	5.6E-09	1.3E-08	1.7E-09	7.1E-09	1.2E-08
3.9E-08	-3.0E-08	1.8E-08	2.4E-08	3.0E-10	-1.6E-08	2.5E-08	-2.2E-08	-9.8E-09	2.3E-08	-1.8E-08	-2.4E-08	1.3E-08	-2.8E-08
-4.6E-08	5.2E-08	-1.1E-08	-3.9E-08	5.4E-09	2.5E-08	-3.4E-08	3.6E-08	1.9E-08	-3.3E-08	3.1E-08	3.3E-08	-1.9E-08	4.1E-08
5.2E-08	-3.2E-08	1.4E-08	3.3E-08	1.3E-08	-2.2E-08	3.6E-08	-2.0E-08	-1.5E-08	3.4E-08	-1.6E-08	-3.2E-08	2.3E-08	-2.9E-08
2.8E-08	-2.1E-08	1.4E-08	1.8E-08	-5.9E-10	-9.8E-09	1.9E-08	-1.5E-08	-5.3E-09	1.7E-08	-1.3E-08	-1.5E-08	1.0E-08	-1.9E-08
-4.3E-08	4.9E-08	-8.8E-09	-3.9E-08	5.6E-09	2.3E-08	-3.3E-08	3.4E-08	1.7E-08	-3.3E-08	2.9E-08	3.0E-08	-2.0E-08	3.8E-08
4.4E-08	-2.5E-08	1.1E-08	2.8E-08	1.3E-08	-1.8E-08	3.1E-08	-1.6E-08	-1.3E-08	2.9E-08	-1.1E-08	-2.7E-08	2.0E-08	-2.3E-08
5.4E-08	-4.4E-08	2.7E-08	3.0E-08	1.7E-09	-2.4E-08	3.3E-08	-3.2E-08	-1.5E-08	3.0E-08	-2.7E-08	-3.8E-08	1.5E-08	-4.1E-08
-2.2E-08	3.4E-08	-1.3E-09	-2.6E-08	7.1E-09	1.3E-08	-1.9E-08	2.3E-08	1.0E-08	-2.0E-08	2.0E-08	1.5E-08	-1.2E-08	2.3E-08
6.1E-08	-4.3E-08	2.0E-08	3.6E-08	1.2E-08	-2.8E-08	4.1E-08	-2.9E-08	-1.9E-08	3.8E-08	-2.3E-08	-4.1E-08	2.3E-08	-3.9E-08
4.3E-08	-3.6E-08	2.3E-08	2.5E-08	9.2E-10	-1.9E-08	2.6E-08	-2.6E-08	-1.1E-08	2.4E-08	-2.2E-08	-3.0E-08	1.2E-08	-3.3E-08
-2.0E-08	3.1E-08	5.9E-10	-2.5E-08	7.3E-09	1.1E-08	-1.9E-08	2.1E-08	9.0E-09	-1.9E-08	1.8E-08	1.2E-08	-1.3E-08	2.0E-08
5.4E-08	-3.8E-08	1.7E-08	3.2E-08	1.3E-08	-2.5E-08	3.6E-08	-2.5E-08	-1.7E-08	3.3E-08	-1.9E-08	-3.7E-08	2.0E-08	-3.4E-08
1.0E-07	-8.4E-08	4.7E-08	5.4E-08	4.9E-09	-5.0E-08	6.0E-08	-6.1E-08	-3.3E-08	5.4E-08	-5.0E-08	-7.7E-08	2.5E-08	-7.8E-08
-1.9E-08	3.5E-08	-3.7E-09	-2.1E-08	6.7E-09	1.4E-08	-1.5E-08	2.4E-08	1.0E-08	-1.5E-08	2.1E-08	1.6E-08	-7.3E-09	2.5E-08
9.1E-08	-7.0E-08	3.1E-08	5.5E-08	9.5E-09	-4.1E-08	6.0E-08	-5.0E-08	-2.8E-08	5.6E-08	-4.0E-08	-6.1E-08	3.4E-08	-6.5E-08
5.3E-09	-5.6E-09	8.0E-09	3.8E-09	-2.5E-09	2.0E-09	3.5E-09	-4.7E-09	3.1E-09	3.3E-09	-4.3E-09	1.1E-09	1.7E-09	-5.2E-09
1.2E-09	1.0E-08	1.2E-08	-1.7E-08	9.1E-09	-1.4E-09	-8.2E-09	6.1E-09	6.2E-10	-1.0E-08	5.8E-09	-8.0E-09	-1.1E-08	1.0E-09
2.9E-08	-1.8E-08	9.3E-09	1.5E-08	1.6E-08	-1.5E-08	1.8E-08	-9.1E-09	-1.1E-08	1.6E-08	-5.2E-09	-2.2E-08	9.1E-09	-1.5E-08

Menghitung Koreksi Parameter $\delta = N^{-1} * t$

9.4E-10	-1.6E-09	1.9E-09	1.8E-09	-9.8E-10	3.2E-09	1.7E-10	-1.5E-09	7.5E-10	-5.8E-06
-1.1E-08	5.2E-09	-1.5E-08	-3.0E-08	7.2E-09	-2.6E-08	1.8E-09	-3.2E-09	-8.6E-09	-1.8E-04
-6.2E-08	1.0E-09	-6.7E-08	-1.5E-07	1.4E-08	-1.1E-07	-6.4E-09	-3.9E-08	-4.1E-08	-1.0E-03
-5.1E-08	2.3E-08	-6.9E-08	-1.4E-07	3.2E-08	-1.2E-07	6.3E-09	-1.5E-08	-4.1E-08	-9.0E-04
9.1E-09	-9.4E-09	1.5E-08	1.9E-08	-5.3E-09	2.5E-08	1.2E-09	-8.0E-09	6.8E-09	6.5E-05
2.0E-07	-6.9E-08	2.1E-07	4.7E-07	-9.9E-08	3.6E-07	2.4E-08	4.9E-08	1.3E-07	2.7E-03
3.4E-08	-2.8E-08	1.3E-07	1.0E-07	-3.8E-08	1.6E-07	-9.0E-09	3.4E-09	1.0E-07	8.6E-04
8.6E-09	-5.9E-09	9.5E-09	2.0E-08	-5.0E-09	1.7E-08	1.1E-09	-2.3E-09	3.7E-09	8.6E-05
-3.1E-08	-1.9E-08	4.9E-08	-5.4E-08	-1.9E-08	3.8E-08	1.6E-08	-2.8E-08	5.7E-08	-3.9E-05
-1.4E-07	9.0E-08	-1.3E-07	-3.2E-07	1.1E-07	-2.3E-07	-1.7E-08	1.3E-08	-7.8E-08	-1.5E-03
-4.6E-08	2.9E-08	-4.8E-08	-1.1E-07	3.5E-08	-8.2E-08	-5.4E-09	1.6E-09	-2.8E-08	-5.6E-04
-4.8E-08	3.8E-08	-1.3E-07	-1.3E-07	5.1E-08	-1.8E-07	4.6E-09	-6.1E-12	-9.6E-08	-1.1E-03
-3.5E-08	1.8E-08	-4.1E-08	-8.8E-08	2.3E-08	-6.9E-08	-1.1E-09	-3.1E-09	-2.4E-08	-4.7E-04
4.3E-08	-2.0E-08	5.4E-08	1.0E-07	-1.9E-08	9.1E-08	5.3E-09	1.2E-09	2.9E-08	6.5E-04
-3.6E-08	3.1E-08	-3.8E-08	-8.4E-08	3.5E-08	-7.0E-08	-5.6E-09	1.0E-08	-1.8E-08	-6.4E-04
2.3E-08	5.9E-10	1.7E-08	4.7E-08	-3.7E-09	3.1E-08	8.0E-09	1.2E-08	9.3E-09	3.6E-04
2.5E-08	-2.5E-08	3.2E-08	5.4E-08	-2.1E-08	5.5E-08	3.8E-09	-1.7E-08	1.5E-08	2.4E-04
9.2E-10	7.3E-09	1.3E-08	4.9E-09	6.7E-09	9.5E-09	-2.5E-09	9.1E-09	1.6E-08	-1.6E-04
-1.9E-08	1.1E-08	-2.5E-08	-5.0E-08	1.4E-08	-4.1E-08	2.0E-09	-1.4E-09	-1.5E-08	-2.6E-04
2.7E-08	-1.9E-08	3.6E-08	6.0E-08	-1.5E-08	6.0E-08	3.5E-09	-8.2E-09	1.8E-08	3.5E-04
-2.6E-08	2.1E-08	-2.5E-08	-6.1E-08	2.4E-08	-5.0E-08	-4.7E-09	6.1E-09	-9.1E-09	-5.5E-04
-1.1E-08	9.0E-09	-1.7E-08	-3.3E-08	1.0E-08	-2.8E-08	3.1E-09	6.2E-10	-1.1E-08	-1.6E-04
2.4E-08	-1.9E-08	3.3E-08	5.4E-08	-1.5E-08	5.6E-08	3.3E-09	-1.0E-08	1.6E-08	3.1E-04
-2.2E-08	1.8E-08	-1.9E-08	-5.0E-08	2.1E-08	-4.0E-08	-4.3E-09	5.8E-09	-5.2E-09	-5.0E-04
-3.0E-08	1.2E-08	-3.7E-08	-7.7E-08	1.6E-08	-6.1E-08	1.1E-09	-8.0E-09	-2.2E-08	-4.6E-04
1.2E-08	-1.3E-08	2.0E-08	2.5E-08	-7.3E-09	3.4E-08	1.7E-09	-1.1E-08	9.1E-09	1.5E-04
-3.3E-08	2.0E-08	-3.4E-08	-7.8E-08	2.5E-08	-6.5E-08	-5.2E-09	1.0E-09	-1.5E-08	-7.0E-04
-2.3E-08	9.7E-09	-3.0E-08	-6.1E-08	1.3E-08	-4.9E-08	2.2E-09	-6.0E-09	-1.8E-08	-3.6E-04
9.7E-09	-1.3E-08	1.8E-08	2.0E-08	-7.4E-09	3.0E-08	1.5E-09	-1.3E-08	7.5E-09	9.7E-05
-3.0E-08	1.8E-08	-2.9E-08	-7.0E-08	2.2E-08	-5.7E-08	-4.9E-09	6.1E-10	-1.2E-08	-6.5E-04
-6.1E-08	2.0E-08	-7.0E-08	-1.5E-07	3.0E-08	-1.2E-07	-3.3E-09	-1.8E-08	-4.2E-08	-9.3E-04
1.3E-08	-7.4E-09	2.2E-08	3.0E-08	-2.9E-09	3.7E-08	1.5E-09	-3.9E-09	1.0E-08	2.3E-04
-4.9E-08	3.0E-08	-5.7E-08	-1.2E-07	3.7E-08	-1.0E-07	-6.4E-09	8.9E-10	-3.0E-08	-9.1E-04
2.2E-09	1.5E-09	-4.9E-09	-3.3E-09	1.5E-09	-6.4E-09	6.2E-09	-1.3E-10	-4.0E-09	-3.5E-06
-6.0E-09	-1.3E-08	6.1E-10	-1.8E-08	-3.9E-09	8.9E-10	-1.3E-10	-2.2E-08	-2.9E-09	-1.9E-04
-1.8E-08	7.5E-09	-1.2E-08	-4.2E-08	1.0E-08	-3.0E-08	-4.0E-09	-2.9E-09	1.1E-09	-5.3E-04

