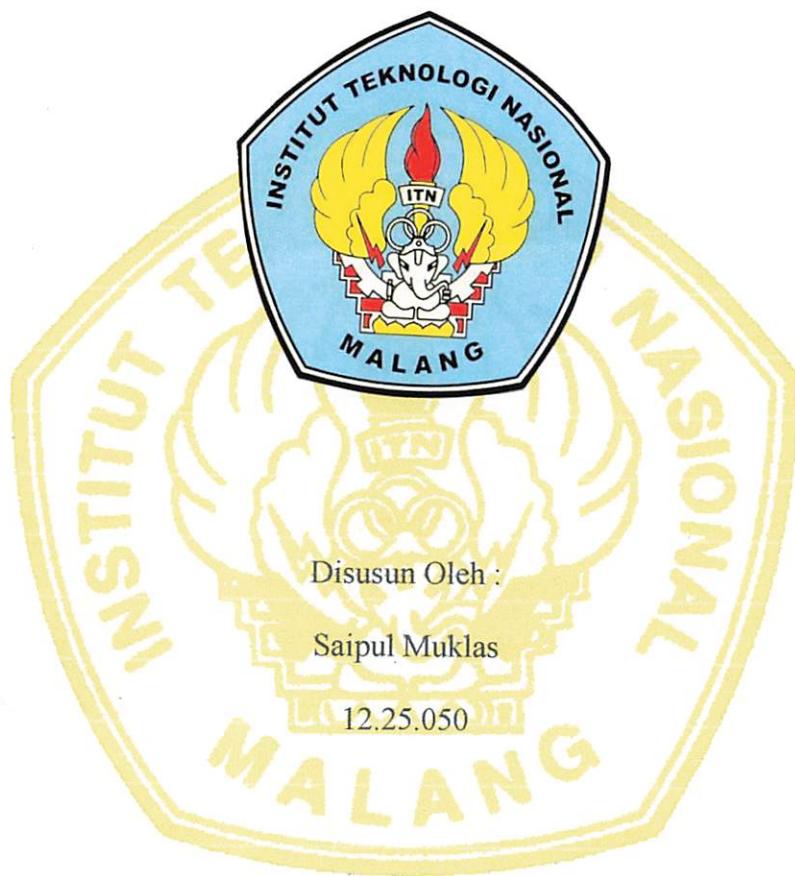


SKRIPSI

PEMBUATAN PERANGKAT LUNAK UNTUK PERHITUNGAN DATA
HASIL PENGUKURAN TERISTRIS BERBASIS WINDOWS



Disusun Oleh :

Saipul Muklas

12.25.050

JURUSAN TEKNIK GEODESI

FAKULTAS TEKNIK SIPIL DAN PERENCANAAN

INSTITUT TEKNOLOGI NASIONAL MALANG

2016

LEMBAR PERSETUJUAN

PEMBUATAN PERANGKAT LUNAK UNTUK PERHITUNGAN DATA HASIL PENGUKURAN TERISTRIS BERBASIS WINDOWS

SKRIPSI

Diajukan untuk memenuhi persyaratan dalam mencapai Gelar Sarjana Teknik (ST)

Strata Satu (S1) Teknik Geodesi S-1

Institut Teknologi Nasional Malang

Oleh:

SAIPUL MUKLAS

12.25.050

Menyetujui,

Dosen Pembimbing Utama

Ir. Pradono Joanes De.Deo., MSI

Dosen Pembimbing Pendamping

Hery Purwanto, ST, MSc

Mengetahui,

Ketua Jurusan Teknik Geodesi S-1



M. Edwin Tjahjadi, ST, M.GeoM.Sc., Ph.D.



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG

INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

BERITA ACARA UJIAN SEMINAR HASIL SKRIPSI FAKULTAS TEKNIK SIPIL DAN PERENCANAAN

NAMA : SAIPUL MUKLAS

NIM : 12.25.050

JURUSAN : TEKNIK GEODESI S-1

JUDUL :

PEMBUATAN PERANGKAT LUNAK UNTUK PERHITUNGAN DATA
HASIL PENGUKURAN TERISTRIS BERBASIS WINDOWS

Telah Dipertahankan di Hadapan Panitia Pengujian Skripsi Jenjang
Strata-1 (S-1)

Pada Hari : Selasa

Tanggal : 23 Agustus 2016

Dengan nilai :

Panitia Ujian Skripsi

Ketua

Bagus Subakti, ST., M.Eng

Pengaji I

Dosen Pendamping

Pengaji II

Ir. Agus Darpono.,MT

Hery Purwanto,ST.,Msc

M. Edwin Tjahjadi, ST, M.Ggeom.Sc.,Ph.D

**PEMBUATAN PERANGKAT LUNAK UNTUK PERHITUNGAN DATA HASIL
PENGUKURAN TERISTRIS BERBASIS WINDOWS**

Saipul Muklas 12.25.050

Dosen Pembimbing I : Ir. Pradono Joanes De Deo. MSi
Dosen Pembimbing II : Hery Purwanto.ST.,Msc

ABSTRAKSI

Dengan semakin pesatnya dunia survey pemetaan saat ini dan semakin kompleksnya permasalahan yang ada, perhitungan data pengukuran merupakan salah satu bagian dari masalah yang dapat menghambat efisiensi waktu dalam pekerjaan. Masalah yang sering dihadapi dalam kegiatan pemetaan topografi adalah perhitungan data hasil pengukuran di lapangan. Kurang tersedianya program perhitungan poligon, levelling beserta detail situasi yang aplikatif, efektif, dan efisien menyebabkan proses perhitungan data memakan waktu yang cukup lama. Selain itu, program yang dibuat pun sebatas hanya pada beberapa titik kontrol saja.

Dengan dibuatnya program ini diharapkan dapat membantu mempermudah dalam melakukan pengolahan data hasil pengukuran di lapangan sehingga menghemat efisiensi waktu yang ada. Mempermudah pengguna dalam melakukan *plotting* koordinat hasil perhitungan ke dalam *software* pemetaan seperti Autocad Land Desktop.

SURAT PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertandatangan di bawah ini :

Nama : Saipul Muklas

NIM : 12.25.050

Program Studi : Teknik Geodesi S-1

Fakultas : Teknik Sipil dan Perencanaan

Menyatakan dengan sesungguhnya bahwa Skripsi saya yang berjudul

“Pembuatan Perangkat Lunak Untuk Perhitungan Data Hasil

Pengukuran Teristris Berbasis *Windows*”

adalah hasil karya saya sendiri dan bukan menjiplak atau menduplikat serta tidak mengutip atau menyadur dari hasil karya orang lain kecuali disebutkan sumbernya.

Malang, 23 Agustus 2016

Yang membuat pernyataan



Saipul Muklas

NIM : 12.25.050

LEMBAR PERSEMPAHAN

“Tragedi terbesar dalam kehidupan bukanlah sebuah kematian, tapi hidup tanpa tujuan.

Karena itu teruslah bermimpi untuk menggapai tujuan dan harapan supaya hidup bisa lebih bermakna”

Dengan segala puji dan syukur kepada Allah SWT, atas dukungan dan doa dari orang-orang tercinta akhirnya skripsi ini bisa diselesaikan dengan baik dan tepat pada waktunya. Oleh karena itu dengan rasa bangga dan bahagia saya khatulkan rasa syukur dan terimakasih saya kepada:

Allah SWT, karena hanya atas izin dan karunia-Nya lah skripsi ini dapat dibuat dan selesai pada waktunya.

Abah dan Umi saya, yang telah memberikan dukungan moral maupun material serta doa yang tiada henti untuk kesuksesan saya.

Bapak dan Ibu dosen pembimbing, pengaji, dan pengajar yang selama ini telah tulus dan ikhlas mendidik saya, menuntun, dan mengarahkan. Jasa kalian akan selalu terpatri dalam sanubari.

Kakak saya Arif Nurrahman atau ambon, adik saya Ana Mas’udah atau kuplek, walau sering berantem tapi kalian satu darah dengan saya dan peran kalian tidak bisa digantikan dengan siapapun dalam hidup ini.

Pak dhe dan Bu dhe saya yang di Tlogomas, selama 4 tahun ini selalu memberikan kasih sayangnya dan berperan sebagai pengganti orang tua saya dirumah.

Untuk semua teman-teman angkatan 2012, terimakasih atas canda tawanya selama ini, dan semua persaingan hebat selama masa kuliah, pressure kalian sangat hebat dan harus terus di imbangi wkwk.

Untuk semua teman-teman yang sering saya singgahi kost nya seperti kost Pondok alam, Kost dan kontrakan tlogomas, dan masih banyak lagi yang tidak bisa saya

sebutkan, terimakasih, kamar dan wifi kalian sangat berarti bagi perjalanan kuliah dan hidup saya wkwk.

Untuk semua wanita yang pernah datang dan pergi, walau kadang tak sempat dimiliki, kalian terbaik, ibarat puzzle kalian bagian dari kepingan perjalanan dan saling melengkapi wkwk.

Untuk satu wanita yang walau tak saya sebutkan namanya disini tapi selalu saya konsultasikan ke Tuhan dan orangtua saya, kamu terbaik diantara wanita-wanita terbaik di atas.

KATA PENGANTAR

Alhamdulillah, puji syukur kehadirat ALLAH S.W.T atas segala rahmat dan hidayah-Nya, sehingga penelitian berjudul **Pembuatan Perangkat Lunak Untuk Perhitungan Data Hasil Pengukuran Teristris Berbasis Windows** dapat terselesaikan. Penelitian ini dibuat untuk memenuhi salah satu syarat dalam memperoleh gelar sarjana teknik. Ucapan terima kasih yang sebesar-besarnya saya sampaikan pada:

1. Bapak M. Edwin Tjahjadi, S.T., M.Geo.Sc., Ph.D., selaku Ketua Jurusan Teknik Geodesi ITN Malang.
2. Bapak Pradono Joanes De Deo.Msi dan Bapak Hery Purwanto.ST.Msc selaku dosen pembimbing sekaligus pengarah dalam menyelesaikan penelitian ini.
3. Kedua orang tua beserta keluarga besar Penulis yang selalu memberikan do'a dan dukungan untuk Penulis.
4. Teman-teman satu jurusan dan angkatan 2012 yang telah banyak berbagi dalam berbagai hal.

Akhir kata penulis menyadari bahwa dalam penulisan penelitian Skripsi ini masih terdapat banyak kekurangan, untuk itu penulis mengharapkan adanya kritik dan saran yang membangun untuk perbaikan penelitian ini.

Malang, 24 Agustus 2016

Saipul Muklas

NIM 12.25.050

- b. Membuat program perhitungan posisi horisontal, vertikal, dan detail/situasi yang jumlah titik kontrolnya sesuai dengan yang diinginkan oleh si pengguna program.

I.5 Manfaat Penelitian

Adapun manfaat dari penelitian tugas akhir ini yang ingin dicapai adalah:

- a. Mempercepat dalam mengolah data posisi horisontal, vertikal, dan detail situasi dari pengukuran langsung di lapangan.
- b. Memberikan teknik pengolahan data posisi horisontal, vertikal, dan detail situasi kepada pengguna program berupa metode metode *Bowditch*.
- c. Mempermudah pengguna dalam melakukan input data karena bisa dihubungkan dengan microsoft excel dengan file format *.csv ataupun *.txt.

BAB II

DASAR TEORI

Pengukuran Terestris adalah pengambilan data dengan cara melakukan survey lapangan untuk mendapatkan hasil titik-titik pengukuran lapangan berupa X,Y dan Z dimana hasilnya nanti dibentuk kontur dan data topografi lainnya. Dalam proses pembentukan kontur menggunakan sistem TIN (Triangulated irregular network), dimana dalam pemanfaatan lebih lanjut dapat dipakai untuk perhitungan Cut & Fill, Staking Out, dll. (Rais J, 1979).

II.1 Kerangka Kontrol Horisontal

Kerangka dasar horizontal merupakan kumpulan titik-titik yang telah diketahui atau ditentukan posisi horizontalnya berupa koordinat pada bidang datar (X, Y) dalam sistem proyeksi tertentu. Bila dilakukan dengan cara teristris, pengadaan kerangka horizontal bisa dilakukan menggunakan cara triangulasi, trilaterasi atau poligon. Pemilihan cara dipengaruhi oleh bentuk medan lapangan dan ketelitian yang dikehendaki (Purworhardjo, 1986).

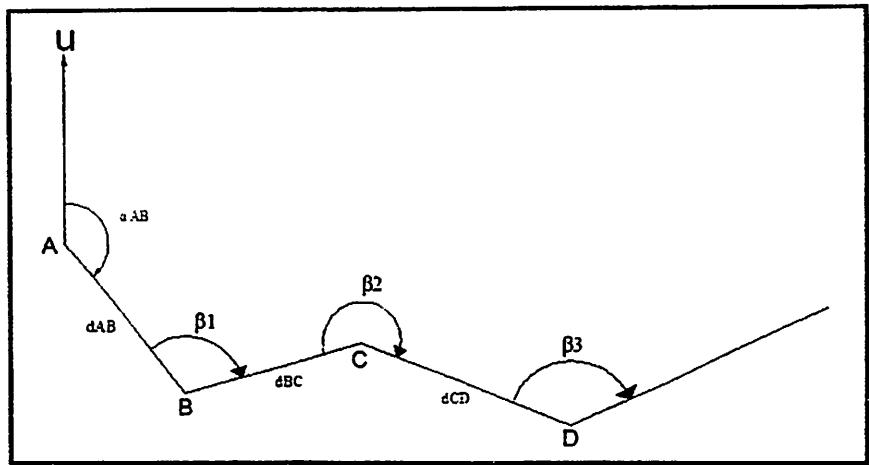
II.1.1 Poligon

Poligon adalah suatu rangkaian garis lurus yang berurutan menghubungkan titik – titik yang berkoordinat satu dengan lainnya menjadi bentuk tertentu (segi banyak beraturan atau segi banyak tidak beraturan/tidak bersegi).

Berdasarkan bentuknya poligon terbagi menjadi dua, poligon terbuka dan poligon tertutup.

1. Poligon terbuka

Poligon terbuka adalah poligon yang titik awal dan titik akhirnya merupakan titik yang berlainan (tidak bertemu pada satu titik).



Gambar 2.1 : poligon terbuka

Keterangan gambar :

- | | |
|---------------|-------------------------------------|
| α | = Azimut awal. |
| A, B, C dan D | = Titik poligon |
| β | = Besarnya sudut. |
| d AB | = Jarak antara titik A dan titik B. |

Syarat poligon terbuka terikat sempurna.

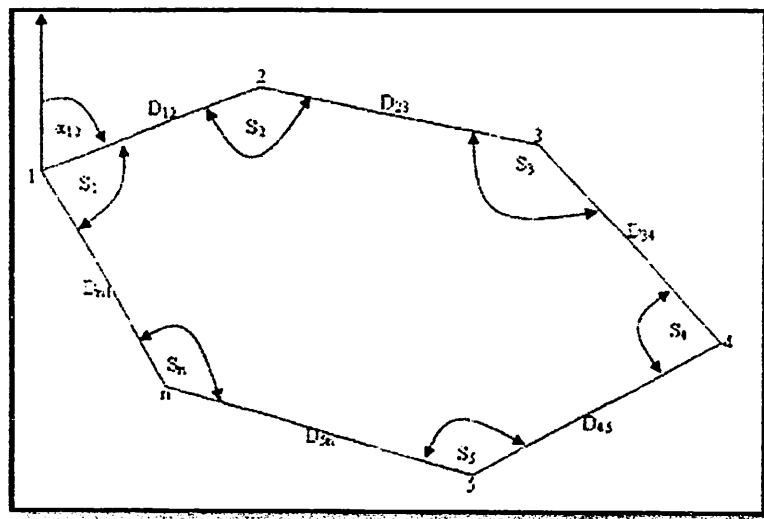
$$\sum s + f(s) = (a_{akhir} - a_{awal}) + (n - 1) * 180^\circ \dots \dots \dots [II.1]$$

Pengolahan data koordinat poligon tertutup

$$X_2 = X_1 + d_{12} \cdot \sin \alpha_{12} + f(x)' = \frac{d_{12}}{\Sigma d} * f(x) \dots [II.2]$$

2. Poligon Tertutup

Poligon tertutup atau kring adalah poligon yang titik awal dan titik akhirnya bertemu pada satu titik yang sama. Dalam perhitungan poligon dapat menggunakan sudut dalam ataupun sudut luar namun syarat masing-masing sudut tersebut berbeda.



