BAB III

METODE PENELITIAN

3.1 Penerapan Ant Colony Algorithm Pada Masalah Transportasi

Masalah Transportasi berisikan proses pendistribusian beberapa komoditi yang homogen dari berbagai tempat (*source*) ke sekumpulan tempat tujuan (*destination*), setiap permintaan dispesifikasikan dengan tingkat komoditi tersebut. Tujuan masalah ini untuk mengalokasikan persediaan yang tersedia pada setiap tempat asal (*source*) dalam memenuhi permintaan pada setiap tempat tujuan (*destination*) agar optimal. Fungsi obyektif yang paling umum adalah untuk meminimalisasi total biaya transportasi atau total jarak yang ditempuh dari proses pengalokasian.

Diberikan m tempat sumber (*source*) dan n tempat tujuan (*destination*) sehingga problem dapat dirumuskan sebagai sebuah model program linier:

$$\text{Min } Z = \sum_{i=1}^{m} \sum_{j=}^{n} C_{ij} X_{ij}$$

dengan kendala $\sum_{j=1}^{n} X_{ij} = s_i$, untuk i= 1, 2,…, m

$\sum_{i=1}^{n} X_{ij} = d_j$, untuk i= 1, 2,…, n

$X_{ij} \geq 0$, untuk semua i dan j

Dimana Z adalah biaya distribusi total dan

X_{ij} (i = 1, 2, …, m; j = 1, 2, …, n) adalah jumlah unit yang harus distribusi dari sumber i ke tujuan j.

3.2 Fisibiliti (Kemungkinan) Dari Masalah Transportasi

Perumusan diatas beranggapan bahwa total unit yang tersedia dan yang diminta adalah sama satu dengan lainnya, dapat dituliskan

$$\sum_{i=1}^{m} ai = \sum_{j=1}^{n} bj$$

Dibawah asumsi kondisi berimbang, problem transportasi selalu memiliki solusi yang fleksibel. Dengan mudah dapat dilihat bahwa persamaan dibawah ini menghasilkan solusi yang fleksibel.

$$X_{ij} = \frac{a_i\, b_j}{\Sigma_i a_I}, \quad i = 1, 2,\ldots, m \;\&\; j = 1, 2, \ldots, n$$

Untuk catatan, setiap komponen Xij harus memenuhi syarat

$$0 \le X_{ij} \le \min \{a_i, b_j\}$$

Problem transportasi biasanya ditampilkan dalam tabel transportasi seperti dapat dilihat pada gambar dibawah ini, dimana baris mewakili bagian *source*, kolom mewakili bagian *destination* dan setiap sel pada baris ke-i dan kolom ke-j mewakili variable keputusan X_{ij}. Kotak dipojok kanan atas pada sel (i,j) mewakili nilai biaya.

Tabel 3.1 Problem transportasi

From \ To	1	2	….	N	Supply
1	$\llcorner^{C11}$ X_{11}	$\llcorner^{C12}$ X_{11}		$\llcorner^{C1n}$ X_{1n}	A_1
2	$\llcorner^{C21}$ X_{21}	$\llcorner^{C22}$ X_{22}		$\llcorner^{C2n}$ X_{2n}	A_2
…					…
M	$\llcorner^{Cm1}$ X_{m1}	$\llcorner^{Cm2}$ X_{m2}		$\llcorner^{Cmn}$ X_{mn}	A_m
Demand	B_1	B2	…	Bn	

Sebuah matriks biasanya digunakan untuk mewakili solusi dari problem transportasi. Alokasi matriks dari problem transportasi dapat ditulis demikian

$$X_p = \begin{pmatrix} X_{11} & X_{12} & \dots & X_{1n} \\ X_{21} & X_{22} & \dots & X_{2n} \\ \dots & \dots & \dots & \dots \\ X_{m1} & X_{m2} & \dots & X_{mn} \end{pmatrix}$$