Hasil iterasi terakhir untuk masing-masing parameter EO dan Koordinat Titik Objek serta Presisi masing-masing titik Sebagai berikut :
 iterasi ke 9

Perhitungan Excel						
parameter EO						
photo	$\hat{\omega}$	$\hat{\phi}$	$\hat{\kappa}$	\hat{X}_L	\hat{Y}_L	\hat{Z}_L
1	-0.0368	0.1919	0.0546	0.0083	0.0017	-0.0075
2	82.3571	-73.2044	99.1786	-2.9695	-0.7747	-2.4770

Point	Koordinat			Presisi		
	\hat{X}	\hat{Y}	\hat{Z}	σ_X	σ_Y	σ_Z
1	-0.30413	0.15535	-2.95277	0.000019	0.000016	0.000023
2	0.31216	0.36909	-2.30718	0.000012	0.000011	0.000034
3	-0.04347	0.14498	-2.64125	0.000015	0.000013	0.000027
4	0.03066	0.17080	-2.56169	0.000014	0.000013	0.000027
5	-0.04168	-0.01952	-2.57646	0.000012	0.000010	0.000018
6	0.02682	0.00369	-2.50287	0.000012	0.000010	0.000015
7	-0.30337	-0.19764	-2.82425	0.000012	0.000010	0.000017
8	0.29750	0.01230	-2.19093	0.000009	0.000007	0.000013

sd 0.000017

Lampiran B

Source Code Bundle Adjustment Pada Console

Application Dalam Bahasa Pemograman

Visual Studio C#

Source Code Bundle Adjustment Pada Bahasa Pemrograman C#

B.1 *Class BA*

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.IO;
using System.Text;

namespace BAcs
{
    public class BA
    {
        private const string FILE_NAME_POINT = "8utaral.txt";
        private const string saveproject = "utaral.txt";
        public static void Main(string[] args)
        { Bundle bundle=newBundle(FILE_NAME_POINT,saveproject);
          Console.WriteLine(bundle.ToString());
        }
    }
}
```

B.2 *Struct Photo Pair EO*

```
namespace BAcs
{
    public struct PhotoPairCams
    {
        string _labelc; double _omega; double _phi;
        double _kappa; double _xL; double _yL; double _zL;
        public PhotoPairCams(string lblc,double om,double pi, double
            kp,double xL,double yL,double zL)
        {
            _labelc = lblc; _omega = om; _phi = pi;
            _kappa = kp; _xL = xL; _yL = yL; _zL = zL;
        }
        public string Labelc
        { get { return _labelc; }
          set { _labelc = value; }
        }
        public double Omega
        { get { return _omega; }
          set { if (_omega != value)
              _omega = value; }
        }
        public double Phi
        { get { return _phi; }
          set { if (_phi != value)
              _phi = value; }
        }
        public double Kappa
        { get { return _kappa; }
          set { if (_kappa != value)
              _kappa = value; }
        }
    }
}
```



```

    }
    public double XL
    { get { return _xL; }
      set { if (_xL != value)
            _xL = value; }
    }
    public double YL
    { get { return _yL; }
      set { if (_yL != value)
            _yL = value; }
    }
    public double ZL
    { get { return _zL; }
      set { if (_zL != value)
            _zL = value; }
    }

    public override string ToString()
    { string cams=String.Format("{0}{1}{2}{3}{4}{5}{6}", _labelc,
      _omega, _phi, _kappa, _xL, _yL, _zL);
      return cams; }
    }
}

```

B.3 Struct Photo Pair Point

```

namespace BAcs
{
    public struct PhotoPairPoint
    {
        string _labelp; double _Xc; double _Yc; double _Zc;
        double _x1; double _y1; double _x2; double _y2;
        public PhotoPairPoint(string lblp, double xc, double yc,
            double zc, double x1, double y1, double x2, double y2)
        {
            _labelp = lblp; _Xc = xc; _Yc = yc; _Zc = zc;
            _x1 = x1; _y1 = y1; _x2 = x2; _y2 = y2;
        }
        public string Labelp
        { get { return _labelp; }
          set { _labelp = value; } }
        public double Xc
        { get { return _Xc; }
          set { if (_Xc != value) _Xc = value; } }
        public double Yc
        { get { return _Yc; }
          set { if (_Yc != value) _Yc = value; } }
        public double Zc
        { get { return _Zc; }
          set { if (_Zc != value) _Zc = value; } }
        public double X1
        { get { return _x1; }
          set { if (_x1 != value) _x1 = value; } }
        public double Y1
        { get { return _y1; }
          set { if (_y1 != value) _y1 = value; } }
        public double X2
        { get { return _x2; }

```

```

        set { if (_x2 != value) _x2 = value; } }
    public double Y2
    { get { return _y2; }
      set { if (_y2 != value) _y2 = value; } }

    public override string ToString()
    { string points = String.Format("{0}{1}{2}{3}{4}{5}{6}{7}",
      _labelp, _Xc, _Yc, _Zc, _x1, _y1, _x2, _y2);
      return points;
    }
}
}

```

B.4 Struct Photo Pair Residu

```

namespace BAcS
{
    public struct PhotoPairResidu
    {
        string _labelr; double _xresidu1; double _yresidu1;
        double _xresidu2; double _yresidu2;
        public PhotoPairResidu(string lblr, double xresidu1, double
        yresidu1, double xresidu2, double yresidu2)
        {
            _labelr = lblr; _xresidu1 = xresidu1;
            _yresidu1 = yresidu1; _xresidu2 = xresidu2;
            _yresidu2 = yresidu2;
        }
        public string Labelr
        { get { return _labelr; }
          set { _labelr = value; } }
        public double Xresidu1
        { get { return _xresidu1; }
          set { if (_xresidu1 != value)
            _xresidu1 = value; }
        }
        public double Yresidu1
        { get { return _yresidu1; }
          set { if (_yresidu1 != value)
            _yresidu1 = value; }
        }
        public double Xresidu2
        { get { return _xresidu2; }
          set { if (_xresidu2 != value)
            _xresidu2 = value; }
        }
        public double Yresidu2
        { get { return _yresidu2; }
          set { if (_yresidu2 != value)
            _yresidu2 = value; }
        }
        public override string ToString()
        { string residu = String.Format("{0}{1}{2}{3}{4}", _labelr,
          _xresidu1, _yresidu1, _xresidu2, _yresidu2);
          return residu;
        }
    }
}
}

```

B.5 Class Proses Bundle Adjustment