Gambar 2.2: poligon tertutup sudut dalam

Keterangan:

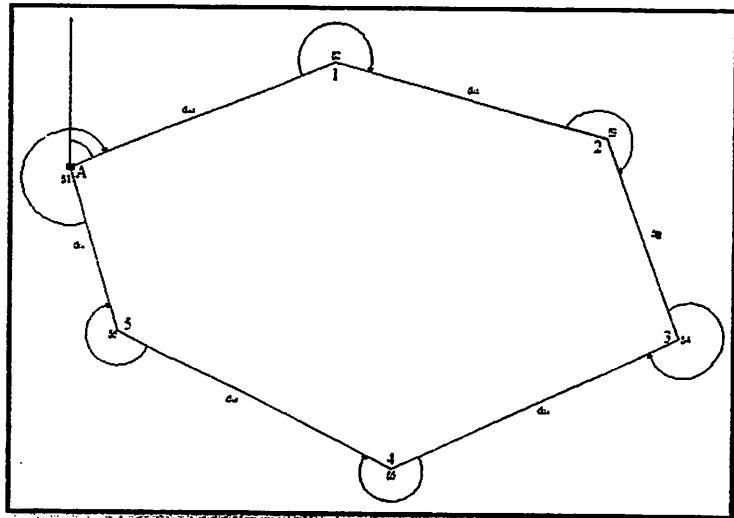
1, 2, 3, ..., n : titik kontrol poligon

D₁₂, D₂₃, ..., D_{n1} : jarak pengukuran sisi poligon

S₁, S₂, S₃, ..., S_n : sudut

Syarat poligon tertutup

$$\sum s + f(s) = (n - 2) * 180^\circ, \text{ untuk sudut dalam} \dots \dots \dots \text{[II.4]}$$



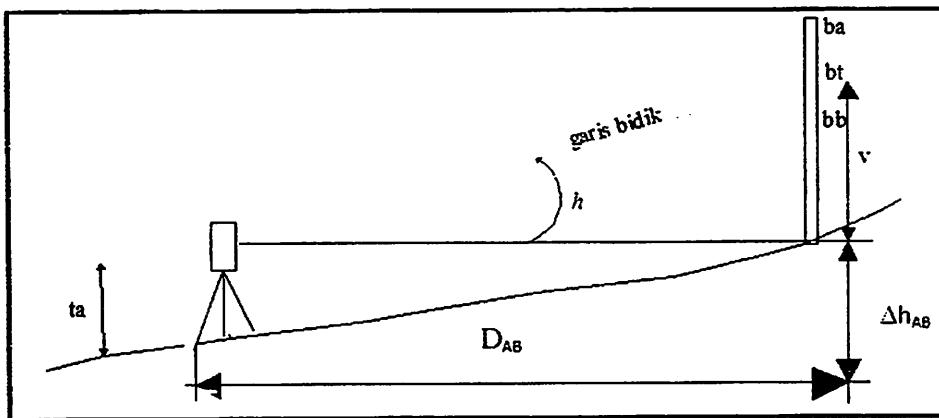
Gambar 2.3: sket poligon tertutup sudut luar

II.3 Pengukuran detail

Dalam pengukuran titik - titik detail prinsipnya adalah menentukan koordinat dan tinggi titik - titik detail dari titik-titik ikat. Metode yang digunakan dalam pengukuran titik - titik detail adalah metode offset dan metode tachymetri. Hasil yang diperoleh dari pengukuran tachymetri adalah posisi planimetris X, Y dan ketinggian Z. (Iskandar, 2011)

II.3.1 Metode Pengukuran Tachymetri

Metode tachymetri adalah pengukuran menggunakan alat-alat optis, elektronis, dan digital. Pengukuran detail cara tachymetri dimulai dengan penyiapan alat ukur di atas titik ikat dan penempatan rambu di titik bidik. Setelah alat siap untuk pengukuran, dimulai dengan perekaman data di tempat alat berdiri, pembidikan ke rambu ukur, pengamatan azimuth dan pencatatan data di rambu BT, BA, BB serta sudut miring . Metode tachymetri didasarkan pada prinsip bahwa pada segitiga-segitiga sebangun, sisi yang sepihak adalah sebanding.



Gambar 2.7: Metode pengukuran Tachymetri

Keterangan gambar:

D_{AB}	: jarak horisontal dari titik A ke titik B
H	: sudut helling
Ba	: benang atas
Bb	: benang bawah
Bt	: benang tengah
t_A	: tinggi alat

Rumus hitungan detil dengan metode tachimetri adalah sebagai berikut :

1. Jarak dengan menggunakan rumus:

$$d_{AB} = 100(ba - bb) \cos^2 \dots \dots \dots [II.7]$$

2. Jarak vertikal antara garis sejajar sumbu II dengan garis sejajar bt:

- ### 3. Beda tinggi titik detil:

4. Tinggi titik tiap detil (nilai Z):

II.4 Microsoft Visual Basic 6.0

Visual Basic adalah bahasa pemrograman berbasis *Windows*. Saat ini, *Visual Basic* adalah salah satu bahasa pemrograman yang terbaik. *Visual Basic* merupakan pengembangan dari *Basic*. *Basic (Beginner's All-purpose Symbolic Instruction Code)* adalah sebuah bahasa pemrograman “kuno” yang merupakan awal dari bahasa-bahasa pemrograman tingkat tinggi lainnya.

II.4.1 Komponen Microsoft Visual Basic

Dalam *software Microsoft Visual Basic* memiliki beberapa komponen yang perlu diketahui oleh penggunanya agar dapat mudah digunakan atau dioperasikan. Komponen – komponen tersebut antara lain jendela *toolbox*, jendela *form*, jendela kode, jendela *project*, jendela *properties*.

- Jendela *Toolbox*
 - Jendela *Form*
 - Jendela *Kode*
 - Jendela *Properties*

II.5 Comma Separated Values (CSV)

CSV atau *Comma-Separated Value* merupakan suatu format penyajian data teks dimana setiap data (*field*) dipisahkan dengan suatu pemisah koma (,) dan setiap baris *record* dipisahkan dengan baris baru (*line-break*). Selain menggunakan pemisah (*delimiter*) koma terkadang jenis file CSV juga dapat dipisahkan dengan titik-koma (;) atau karakter <tab>. Jenis file CSV sendiri, dapat dibaca oleh semua software pengolah data (*spreadsheet*) seperti Microsoft Excel dan OpenOffice Calc, AutoCadLand Development dan sebagainya.

BAB III

METODE PENELITIAN

III.1 Kebutuhan Alat dan Bahan

Peralatan yang akan digunakan dalam penelitian ini meliputi,

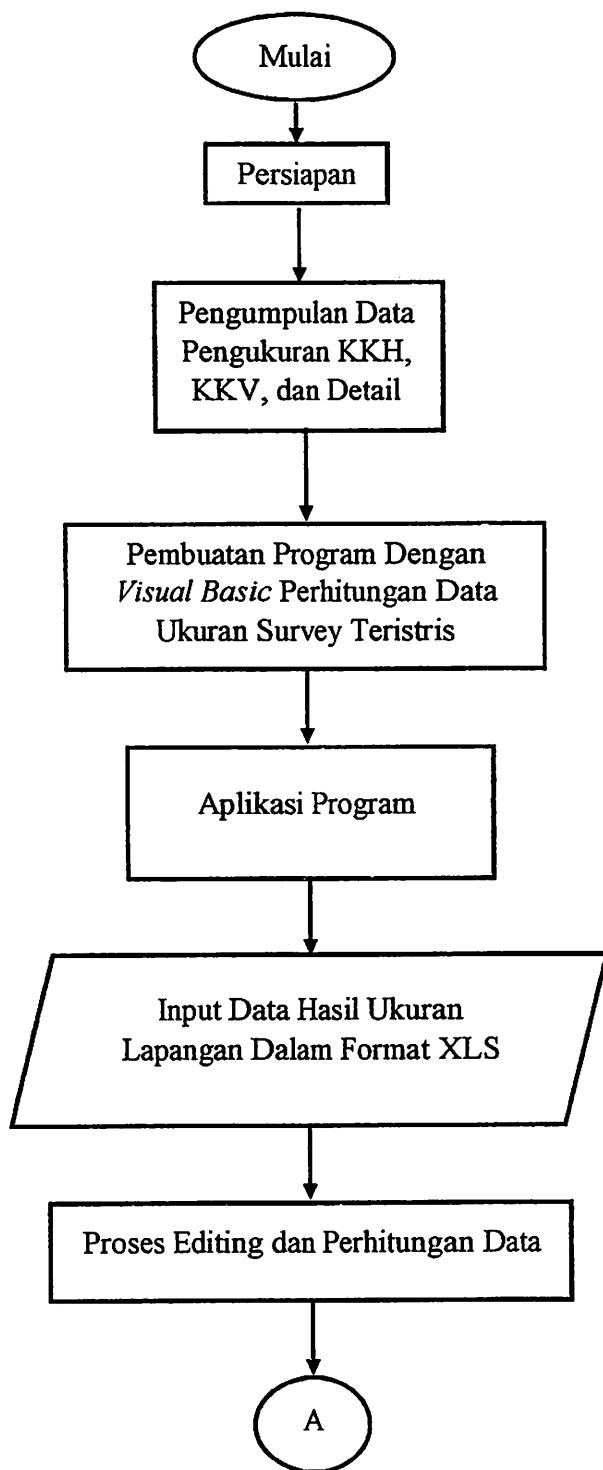
1. Perangkat keras:
 - a. Seperangkat komputer dengan *processor i3 2,30 Ghz* dan RAM 2 GB
 - b. Printer A4
2. Perangkat lunak:
 - a. *Software Visual Basic 6.0*
 - b. *Microsoft Office 2013*
 - c. *Autodesk Land Development 2009*

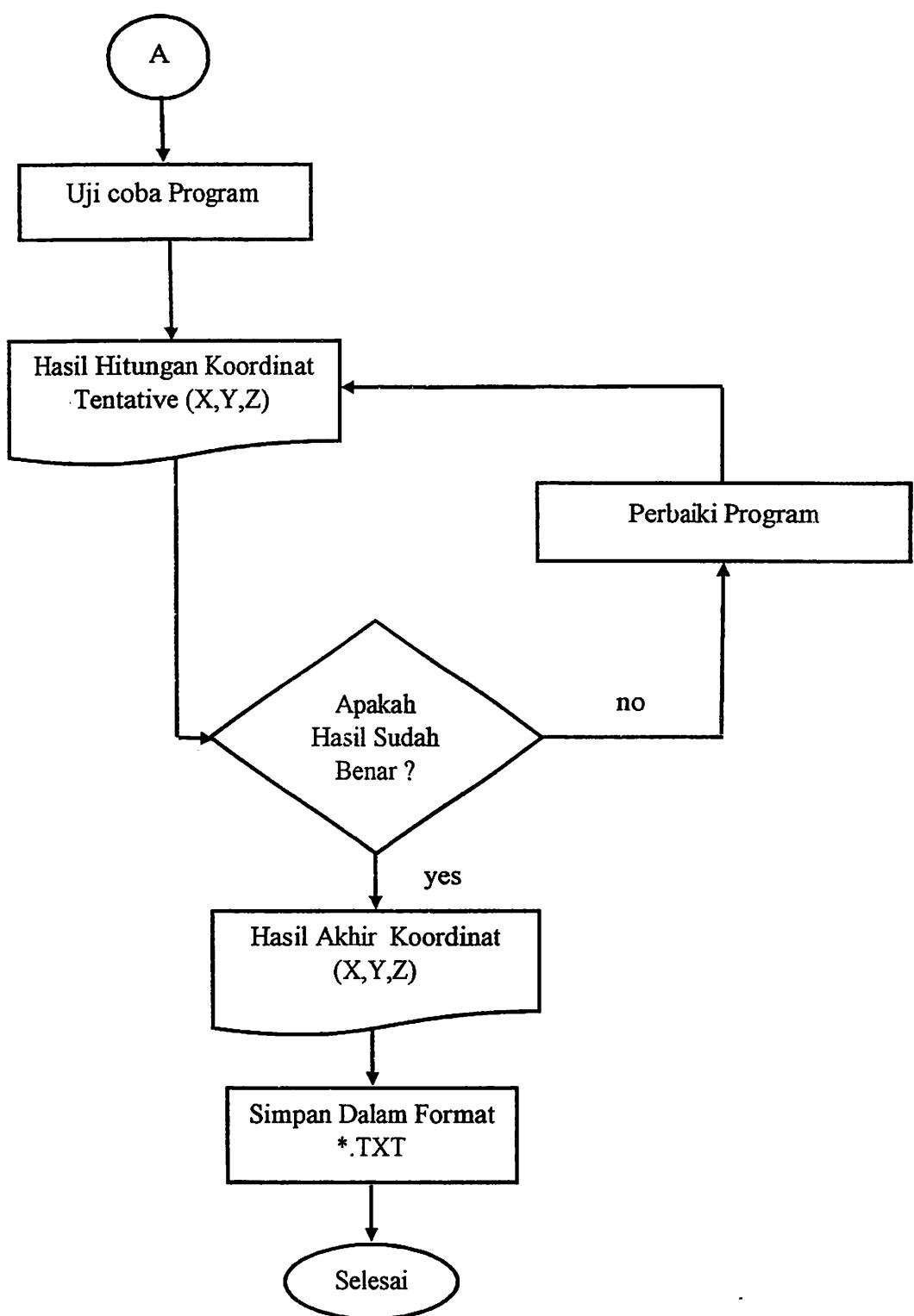
Bahan yang akan digunakan dalam penelitian ini meliputi;

- a. Data hasil pengukuran poligon
- b. Data hasil pengukuran levelling
- c. Data hasil pengukuran detail

III.2 Tahapan Penelitian

Berikut ini diagram alir yang menggambarkan tahapan dalam penelitian sebagai berikut;





Gambar 3.1: Diagram alir

Dari diagram alir penelitian pada bagan (3.1) secara garis besar pekerjaan ini meliputi:

1. Persiapan

Tahapan persiapan meliputi penyiapan referensi dan serta pengumpulan bahan dan peralatan yang akan digunakan. Dari referensi yang telah dikumpulkan, dilakukan studi literature untuk penentuan metode penelitian yang berkaitan dengan metode pengolahan dan analisa data.

2. Pengumpulan data

Data yang dikumpulkan meliputi data hasil ukuran lapangan (poligon) yang diukur dengan alat *Total Station* dan data hasil pengukuran levelling menggunakan alat waterpass beserta data detail situasinya. Dengan diperolehnya data penelitian maka kita bisa melakukan tahap selanjutnya.

3. Pembuatan visualisasi program

Pada tahap ini kerangka atau model muka dari program akan di tentukan agar mengetahui input dan output program ini seperti apa nantinya, serta memasukkan rumus-rumus perhitungan poligon dan dirubah menjadi bahasa pemrograman.