Dimana X_p = melambangkan semut ke-p

X_{ij} = variable keputusan yang berhubungan

3.3 Membuat Kemungkinan Solusi

Membuat kemungkinan solusi pada ACO dapat mengikuti sebagai berikut: Dimulai pada saat t = 0 rangkaian truk dengan pesanan. Setelah setiap pesanan ditempatkan, waktu t akan di update, ini digunakan untuk menentukan waktu pengiriman sebenarnya dari penempatan terakhir. Penempatan pesanan akan berlanjut hingga akhir dari rencana jangkauan T terpenuhi, atau tidak ada lagi penenpatan pesanan selanjutnya. Ini berarti kendaraan lain membawanya hingga digunakan, t berubah menjadi t = 0 dan penepatan lain masih berlanjut. Prosedur ini berjalan hingga semua pesanan ditempatkan.

Untuk pesanan terpilih yang belum ditempatkan kedalam truk, dua aspek yang perlu diperhitungkan: bagaimana memberikan harapan tentang pesanan secara umum, dan seberapa baik pilihan tersebut untuk pesanan pada iterasi algoritma sebelumnya. Informasi yang terlebih dahulu adalah jarak penglihatan (visibility), kemudian informasi feromon.

a. Visibility (Kemungkinan).

Informasi *visibility* disimpan kedalam matrik $\eta(t)$. Tiap matrik $\eta ij(t)$ adalah positif, jika dan hanya jika penempatan pesanan *j* setelah pesanan *i* dikerjakan. Penempatan pesanan *j* dikerjakan , jika pesanan dapat terjadwal kedalam kendaraan yang pasti tanpa pelanggaran waktu. Oleh sebab itu telah jelas η bergantung pada waktu. Catatan bahwa setiap iterasi hanya merupakan gabungan dari baris dengan penempatan pesanan pada iterasi sebelumnya telah dihitung. Nilai *visibility* sebenarnya dari pesanan *j* bergantung pada aturan prioritas pada algoritma.

b. Informasi Feromon

Informasi feomon dapat dibaca melalui dua cara. Yang pertama, nilai τij yang bergabung dengan *(vi,vj)* mewakili informasi feromon yang sebenarnya. Keduanya menguraikan pola $i \leq n, j \leq n$, nilai τij mewakili informasi feromon dari penempatan pesanan *j* ditengah-tengah pesanan *i*. Pada penguraian pertama ini nilai $\tau ij, \quad i \leq n, j \geq n + i$ menggambarkan informasi feromon untuk memulai suatu perjalanan dengan truk lain pada gudang *j* ditetapkan bahwa pesanan *i* adalah pesanan terakhir di kendaraan.

Kebalikannya, metode penguraian kedua, informasi feromon untuk memilih gudang untuk kendaraan telah ada pada suatu extra array dan berdiri sendiri dari pesanan terakhir yang diberikan olek kendaraan lain dan gudang dari semua kendaraan lainnya.

Akhirnya, untuk kedua penguraian $\tau ij, \quad i \leq n, j \geq n + i$ menggambarkan informasi feromon untuk pesanan *j* bersamaan pesanan pertama dimulai dari gudang *i*.

c. Aturan Keputusan

Dari visibility dan informasi feromon yang telah dijelaskan, dan $\Omega_i(t) = \{j \in J \cup D : \exists \eta_{ij}(t) > 0\}$, pesanan atau gudang j terpilih akan segera dikunjungi setelah pesanan atau gudang *i* memberikan ukuran aturan secara acak:

$$P_{ij}(t) = \begin{cases} \dfrac{[\tau_{ij}]^{\alpha}\,[\eta_{ij}(t)]^{\beta}}{\sum_{h \in \Omega}[\tau_{ih}]^{\alpha}[\eta_{ih}(t)]^{\beta}} & \text{jika } j \in \Omega_i(t) \\ 0 & \text{sebaliknya} \end{cases}$$

Distribusi probabilitas ini dibiaskan oleh parameter αdan β yang menetukan dampak relatif dari jejak dan visibility, secara tepat.