```
using System;
using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.Linq;
using System.Text;
using System.IO;
namespace BAcs
{
    public class Bundle
    {
        private int _imageNumber; private int _pointNumber;
        private int _residuNumber; private double _focus;
        private double _xo; private double _yo;

        Collection<CSML.Matrix> Nicol = new Collection<CSML.Matrix>();
        Collection<CSML.Matrix> Njcol = new Collection<CSML.Matrix>();
        Collection<CSML.Matrix> ticol = new Collection<CSML.Matrix>();
        Collection<CSML.Matrix> tjcol = new Collection<CSML.Matrix>();
        Collection<CSML.Matrix> Nijcol = new Collection<CSML.Matrix>();
        Collection<CSML.Matrix> Njicol = new Collection<CSML.Matrix>();
        public Collection<CSML.Matrix> EOparam = new
            Collection<CSML.Matrix> ();
        public Collection<CSML.Matrix> pointcol = new
            Collection<CSML.Matrix>();

        Collection<Collection<CSML.Matrix>> nicol2 = new
            Collection<Collection<CSML.Matrix>>();
        Collection<Collection<CSML.Matrix>> njcol2 = new
            Collection<Collection<CSML.Matrix>>();
        Collection<Collection<CSML.Matrix>> ticol2 = new
            Collection<Collection<CSML.Matrix>>();
        Collection<Collection<CSML.Matrix>> tjcol2 = new
            Collection<Collection<CSML.Matrix>>();
        Collection<Collection<CSML.Matrix>> nijcol2 = new
            Collection<Collection<CSML.Matrix>>();
        Collection<Collection<CSML.Matrix>> njicol2 = new
            Collection<Collection<CSML.Matrix>>();

        List<PhotoPairPoint> _photopairlistpoint;
        List<PhotoPairCams> _photopairlistcams;
        List<PhotoPairResidu> _photopairlistresidu;

        public Bundle()
        {
            _imageNumber = 0; _focus = 0; _xo = 0; _yo = 0;
            _pointNumber = 0; _residuNumber = 0;
            _photopairlistcams = new List<PhotoPairCams>();
            _photopairlistpoint = new List<PhotoPairPoint>();
            _photopairlistresidu = new List<PhotoPairResidu>();
        }
        public Bundle(int pointc,int pointp,int pointr, double f, double
            xo,double yo)
        {
            _imageNumber = pointc; _pointNumber = pointp;
            _residuNumber = pointr; _focus = f; _xo = xo;_yo = yo;
            _photopairlistcams = new List<PhotoPairCams>();
        }
    }
}
```



```

    _photopairlistpoint = new List<PhotoPairPoint>();
    _photopairlistresidu = new List<PhotoPairResidu>();
}
public void listcollection(CSML.Matrix Nilist, CSML.Matrix
    Njlist, CSML.Matrix Tjlist, CSML.Matrix Tilist, CSML.Matrix
    NIJlist, CSML.Matrix NJIlist)
{
    //tjcol.Add(tj);
    Nicol.Add(Nilist); foreach (CSML.Matrix a in Nicol){ }
}
public void listcollection2(Collection<CSML.Matrix> Nicol2,
    Collection<CSML.Matrix> Njcol2, Collection<CSML.Matrix>
    Tjcol2, Collection<CSML.Matrix> Ticol2,
    Collection<CSML.Matrix> Nijcol2, Collection<CSML.Matrix>
    Njicol2)
{
    nicol2.Add(Nicol2); foreach (Collection<CSML.Matrix> a in
    nicol2) { }
}
private void EOparamater(CSML.Matrix EO)
{ EOparam.Add(EO);
    foreach (CSML.Matrix f in EOparam) { }
}
private void Pointlist(CSML.Matrix point)
{ pointcol.Add(point);
    foreach (CSML.Matrix f in pointcol) { }
}
private void X1Y1X2Y2(CSML.Matrix XY)
{ X1Y1.Add(XY);
    foreach (CSML.Matrix f in pointcol) { }
}
private void Residual(CSML.Matrix res)
{ residu.Add(res);
    foreach (CSML.Matrix f in residu) { }
}

public Bundle(string fileName, string savedata)
{
    if (!File.Exists(fileName))
    {
        Console.WriteLine("{0} does not exist.", fileName);
        return;
    }
    _photopairlistpoint = new List<PhotoPairPoint>();
    _photopairlistcams = new List<PhotoPairCams>();
    _photopairlistresidu = new List<PhotoPairResidu>();
    Collection<CSML.Matrix> Nisheet = new Collection<CSML.Matrix>();
    Collection<CSML.Matrix> Tisheet = new Collection<CSML.Matrix>();
    ReadFile(ref fileName);
    double[,] p_to_m = new double[_pointNumber, 2 * _imageNumber];
    PointtoMatrix(ref p_to_m);
    double[,] r_to_m = new double[_residuNumber, 2 * _imageNumber];
    RetoMatrix(ref r_to_m);
    for (int i = 0; i < ImageNumber; i++)
    {
        int l = i + 1;
        System.Diagnostics.Trace.WriteLine("\nImage : " + l.ToString());
        double omega = _photopairlistcams.ElementAt(i).Omega;
        System.Diagnostics.Trace.WriteLine("omega :"+ omega.ToString());
    }
}

```

```

double phi = _photopairlistcams.ElementAt(i).Phi;
double kappa = _photopairlistcams.ElementAt(i).Kappa;
double xl = _photopairlistcams.ElementAt(i).XL;
double yl = _photopairlistcams.ElementAt(i).YL;
double zl = _photopairlistcams.ElementAt(i).ZL;
double focus = _focus;
//CSML.Matrix Nii = new CSML.Matrix(6, 6);
CSML.Matrix Ni = new CSML.Matrix(6, 6);
CSML.Matrix Nj = new CSML.Matrix(3, 3);
CSML.Matrix ti = new CSML.Matrix(6, 1);
CSML.Matrix tj = new CSML.Matrix(3, 1);
CSML.Matrix Nij = new CSML.Matrix(6, 3);
CSML.Matrix Nji = new CSML.Matrix(3, 6);
//Nijcol.Count<i>;
double[] Parameter = new double[6] {omega,phi,kappa,xl,yl,zl};
CSML.Matrix param = new CSML.Matrix(Parameter);
CSML.Matrix param1 = param.Transpose();
PrintVectorDebug(ref Parameter, "param");
for (int j = 0; j < PointNumber; j++)
{
int k = j + 1;
System.Diagnostics.Trace.WriteLine("Point : " + k.ToString());
double[,] w = new double[2, 2];
double[,] f = new double[2, 1];
double xc = _photopairlistpoint.ElementAt(j).Xc;
double yc = _photopairlistpoint.ElementAt(j).Yc;
double zc = _photopairlistpoint.ElementAt(j).Zc;
double[,]objectsp=new double[3,1] {{ xc},{yc},{zc} };
CSML.Matrix spacepo = new CSML.Matrix(objectsp);
CSML.Matrix spapoint = spacepo.Transpose();
Nisheet.Add(Ni); foreach (CSML.Matrix d in Nisheet){ }
Tisheet.Add(ti);
double[,] M = new double[3,3];
M = ProjectRotation(ref omega, ref phi, ref kappa);
PrintMatrixDebug(ref M, "Rotasi");
double[] rsq = new double[1];
rsq = ProjectUVW(ref omega, ref phi, ref kappa, ref xl, ref yl,
ref zl, ref xc, ref yc, ref zc);
//Matrix A1
double[,] a1 = new double[2, 6];
a1 = ProjectA1(ref omega, ref phi, ref kappa, ref xl, ref yl,
ref zl, ref xc, ref yc, ref zc, ref focus);
CSML.Matrix A1_csml = new CSML.Matrix(a1);
//Matrix A2
double[,] a2 = new double[2, 3];
a2 = ProjectA2(ref omega, ref phi, ref kappa, ref xl, ref yl,
ref zl, ref xc, ref yc, ref zc, ref focus);
CSML.Matrix A2_csml = new CSML.Matrix(a2);
//Mantrix f
f[0,0]=(p_to_m[j, (2*i)]-_xo)+(focus*(rsq[0]/rsq[2]));
f[1,0]=(p_to_m[j, (2*i)+1]-_yo)+(focus*(rsq[1]/ rsq[2]));
CSML.Matrix F_csml = new CSML.Matrix(f);
//PrintMatrixDebug(ref f, "ti");
//Matrix W(bobot)
w[0,0] = 1/(r_to_m[j, (2*i)]*r_to_m[j, (2*i)]);
w[0,1] = 0;
w[1,0] = 0;
w[1,1] = 1/(r_to_m[j, (2*i)+ 1]*r_to_m[j, (2*i)+1]);
CSML.Matrix W_csml = new CSML.Matrix(w);

```