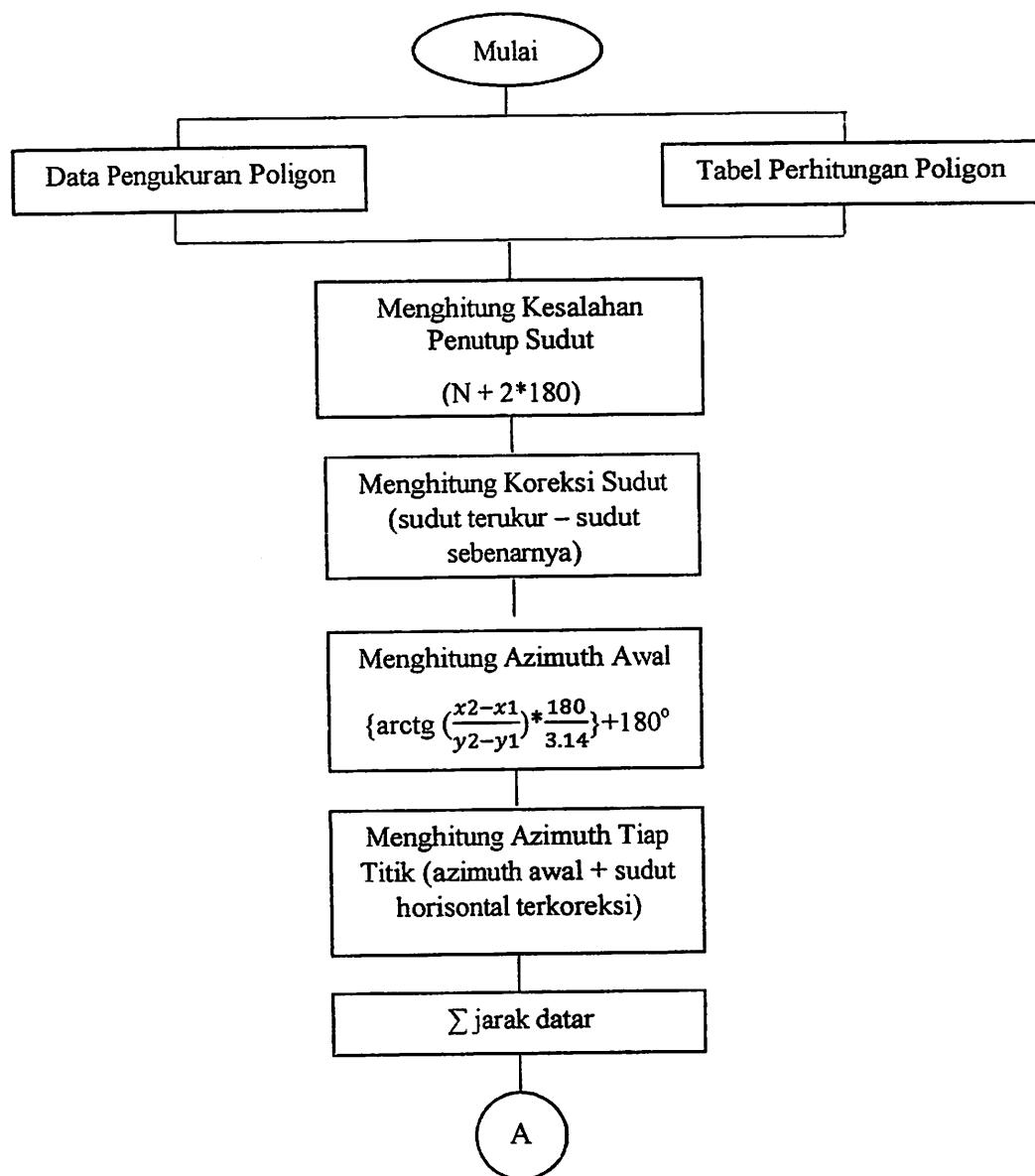
4. *Input* data

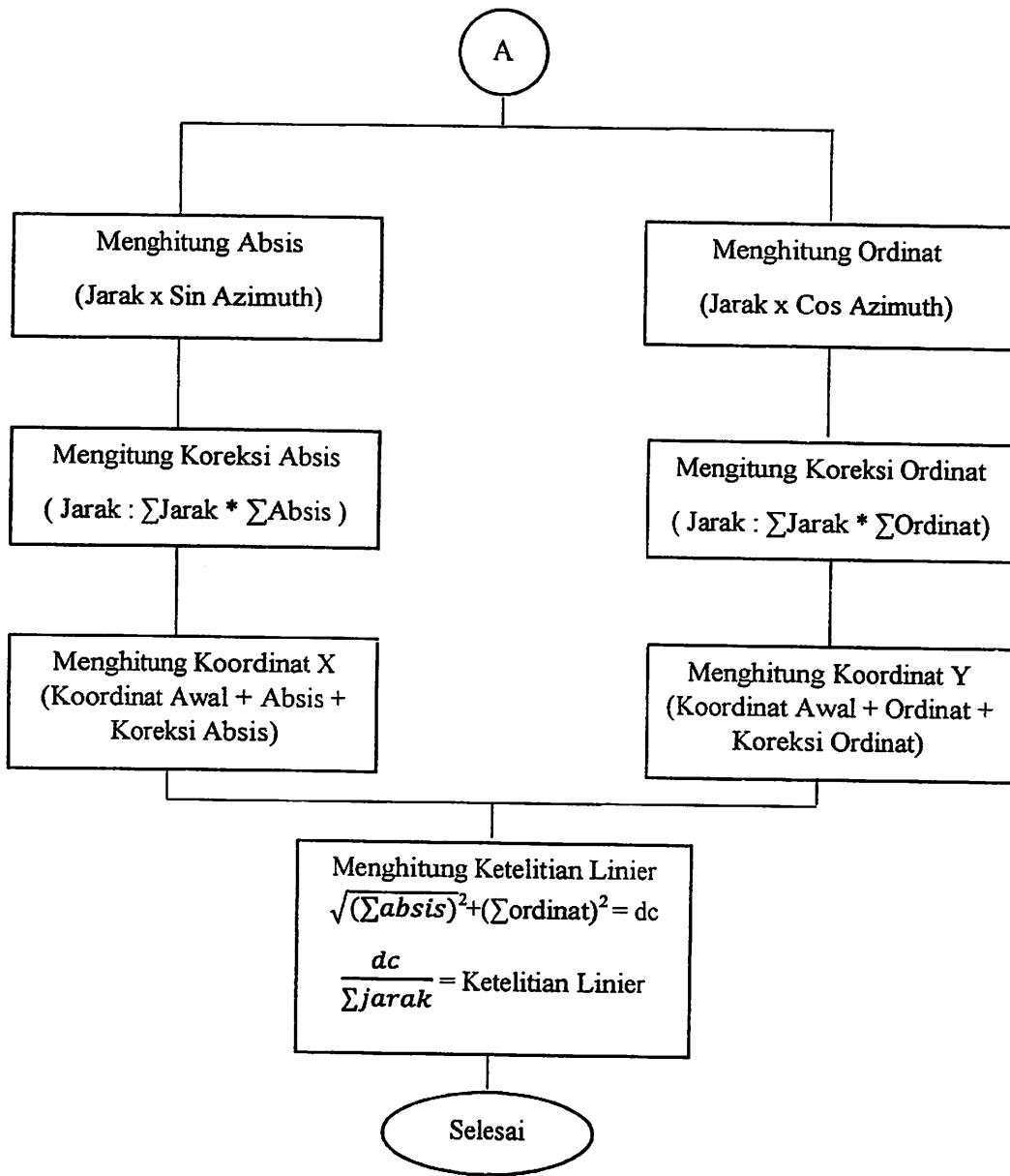
Pada tahap ini data hasil ukuran dilapangan di inputkan ke dalam program yang telah dibuat, agar dapat mengetahui kesalahan atau *system error* yang ada pada program yang telah dibuat.

5. Hasil

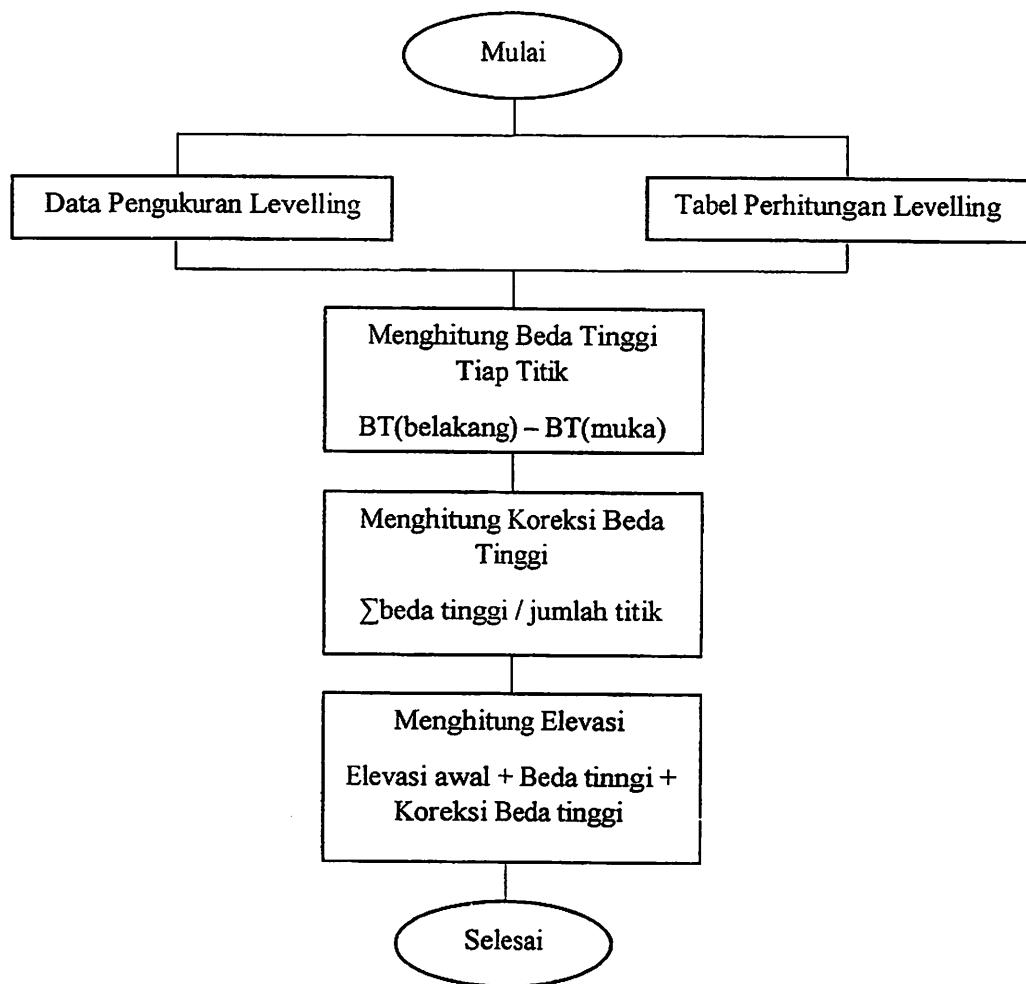
Bila program sudah bisa berjalan dengan baik, maka program bisa dikatakan sukses, dan apabila masih ada kekurangan dalam program maka kembali pada tahap perbaikan.

III.3 Diagram Alir Perhitungan Poligon Dengan Metode Bowditch

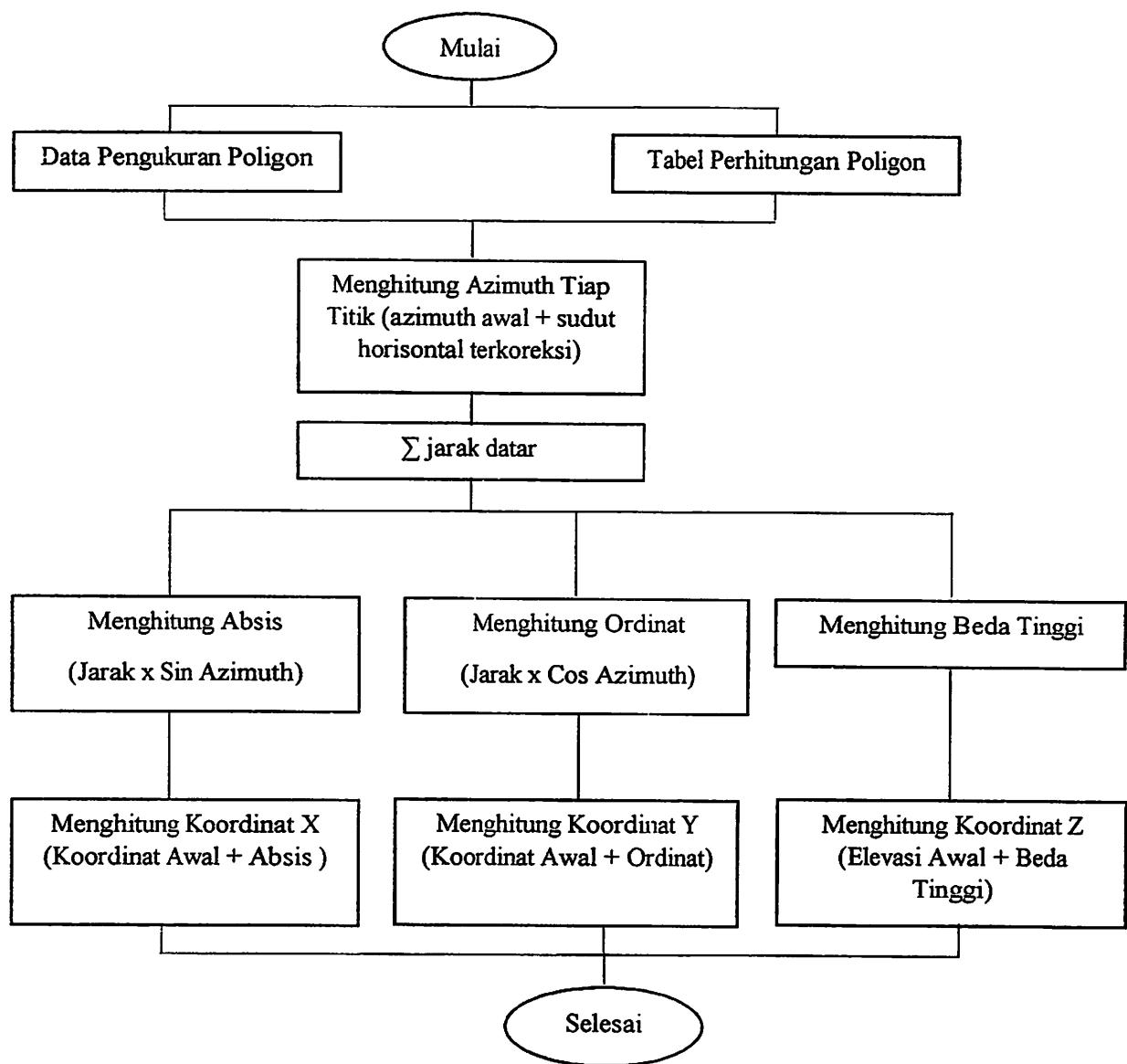




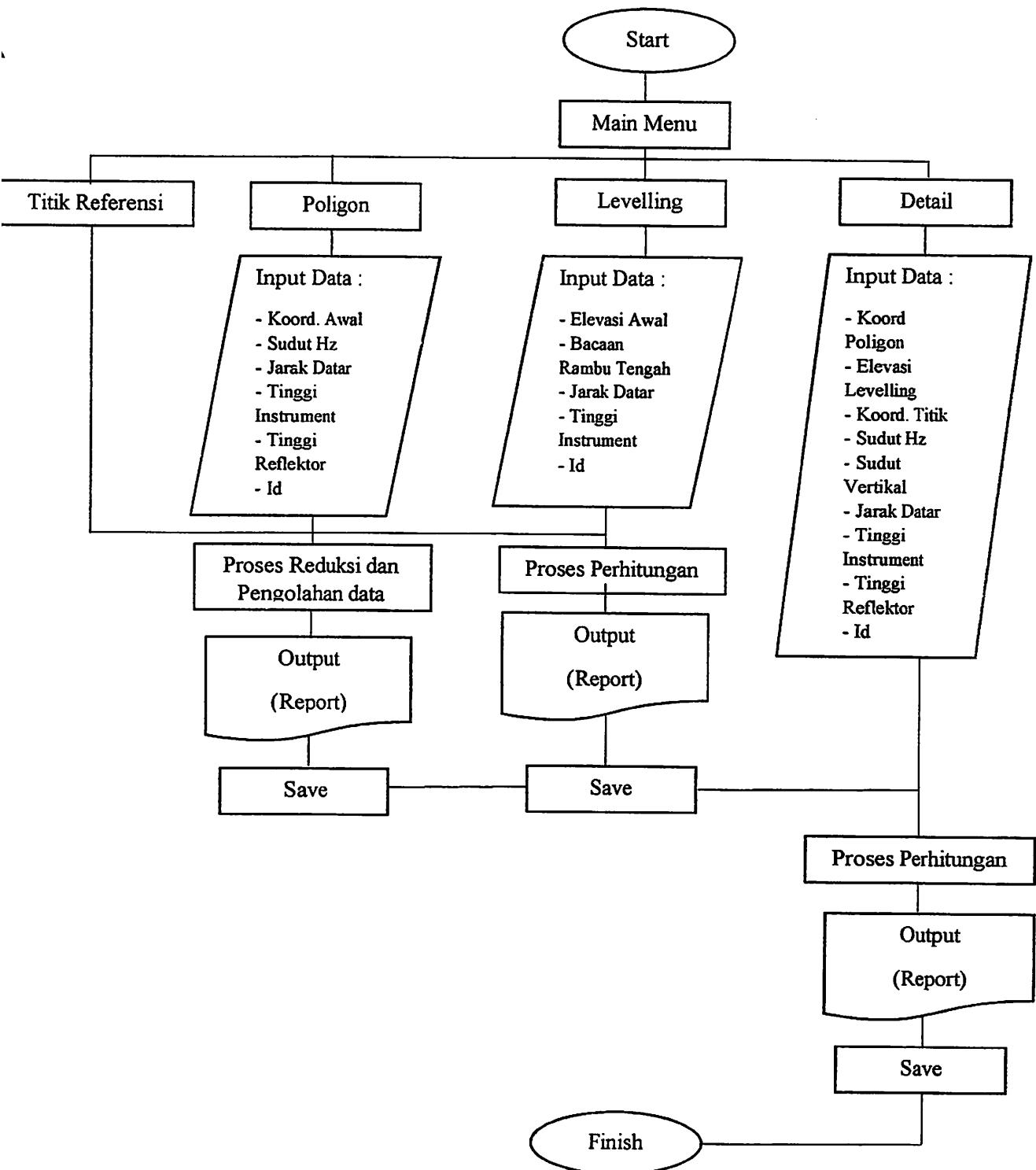
III.4 Diagram Alir Perhitungan Levelling