```

Ni = Ni + A1_csml.Transpose() * W_csml * A1_csml;
System.Diagnostics.Trace.WriteLine("Ni : \n" + Ni.ToString());
//Matrix ti
ti = ti + A1_csml.Transpose() * W_csml * F_csml;
//Matrix Nj
Nj = A2_csml.Transpose() * W_csml * A2_csml;
System.Diagnostics.Trace.WriteLine("Nj : \n" + Nj.ToString());
//Matrix tj
tj = A2_csml.Transpose() * W_csml * F_csml;
System.Diagnostics.Trace.WriteLine("tj : \n" + tj.ToString());
//Matrik Nij
Nij = A1_csml.Transpose() * W_csml * A2_csml;
System.Diagnostics.Trace.WriteLine("Nij : \n" + Nij.ToString());
//Matrik Nji
Nji = Nij.Transpose();
System.Diagnostics.Trace.WriteLine("Nji : \n" + Nji.ToString());
listcollection(Ni, Nj, tj, ti, Nij, Nji);
listcollection2(Nicol, Njcol, tjcol, ticol, Nijcol, Njicol);
EOparamater(param1);
Pointlist(spapoint);
}
double[,] Cb = new double[6, 6] {{0,0,0,0,0,0}, {0,0,0,0,0,0},
    {0,0,0,0,0,0}, {0,0,0,0,0,0}, {0,0,0,0,0,0}, {0,0,0,0,0,0}};
double[,] Cc = new double[3, 6] {{0,0,0,0,0,0},
    {0,0,0,0,0,0}, {0,0,0,0,0,0}};
CSML.Matrix Cbc = new CSML.Matrix(Cb);
double[,] Cd = new double[3, 1] {{0}, {0}, {0}};
#region raisebundle
CSML.Matrix sigmaP11P21 = new CSML.Matrix();
#region matrik Nj
CSML.Matrix P11 = njcol2[0][0];

#endregion
CSML.Matrix NiP1 = Nisheet[7];
CSML.Matrix NiP2 = Nisheet[15];

CSML.Matrix NjilP1 = njicol2[0][0];
sigmaP11P21 = P11 + P21;
#region Pelengkap
double[,] compliment1 = new double[3,21]{{0,0,0,0,0,0, 0,0,0,0,
    0,0,0,0,0,0,0,0,0,0}, {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0}, {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}};
CSML.Matrix Ppell = new CSML.Matrix(pell);
CSML.Matrix jper1 = CSML.Matrix.HorizontalConcat
    ((CSML.Matrix.HorizontalConcat((CSML.Matrix.HorizontalConcat
    (NjilP1.Column(1),NjilP2.Column(1))), (CSML.Matrix.Horizonta
    lConcat(NjilP3.Column(1),NjilP4.Column(1))))), (CSML.Matrix.
    HorizontalConcat((CSML.Matrix.HorizontalConcat(NjilP5.Column
    (1),NjilP6.Column(1))), (CSML.Matrix.HorizontalConcat(NjilP7
    .Column(1), NjilP8.Column(1))))));

CSML.Matrix sper1 = CSML.Matrix.HorizontalConcat (vper1.
    Column(1), vper2.Column(1));

CSML.Matrix kper1 = CSML.Matrix.VerticalConcat ((CSML.Matrix.
    VerticalConcat((CSML.Matrix.VerticalConcat(sper1, sper2))
    , (CSML.Matrix.VerticalConcat ((CSML.Matrix. VerticalConcat
    (sper3, sper4)), (CSML.Matrix.VerticalConcat (sper5, sper6)))
    )), CSML.Matrix.VerticalConcat(sper7, sper8));

```



```

CSML.Matrix kper2 = CSML.Matrix.VerticalConcat
    ((CSML.Matrix.VerticalConcat((CSML.Matrix.VerticalConcat(sper
r9, sper10)), (CSML.Matrix.VerticalConcat
    ((CSML.Matrix.VerticalConcat(sper11, sper12)),
    (CSML.Matrix.VerticalConcat(sper13, sper14))))),
    CSML.Matrix.VerticalConcat(sper15, sper16));

CSML.Matrix koril = CSML.Matrix.HorizontalConcat
    (hori1.Column(1), hori2.Column(1));
CSML.Matrix kori2 = CSML.Matrix.HorizontalConcat
    (hori3.Column(1), hori4.Column(1));

CSML.Matrix bundleNjsip = CSML.Matrix.VerticalConcat
    ((CSML.Matrix.VerticalConcat(vori1, vori2)),
    (CSML.Matrix.VerticalConcat((CSML.Matrix.VerticalConcat(vori
3, vori4)), (CSML.Matrix.VerticalConcat (vori5, vori6)))));
#endregion

#region susun bundle
CSML.Matrix seril = CSML.Matrix.HorizontalConcat
    (bundleNisip.Column(1), bundleNjisip.Column(1));

CSML.Matrix Matrix_bundle = CSML.Matrix.VerticalConcat
    (bundleNi_NJi, bundleNij_Nj);
double[,] matrixbundledo=CSML2Array(ref Matrix_bundle);
double[,] matrikNJ = CSML2Array(ref bundleNjsip);
double[,] InvbundleNj =MatrixLibrary.Matrix.Inverse (matrikNJ);
CSML.Matrix tes = new CSML.Matrix(InvbundleNj);
CSML.Complex trace = tes.Trace();
CSML.Complex trace1 = trace/(3*8);
double[,] inversebundle = MatrixLibrary.Matrix.Inverse
    (matrixbundledo);
CSML.Matrix bundleinv =new CSML.Matrix(inversebundle);
System.Diagnostics.Trace.WriteLine("InvbundleNj :\n" +
    trace1.ToString());
#endregion

#region matric_L
CSML.Matrix Ti1 = Tisheet[7];
CSML.Matrix Ti2 = Tisheet[15];

CSML.Matrix matrix_L = CSML.Matrix.HorizontalConcat
    ((CSML.Matrix.HorizontalConcat((CSML.Matrix.HorizontalConcat
    ((CSML.Matrix.HorizontalConcat(Ti1.Column(1), Ti2.Column(1))
    ), (CSML.Matrix.HorizontalConcat (Tj1P1.Column(1), Tj1P2.
    Column(1))))), (CSML.Matrix.HorizontalConcat((CSML.Matrix.
    HorizontalConcat(Tj1P3.Column(1), Tj1P4.Column(1))), ((CSML.
    Matrix.HorizontalConcat(Tj1P5.Column(1), Tj1P6.Column(1))))))
    )), CSML.Matrix.HorizontalConcat(Tj1P7, Tj1P8));

#endregion
CSML.Matrix final = bundleinv * matrix_L;
#endregion

#region OUtput
CSML.Matrix koreksiEoimagel = CSML.Matrix.VerticalConcat((
    CSML.Matrix.VerticalConcat(final.Row(1), final.Row(2))),
    (CSML.Matrix.VerticalConcat((CSML.Matrix.VerticalConcat(fina

```

```

1.Row(3),final.Row(4))), (CSML.Matrix. VerticalConcat
(final.Row(5), final.Row(6)))));
CSML.Matrix koreksipoint1 = CSML.Matrix.VerticalConcat((
CSML.Matrix.VerticalConcat(final.Row(13), final.Row(14))),
final.Row(15));
CSML.Matrix EOfinalimage1=EOparam[0]+koreksiEoimage1;
CSML.Matrix EOfinalimage2=EOparam[8]+koreksiEoimage2;
CSML.Matrix I1P1_final = pointcol[0] + koreksipoint1;
CSML.Matrix I1P2_final = pointcol[1] + koreksipoint2;

double[,] bundle = CSML2Array(ref final);
double[,]EO_final_image1=CSML2Array(ref EOfinalimage1);
double[,]EO_final_image2=CSML2Array(ref EOfinalimage2);
double[,] ObjSpcI1P1 = CSML2Array(ref I1P1_final);
double[,] ObjSpcI1P2 = CSML2Array(ref I1P2_final);

using (StreamWriter sw = File.CreateText(save))
{
    PrintMatrixDebugsd(ref bundle,"Final bundle",sw);
    PrintMatrixDebugsd(ref EO_final_image1, "EO_final_image1",
        sw);
    PrintMatrixDebugsd(ref EO_final_image2, "EO_final_image2",
        sw);
    PrintMatrixDebugsd(ref ObjSpcI1P1,"ObjSpcI1P1", sw);
    PrintMatrixDebugsd(ref ObjSpcI1P2,"ObjSpcI1P2", sw);
    sw.WriteLine();
    sw.WriteLine("kovarian:" + tracel.ToString());
        sw.Close();
}
#endregion

System.Diagnostics.Trace.WriteLine("final
:\n"+final.ToString());
System.Diagnostics.Trace.WriteLine("koreksiEoimage1 :\n" +
    koreksiEoimage1.ToString());
System.Diagnostics.Trace.WriteLine("EOfinalimage1 :\n" +
    EOfinalimage1.ToString());
}
#region Properties
public int ImageNumber
public int PointNumber
public int ResiduNumber
public double Focal
public double Xo
public double Yo
public List<PhotoPairPoint> PhotoPairListPoint
public List<PhotoPairCams> PhotoPairListCams
public List<PhotoPairResidu> PhotoPairListResidu
#endregion

private double[,] PointtoMatrix(ref double[,] pp)
{
    int k = _imageNumber;
    for (int i = 0; i < _pointNumber; i++)
    {
        pp[i, 0]=_photopairlistpoint.ElementAt(i).X1;
        pp[i, 1]=_photopairlistpoint.ElementAt(i).Y1;
        pp[i, 2]=_photopairlistpoint.ElementAt(i).X2;
        pp[i, 3]=_photopairlistpoint.ElementAt(i).Y2;
    }
}

```



```

}
return pp;
}
private double[,] RetoMatrix(ref double[,] rr)
{
    int k = _imageNumber;
    for (int i = 0; i < _residuNumber; i++)
    {
        rr[i, 0]=_photopairlistresidu.ElementAt(i).Xresidu1;
        rr[i, 1]=_photopairlistresidu.ElementAt(i).Yresidu1;
        rr[i, 2]=_photopairlistresidu.ElementAt(i).Xresidu2;
        rr[i, 3]=_photopairlistresidu.ElementAt(i).Yresidu2;
    }
    return rr;
}

private double[,] ProjectA2(ref double Omega, ref double
    Phi, ref double Kappa, ref double XL, ref double YL,
    ref double ZL, ref double Xc, ref double Yc, ref double
    Zc, ref double foc)
{
    double[,] A2 = new double[2, 3];
    double[,] M_rot = new double[3, 3];
    M_rot = ProjectRotation(ref Omega, ref Phi, ref Kappa);
    double[] Rsq = new double[3];
    Rsq = ProjectUVW(ref Omega, ref Phi, ref Kappa, ref XL,
        ref YL, ref ZL, ref Xc, ref Yc, ref Zc);
    A2[0,0]=(foc/(Rsq[2]*Rsq[2]))*((Rsq[0]*M_rot[2,0])-
        (Rsq[2]*M_rot[0,0]));
    A2[0,1]=(foc/(Rsq[2]*Rsq[2]))*((Rsq[0]*M_rot[2,1])-
        (Rsq[2]*M_rot[0,1]));
    A2[0,2]=(foc/(Rsq[2]*Rsq[2]))*((Rsq[0]*M_rot[2,2])-
        (Rsq[2]*M_rot[0,2]));
    A2[1,0]=(foc/(Rsq[2]*Rsq[2]))*((Rsq[1]*M_rot[2,0])-
        (Rsq[2]*M_rot[1,0]));
    A2[1,1]=(foc/(Rsq[2]*Rsq[2]))*(Rsq[1]*M_rot[2,1])-
        (Rsq[2]*M_rot[1,1]));
    A2[1,2]=(foc/(Rsq[2]*Rsq[2]))*((Rsq[1]*M_rot[2,2])-
        (Rsq[2]*M_rot[1,2]));
    return A2;
}

private double[,] ProjectA1(ref double Omega, ref double
    Phi, ref double Kappa, ref double XL, ref double YL,
    ref double ZL, ref double Xc, ref double Yc, ref
    double Zc, ref double foc)
{
    double a = Math.PI / 180;
    double[,] A1 = new double[2, 6];
    double[,] M_rot = new double[3, 3];
    M_rot = ProjectRotation(ref Omega, ref Phi, ref Kappa);
    double[] Rsq = new double[3];
    Rsq = ProjectUVW(ref Omega, ref Phi, ref Kappa, ref XL,
        ref YL, ref ZL, ref Xc, ref Yc, ref Zc);
    A1[0,0]=(foc/(Rsq[2]*Rsq[2]))*((Rsq[0]*((-M_rot[2,2]*
        (Yc-YL))+M_rot[2,1]*(Zc-ZL)))-(Rsq[2]*((-
        M_rot[0,2]*(Yc-YL))+M_rot[0,1]*(Zc-ZL))));
    A1[0,1]=(foc/(Rsq[2]*Rsq[2]))*((Rsq[0]*((Math.Cos(Phi
        *a)*(Xc-XL))+(Math.Sin(Omega*a)*Math.Sin(Phi*a)*
        (Yc-YL))-(Math.Cos(Omega*a)*Math.Sin(Phi*a)*(Zc-

```

```

        ZL))) - Rsq[2] * ((-Math.Sin(Phi*a) * Math.Cos(Kappa*a) *
        (Xc-XL)) + (Math.Sin(Omega*a) * Math.Cos(Phi*a) *
        Math.Cos(Kappa*a) * (Yc-YL)) - (Math.Cos(Omega*a) *
        Math.Cos(Phi*a) * Math.Cos(Kappa * a) * (Zc-ZL)));
A1[0,2] = (-foc/Rsq[2]) * ((M_rot[1,0] * (Xc-
        XL)) + (M_rot[1,1] * (Yc -YL)) + (M_rot[1,2] * (Zc-ZL)));
A1[0,3] = (foc/(Rsq[2]*Rsq[2])) * ((Rsq[0]*M_rot[2,0]) -
        Rsq[2]*M_rot[0,0]);
A1[0,4] = -(foc/(Rsq[2]*Rsq[2])) * ((Rsq[0]*M_rot[2,1]) -
        (Rsq[2]*M_rot[0,1]));
A1[0,5] = -(foc/(Rsq[2]*Rsq[2])) * ((Rsq[0]*M_rot[2,2]) -
        (Rsq[2]*M_rot[0,2]));
A1[1,0] = (foc/(Rsq[2]*Rsq[2])) * (Rsq[1] * ((-M_rot[2,2] *
        (Yc-YL)) + (M_rot[2,1] * (Zc-ZL))) - Rsq[2] * ((-
        M_rot[1,2] * (Yc-YL)) + (M_rot[1,1] * (Zc-ZL))));
A1[1,1] = (foc/(Rsq[2]*Rsq[2])) * (Rsq[1] * ((Math.Cos(Phi
        *a) * (Xc-XL)) + (Math.Sin(Omega*a) * Math.Sin(Phi*a) *
        (Yc-YL)) - (Math.Cos(Omega*a) * Math.Sin(Phi*a) * (Zc-
        ZL))) - Rsq[2] * ((Math.Sin(Phi*a) * Math.Sin(Kappa*a) *
        (Xc-XL)) - (Math.Sin(Omega*a) * Math.Cos(Phi*a) *
        Math.Sin(Kappa*a) * (Yc-YL)) + (Math.Cos(Omega*a) *
        Math.Cos(Phi*a) * Math.Sin(Kappa * a) * (Zc-ZL))));
A1[1,2] = (foc/Rsq[2]) * ((M_rot[0,0] * (Xc-XL)) + (M_rot[0,1]
        * (Yc-YL)) + (M_rot[0,2] * (Zc - ZL)));
A1[1,3] = -(foc/(Rsq[2]*Rsq[2])) * ((Rsq[1]*M_rot[2,0]) -
        (Rsq[2]*M_rot[1,0]));
A1[1,4] = -(foc/(Rsq[2]*Rsq[2])) * ((Rsq[1]*M_rot[2,1]) -
        (Rsq[2]*M_rot[1,1]));
A1[1,5] = -(foc/(Rsq[2]*Rsq[2])) * ((Rsq[1]*M_rot[2,2]) -
        (Rsq[2]*M_rot[1,2]));
    return A1;
}
private double[] ProjectUVW(ref double Omega, ref double
    Phi, ref double Kappa, ref double XL, ref double YL,
    ref double ZL, ref double Xc, ref double Yc, ref
    double Zc)
{
    double[] rsq = new double[3];
    double[,] M_rot = new double[3, 3];
    M_rot = ProjectRotation(ref Omega, ref Phi, ref Kappa);
    rsq[0] = (M_rot[0,0] * (Xc-XL)) + (M_rot[0,1] * (Yc-YL)) +
        (M_rot[0,2] * (Zc-ZL));
    rsq[1] = (M_rot[1,0] * (Xc-XL)) + (M_rot[1,1] * (Yc-YL)) +
        (M_rot[1,2] * (Zc-ZL));
    rsq[2] = (M_rot[2,0] * (Xc-XL)) + (M_rot[2,1] * (Yc-YL)) +
        (M_rot[2,2] * (Zc-ZL));
    return rsq;
}
private double[,] ProjectRotation(ref double Omega, ref
    double Phi, ref double Kappa)
{
    double[,] M = new double[3, 3];
    double a = Math.PI / 180;
    M[0,0] = Math.Cos(Phi*a) * Math.Cos(Kappa*a);
    M[0,1] = Math.Cos(Omega*a) * Math.Sin(Kappa*a) + Math.Sin
        (Omega*a) * Math.Sin(Phi*a) * Math.Cos(Kappa*a);
    M[0,2] = Math.Sin(Omega*a) * Math.Sin(Kappa*a) - Math.Cos
        (Omega*a) * Math.Sin(Phi*a) * Math.Cos(Kappa*a);
    M[1,0] = -Math.Cos(Phi*a) * Math.Sin(Kappa*a);

```