III.5 Diagram Alir Perhitungan Detail / Situasi



III.6 Diagram Alir Program



III.7 Pembuatan Program

Program ini memiliki tampilan menu utama yang terdiri dari Poligon , Levelling, Detail, Hasil perhitungan, Bantuan, Cetak report, dan Keluar. Setiap menu memiliki kemudahan baik dalam segi tampilan maupun pengoperasiannya sehingga program ini benar-benar dapat memudahkan penggunanya (*user friendly*).

III.7.1 Program data input

Dalam program ini ada 3 data input yang berbeda untuk setiap program perhitungan seperti dijelaskan pada table di bawah;

Data Input		
Poligon	Levelling	Detail
- Koord. Awal - Sudut Hz - Jarak Datar - Tinggi Instrument - Tinggi Reflektor - Id	- Elevasi Awal - Bacaan Rambu Tengah - Jarak Datar - Tinggi Instrument - Id	- Koord Poligon - Elevasi Levelling - Koord. Titik - Sudut Hz - Sudut Vertikal - Jarak Datar - Tinggi Instrument - Tinggi Reflektor - Id

Untuk menyusun data-data input setiap perhitungan dapat dilakukan dengan cara sebagai berikut :

1. Poligon : Data ditulis menggunakan Notepad dengan format file (*.txt) atau menggunakan Microsoft Excel dengan format (*.xls), aturan penulisan, nomer titik, deskripsi, tinggi alat, sudut horizontal dan titik referensi (koordinat awal).
2. Levelling : data ditulis menggunakan Notepad dengan format file (*.txt) dengan syarat penyusunan, nomer titik, titik target, bacaan rambu belakang, bacaan rambu depan, elevasi awal.
3. Detail : data ditulis menggunakan Notepad dengan format file (*.txt) dengan syarat penulisan nomer titik, deskripsi, tinggi alat, sudut horizontal dan titik referensi (koordinat awal).

III.7.2 Proses Perhitungan Program

Proses perhitungan untuk mendapatkan koordinat X,Y, dan Z dalam suatu pengukuran diawali dengan menghitung data poligon, levelling, kemudian detail, adapun langkahnya sebagai berikut :

1. Poligon

a. Menghitung jumlah sudut

$$f\delta = \Sigma \delta \text{ hasil pengukuran} - (n - 2) \cdot 180$$

Apabila selisih sudut tersebut masuk toleransi, maka perhitungan dapat dilanjutkan tetapi jika selisih sudut tersebut tidak masuk toleransi maka akan dilakukan cek lapangan atau pengukuran ulang.

b. Mengitung koreksi pada tiap-tiap sudut ukuran ($k\delta_i$)

$$k\delta_i = f\delta_i / n \quad (\text{jika kesalahan penutup sudut bertanda negatif } (-) \text{ maka koreksinya positif } (+), \text{ begitu juga sebaliknya.})$$

- c. Menghitung sudut terkoreksi

$$\delta_i = \delta_1 + k\delta_1$$

- d. Menghitung azimuth sisi poligon (α)

misal diketahui azimuth awal (α_{1-2})

$$\alpha_{2-3} = \alpha_{1-2} + 180^\circ - \delta_2 \text{ (untuk sudut dalam)}$$

$$\alpha_{2-3} = \alpha_{1-2} - 180^\circ + \delta_2 \text{ (untuk sudut luar)}$$

Dengan catatan, apabila azimuth lebih dari 360° , maka :

$$\alpha_{2-3} = (\alpha_{1-2} + 180^\circ - \delta_2) - 360^\circ$$

apabila azimuth kurang dari 0° , maka :

$$\alpha_{2-3} = (\alpha_{1-2} + 180^\circ - \delta_2) + 360^\circ$$

- e. Menghitung selisih absis dan selisih ordinat (ΔX dan ΔY)

$$\Delta X_{1-2} = d_{1-2} \cdot \sin \alpha_{1-2}$$

$$\Delta Y_{1-2} = d_{1-2} \cdot \cos \alpha_{1-2}$$

- f. Melakukan koreksi pada tiap-tiap kesalahan absis dan ordinat ($k\Delta X_i$ dan $k\Delta Y_i$)

$$k\Delta X_i = (d_i / \Sigma d) \cdot f\Delta X \quad \text{dalam hal ini } f\Delta X = \Sigma \Delta X$$

$$k\Delta Y_i = (d_i / \Sigma d) \cdot f\Delta Y \quad f\Delta Y = \Sigma \Delta Y$$

jika kesalahan absis dan ordinat bertanda negatif (-) maka koreksinya positif (+) begitu juga sebaliknya.

- g. Menghitung selisih absis (ΔX) dan ordinat (ΔY) terkoreksi

$$\Delta X_{1-2} = \Delta X_{1-2} + k\Delta X_{1-2}$$

$$\Delta Y_{1-2} = \Delta Y_{1-2} + k\Delta Y_{1-2}$$

- h. Koordinat (X,Y)

Misal diketahui koordinat awal (X_1 , Y_1) maka :

$$X_2 = X_1 + \Delta X_{1-2}$$

$$Y_2 = Y_1 + \Delta Y_{1-2}$$

Keterangan :

- $\Sigma\delta$ = jumlah sudut ukuran
n = jumlah titik pengukuran
 $f\delta$ = kesalahan penutup sudut ukuran
 $\Sigma\Delta X$ = jumlah selisih absis (X)
 $\Sigma\Delta Y$ = jumlah selisih ordinat (Y)
 $f\Delta X$ = kesalahan absis (X)
 $f\Delta Y$ = kesalahan ordinat (Y)
D = jarak / sisi poligon
 α = azimuth

2. Levelling

- a. Menghitung beda tinggi setiap titik: Beda tinggi setiap titik dihitung dengan cara mengurangi bacaan benang tengah rambu belakang dengan bacaan benang tengah rambu muka.
- b. Menghitung koreksi beda tinggi : Pada perhitungan koreksi beda tinggi dapat dilakukan dengan cara menjumlahkan semua beda tinggi kemudian membaginya dengan jumlah titik yang diukur .
- c. Menghitung Elevasi : Saat akan menghitung elevasi perlu diingat bahwa hal yang paling penting adalah kita harus mempunyai data elevasi awal , kemudian elevasi awal ditambahkan dengan beda tinggi dan koreksi beda tinggi.

3. Detail

- Menghitung azimuth
langkah awal dalam perhitungan detail / situasi adalah menghitung azimuth tiap titik dengan cara azimuth awal di tambahkan sudut horisontal +- 180 atau 360, dengan catatan :

- a. Bila hasil pertambahan azimuth awal dengan sudut horisontal kurang dari 180 maka ditambah 180
- b. Bila hasilnya lebih dari 180 maka di kurang 180
- c. Dan bila hasilnya lebih dari 360 maka di kurang 360
- Menghitung jarak datar
- Menghitung absis

Dalam menghitung absis dapat dilakukan dengan cara Menghitung Koordinat X (Koordinat Awal + Absis)

- Menghitung ordinat

Ordinat dapat dihitung dengan rumus (Jarak x Cos Azimuth)

- Menghitung beda tinggi
- Menghitung koordinat X
- Menghitung koordinat Y
- Menghitung koordinat Z

4. Prinsip kerja pada program ini adalah sebagai berikut :

- a. Melakukan pemilihan program yang akan digunakan, seperti poligon, levelling, atau detail.
- b. Melakukan input data yang telah disusun dalam format *.txt.
- c. Memasukkan informasi tentang pengukuran, seperti :
 - Waktu pengukuran yang berupa hari, tanggal, dan tahun.
 - Memasukkan tempat pengukuran untuk melengkapi report data.
 - Masukkan lokasi pengukuran.

BAB IV

HASIL DAN ANALISA

4.1 Hasil Penelitian

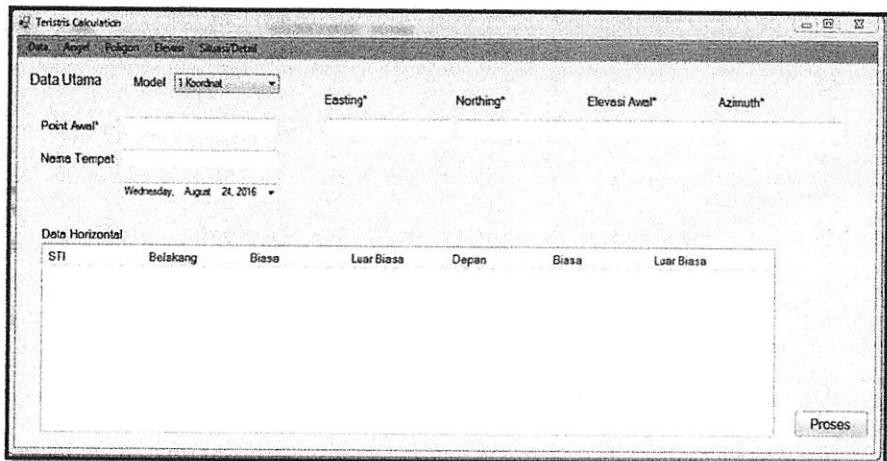
Dari hasil program yang dibuat maka dapat diketahui tampilan dari program adalah sebagai berikut:

1. Hasil data input

Data input dari program berformat ekstensi *.txt maka data mentah dari alat yang masih berformat RAW data di konversi menjadi txt dahulu kemudian di export ke dalam format txt.

2. Tampilan interface program

Tampilan interface program atau main menu sangat sederhana, hal ini dibuat agar lebih memudahkan pengguna dalam menjalankan program.



Gambar 4.1 tampilan program

3. Hasil data output

Data output yang dihasilkan pada program ini ada 2 , yang pertama program ini mampu mengeluarkan report data hasil ukuran, dan yang kedua program ini mampu mengeluarkan nilai koordinat dalam format *.txt.

4.2 Uji Ketelitian

Pengujian program dilakukan dengan cara membandingkan hasil pengolahan program perhitungan teristris dengan hasil pengolahan software Microsoft Excel menggunakan data yang sama. Dalam penelitian ini data yang digunakan meliputi :

1. Data poligon bacaan satu seri.
2. Data levelling menggunakan sifat datar
3. Data pengukuran detail / situasi.

4.2.1 Lokasi : Kampus 2 ITN Malang

Data : Poligon Tertutup

Perbandingan perhitungan hasil data poligon menggunakan program perhitungan teristris dengan software Microsoft Excel dapat dilihat dalam tabel di bawah ini:

Hasil report program perhitungan poligon dan Microsoft excel dapat dilihat pada *lampiran 1*

Tabel 4.1 Perbandingan antara program perhitungan teristris dengan Microsoft Excel

No	Keterangan	Hasil Analisa		Selisih
		Program Perhitungan Teristris	Perhitungan Microsoft Excel	
1	Total Sudut	359°59'51.97"	359°59'56.976"	-5.004"
2	Total Absis	-0.00908966	-0.00909609	0.00000643
3	Total Ordinat	0.00194089	-0.00202975	0.00397064
4	Ketelitian Linier	16417.61930	16373.17701	44.44229

Tabel 4.1 Perbandingan antara hasil koordinat program perhitungan teristris dengan Microsoft excel

No	Deskripsi Point	Hasil Analisa				Selisih (m)	
		Program Perhitungan Teristris (m)		Perhitungan Microsoft Excel (m)			
		X	Y	X	Y		
1	P1	679980.963	9124509.473	679980.964	9124509.473	-0.001	
2	P2	680013.890	9124536.057	680013.891	9124536.058	-0.001	
3	CP5	680005.255	9124572.049	680005.256	9124572.049	-0.001	
4	BM5	679974.733	9124528.957	679974.733	9124528.957	0	

4.2.2 Lokasi : Kampus 2 ITN Malang

Data : Levelling Memanjang Pulang Pergi

Perbandingan perhitungan hasil data levelling menggunakan program perhitungan teristris dengan software Microsoft Excel dapat dilihat dalam tabel di bawah ini:

Tabel 4.1 Perbandingan antara hasil perhitungan levelling menggunakan program perhitungan teristris dengan microsoft excel.

No	Keterangan	Hasil Analisa			Selisih
		Program Perhitungan Teristris	Perhitungan Microsoft Excel		
1	Beda Tinggi	-0.208	-0.208	0	
		-0.134	-0.134	0	
		0.135	0.135	0	
		0.205	0.205	0	
2	Sigma Beda Tinggi	-0.002	-0.002	0	

Tabel 4.1 Perbandingan elevasi hasil perhitungan levelling menggunakan program perhitungan teristris dengan microsoft excel.

No	Nama Titik	Hasil Analisa		Selisih
		Program Perhitungan Teristris	Perhitungan Microsoft Excel	
		Elevasi		
1	CP5	530.456	530.456	0
2	BM5	530.247	530.247	0
3	P1	530.113	530.113	0
4	P2	530.247	530.247	0

4.2.3 Lokasi : Kampus 2 ITN Malang

Data : Detail / Situasi

Perbandingan perhitungan koordinat x,y, dan z dari data pengukuran detail menggunakan program perhitungan teristris dengan software Microsoft Excel dapat dilihat dalam tabel di bawah ini:

Tabel 4.1 Perbandingan koordinat x,y, dan z antara program perhitungan dengan Microsoft excel.