```

M[1,1] = Math.Cos(Omega*a)*Math.Cos(Kappa*a)- Math.Sin
(Omega*a)*Math.Sin(Phi*a)*Math.Sin(Kappa*a);
M[1,2] = Math.Sin(Omega*a)*Math.Cos(Kappa*a)+Math.Cos
(Omega*a)*Math.Sin(Phi*a)*Math.Sin(Kappa*a);
M[2,0] = Math.Sin(Phi*a);
M[2,1] = -Math.Sin(Omega*a)*Math.Cos(Phi*a);
M[2,2] = Math.Cos(Omega*a)*Math.Cos(Phi*a);
return M;
}
private void ReadFile(ref string file)
{
    string[] readText = File.ReadAllLines(file);
    System.Diagnostics.Trace.WriteLine("readtext: " +
readText.Count());
    for (int i = 0; i < readText.Count(); i++)
    {
        if (i == 0)
        {
            String[] firstlineText = readText[0].Split(new
            char[] { ' ', '\t' });
            if (firstlineText.Length == 4)
            {
                ImageNumber = int.Parse(firstlineText[0]);
                Focal = Double.Parse(firstlineText[1]);
                Xo = Double.Parse(firstlineText[2]);
                Yo = Double.Parse(firstlineText[3]);
            }
        }
        else
        {
            String[] textc = readText[i].Split(new char[] {
            ' ', '\t' });
            if (textc.Length == 7)
            {
                PhotoPairCams pairc = new PhotoPairCams();
                pairc.Labelc = textc[0];
                pairc.Omega = Double.Parse(textc[1]);
                pairc.Phi = Double.Parse(textc[2]);
                pairc.Kappa = Double.Parse(textc[3]);
                pairc.XL = Double.Parse(textc[4]);
                pairc.YL = Double.Parse(textc[5]);
                pairc.ZL = Double.Parse(textc[6]);
                PhotoPairListCams.Add(pairc);
            }
        }
        int im = _imageNumber + 1;
        if (i == im)
        {
            String[] pointNumberline = readText[im].Split
            (new char[] { ' ', '\t' });
            if (pointNumberline.Length == 1)
            {
                PointNumber = int.Parse(pointNumberline[0]);
            }
        }
        else //from the second line and beyond
        {
            String[] textp = readText[i].Split(new char[]
            { ' ', '\t' });
            if (textp.Length == 8)
            {

```



```

        PhotoPairPoint pairp = new PhotoPairPoint();
        pairp.Labelp = textp[0];
        pairp.Xc = Double.Parse(textp[1]);
        pairp.Yc = Double.Parse(textp[2]);
        pairp.Zc = Double.Parse(textp[3]);
        pairp.X1 = Double.Parse(textp[4]);
        pairp.Y1 = Double.Parse(textp[5]);
        pairp.X2 = Double.Parse(textp[6]);
        pairp.Y2 = Double.Parse(textp[7]);
        PhotoPairListPoint.Add(pairp);
    }
}
int ip = _imageNumber + 1 + _pointNumber + 1;
if (i == ip)
{
    String[] residuNumberline =
        readText[ip].Split(new char[] { ' ', '\t' });
    if (residuNumberline.Length == 1)
        {ResiduNumber = int.Parse(residuNumberline[0]);}
}
else
{
    String[] textr = readText[i].Split(new char[] {
        ' ', '\t' });
    if (textr.Length == 5)
    {
        PhotoPairResidu pairr = new PhotoPairResidu();
        pairr.Labelr = textr[0];
        pairr.Xresidu1 = Double.Parse(textr[1]);
        pairr.Yresidu1 = Double.Parse(textr[2]);
        pairr.Xresidu2 = Double.Parse(textr[3]);
        pairr.Yresidu2 = Double.Parse(textr[4]);
        PhotoPairListResidu.Add(pairr);
    }
}
}
}
public override string ToString()
{
    string text1 = String.Format("imageNumber:{0}
        focal:{1} xo:{2} yo:{3}\n", _imageNumber, _focus,
        _xo, _yo);
    string text2 = String.Empty;
    string text3 = String.Format("pointNumber:{0}\n",
        _pointNumber);
    string text4 = String.Empty;
    string text5 = String.Format("residuNumber: {0}\n",
        _residuNumber);
    string text6 = String.Empty;
    string tempc = String.Empty;
    string tempp = String.Empty;
    string tempr = String.Empty;
    foreach (PhotoPairCams c in _photopairlistcams)
    {
        text2 = c.ToString() + "\n";
        tempc = tempc + text2;
    }
    foreach (PhotoPairPoint p in _photopairlistpoint)
    {
        text4 = p.ToString() + "\n";
        tempp = tempp + text4;
    }
}
}

```

```

    }
    foreach (PhotoPairResidu r in _photopairlistresidu)
    {
        text6 = r.ToString() + "\n";
        tempr = tempr + text6;
    }
    string text =text1+ tempr+ text3+ tempr+ text5+ tempr;
    return text;
}
public void PrintVectorDebug(ref double[] v, string name)
{
    int lower = v.GetLowerBound(0);
    int upper = v.GetUpperBound(0);
    int length = v.Length;
    for (int i = lower; i <= upper; i++)
    {}
}
public void PrintMatrixDebug(ref double[,] a, string name)
{
    int lower1 = a.GetLowerBound(0);
    int upper1 = a.GetUpperBound(0);
    int lower2 = a.GetLowerBound(1);
    int upper2 = a.GetUpperBound(1);
    int length = a.Length;
    System.Diagnostics.Trace.WriteLine("\n" + name + ":");
    for (int i = lower1; i <= upper1; i++)
    {
        for (int j = lower2; j <= upper2; j++)
        {}
    }
}
public void PrintMatrixDebugsd(ref double[,] a, string
name, StreamWriter sc)
{
    int lower1 = a.GetLowerBound(0);
    int upper1 = a.GetUpperBound(0);
    int lower2 = a.GetLowerBound(1);
    int upper2 = a.GetUpperBound(1);
    int length = a.Length;
    sc.WriteLine( name + ":");
    for (int i = lower1; i <= upper1; i++)
    {
        for (int j = lower2; j <= upper2; j++)
        { sc.WriteLine(a[i, j].ToString() + "\t"); }
        sc.WriteLine();
    }
}
private double[,] CSML2Array(ref CSML.Matrix m)
{
    int rowCount = m.RowCount;
    int columnCount = m.ColumnCount;
    double[,] t = new double[rowCount, columnCount];
    for (int i = 1; i <= rowCount; i++)
    {
        for (int j = 1; j <= columnCount; j++)
            t[i - 1, j - 1] = (m[i, j]).Re;
    }
    return t;
}
}
}
}

```

Lampiran C

Source Code Bundle Adjustment Pada Bahasa

Pemograman MATLAB 2008

Source Code Bundle Adjustment Pada Bahasa MATLAB 2008

C.1 Program Utama Bundle Adjustment

```
% Demo program for ba's MEX-file interface
camsfname = '../demo/utara1_2.txt'; %'/home/lourakis/cmp/bt.cams';
ptsfname = '../demo/utara1_52.txt'; %'/home/lourakis/cmp/bt.pts';
residufname = '../demo/utara1_52re.txt';
%'/home/lourakis/cmp/bt.pts';
xo = input('Masukan Nilai xo = ');
yo = input('Masukan Nilai yo = ');
foc = input('Masukan Nilai fokus = ');

% read in camera parameters
[p1, p2, p3, p4, p5, p6, p7] = textread(camsfname,
'%s%f%f%f%f%f', 'commentstyle', 'shell');
ncams=length(p1);
cams=[p2, p3, p4, p5, p6, p7];
cnp=size(cams, 2);
cams;

% Membaca titik-titik pada notepad tiap line
spmask = sparse([], [], [], 1, ncams);
dmask = zeros(1, ncams);
npts = 0;
pts2D = [];
fid = fopen(ptsfname);

% Membaca titik-titik pada notepad line per line
while ~feof(fid),line=fgets(fid);
    ~feof(fid);
    [A, count, errmsg, nextindex]=sscanf(line, '%f%f%f', [1,
        4]); % membaca X, Y, Z, nframes
    if(size(A, 2)>0) % did we read anything?
        npts = npts+1;
        pts3D(npts, :) = A(1:3); % store X, Y, Z
        nframes = A(4);
        projs = zeros(1, 2*ncams); %preallocate
```

```

ptvis = zeros(1, ncams); %preallocate
for i=1:nframes % read "nframes" id, x, y triplets
    [A, count, errmsg,j]=sscanf(line(nextindex:length
        (line)), '%f%f%f', [1, 3]); % read id, x, y
    nextindex=nextindex+j; %skip the already read line
        prefix
    %store id, remember that matlab indices start from 1
    ptvis(A(1)+1)=1;
    spmask(npts, A(1)+1)=1;
    projs((i-1)*2+1:i*2)=A(2:3); % store x, y
end
pts2D=[pts2D projs];
dmask(npts, :)=ptvis;
end

end
fclose(fid);
pnp = size(pts3D, 2);
pts2D;
pts3D;

spsmask2 = sparse([], [], [], 1, ncams);
dmask2 = zeros(1, ncams);
npts2 = 0;
residu = [];
fresidu = fopen(residufname);
% membaca residu file line by line
while ~feof(fresidu),line=fgets(fresidu);
    [B, count, errmsg, nextindex]=sscanf(line, '%f', [1, 1]); %
        read nframes
    if(size(B, 2)>0) % did we read anything?
        npts2=npts2+1;
        nframes=B(1);
        projs2=zeros(1, 2*ncams); % preallocate
        ptvis2=zeros(1, ncams); % preallocate
        for i=1:nframes % read "nframes" id, x, y triplets
            [B,count,errmsg,j]=sscanf(line(nextindex:length
                (line)),
                '%f%f%f', [1, 3]); % read id, xres, yres
            nextindex=nextindex+j;%skip the already read line prefix
            %store id, remember that matlab indices start from 1
            ptvis2(B(1)+1)=1;

```



```

        spmask2(npts2, B(1)+1)=1;
        projs2((i-1)*2+1:i*2)=B(2:3); % store xres, yres
    end
    % "projs2" now has the image projections of point "npts2"
    % and "ptvis2" is its visibility mask
    residu=[residu projs2];
    dmask2(npts2, :)=ptvis2;
end
end
fclose(fresidu);

%Syarat bundle
m=size(cams,1);
n=size(pts3D,1);
% m=2
% n=4
if ((n<=3+(2/(2*m-3))) || (m<=1.5+(1/n-3)))error('point : &d/5',n)
    |error('image min 2 :&d/2',m);
end;

% initial point & structure estimates vector
p0 = [reshape(cams', 1, ncams*cnp) reshape(pts3D', 1, npts*pnp)
    reshape(pts2D', 1, npts*2*ncams)];
p02 = [reshape(residu', 1,npts*2*ncams)];

%struktur nilai residu
residu = reshape(p02(1:npts*2*ncams), 2*ncams, npts2)';

%struktur omega,phi,kappa,XL,YL,ZL
cams = reshape(p0(1:ncams*cnp), cnp, ncams)';
om = cams(:,1);
ph = cams(:,2);
kp = cams(:,3);
XL = cams(:,4);
YL = cams(:,5);
ZL = cams(:,6);
%struktur X,Y,Z
pts3D = reshape(p0(ncams*cnp+1:ncams*cnp+npts*pnp), pnp, npts)';
X = pts3D(:,1);

```

```

Y = pts3D(:,2);
Z = pts3D(:,3);

%struktur x,y,...
pts2D = reshape(p0(ncams*cnp+npts*pn+1:ncams*cnp+npts*pn+npts*2*
    ncams), 2*ncams, npts)';

%struktur matrik w
W = zeros(2,1); % preallocate
%struktur matrik bobot
P = zeros(2,2);

Nki = zeros(6*ncams,6*ncams);
Nkij = zeros(3*npts,6*ncams);
Cki = zeros(6*ncams,1);
Nkj = zeros(3*npts,3*npts);

for iter = 1:10
    iterasi =iter
    %Struktur N1=AT1PA1 (Struktur untuk parameter EO)
    for i=1:ncams
        i;
        om(i); ph(i); kp(i); XL(i); YL(i); ZL(i);
        Ni=zeros(6,6);
        Ci=zeros(6,1);
        for j=1:npts
            j;
            n=1;
            uvw = projectUVW(om(i),ph(i),kp(i),XL(i),YL(i),ZL(i),
                X(j),Y(j),Z(j)));
            A1=projectAR1(om(i),ph(i),kp(i),XL(i),YL(i),ZL(i),foc,
                X(j),Y(j),Z(j)));
            A2=projectAR2(om(i),ph(i),kp(i),XL(i),YL(i),ZL(i),foc,
                X(j),Y(j),Z(j)));
            a(n,1)=A1(1); a(n,2)=A1(2); a(n,3)=A1(3);
            a(n,4)=-A1(4); a(n,5)=-A1(5); a(n,6)=-A1(6);
            a(n+1,1)=A1(7); a(n+1,2)=A1(8); a(n+1,3)=A1(9);
            a(n+1,4)=-A1(10); a(n+1,5)=-A1(11); a(n+1,6)=-A1(12);
            a2(n,1)=A2(1); a2(n,2)=A2(2); a2(n,3)=A2(3);
            a2(n+1,1)=A2(4); a2(n+1,2)=A2(5); a2(n+1,3)=A2(6);

```

```

W(1,1)=pts2D(j,i*2-1)-xo+(foc*uvw(1)/uvw(3));
W(2,1)=pts2D(j,i*2)-yo+(foc*uvw(2)/uvw(3));

%matrik bobot
P(1,1)=1/(residu(j,i*2-1))^2;
P(1,2)=0;
P(2,1)=0;
P(2,2)=1/(residu(j,i*2))^2;
%-----
Ni=Ni+(a'*P*a);      %Ni=ATPA
Ci=Ci+(a'*P*W);      %Ci=AT1PW
Nij=a2'*P*a;         %Nij=A2TPA

k=(j-1)+(j*2)-1;
N23(k:k+2,1:6)=reshape(reshape(Nij,1,[])',3,6);
end;
i;
k=6*i-5;
l=((3*npts)-1)*i-((3*npts)-1)+(i*1);
%Matrik diagonal Ni
Nki(k:k+5,k:k+5)=reshape(reshape(Ni,1,[])',6,6); %12x12
%matrik Nij
Nkij(1:3*npts,k:k+5)=reshape(reshape(N23,1,[])',(3*npts),6);
%24x12
%matrik Ci
Cki(k:k+5,1:1)=reshape(reshape(Ci,1,[])',6,1);
end;
%Struktur N2=AT2PA2(hitung nilai koord. object space)
for j=1:npts
    Nj=zeros(3,3);
    Cj=zeros(3,1);
    W=zeros(2,1);
    for i=1:ncams
        n=1;
        uvw=projectUVW(om(i),ph(i),kp(i),XL(i),YL(i),ZL(i),X(j),Y(j),Z(j)));
        A2=projectAR2(om(i),ph(i),kp(i),XL(i),YL(i),ZL(i),foc,X(j),Y(j),Z(j)));
        a2(n,1)=A2(1); a2(n,2)=A2(2); a2(n,3)=A2(3);
        a2(n+1,1)=A2(4); a2(n+1,2)=A2(5); a2(n+1,3)=A2(6);
    end
end

```

```

W(1,1)=pts2D(j,i*2-1)-xo+(foc*uvw(1)/uvw(3));
W(2,1)=pts2D(j,i*2)-yo+(foc*uvw(2)/uvw(3));

%matrik bobot
P(1,1)=1/(residu(j,i*2-1))^2;
P(1,2)=0;
P(2,1)=0;
P(2,2)=1/(residu(j,i*2))^2;

Nj=Nj+(a2'*P*a2);    %Nj=AT2PA2
Cj=Cj+(a2'*P*W);    %Ci=AT2Pw
end;
j;
k=(3*j)-2;
%Matrik diagonal Nj
Nkj(k:k+2,k:k+2)=reshape(reshape(Nj,1,[])',3,3); %24x24
%Matrik Cj
Ckj(k:k+2,1:1)=reshape(reshape(Cj,1,[])',3,1);
end;
%membentuk matrik N
N(1:12,1:(3*npts+6*ncams))=[reshape(Nki,12,12) reshape(Nkij',1
    2,(3*npts))];
N(13:(3*npts+6*ncams),1:(3*npts+6*ncams))=[reshape(Nkij,(3*np
    ts),12) reshape(Nkj,(3*npts),(3*npts))];
%membentuk matrik C
C(1:12,1:1)=Cki;
C(13:(3*npts+6*ncams),1:1)=Ckj;
%hitung Matrik koreksi
invP=pinv(N);
xx=pinv(N)*C;
%hitung parameter EO update
for i=1:ncams
    om(i,1)=om(i,1)+xx(6*i-5);
    ph(i,1)=ph(i,1)+xx(6*i-4);
    kp(i,1)=kp(i,1)+xx(6*i-3);
    XL(i,1)=XL(i,1)+xx(6*i-2);
    YL(i,1)=YL(i,1)+xx(6*i-1);
    ZL(i,1)=ZL(i,1)+xx(6*i);
end;

```