No	Deskripsi Point	Hasil Analisa					
		Program Perhitungan Teristris (m)			Perhitungan Microsoft Excel (m)		
		X	Y	Z	X	Y	Z
1	Jl	680003.352	9124560.167	529.998	680003.351	9124560.169	529.998
2	Jl	680011.314	9124564.081	530.065	680011.313	9124564.083	530.065
3	Jl	680014.458	9124562.157	530.100	680014.458	9124562.159	530.100
4	Jl	680022.016	9124557.223	529.891	680022.015	9124557.225	529.892
5	Jl	679997.606	9124552.101	529.950	679997.605	9124552.102	529.951

6	Jl	680001.213	9124550.056	530.037	680001.212	9124550.058	530.038
7	Jl	680004.193	9124547.772	530.122	680004.193	9124547.774	530.123
8	Jl	680011.100	9124542.158	529.973	680011.100	9124542.158	529.974
9	Jl	679996.288	9124521.381	530.083	679996.288	9124521.381	530.083
10	Jl	679989.411	9124526.988	530.110	679989.411	9124526.988	530.111

No	Deskripsi Point	Selisih (m)		
		X	Y	Z
1	Jl	0.001	-0.002	0.000
2	Jl	0.001	-0.002	0
3	Jl	0	-0.002	0
4	Jl	0.001	-0.002	-0.001
5	Jl	0.001	-0.001	-0.001
6	Jl	0.001	-0.002	-0.001
7	Jl	0	-0.002	-0.001
8	Jl	0	0	-0.001
9	Jl	0	0	0
10	Jl	0	0	-0.001

4.3 Analisa Hasil Perhitungan Program

4.3.1 Poligon

Dari tabel perbandingan hasil perhitungan poligon di atas maka dapat dianalisa sebagai berikut :

- Hasil total sudut yang dihitung menggunakan program perhitungan teristris dengan Microsoft excel hampir mendekati sama hanya selisih -0.00139, selisih tersebut terjadi karena jumlah pembulatan angka di belakang koma.

2. Hasil penjumlahan absis yang dihitung oleh program perhitungan teristris dengan Microsoft excel apabila diambil 5 digit di belakang koma maka hasilnya sama namun bila 6 digit dibelakang koma maka hasilnya sudah berbeda, hal ini terjadi karena pembulatan angka dibelakang koma.
3. Hasil analisa penjumlahan ordinat yang dihitung dengan program perhitungan teristris dengan Microsoft excel sama seperti analisa pada ordinat, hanya selisih 0.003, hal ini terjadi juga karena pembulatan angka di belakang koma.
4. Hasil ketelitian linier yang dihasilkan program perhitungan teristris dengan Microsoft excel juga mengalami selisih, namun selisih yang didapat masih dalam konteks yang diizinkan karena hanya selisih 44.4 dari bilangan 16417.61930.

4.3.2 Levelling

1. Pada proses perhitungan levelling, semua hasil dari program perhitungan teristris dengan Microsoft excel sama dan tidak mengalami selisih sedikitpun, mulai dari beda tinggi, sigma beda tinggi, dan elevasi semua sama, hal ini terjadi karena proses perhitungan dalam levelling tidak terlalu rumit dan rumus yang digunakan sederhana.

4.3.3 Detail / Situasi

1. Sama halnya dengan analisa perhitungan poligon, pada perhitungan detail/situasi dalam program ini mengalami selisih, hal ini terjadi karena rumus atau pola perhitungan yang ada di dalam perhitungan detail/situasi ini sama dengan dengan perhitungan poligon, dan selisih terjadi karena pembulatan angka di belakang koma.

4.4 Hasil Analisa Program Data Input

Berdasarkan program data input yang telah dibuat, maka dapat dianalisa kelebihan dan kekurangan dari program tersebut, antara lain sebagai berikut:

- Kelebihan data input dalam program :
 1. Program dapat mencari dan memasukan file berekstensi *.txt dan *.excel.
 2. Data inputan merupakan data mentah dari alat dan langsung dapat diolah.
 3. Program langsung dapat mendeteksi jika ada kesalahan dalam penginputan data.
 4. Data input dapat terlihat di layar program sehingga apabila ada kesalahan data dapat langsung dilakukan pengeditan.
 5. Tampilan input data dibuat semudah mungkin agar pengguna lebih mudah dalam mengoperasikan program.
- Kekurangan data input dalam program :
 1. Tidak bisa menginputkan file dengan format ekstensi lain seperti *.csv dan lain-lain.
 2. Input data koordinat awal ataupun azimuth awal masih dilakukan secara manual.
 3. Masih ada ketidak sempurnaan dalam membaca data mentah sehingga perlu dilakukan penyempurnaan.

4.5 Hasil Analisa Proses Perhitungan Program

Berdasarkan program proses perhitungan yang telah dibuat, maka dapat dianalisa kelebihan dan kekurangan dari proses perhitungan program tersebut, antara lain :

- Kelebihan proses perhitungan dalam program :
 1. Program langsung melakukan proses melalui main program data inputan.
 2. Tidak perlu memasukkan faktor pengali secara manual karena semua faktor pengali sudah dimasukkan kedalam program.
 3. Tidak perlu menghitung azimuth awal karena program secara otomatis sudah dapat mengitung azimuth awal.

4. Perhitungan dalam program sudah sesuai dengan ketelitian yang ditoleransi.
 5. Program dapat menampilkan ketelitian yang diinginkan.
- Kekurangan Program :
 1. Program hanya dapat menghitung data poligon tertutup.
 2. Program hanya mampu membaca data satu seri.

4.6 Hasil Analisa Program Data Output

Berdasarkan program data output yang telah dibuat, maka dapat dianalisa kekurangan dan kelebihan dari program tersebut, anatara lain sebagai berikut :

- Kelebihan data output dalam program:
 1. Hasil dan proses perhitungan dapat langsung dilihat dilayar komputer setelah menekan pilihan perhitungan.
 2. Hasil perhitungan terdiri dari 3 macam yaitu hasil perhitungan poligon, levelling, dan detail.
 3. Setiap hasil perhitungan memiliki report sebagai laporan hasil perhitungan.
 4. Setiap hasil perhitungan, koordinat dapat disimpan dalam format *.txt sehingga dapat langsung di import kedalam software pemetaan.
- Kekurangan data output program
 1. Output program hanya bisa dicetak dengan ekstensi *.pdf.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang sudah dilakukan maka dapat diambil suatu kesimpulan sebagai berikut:

1. Berdasarkan analisa dari hasil program perhitungan teristris, dapat diketahui bahwa perhitungan program ini hampir sepenuhnya sesuai dengan perhitungan manual menggunakan Microsoft excel, hanya selisih pada pembulatan angka dibelakang koma.
2. Program yang dibuat memiliki kelebihan diantaranya adalah program dapat menampilkan data inputan di main menu dan dapat langsung diedit, sehingga lebih memudahkan pengguna dalam melakukan cek data.
3. Input data sangat *friendly* dan jumlah titik tidak dibatasi.
4. Terdapat output report data yang dihasilkan dari setiap hasil perhitungan.

5.2 Program mampu meng-export nilai koordinat x,y, dan z dalam format *.txt.Saran

Berdasarkan penelitian yang sudah dilakukan maka dapat diambil suatu saran sebagai berikut:

1. Penambahan metode perhitungan poligon seperti poligon terbuka kedepannya sangat dibutuhkan, mengingat dinamisnya pengukuran teristris saat ini.
2. Diharap kedepannya program ini dapat membaca data satu seri rangkap, untuk kepentingan yang sangat teliti.
3. Perlunya penambahan sket pengukuran levelling agar lebih memudahkan pengguna dalam melihat beda tinggi suatu titik.

LAMPIRAN

REPORT POLIGON

Tanggal : 8/22/2016

Tempat : ITN2

Koordinat Poligon (X,Y)

STI	X	Y
CP05	680005.248	9124572.044
BM05	679974.733	9124528.957
P01	679980.963813453	9124509.47308804
P02	680013.890493175	9124536.05770296
CP05	680005.255352889	9124572.04926918
BM05	679974.733	9124528.957

Sigma Sudut Terukur : 359.997777777778

Sudut Sebenarnya : 360

Kesalahan Penutupan Sudut : 0.0022222222222581

Sigma Absis : 0.00908966858284543

Sigma Ordinat : -0.00194089544972087

Ketelitian Linear : 16417.6193027846

No	Keterangan	Hasil Hitungan Poligon Pada Program
1	Total Sudut	359.99777
2	Total Absis	-0.00908966
3	Total Ordinat	0.00194089
4	Ketelitian Linier	16417.61930

No	Deskripsi Point	Hasil Hitungan Poligon Pada Program	
		X	Y
1	P1	679980.963	9124509.473
2	P2	680013.890	9124536.057
3	CP5	680005.255	9124572.049
4	BM5	679974.733	9124528.957

No	Keterangan	Hasil Hitungan Levelling Pada Program
1	Beda Tinggi	-0.208
		-0.134
		0.135
		0.205
2	Sigma Beda Tinggi	-0.002

No	Nama Titik	Hasil Hitungan Levelling Pada Program	
		Elevasi	
1	CP5	530.456	
2	BMS	530.247	
3	P1	530.113	
4	P2	530.247	

Hasil Hitungan Detail Pada Program

no	x	y	z	deskripsi
1	679974.733	9124528.957	530.2475	BM5
2	679974.733	9124528.957	532.248	JL
3	679974.733	9124528.957	533.248	JL
4	679974.733	9124528.957	534.248	JL
5	679974.733	9124528.957	535.248	JL
6	679974.733	9124528.957	536.248	JL
7	679974.733	9124528.957	537.248	JL
8	679974.733	9124528.957	538.248	JL
9	679974.733	9124528.957	539.248	JL
10	679974.733	9124528.957	540.248	JL
11	679974.733	9124528.957	541.248	JL
12	679974.733	9124528.957	542.248	JL
13	679974.733	9124528.957	543.248	JL
14	679974.733	9124528.957	544.248	JL
15	679974.733	9124528.957	545.248	JL
16	679974.733	9124528.957	546.248	JL
17	679974.733	9124528.957	547.248	PG
18	679974.733	9124528.957	548.248	PG
19	679974.733	9124528.957	549.248	PG
20	679974.733	9124528.957	550.248	PG
21	679974.733	9124528.957	551.248	PG
22	679974.733	9124528.957	552.248	PG
23	679974.733	9124528.957	553.248	PG
24	679974.733	9124528.957	554.248	PG
25	679974.733	9124528.957	555.248	PG
26	679974.733	9124528.957	556.248	PG
27	679974.733	9124528.957	557.248	PG
28	679974.733	9124528.957	558.248	SH
29	679974.733	9124528.957	559.248	SH
30	679974.733	9124528.957	560.248	SH
31	679974.733	9124528.957	561.248	SH
32	679974.733	9124528.957	562.248	SH
33	679974.733	9124528.957	563.248	SH
34	679974.733	9124528.957	564.248	SH
35	679974.733	9124528.957	565.248	SH
36	679974.733	9124528.957	566.248	SH
37	679974.733	9124528.957	567.248	SH
38	679980.964	9124509.473	530.153	P01
39	679980.964	9124509.473	569.153	SH
40	679980.964	9124509.473	570.153	SH
41	679980.964	9124509.473	571.153	SH
42	679980.964	9124509.473	572.153	SH
43	679980.964	9124509.473	573.153	SH

HASIL HITUNGAN MICROSOFT EXCEL

No.	Tinggi alat	STA	Hor.angle			
			d	m	s	deg
1	1.212	CP5				
		BM5				
		P1	126	57	22	126.9561111
		P2	68	49	21	68.8225000
		CP5	115	25	32	115.4255556
		BM5	48	47	42	48.7950000
2	1.52	CP5				
		BM5				
		P1	126	57	22	126.9561111
		P2	68	49	21	68.8225000
		CP5	115	25	32	115.4255556
		BM5	48	47	42	48.7950000
3	1.405	CP5				
		BM5				
		P1	126	57	22	126.9561111
		P2	68	49	21	68.8225000
		CP5	115	25	32	115.4255556
		BM5	48	47	42	48.7950000
4	1.21	CP5				
		BM5				
		P1	126	57	22	126.9561111
		P2	68	49	21	68.8225000
		CP5	115	25	32	115.4255556
		BM5	48	47	42	48.7950000

(1)

SUM

359 | 59 | 57

359.9991667

(2)

 $(n+2) \times 180 =$

2 | X | 180

360.0000000

(3)

 $(\alpha_{Start} - \alpha_{End}) - (\sum \beta - n \cdot 180)$

0.0008333

ini

Corr	Hor.Angle	Azimuth		Azimuth		slop Distance	disten	prisma
deg	deg	deg	d	m	s	m		
0.0002083	126.9563194	215.3071363	215	18	26	20.457	20.457	1.158
0.0002083	68.8227083	162.2634557	162	15	48	42.322	42.321	1.405
0.0002083	115.4257639	51.0861640	051	05	10	37.012	37.012	1.21
0.0002083	48.7952083	346.5119279	346	30	43	52.806	52.806	1.195
		215.3071363	215	18	26			

0.0008333 360.0000000
360.0000000
0.0000000

Total :

152.595

Digunakan Tidak?

	vertikal			DE	Corr.DE	DN	Corr.DN	DZ	Corr.DZ	Easting
d	m	s		m	m	m	m		m	m
90	22	19.5	90.3720833	6.232	-0.001	-19.484	0.000	-0.0789	-0.041	680005.248
89	33	33	89.5591667	32.929	-0.003	26.584	0.001	0.441	-0.086	679974.733
90	3	52	90.0644444	-8.633	-0.002	35.991	0.000	0.155	-0.075	679980.964
90	14	44	90.2455556	-30.519	-0.003	-43.093	0.001	-0.208	-0.107	680013.891
										680005.256

0.009 -0.009 -0.002 0.002 0.309 -0.309 679974.733

0.000

-0.009 0.002 -0.309

f(d) = S

0.000

0.001

FORMULIR SIPAT DATAR MEMANJANG PERGI

Laboratorium Ilmu Ukur Tanah Institut Teknologi Nasional Malang		LEVELLING	Halaman :				
			Lokasi : ITN 2				
Diukur		Instrumen No.	Tanggal	Diperiksa			
Kelompok							
No.Titik	Target	Pembacaan		Δh	Koreksi	Elevasi	Jarak
		Rambu Belakang	Rambu Muka				Keterangan
I	CP05	1.181					
		1.044				530.456	27.3
		0.908					pergi
	BM05		1.393				
II	BM05		1.252	-0.208	-0.00050	530.248	28.2
			1.111				
		0.993					
	P01	0.957					7.2
		0.921					pergi
III	P01		1.164				
			1.091	-0.134	-0.00050	530.113	14.5
			1.019				
	P02	1.712					
		1.604					21.400
		1.498					pergi
IV	P02		1.567				
			1.469	0.135	-0.00050	530.248	19.600
			1.371				
	CP05	1.417					
		1.324					18.6
		1.231					pergi
				sigma =	-0.002		

sigma jarak= 155.4

SKET GAMBAR

FORMULIR SIPAT DATAR MEMANJANG PERGI

Laboratorium Ilmu Ukur Tanah Institut Teknologi Nasional Malang		LEVELLING	Halaman :				
			Lokasi : ITN 2				
Diukur		Instrumen No.	Tanggal			Diperiksa	
Kelompok							
No.Titik	Target	Pembacaan		Δh	Koreksi	Elevasi	Jarak
		Rambu Belakang	Rambu Muka				
I	CP05	1.109					
		1.015					530.452 18.8
		0.921					
	P02		1.312				
				1.221 -0.206	0.00050	530.247	17.3
II	P02	1.650					
		1.553					19.4
		1.456					
	P01		1.797				
				1.688 -0.135	0.00050	530.112	21.6
III	P01	1.352					
		1.280					14.400
		1.208					
	BM05		1.183				
				1.148 0.132	0.00050	530.245	7.200
IV	BM05		1.111				
		1.240					
	CP05	1.110					28.0
		0.960					
			1.047				
				0.899 0.211	0.00050	530.456	31.5
			0.732				
				sigma= 0.002			
					0.002		

sigma jarak = 158.2

Azimuth	Jarak Miring	Jarak Datar	Tinggi Prisma	Bacaan Ver
Azimuth	Miring	Datar	Prisma	
162	15	48.44	162.2634557	
140	37	51.44	140.63096	13.603
146	37	48.44	146.63012	5.863
162	49	49.44	162.83040	16.408
232	22	28.44	232.37457	4.141
153	29	43.44	153.49540	10.214
172	26	43.44	172.44540	18.698
180	59	3.44	180.98429	14.21
262	41	39.44	262.69429	5.731
132	31	40.44	132.52790	10.241
109	48	54.44	109.81512	9.767
183	35	43.44	183.59540	14.809
99	33	31.44	99.55873	6.864
293	55	29.44	293.92484	7.689
192	32	10.44	192.53623	32.594
83	7	35.44	83.12651	6.7
111	0	48.44	111.01346	13.377
187	43	39.44	187.72762	13.249
183	36	18.44	183.60512	14.037
191	41	19.44	191.68873	30.179
300	10	47.44	300.17984	10.517
197	52	44.44	197.87901	17.702
200	10	0.44	200.16679	17.123
191	21	24.44	191.35679	26.771
189	20	27.44	189.34096	20.774
204	48	28.44	204.80790	20.144
208	54	14.44	208.90401	15.141
230	33	41.44	230.56151	16.994
207	53	15.44	207.88762	15.394
207	19	25.44	207.32373	28.252
299	9	49.44	299.16373	18.416
243	57	2.44	243.95068	20.07