```

%hitung parameter terkoreksi XYZ
for j=1:npts
    X(j,1)=X(j,1)+xx(13+((j-1)*3),1);
    Y(j,1)=Y(j,1)+xx(14+((j-1)*3),1);
    Z(j,1)=Z(j,1)+xx(15+((j-1)*3),1);
end;
%pts3D=[X Y Z];
%hitung presisi data
sd=sqrt(trace(inv(Nkj))/(3*npts))
pre=sqrt(diag(inv(Nkj)));
for j=1:npts
    Xpre(j,1)=pre(1+((j-1)*3),1);
    Ypre(j,1)=pre(2+((j-1)*3),1);
    Zpre(j,1)=pre(3+((j-1)*3),1);
end;
if (norm(xx)<8e-010)
    break
end;
end;
ExtFinal=[om ph kp XL YL ZL]
pts3=[X Y Z]
presisi=[Xpre Ypre Zpre]
plot3(pts3(:,1),pts3(:,2),pts3(:,3),'or');

% Save file data akhir
fsave='../demo/saveutara2.txt';
fout=fopen(fsave,'w');

fprintf(fout,'Copyright : Kadok Dody Aris Pramana\n');
fprintf(fout,'\n');
fprintf(fout,'***** Hasil Akhir dari Proses Bundle Adjustment
pada Iterasi Terakhir *****\n');
fprintf(fout,'\n');
fprintf(fout,'%s','Jumlah Foto = ');
fprintf(fout,'%i\n',ncams);
fprintf(fout,'%s','Jumlah Titik-titik = ');
fprintf(fout,'%i\n',npts);
fprintf(fout,'\n');
fprintf(fout,'%s','Iterasi ke ');
fprintf(fout,'%i\n',iterasi);

```

```

fprintf(fout, '%s', 'Standar Deviasi = ');
fprintf(fout, '%f\n', sd);
fprintf(fout, '\n');
fprintf(fout, '***** Parameter Eksterior
  Kamera *****\n');
fprintf(fout, '%4s %8s %12s %12s %10s %12s
  %12s\n', 'Foto', 'Omega', 'Phi', 'Kappa', 'XL', 'YL', 'ZL');
for i=1:ncams
  fprintf(fout, '%2i %12f %12f %12f %12f %12f
  %12f\n', i, ExtFinal(i,1), ExtFinal(i,2), ExtFinal(i,3),
  ExtFinal(i,4), ExtFinal(i,5), ExtFinal(i,6));
end
fprintf(fout, '\n');
fprintf(fout, '\n');

fprintf(fout, '***** Koordinat Titik-titik
  Objek *****Presisi*****\n');
fprintf(fout, '%2s %8s %12.5s %17s %13s %13s %13s\n', 'Titik', 'X',
  'Y', 'Z', 'Presisi X', 'Presisi Y', 'Presisi Z');
for j=1:npts
  fprintf(fout, '%3i %13f %13f %13f %13f %13f %13f\n', j, pts3
  (j,1), pts3(j,2), pts3(j,3), presisi(j,1), presisi(j,2), presisi(j,3)
  ));
end
fclose(fout);

```

C.2 Source Code Fungsi Elemen Matrik A1

```

function A1=projectAR1(om, ph, kp, XL, YL, ZL, foc, X, Y, Z)
m=projectROTM(om, ph, kp);
uvw=projectUVW(om, ph, kp, XL, YL, ZL, X, Y, Z);

A1(1)=(foc/(uvw(3))^2)*((uvw(1))*((-m(3,3)*(Y-YL))+m(3,2)*(Z-ZL))
  )-(uvw(3))*((-m(1,3)*(Y-YL))+m(1,2)*(Z-ZL)));
A1(2)=(foc/(uvw(3))^2)*((uvw(1))*((cos(ph*pi/180)*(X-XL))+sin(om*
  pi/180)*sin(ph*pi/180)*(Y-YL))-cos(om*pi/180)*sin(ph*pi
  /180)*(Z-ZL)))-uvw(3)*((-sin(ph*pi/180)*cos(kp*pi/180)*(X-
  XL))+sin(om*pi/180)*cos(ph*pi/180)*cos(kp*pi/180)*(Y-YL))-
  (cos(om*pi/180)*cos(ph*pi/180)*cos(kp*pi/180)*(Z-ZL)));

```



```

A1(3)=(-foc/(uvw(3)))*(m(2,1)*(X-XL))+m(2,2)*(Y-YL)+m(2,3)*(Z-
    ZL));
A1(4)=(foc/(uvw(3))^2)*(((uvw(1))*m(3,1))-((uvw(3))*m(1,1)));
A1(5)=(foc/(uvw(3))^2)*(((uvw(1))*m(3,2))-((uvw(3))*m(1,2)));
A1(6)=(foc/(uvw(3))^2)*(((uvw(1))*m(3,3))-((uvw(3))*m(1,3)));
A1(7)=(foc/(uvw(3))^2)*((uvw(2))*((-m(3,3)*(Y-YL))+m(3,2)*(Z-ZL)
    )-(uvw(3))*((-m(2,3)*(Y-YL))+m(2,2)*(Z-ZL))));
A1(8)=(foc/(uvw(3))^2)*((uvw(2))*((cos(ph*pi/180)*(X-XL))+sin(om*
    pi/180)*sin(ph*pi/180)*(Y-YL))-cos(om*pi/180)*sin(ph*pi/
    180)*(Z-ZL)))-((uvw(3))*((sin(ph*pi/180)*sin(kp*pi/180)*(X-
    XL))-sin(om*pi/180)*cos(ph*pi/180)*sin(kp*pi/180)*(Y-YL)
    +cos(om*pi/180)*cos(ph*pi/180)*sin(kp*pi/180)*(Z-ZL))));
A1(9)=(foc/(uvw(3)))*(m(1,1)*(X-XL))+m(1,2)*(Y-YL)+m(1,3)*(Z-
    ZL));
A1(10)=(foc/(uvw(3))^2)*(((uvw(2))*m(3,1))-((uvw(3))*m(2,1)));
A1(11)=(foc/(uvw(3))^2)*(((uvw(2))*m(3,2))-((uvw(3))*m(2,2)));
A1(12)=(foc/(uvw(3))^2)*(((uvw(2))*m(3,3))-((uvw(3))*m(2,3)));

```

C.3 Source Code Fungsi Elemen Matrik A2

```

function A2=projectAR2(om,ph,kp,XL,YL,ZL,foc,X,Y,Z)
m=projectROTM(om,ph,kp);
uvw=projectUVW(om,ph,kp,XL,YL,ZL,X,Y,Z);
A2(1)=(foc/(uvw(3))^2)*(((uvw(1))*m(3,1))-((uvw(3))*m(1,1)));
A2(2)=(foc/(uvw(3))^2)*(((uvw(1))*m(3,2))-((uvw(3))*m(1,2)));
A2(3)=(foc/(uvw(3))^2)*(((uvw(1))*m(3,3))-((uvw(3))*m(1,3)));
A2(4)=(foc/(uvw(3))^2)*(((uvw(2))*m(3,1))-((uvw(3))*m(2,1)));
A2(5)=(foc/(uvw(3))^2)*(((uvw(2))*m(3,2))-((uvw(3))*m(2,2)));
A2(6)=(foc/(uvw(3))^2)*(((uvw(2))*m(3,3))-((uvw(3))*m(2,3)));

```

C.4 Source Code Fungsi UVW

```

function uvw=projectUVW(om,ph,kp,XL,YL,ZL,X,Y,Z)
m=projectROTM(om,ph,kp);
uvw(1)=m(1,1)*(X-XL)+m(1,2)*(Y-YL)+m(1,3)*(Z-ZL);
uvw(2)=m(2,1)*(X-XL)+m(2,2)*(Y-YL)+m(2,3)*(Z-ZL);
uvw(3)=m(3,1)*(X-XL)+m(3,2)*(Y-YL)+m(3,3)*(Z-ZL);

```

C.5 Source Code Fungsi Matrik Rotasi

```
function m=projectROTM(om, ph, kp)

m11=cos(ph*pi/180).*cos(kp*pi/180);
m12=(sin(om*pi/180).*sin(ph*pi/180).*cos(kp*pi/180))+(cos(om*pi
    /180).*sin(kp*pi/180));
m13=-(cos(om*pi/180).*sin(ph*pi/180).*cos(kp*pi/180))+(sin(om*pi/
    180).*sin(kp*pi/180));
m21=-cos(ph*pi/180).*sin(kp*pi/180);
m22=-(sin(om*pi/180).*sin(ph*pi/180).*sin(kp*pi/180))+(cos(om*pi/
    180).*cos(kp*pi/180));
m23=(cos(om*pi/180).*sin(ph*pi/180).*sin(kp*pi/180))+(sin(om*pi/18
    0).*cos(kp*pi/180));
m31=sin(ph*pi/180);
m32=-sin(om*pi/180).*cos(ph*pi/180);
m33=cos(om*pi/180).*cos(ph*pi/180);
m=[m11 m12 m13;m21 m22 m23;m31 m32 m33];
```