205	55	48.44	205.93012	26.362	26.133	1.6	97	33
209	41	32.44	209.69234	12.869	11.544	1.6	116	13
270	13	54.44	270.23179	17.709	17.142	1.6	104	32
273	47	18.44	273.78846	22.24	21.792	1.6	101	31
234	42	38.44	234.71068	24.251	24.082	1.6	96	45
256	25	57.44	256.43262	26.399	26.251	1.6	96	3
224	44	7.44	224.73540	13.016	11.770	1.6	115	16
232	35	27.44	232.59096	24.282	24.195	1.6	94	51
258	21	0.44	258.35012	26.606	26.411	1.6	96	56
256	7	17.44	256.12151	29.065	28.993	1.6	94	1
244	46	24.44	244.77346	17.436	16.650	1.6	107	16
252	8	10.44	252.13623	29.825	29.785	1.6	92	58
261	10	28.44	261.17457	27.728	27.431	1.6	98	23
261	4	26.44	261.07401	27.758	27.464	1.6	98	20
245	56	4.44	245.93457	6.286	5.967	2.55	108	19
249	54	19.44	249.90540	42.064	42.050	1.6	89	11
260	19	59.44	260.33318	24.197	23.831	2.55	99	59
200	22	5.44	200.36818	50.118	50.117	1.6	89	39
197	26	13.44	197.43707	41.038	41.036	1.6	89	29
197	18	39.44	197.31096	40.128	40.127	1.6	90	25
261	2	35.44	261.04318	24.613	23.967	2.55	103	9
211	49	42.44	211.82846	39.704	39.702	1.6	89	30
219	53	58.44	219.89957	41.322	41.319	1.6	89	19
219	58	49.44	219.98040	40.846	40.845	1.6	90	20
190	4	44.44	190.07901	13.67	12.602	2.55	112	47
230	33	57.44	230.56596	42.898	42.895	1.6	89	20
230	34	0.44	230.56679	42.805	42.804	1.6	89	40
230	36	0.44	230.60012	42.607	42.607	1.6	89	47
198	3	53.44	198.06484	16.112	15.236	2.55	108	58
236	56	44.44	236.94568	44.156	44.150	1.6	89	4
180	7	54.44	180.13179	23.11	22.460	2.55	103	37
179	33	23.44	179.55651	18.196	17.457	2.55	106	23
178	33	30.44	178.55846	17.806	17.278	2.55	103	59
167	56	28.44	167.94123	19.335	19.070	2.55	99	29

172	34	24.44	172.57346
209	40	7.44	209.66873
209	51	50.44	209.86401
228	15	14.44	228.25401
228	29	39.44	228.49429
244	12	45.44	244.21262
244	48	26.44	244.80734
260	11	37.44	260.19373
247	43	29.44	247.72484
247	51	41.44	247.86151
246	26	28.44	246.44123
234	10	35.44	234.17651
234	42	8.44	234.70234
216	34	29.44	216.57484
216	6	56.44	216.11568
206	39	8.44	206.65234
204	51	57.44	204.86596

20.94	20.294
32.403	32.385
31.551	31.485
32.016	32.001
31.618	31.576
34.561	34.559
34.176	34.146
18.783	18.075
28.945	28.892
28.756	28.619
5.886	5.693
24.685	24.613
24.381	24.215
25.612	25.530
24.845	24.597
28.382	28.306
27.035	26.822

2.55		104	15
1.6		91	56
1.6		93	41
1.6		91	45
1.6		92	57
1.6		90	35
1.6		92	24
2.55		105	46
1.6		93	27
1.6		95	35
4.5		104	42
1.6		94	22
1.6		96	41
1.6		94	34
1.6		98	6
1.6		94	12
1.6		97	11

No.	Ttk	Elevasi Z	Koordinat Y	X	Beda Tinggi	Ordinat J'x Cos A	Absis J'x Sin A	Sudut Vertical	Koordinat	
									X	Y
						679980.964	9124509.473	530.153	P01	
26	-6.92389	8.629	-10.516	-1.732	679989.592	9124498.957	528.421	SH		
47	-16.16306	3.225	-4.896	-1.779	679984.188	9124504.577	528.374	SH		
31	-12.07528	4.844	-15.677	-3.590	679985.807	9124493.797	526.563	SH		
35	-31.24306	-2.804	-2.161	-2.228	679978.160	9124507.312	527.925	SH		
22	-12.70611	4.447	-8.917	-2.327	679985.410	9124500.556	527.826	SH		
26	-13.62389	2.389	-18.014	-4.484	679983.353	9124491.459	525.668	SH		
32	-16.24222	-0.234	-13.641	-4.055	679980.729	9124495.832	526.098	SH		
50	-25.04722	-5.150	-0.660	-2.506	679975.814	9124508.813	527.646	SH		
5	-9.90139	7.435	-6.819	-1.841	679988.398	9124502.654	528.312	SH		
30	-4.35833	9.162	-3.301	-0.822	679990.126	9124506.172	529.331	SH		
14	-18.42056	-0.881	-14.023	-4.759	679980.083	9124495.451	525.393	SH		
52	-3.53111	6.756	-1.138	-0.503	679987.720	9124508.335	529.650	SH		
12	-15.28667	-6.780	3.008	-2.107	679974.184	9124512.481	528.046	SH		
55	-3.36528	-7.063	-31.762	-1.993	679973.901	9124477.711	528.159	SH		
27	0.74250	6.651	0.802	0.007	679987.615	9124510.275	530.160	SH		
43	-1.27861	12.484	-4.796	-0.378	679993.448	9124504.677	529.774	SH		
7	-21.31861	-1.660	-12.230	-4.897	679979.304	9124497.243	525.256	SH		
20	-19.80556	-0.830	-13.181	-4.836	679980.133	9124496.293	525.317	SH		
56	-6.68222	-6.073	-29.352	-3.592	679974.891	9124480.121	526.561	SH		
2	-12.13389	-8.888	5.169	-2.291	679972.075	9124514.642	527.862	SH		
34	-17.20944	-5.191	-16.093	-5.317	679975.772	9124493.380	524.835	SH		
10	-16.91944	-5.648	-15.378	-5.063	679975.316	9124494.096	525.089	SH		
45	-8.01250	-5.220	-25.991	-3.812	679975.743	9124483.482	526.341	SH		
2	-15.81722	-3.244	-19.722	-5.742	679977.720	9124489.751	524.410	SH		
21	-9.65583	-8.332	-18.026	-3.459	679972.631	9124491.447	526.694	SH		
33	-20.74250	-6.844	-12.396	-5.442	679974.120	9124497.077	524.710	SH		
2	-11.66722	-12.853	-10.572	-3.517	679968.110	9124498.901	526.636	SH		
33	-23.22583	-6.617	-12.504	-6.151	679974.347	9124496.970	524.002	SH		
1	-4.06694	-12.936	-25.037	-2.084	679968.028	9124484.436	528.069	SH		
42	-7.31167	-15.951	8.901	-2.424	679965.013	9124518.374	527.729	SH		
18	-9.13833	-17.802	-8.702	-3.267	679963.161	9124500.771	526.885	SH		

50	-7.56389	-11.427	-23.502	-3.550	679969.537	9124485.971	526.603	SH
39	-26.22750	-5.718	-10.028	-5.767	679975.245	9124499.445	524.385	SH
37	-14.54361	-17.141	0.069	-4.527	679963.822	9124509.542	525.626	SH
32	-11.52556	-21.744	1.440	-4.524	679959.220	9124510.913	525.629	SH
32	-6.75889	-19.657	-13.913	-2.934	679961.306	9124495.561	527.219	SH
40	-6.06111	-25.519	-6.158	-2.867	679955.445	9124503.315	527.285	SH
12	-25.27000	-8.284	-8.361	-5.636	679972.679	9124501.112	524.516	SH
48	-4.86333	-19.218	-14.698	-2.139	679961.745	9124494.775	528.014	SH
39	-6.94417	-25.867	-5.333	-3.297	679955.097	9124504.140	526.856	SH
48	-4.03000	-28.147	-6.954	-2.123	679952.817	9124502.519	528.030	SH
19	-17.27194	-15.062	-7.096	-5.257	679965.902	9124502.377	524.896	SH
10	-2.96944	-28.349	-9.137	-1.625	679952.615	9124500.336	528.528	SH
19	-8.38861	-27.107	-4.209	-4.125	679953.857	9124505.264	526.028	SH
43	-8.34528	-27.131	-4.261	-4.109	679953.832	9124505.212	526.044	SH
43	-18.32861	-5.448	-2.433	-3.007	679975.515	9124507.040	527.146	SH
52	0.80222	-39.500	-14.451	0.509	679941.464	9124495.023	530.662	SH
20	0.34444	-17.443	-46.984	0.221	679963.520	9124462.490	530.374	SH
0	-9.98333	-23.492	-4.002	-5.225	679957.471	9124505.471	524.928	SH
53	0.50194	-12.297	-39.151	0.280	679968.667	9124470.322	530.432	SH
17	-0.42139	-11.940	-38.309	-0.375	679969.024	9124471.164	529.778	SH
34	-13.15944	-23.674	-3.731	-6.633	679957.289	9124505.742	523.519	SH
7	0.49806	-20.938	-33.732	0.265	679960.025	9124475.741	530.418	SH
22	0.67722	-26.504	-31.699	0.408	679954.460	9124477.774	530.561	SH
33	-0.34250	-26.244	-31.298	-0.324	679954.720	9124478.175	529.829	SH
48	-22.79667	-2.205	-12.408	-6.327	679978.758	9124497.065	523.826	SH
3	0.66583	-33.130	-27.247	0.419	679947.833	9124482.227	530.571	SH
16	0.32889	-33.061	-27.188	0.166	679947.903	9124482.285	530.318	SH
6	0.21500	-32.924	-27.044	0.080	679948.040	9124482.429	530.233	SH
38	-18.97722	-4.725	-14.485	-6.269	679976.239	9124494.988	523.883	SH
24	0.92667	-37.005	-24.081	0.634	679943.959	9124485.392	530.787	SH
1	-13.61694	-0.052	-22.460	-6.471	679980.912	9124487.013	523.682	SH
0	-16.38333	0.135	-17.457	-6.162	679981.099	9124492.016	523.990	SH
28	-13.99111	0.435	-17.272	-5.335	679981.398	9124492.201	524.818	SH
33	-9.49250	3.984	-18.649	-4.219	679984.948	9124490.824	525.934	SH

48	-14.26333	2.623	-20.124	-5.189	679983.587	9124489.349	523.964	SH
4	-1.93444	-16.030	-28.139	-1.174	679964.934	9124481.334	528.979	SH
35	-3.69306	-15.678	-27.305	-2.112	679965.286	9124482.169	528.041	SH
36	-1.76000	-23.876	-21.307	-1.063	679957.088	9124488.166	529.089	SH
12	-2.95333	-23.647	-20.925	-1.709	679957.317	9124488.548	528.444	SH
35	-0.59306	-31.118	-15.034	-0.438	679949.846	9124494.439	529.715	SH
34	-2.40944	-30.898	-14.535	-1.517	679950.066	9124494.938	528.636	SH
50	-15.78056	-17.811	-3.078	-6.138	679963.153	9124506.395	524.015	SH
33	-3.45917	-26.736	-10.952	-1.826	679954.228	9124498.521	528.326	SH
36	-5.59333	-26.509	-10.785	-2.883	679954.455	9124498.688	527.270	SH
15	-14.70417	-5.219	-2.276	-4.474	679975.745	9124507.198	525.679	SH
44	-4.37889	-19.957	-14.406	-1.965	679961.007	9124495.067	528.188	SH
21	-6.68917	-19.763	-13.992	-2.920	679961.200	9124495.481	527.233	SH
57	-4.58250	-15.213	-20.503	-2.126	679965.751	9124488.970	528.026	SH
32	-8.10889	-14.498	-19.870	-3.585	679966.466	9124489.603	526.568	SH
17	-4.20472	-12.697	-25.298	-2.161	679968.266	9124484.175	527.992	SH
28	-7.19111	-11.279	-24.336	-3.464	679969.685	9124485.137	526.689	SH

No	Azimuth Datar	Azimuth Miring	Jarak Datar	Jarak Miring	Tinggi Prisma	Bacaan Ver
42	31	3.69	42.51769	42.346	42.346	1.6
46	9	42.69	46.16186	50.715	50.715	1.6
50	6	41.69	50.11158	51.773	51.772	1.6
59	7	35.69	59.12658	55.088	55.088	1.6
44	39	37.69	44.66047	32.54	32.540	1.6
51	26	58.69	51.44964	33.859	33.859	1.6
57	25	57.69	57.43269	34.957	34.956	1.6
70	3	1.69	70.05047	38.689	38.689	1.6
109	21	53.69	109.36491	22.849	22.848	1.6
97	38	27.69	97.64103	14.812	14.810	1.6
88	44	37.69	88.74380	11.609	11.608	1.6
65	31	53.69	65.53158	7.737	7.737	1.6
95	25	1.69	95.41714	10.923	10.921	1.6
102	35	17.69	102.58825	12.075	12.073	1.6
101	57	23.69	101.95658	14.224	14.222	1.6
115	3	17.69	115.05491	29.632	29.631	1.6
111	12	59.69	111.21658	24.37	24.369	1.6
110	51	47.69	110.86325	23.939	23.935	1.6
70	30	24.69	70.50686	49.13	49.130	1.6
65	42	54.69	65.71519	47.41	47.410	1.6
65	7	30.69	65.12519	47.256	47.256	1.6
204	42	26.69	204.70741	15.258	15.248	1.6
209	6	25.69	209.10714	26.115	26.109	1.6
36	40	9.69	36.66936	21.664	21.663	1.6
35	56	46.69	35.94630	38.511	38.510	1.6
229	17	27.69	229.29103	11.575	11.568	1.6
211	56	28.69	211.94130	10.458	10.421	1.6
165	46	11.69	165.76991	20.2	20.198	1.6
184	7	13.69	184.12047	12.418	12.397	1.6
174	24	58.69	174.41630	14.629	14.602	1.6
228	22	45.69	228.37936	11.248	11.161	1.6

234	34	13.69	234.57047	4.777	4.775	1.6	91	49
170	15	21.69	170.25603	6.975	6.975	1.6	90	7
157	28	9.69	157.46936	11.922	11.922	1.6	90	14
151	50	56.69	151.84908	14.768	14.767	1.6	89	23

No.	Sudut Vertical	Absis $J \times \sin A$	Ordinat $J \times \cos A$	Beda Tinggi	Koordinat		Elevasi Z	No. Ttk
					X	Y		
							530.2475	BM5
45	0.18750	28.618	31.212	-0.249	680003.351	9124560.169	529.998	JL
3	0.23250	36.580	35.126	-0.182	680011.313	9124564.083	530.065	JL
0	0.26667	39.725	33.201	-0.147	680014.458	9124562.159	530.100	JL
0	0.03333	47.282	28.268	-0.356	680022.015	9124557.225	529.892	JL
22	0.16056	22.872	23.145	-0.297	679997.605	9124552.102	529.951	JL
55	0.30139	26.479	21.101	-0.210	680001.212	9124550.058	530.038	JL
6	0.43167	29.460	18.816	-0.125	680004.193	9124547.774	530.123	JL
52	0.16889	36.367	13.200	-0.274	680011.100	9124542.158	529.974	JL
18	0.56167	21.555	-7.576	-0.164	679996.288	9124521.381	530.083	JL
41	0.97194	14.678	-1.969	-0.137	679989.411	9124526.988	530.111	JL
22	0.89389	11.605	0.254	-0.207	679986.338	9124529.212	530.041	JL
12	0.59667	7.042	3.204	-0.307	679981.775	9124532.162	529.940	JL
52	1.00222	10.873	-1.031	-0.197	679985.606	9124527.926	530.051	JL
19	1.12806	11.782	-2.631	-0.150	679986.515	9124526.326	530.097	JL
2	0.88278	13.914	-2.946	-0.169	679988.647	9124526.011	530.079	JL
43	0.43806	26.843	-12.548	-0.161	680001.576	9124516.409	530.086	PG
53	-0.41472	22.718	-8.819	-0.564	679997.451	9124520.138	529.683	PG
23	1.02694	22.366	-8.524	0.041	679997.099	9124520.433	530.289	PG
2	0.18278	46.314	16.394	-0.231	680021.047	9124545.352	530.016	PG
13	-0.17028	43.215	19.498	-0.529	680017.948	9124548.456	529.719	PG
0	0.23333	42.872	19.877	-0.196	680017.605	9124548.835	530.052	PG
7	2.11472	-6.373	-13.852	0.175	679968.360	9124515.106	530.423	PG
41	1.22194	-12.701	-22.812	0.169	679962.032	9124506.146	530.416	PG
9	0.48083	12.937	17.376	-0.206	679987.670	9124546.333	530.041	PG
2	0.31611	22.607	31.177	-0.176	679997.340	9124560.134	530.072	PG
47	1.98694	-8.769	-7.545	0.013	679965.964	9124521.412	530.261	PG
11	-4.85306	-5.513	-8.843	-1.273	679969.220	9124520.115	528.975	SH
59	-0.74972	4.965	-19.579	-0.652	679979.698	9124509.379	529.595	SH
19	-3.35528	-0.891	-12.365	-1.115	679973.842	9124516.593	529.133	SH
47	-3.51306	1.421	-14.532	-1.284	679976.154	9124514.425	528.963	SH
48	-7.14667	-8.343	-7.413	-1.787	679966.390	9124521.544	528.460	SH

-	679981.7	9124515.936	530.015	SH
-	679984.769	9124506.721	529.317	SH
P01	679980.964	9124509.473	530.113	P01
-	679989.593	9124498.957	528.381	SH
-	679984.189	9124504.577	528.334	SH
-	679985.808	9124493.797	526.523	SH
-	679978.16	9124507.311	527.885	SH
-	679985.411	9124500.557	527.786	SH
-	679983.354	9124491.459	525.629	SH
-	679980.73	9124495.832	526.058	SH
-	679975.814	9124508.813	527.607	SH
-	679988.399	9124502.654	528.272	SH
-	679990.126	9124506.172	529.291	SH
-	679980.084	9124495.45	525.354	SH
-	679987.72	9124508.336	529.61	SH
-	679974.184	9124512.481	528.006	SH
-	679973.903	9124477.711	528.12	SH
-	679987.615	9124510.275	530.12	SH
-	679993.448	9124504.678	529.735	SH
-	679979.305	9124497.243	525.216	SH
-	679980.134	9124496.292	525.277	SH
-	679974.893	9124480.12	526.521	SH
-	679972.075	9124514.641	527.822	SH
-	679975.773	9124493.38	524.796	SH
-	679975.317	9124494.095	525.05	SH
-	679975.745	9124483.482	526.301	SH
-	679977.721	9124489.751	524.371	SH
-	679972.633	9124491.447	526.654	SH
-	679974.121	9124497.077	524.671	SH
-	679968.111	9124498.9	526.596	SH
-	679974.348	9124496.969	523.962	SH
-	679968.03	9124484.436	528.029	SH
-	679965.013	9124518.373	527.689	SH
-	679963.162	9124500.77	526.846	SH
-	679969.538	9124485.971	526.563	SH
-	679975.246	9124499.444	524.346	SH
-	679963.822	9124509.541	525.586	SH
-	679959.22	9124510.912	525.589	SH
-	679961.307	9124495.559	527.179	SH
-	679955.445	9124503.313	527.246	SH
-	679972.68	9124501.111	524.477	SH
-	679961.746	9124494.774	527.974	SH
-	679955.097	9124504.138	526.816	SH
-	679952.818	9124502.517	527.99	SH
-	679965.902	9124502.376	524.856	SH

-	679952.615	9124500.335	528.488	SH
-	679953.857	9124505.263	525.988	SH
-	679953.833	9124505.21	526.004	SH
-	679975.516	9124507.039	527.106	SH
-	679941.465	9124495.02	530.622	SH
-	679963.523	9124462.488	530.334	SH
-	679957.472	9124505.47	524.888	SH
-	679968.669	9124470.322	530.393	SH
-	679969.026	9124471.163	529.738	SH
-	679957.29	9124505.74	523.48	SH
-	679960.028	9124475.739	530.378	SH
-	679954.462	9124477.773	530.521	SH
-	679954.722	9124478.173	529.789	SH
-	679978.759	9124497.065	523.786	SH
-	679947.835	9124482.225	530.532	SH
-	679947.905	9124482.283	530.279	SH
-	679948.042	9124482.427	530.193	SH
-	679976.24	9124494.988	523.844	SH
-	679943.961	9124485.39	530.747	SH
-	679980.913	9124487.013	523.642	SH
-	679981.1	9124492.016	523.951	SH
-	679981.4	9124492.201	524.778	SH
-	679984.949	9124490.824	525.894	SH
-	679983.588	9124489.349	523.924	SH
-	679964.936	9124481.333	528.939	SH
-	679965.287	9124482.168	528.001	SH
-	679957.089	9124488.165	529.05	SH
-	679957.318	9124488.546	528.404	SH
-	679949.847	9124494.437	529.675	SH
-	679950.067	9124494.937	528.596	SH
-	679963.153	9124506.394	523.975	SH
-	679954.228	9124498.52	528.287	SH
-	679954.455	9124498.686	527.23	SH
-	679975.745	9124507.197	525.639	SH
-	679961.008	9124495.066	528.148	SH
-	679961.201	9124495.48	527.193	SH
-	679965.752	9124488.969	527.987	SH
-	679966.467	9124489.602	526.528	SH
-	679968.268	9124484.174	527.952	SH
-	679969.687	9124485.137	526.649	SH

HASIL HITUNGAN PROGRAM

33	-1.82583	-3.890	-2.768	-0.540	679970.843	9124526.189	529.707	SH
27	-0.12417	1.180	-6.874	-0.403	679975.913	9124522.083	529.844	SH
30	-0.24167	4.568	-11.012	-0.438	679979.301	9124517.945	529.809	SH
53	0.60194	6.967	-13.020	-0.233	679981.700	9124515.937	530.015	SH

BM05 Type	#	Horizontal Circle	Point From	Polygon		Reflector Height	Date	Note	Azimuth Code
				Slope	Distance				
1.212	1	2	CP05B	1	0	0°00'00.0000"	0°00'00.0000"		
	2	2	BKB	32	1.52		126°57'37"	20.457	
90°22'19.9920"	3	2	P01B		1.158		179°59'43"	52.809	
270°09'55.0080"	4	2	CP05LB		1.52		306°57'21"	20.457	
269°37'41.0160"	5	2	P01LB		1.6		7°12'46.0080"	42.346	
89°48'45.0000"	6	2	JL		1.6		10°51'24.9840"	50.715	
89°46'03.0000"	7	2	JL		1.6		14°48'24.0120"	51.773	
89°43'59.9880"	8	2	JL		1.6		23°49'18.0120"	55.088	
89°58'00.0120"	9	2	JL		1.6		9°21'20.0160"	32.54	
89°50'21.9840"	10	2	JL		1.6		16°08'40.9920"	33.859	
89°41'54.9960"	11	2	JL		1.6		22°07'40.0080"	34.957	
89°34'05.9880"	12	2	JL		1.6		34°44'44.0160"	38.689	
89°49'51.9960"	13	2	JL		1.6		74°03'36.0000"	22.849	
89°26'17.9880"	14	2	JL		1.6		62°20'09.9960"	14.812	
89°01'41.0160"	15	2	JL		1.6		53°26'20.0040"	11.609	
89°06'21.9960"	16	2	JL		1.6		30°13'36.0120"	7.737	
89°24'11.9880"	17	2	JL		1.6		60°06'43.9920"	10.923	
88°59'52.0080"	18	2	JL		1.6		67°16'59.9880"	12.075	
88°52'18.9840"	19	2	JL		1.6		66°39'06.0120"	14.224	
89°07'01.9920"	20	2	JL		1.6		79°45'00.0000"	29.632	
89°33'42.9840"	21	2	PG		1.6		75°54'42.0120"	24.37	
90°24'52.9920"	22	2	PG		1.6		75°33'29.9880"	23.939	
88°58'23.0160"	23	2	PG		1.6		35°12'06.9840"	49.13	
89°49'01.9920"	24	2	PG		1.6		30°24'37.0080"	47.41	
90°10'13.0080"	25	2	PG		1.6		29°49'13.0080"	47.256	
89°46'00.0120"	26	2	PG		1.6		349°24'09.0000"	15.258	
87°53'07.0080"	27	2	PG		1.6		353°48'07.9920"	26.115	
88°46'41.0160"	28	2	PG		1.6		1°21'51.9840"	21.664	
89°31'09.0120"	29	2	PG		1.6		0°38'29.0040"	38.511	
89°41'02.0040"	30	2	PG		1.6		193°59'09.9960"	11.575	
88°00'47.0160"	31	2	PG		1.6		176°38'11.0040"	10.458	
94°51'11.0160"	32	2	SH		1.6		130°27'54.0000"	20.2	
90°44'58.9920"	33	2	SH		1.6		148°48'56.0160"	12.418	

P02 # Point From Point To Reflector Height Azimuth								Code
P02	#	Point	From	Point	To	Reflector	Height	
Horizontal Circle				Slope	Distance	Zenith	Date	Note
Type								
1.405	1	44	24	0	0°00'00.0000"	0°00'00.0000"		
		P1B	BKB					
	2	44	14	1.21		115°25'49"		37.012
90°03'43.9920"				CP05B	SS			
	3	44	26	1.52		179°59'57"		42.322
269°30'18.0000"				P1LB	SS			
	4	44	15	1.21		295°25'17"		37.012
269°55'59.9880"				CP05LB	SS			
CP05	#	Point	From	Point	To	Reflector	Height	Azimuth
Horizontal Circle				Slope	Distance	Zenith	Date	Note
Type								

				Polygon		
1.21	1	14	45	0	295°25'54"	0°00'00.0000"
		P02B	BKB			
2		14	28	1.195	48°47'58"	52.806
90°14'34.0080"				BM05B	SS	
3		14	47	1.405	179°59'36"	37.011
270°00'10.0080"				P02LB	SS	
4		14	27	1.195	228°47'47"	52.806
269°45'05.0040"				BM05LB	SS	

SOURCE CODE ANGEL

```

Public Class Angel
    Private myDegree As Double
    Private myMinute As Double
    Private mySecond As Double
    Private myPositive As Boolean

    Sub New(degree As Double, minute As Double, second As
Double)
        Dim checker As Boolean = False

        If (degree <= 0 And minute <= 0 And second <= 0) Then
            checker = True
            Me.myPositive = False
        End If

        If (degree >= 0 And minute >= 0 And second >= 0) Then
            checker = True
            Me.myPositive = True
        End If

        If Not checker Then Throw New Exception("Unknown
Angle. Parameter is invalid horizontal angel")

        Me.myDegree = degree
        Me.myMinute = minute
        Me.mySecond = second
        Me.Normalization()
    End Sub

    Sub New(totalSecond As Double)
        Dim temp As Angel = Me.toAngel(totalSecond)
        Me.myDegree = temp.Degree
        Me.myMinute = temp.Minute
        Me.mySecond = temp.Second
    End Sub

    Public ReadOnly Property Degree As Double
        Get
            Return Me.myDegree
        End Get
    End Property

```

```

    Public ReadOnly Property Degree(decimals As Integer) As
Double
        Get
            Return Me.myDegree
        End Get
    End Property

    Public ReadOnly Property Minute As Double
        Get
            Return Me.myMinute
        End Get
    End Property

    Public ReadOnly Property Second As Double
        Get
            Return Me.mySecond
        End Get
    End Property

    Public ReadOnly Property isPositive() As Boolean
        Get
            Return Me.myPositive
        End Get
    End Property

    Private Function calculate(d As Double, m As Double, s As
Double, type As CalculationType) As Double
        Select Case type
            Case CalculationType.toDegree
                Return s / 3600 + m / 60 + d
            Case CalculationType.toMinute
                Return d * 60 + m + s / 60
            Case CalculationType.toSecond
                Return (d * 60 + m) * 60 + s
            Case Else
                Return -1
        End Select
    End Function

    Private Function reverseValue(value As Double, type As
CalculationType) As Double
        Select Case type
            Case CalculationType.toDegree
                Return 360 + value

```

```
        Case CalculationType.toMinute
            Return 21600 + value
        Case CalculationType.toSecond
            Return 1296000 + value
        Case Else
            Return -1
    End Select
End Function

Public Function toSecond() As Double
    Return Me.toSecond(False)
End Function

Public Function toSecond(fullDegree As Boolean) As Double
    Return Me.toSecond(fullDegree, False)
End Function

Public Function toSecond(fullDegree As Boolean, toPositive
As Boolean) As Double
    Return Me.Convert(fullDegree, toPositive,
CalculationType.toSecond)
End Function

Public Function toMinute() As Double
    Return Me.toMinute(False)
End Function

Public Function toMinute(fullDegree As Boolean) As Double
    Return Me.toMinute(fullDegree, False)
End Function

Public Function toMinute(fullDegree As Boolean, toPositive
As Boolean) As Double
    Return Me.Convert(fullDegree, toPositive,
CalculationType.toMinute)
End Function
Public Function toDegree() As Double
    Return Me.toDegree(False)
End Function

Public Function toDegree(fullDegree As Boolean) As Double
    Return Me.toDegree(fullDegree, False)
End Function
```

```

    Public Function toDegree(fullDegree As Boolean, toPositive
As Boolean) As Double
        Return Me.Convert(fullDegree, toPositive,
CalculationType.toDegree)
    End Function

    Public Function toVDegree() As Double
        Return Me.toVDegree(False)
    End Function

    Public Function toVDegree(fullDegree As Boolean) As Double
        Return Me.toVDegree(fullDegree, False)
    End Function

    Public Function toVDegree(fullDegree As Boolean,
toPositive As Boolean) As Double
        Return 90 - Me.Convert(fullDegree, toPositive,
CalculationType.toDegree)
    End Function

    Private Function Convert(fullDegree As Boolean, toPositive
As Boolean, type As CalculationType) As Double
        Dim res As Double = Me.calculate(Me.Degree, Me.Minute,
Me.Second, type)

        If toPositive Then
            If Not Me.isPositive Then
                res = Me.reverseValue(res, type)
            End If
        End If

        If fullDegree Then
            If Me.isPositive Then

                If Me.Degree = 0 Then
                    res = Me.calculate(360, Me.Minute,
Me.Second, type)
                End If

                End If
            End If
            Return res
        End Function

```

```

Private Sub Normalization()
    Dim temp As Double = Math.Floor(Me.Second / 60)
    If temp < 0 Then temp += 1
    Me.mySecond = Me.Second Mod 60
    Me.myMinute += temp
    temp = Math.Floor(Me.Minute / 60)
    If temp < 0 Then temp += 1
    Me.myMinute = Me.Minute Mod 60
    Me.myDegree = (Me.Degree + temp) Mod 360
End Sub

Private Function toAngel(value As Double) As Angel
    Dim vtemp As Double = value Mod 1296000
    If vtemp < 0 Then vtemp = 1296000 + vtemp
    Dim temp As Double = Math.Floor(vtemp / 60)
    Return New Angel(Math.Floor(temp / 60) Mod 360, temp
Mod 60, vtemp Mod 60)
End Function

Public Function Reverse() As Angel
    Dim value As Double ' = If(Me.isPositive, 1296000 -
Me.toSecond(False), 1296000 + Me.toSecond(False))
    Dim mul As SByte
    If Me.isPositive Then
        value = 1296000 - Me.toSecond(False)
        mul = -1
    Else
        value = 1296000 + Me.toSecond(False)
        mul = 1
    End If
    Dim temp As Double = Math.Floor(value / 60)
    Return New Angel(Math.Floor(temp / 60) * mul, (temp
Mod 60) * mul, (value Mod 60) * mul)
End Function

Public Function Subtraction(point As Angel) As Angel
    Return Me.toAngel(Me.toSecond(False) -
point.toSecond(False))
End Function

Public Function subtractionTo(point As Angel) As Angel
    Return Me.toAngel(point.toSecond(False) -
Me.toSecond(False))
End Function

```

```

    Public Function Addition(point As Angel) As Angel
        Return Me.toAngel(Me.toSecond(False) +
point.toSecond(False))
    End Function

    Public Function additionTo(point As Angel) As Angel
        Return Me.toAngel(point.toSecond(False) +
Me.toSecond(False))
    End Function

    Public Function Division(divisor As Double) As Angel
        Return Me.toAngel(Me.toSecond(False) / divisor)
    End Function

    Public Function Division(divisor As Angel) As Angel
        Return Me.toAngel(Me.toSecond(False) /
divisor.toSecond(False))
    End Function

    Public Function Multiplication(multipler As Double) As
Angel
        Return Me.toAngel(Me.toSecond(False) * multipler)
    End Function

    Public Function Multiplication(multipler As Angel) As
Angel
        Return Me.toAngel(Me.toSecond(False) *
multipler.toSecond(False))
    End Function

    Public Overrides Function toString() As String
        Return Me.Degree & ":" & Me.Minute & ":" & Me.Second &
"***"
    End Function

    Public Shared Function DegreeToAngel(degree As Double) As
Angel
        Dim temp = degree Mod 360
        If temp < 0 Then temp = 360 + temp

        Dim d As Double = Math.Floor(temp)
        Dim m As Double = Math.Floor((temp - d) * 60)
        Dim s As Double = (temp - (d + m / 60)) * 3600

```

```
        Return New Angel(d, m, s)
    End Function

    Public Shared Function StringToAngel(angelString As
String) As Angel
        angelString = angelString.Replace(",", ".")
        Dim degree() As String = angelString.Split("°")
        Dim minute() As String = degree(1).Split("'")
        minute(1) = minute(1).Substring(0, minute(1).Length -
1)
        Return New Angel(Val(degree(0)), Val(minute(0)),
Val(minute(1)))
        'MessageBox.Show(degree(0) & vbCrLf & minute(0) &
vbCrLf & minute(1))
    End Function
End Class

Public Enum CalculationType
    toSecond
    toMinute
    toDegree
End Enum
```

SOURCE CODE HORIZONTAL ANGEL

```
Public Class Angel
    Private myDegree As Double
    Private myMinute As Double
    Private mySecond As Double
    Private myPositive As Boolean

    Sub New(degree As Double, minute As Double, second As
Double)
        Dim checker As Boolean = False

        If (degree <= 0 And minute <= 0 And second <= 0) Then
            checker = True
            Me.myPositive = False
        End If

        If (degree >= 0 And minute >= 0 And second >= 0) Then
            checker = True
            Me.myPositive = True
        End If

        If Not checker Then Throw New Exception("Unknown
Angle. Parameter is invalid horizontal angel")

        Me.myDegree = degree
        Me.myMinute = minute
        Me.mySecond = second
        Me.Normalization()
    End Sub

    Sub New(totalSecond As Double)
        Dim temp As Angel = Me.toAngel(totalSecond)
        Me.myDegree = temp.Degree
        Me.myMinute = temp.Minute
        Me.mySecond = temp.Second
    End Sub

    Public ReadOnly Property Degree As Double
        Get
            Return Me.myDegree
        End Get
    End Property

    Public ReadOnly Property Degree(decimals As Integer) As
Double
```

```

Get
    Return Me.myDegree
End Get
End Property

Public ReadOnly Property Minute As Double
    Get
        Return Me.myMinute
    End Get
End Property

Public ReadOnly Property Second As Double
    Get
        Return Me.mySecond
    End Get
End Property

Public ReadOnly Property isPositive() As Boolean
    Get
        Return Me.myPositive
    End Get
End Property

Private Function calculate(d As Double, m As Double, s As
Double, type As CalculationType) As Double
    Select Case type
        Case CalculationType.toDegree
            Return s / 3600 + m / 60 + d
        Case CalculationType.toMinute
            Return d * 60 + m + s / 60
        Case CalculationType.toSecond
            Return (d * 60 + m) * 60 + s
        Case Else
            Return -1
    End Select
End Function

Private Function reverseValue(value As Double, type As
CalculationType) As Double
    Select Case type
        Case CalculationType.toDegree
            Return 360 + value
        Case CalculationType.toMinute
            Return 21600 + value
    End Select
End Function

```

```
        Case CalculationType.toSecond
            Return 1296000 + value
        Case Else
            Return -1
    End Select
End Function

Public Function toSecond() As Double
    Return Me.toSecond(False)
End Function

Public Function toSecond(fullDegree As Boolean) As Double
    Return Me.toSecond(fullDegree, False)
End Function

Public Function toSecond(fullDegree As Boolean, toPositive
As Boolean) As Double
    Return Me.Convert(fullDegree, toPositive,
CalculationType.toSecond)
End Function

Public Function toMinute() As Double
    Return Me.toMinute(False)
End Function

Public Function toMinute(fullDegree As Boolean) As Double
    Return Me.toMinute(fullDegree, False)
End Function

Public Function toMinute(fullDegree As Boolean, toPositive
As Boolean) As Double
    Return Me.Convert(fullDegree, toPositive,
CalculationType.toMinute)
End Function
Public Function toDegree() As Double
    Return Me.toDegree(False)
End Function

Public Function toDegree(fullDegree As Boolean) As Double
    Return Me.toDegree(fullDegree, False)
End Function

Public Function toDegree(fullDegree As Boolean, toPositive
As Boolean) As Double
```

```

    End Try
    lineCounter += 1
Loop

    objReader.Close()
End If
GC.Collect()
End Sub

Private Sub Setting()
    Me.mySTI = New BakaliCollection(Of StiPoint)
    Me.mySTIRelation = New BakaliCollection(Of pointRelation)
    Me.myCoordinate = New BakaliCollection(Of UTMCoordinate)
    Me.myBearings = New BearingVerticalAngelCollection
    Me.mySituation = New Situasi

    Me.MyLoaded = False
    'Me.AdditionalSub()
End Sub

'Private Sub AdditionalSub()
'    Me.addUTM(New UTMCoordinate(0, 0))
'    Me.addUTM(New UTMCoordinate(0, 0))
'End Sub

Public Sub Process(firstCoor As UTMCoordinate, secondCoor As
UTMCoordinate)
    Me.beginProcess(firstCoor, secondCoor)
    Me.myAzimuth = Me.myHACalculator.getAzimuth(Me.myCoordinate)
    Me.Process()
End Sub

Public Sub Process(secondCoor As UTMCoordinate, azimuth As Double)
    Me.beginProcess(New UTMCoordinate(0, 0), secondCoor)
    Me.myAzimuth = Me.myHACalculator.getAzimuth(azimuth)
    Me.Process()
End Sub

Private Sub beginProcess(firstCoor As UTMCoordinate, secondCoor As
UTMCoordinate)
    Coordinates.Clear()
    Me.addUTM(firstCoor)
    Me.addUTM(secondCoor)
    Me.myHACalculator = New HorizontalAngelCalculation(Me.mySTIRelation)
End Sub

```

```

Private Sub Process()
    Me.myCorrection = New UTMDegreeCollection(Me.Azimuths, Me.Bearings)
    Me.myCorrection.getCoordinates(Me.Coordinates)
    Me.myCorrection.setCorrection(Me.Coordinates, Me.Bearings)
End Sub

Public Sub Calculate(elevation As Elevasi, eResult As ElevasiResult)
    Me.Situasi.Calculate(Me, elevation, eResult)
End Sub

Public Sub addPoint(pointProperties As StiPointProperties, relation As
pointRelation, vertical As BearingVerticalAngel)

    Me.Points.Add(New StiPoint(pointProperties, 0))
    Me.Relations.Add(relation)
    Me.Bearings.Add(vertical)

    MessageBox.Show(Me.Points.Item(Me.Points.Count - 1).ToString)
    MessageBox.Show(relation.ToString)
    MessageBox.Show(vertical.ToString)
End Sub

Public Function EditPoint(index As Integer, pointProperties As StiPointProperties)
As StiPoint
    With Me.Points.Item(index)
        .Name = pointProperties.Name
        .Lat = pointProperties.Lat
        .Lng = pointProperties.Lng
        .Description = pointProperties.Description
    End With
    Return Me.Points.Item(index)
End Function

Public Sub DeletePoint(index As Integer)
    Me.Points.removeByIndex(index)
    Me.Relations.removeByIndex(index)
    Me.Bearings.removeByIndex(index)
End Sub

Public Function EditRelation(index As Integer, ordinary As
DoubleHorizontalAngel, eOrdinary As DoubleHorizontalAngel) As pointRelation
    Me.Relations.Item(index).Change(ordinary, eOrdinary)
    Return Me.Relations.Item(index)
End Function

```

```
Private Sub addUTM(value As UTMCoordinate)
    Me.Coordinates.Add(value)
End Sub

Public Function EditUTM(index As Integer, value As UTMCoordinate) As
UTMCoordinate
    If index > 1 Then Throw New Exception("Just index 0 and 1 that can be
changed")
    Me.Coordinates.Replace(index, value)
    Return Me.Coordinates.Item(index)
End Function

Public Function EditBearing(index As Integer, value As BearingVerticalAngel) As
BearingVerticalAngel
    Me.Bearings.Replace(index, value)
    Return Me.Bearings.Item(index)
End Function
End Class
```

SOURCE CODE LEVELLING

```

Public Class Elevasi
    'Private myStrIndex() As Integer = {0, 1, 5}
    Private myGoPoints As ElevasiPointCollection
    Private myReturnPoints As ElevasiPointCollection

    Public ReadOnly Property GoPoints As ElevasiPointCollection
        Get
            Return Me.myGoPoints
        End Get
    End Property

    Public ReadOnly Property ReturnPoints As ElevasiPointCollection
        Get
            Return Me.myReturnPoints
        End Get
    End Property

    Sub New(filepath As String)
        Me.readFile(filepath)
    End Sub

    Private Sub readFile(filepath As String)
        If Not System.IO.File.Exists(filepath) Then Throw New Exception("File tidak ditemukan. Silahkan pilih kembali")
        If System.IO.File.Exists(filepath) Then
            Dim objReader As New System.IO.StreamReader(filepath)
            Dim sline As String
            Dim line() As String
            Dim lineCounter As Integer = 1
            Dim status As Byte = 0
            Dim goDatas As Integer = 0
            Dim returnDatas As Integer = 0
            Dim tempData As ElevasiPoint

            Me.myGoPoints = New ElevasiPointCollection
            Me.myReturnPoints = New ElevasiPointCollection

            Do While objReader.Peek() <> -1
                sline = objReader.ReadLine().Replace(" ", String.Empty)

                If sline.Replace(vbTab, String.Empty) = String.Empty Then
                    lineCounter += 1
                    Continue Do
                End If
            End Do
        End If
    End Sub

```

```

line = GlobalFunction.getSplit(sline, vbTab, -1)
If line.Length <> 1 And line.Length <> 5 Then
    objReader.Close()
    Throw New Exception("Format data salah pada baris ke " &
lineCounter)
End If

If line(0) = "#pergi" Then
    status = 1
ElseIf line(0) = "#pulang" Then
    status = 2
Else
    If status <> 1 And status <> 2 Then
        objReader.Close()
        Throw New Exception("Data pada baris ke " & lineCounter & " "
ambigu karena tidak diketahui data tersebut merupakan data pergi atau pulang")
    End If

Dim checker As Boolean = False
Do While (Not checker)
    lineCounter += 1
    sline = objReader.ReadLine().Replace(" ", String.Empty)

    If sline = String.Empty Then
        Continue Do
    Else
        checker = True
    End If
Loop

Dim line2() As String = GlobalFunction.getSplit(sline, vbTab, -1)
If line2.Length <> 5 Then
    objReader.Close()
    Throw New Exception("Format data salah pada baris ke " &
lineCounter)
End If

tempData = New ElevasiPoint(
    line(0),
    New ElevasiPointDetail(
        line(1),
        New ElevasiAngel(
            Val(line(2)),
            Val(line(3)),
            Val(line(4)))

```

```

        )
),
New ElevasiPointDetail(
    line2(1),
    New ElevasiAngel(
        Val(line2(2)),
        Val(line2(3)),
        Val(line2(4))
    )
)
)
)
)
)
'MessageBox.Show(tempData.ToString)
If status = 1 Then
    Me.Add(Me.myGoPoints, tempData)
    goDatas += 1
Else
    Me.Add(Me.myReturnPoints, tempData)
    returnDatas += 1
End If
End If
lineCounter += 1
Loop

If goDatas <> returnDatas Then
    Throw New Exception("Jumlah data pergi dan pulang tidak sama sehingga
proses tidak dapat dilakukan")
    objReader.Close()
End If

objReader.Close()
End If
GC.Collect()
End Sub

Private Sub Add(ByRef obj As ElevasiPointCollection, item As ElevasiPoint)
    obj.Add(item)
End Sub

Public Function Calculate(firstElevation As Double) As ElevasiResult
    Dim res As New ElevasiResult

    res.Goes.Add(firstElevation)
    Dim corr As Double = Me.GoPoints.getCorrection
    For i As Integer = 0 To Me.GoPoints.Count - 1
        res.Goes.Add(res.Goes.Item(i) + Me.GoPoints.Item(i).getDeviation + corr)
    Next
End Function

```

Next

res.Returns.Add(res.Goes.Item(res.Goes.Count - 1))

corr = Me.ReturnPoints.getCorrection

For i As Integer = 0 To Me.ReturnPoints.Count - 1

 res.Returns.Add(res.Returns.Item(i) + Me.ReturnPoints.Item(i).getDeviation
+ corr)

 Next

 Return res

End Function

End Class

SOURCE CODE DETAIL/SITUATION

Public Class Situasi

Inherits BakaliCollection(Of SituasiItem)

```
'Private myStrIndex() As Integer = {0}

Sub New()
    MyBase.New
    'If Not System.IO.File.Exists(filepath) Then Throw New Exception("File tidak
ditemukan. Silahkan pilih kembali")
    'If System.IO.File.Exists(filepath) Then
        ' Dim objReader As New System.IO.StreamReader(filepath)
        ' Dim sline As String
        ' Dim line() As String
        ' Dim lineCounter As Integer = 1
        ' Dim temp As Double
        ' Dim activeindex As Integer
        ' Dim name As String = String.Empty
        ' Dim check As Boolean

        ' Do While objReader.Peek() <> -1
        '     sline = objReader.ReadLine().Replace(" ", String.Empty)
        '     If sline = String.Empty Then Continue Do
        '     line = sline.Split(",")
        '     If line.Length <> 1 And line.Length <> 9 Then
        '         objReader.Close()
        '         Throw New Exception("Format data salah pada baris ke " &
lineCounter)
        '     End If

        '     If line.Length = 1 Then
        '         activeindex = -1
        '         'get Name
        '         name = line(0).Substring(1)

        '         'check if exist name
        '         For i As Integer = 0 To Me.Count - 1
        '             If name.ToLower = Me.Item(i).Name.ToLower Then
        '                 activeindex = i
        '                 Exit For
        '             End If
        '         Next

        '         'registering name if not exist
        '         If activeindex = -1 Then
        '             Me.Add(New SituasiItem(name))
        '             activeindex = Me.Count - 1
        '         End If
        '     End If
    End If
```