3.4 Pengubahan Jejak

Setelah peta semut telah dibuat, pengupdetan feromon dilakukan dengan prosedure:

$$\tau_{ij}^{new} = \rho \,.\, \tau_{ij}^{old} + \sum_{\lambda=1}^{\Lambda} \Delta\tau_{ij}^{\lambda} \qquad \forall vi,vj \in V$$

dimana ρ adalah jejak (dengan $0 \le \rho \le 1$). Hanya semut deangan Λ terbaik yang akan mengupdate informasi feromon. Jika arc (vi,vj) digunakan oleh λ semut terbaik jejak feromon meningkat dengan nilai $\Delta\tau_{ij}^{\lambda}$

nilai peng-update-tan dapat digambarkan sebagai berikut:

$$\Delta \tau_{ij}^{\lambda} = \begin{cases} 1 - \frac{\lambda - 1}{\Lambda} & \text{jika } 1 \le \lambda \le \Lambda \\ 0 & \text{sebaliknya} \end{cases}$$

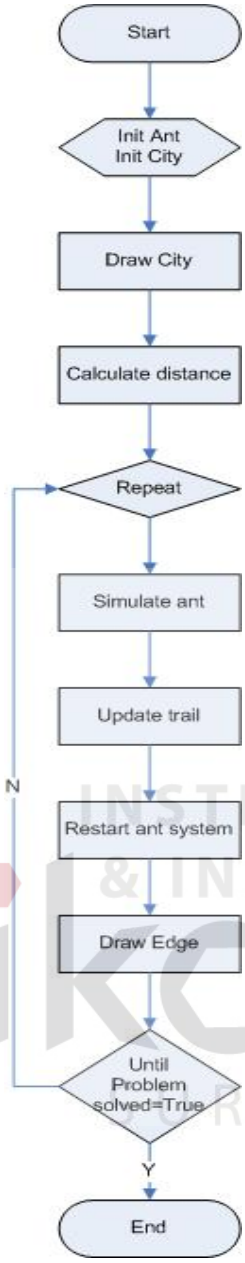
3.5 Algoritma Ant Colony Untuk Masalah Transportasi

Pada saat penginisialisasian, jumlah semut dan parameter-parameter lain telah ditentukan. Kemudian kedua tahap dasar – membangun perjalanan dan mengubah jejak, akan dieksekusi agar memberikan bilangan iterasi. Sebelumnya, procedur ant colony system dapat dijelaskan sebagai berikut:

```
procedure TFAntColonySystem.starting;
var _simulated : integer;
label go;
begin
  goto go;
  _counter := 0;
  StatusBar1.Panels[0].Text := 'Computing';
  repeat
   // StatusBar1.Panels[0].Text := 'Enter Simulate';
    _simulated :=  simulateAnts;
```

```
    showmessage(intToStr(_simulated));
    StatusBar1.Panels[0].Text := intToStr(_simulated);
    updateTrail;
    if _simulated = 1 then
    begin
      //updateTrail;
      restartAntsSystem;
      _counter := _counter + 1;
     // StatusBar1.Panels[3].Text :=  intToStr(_Counter)+' Restarting
Ants System' ;
      _restart := _restart + 1;
      showmessage('Ant System Restarted ...');
      checkContent(0);
     // drawEdges;
     // _problemSolved := True;
    end ;
    //Animate;
    //drawEdges;
    sleep(50);
  until _problemSolved = True;
 // StatusBar1.Panels[0].Text := 'Job Done....';
  {MessageDlg('Job Done !!!', mtInformation, [mbOK], 0);}
  sleep(1000);
  go:
  TriyingToCalculate;
  grProgress.Visible := false;
  CubeSpin1.Continuous := false;
  ShowMessage('Finish....!!!');
  ExitThread(0);
end;
```

Sedangkan algoritma untuk masalah transportasi dapat digambarkan pada flowchart sebagai berikut:



Gambar 3.1. Flow Chart algoritma ACO untuk Transportasi

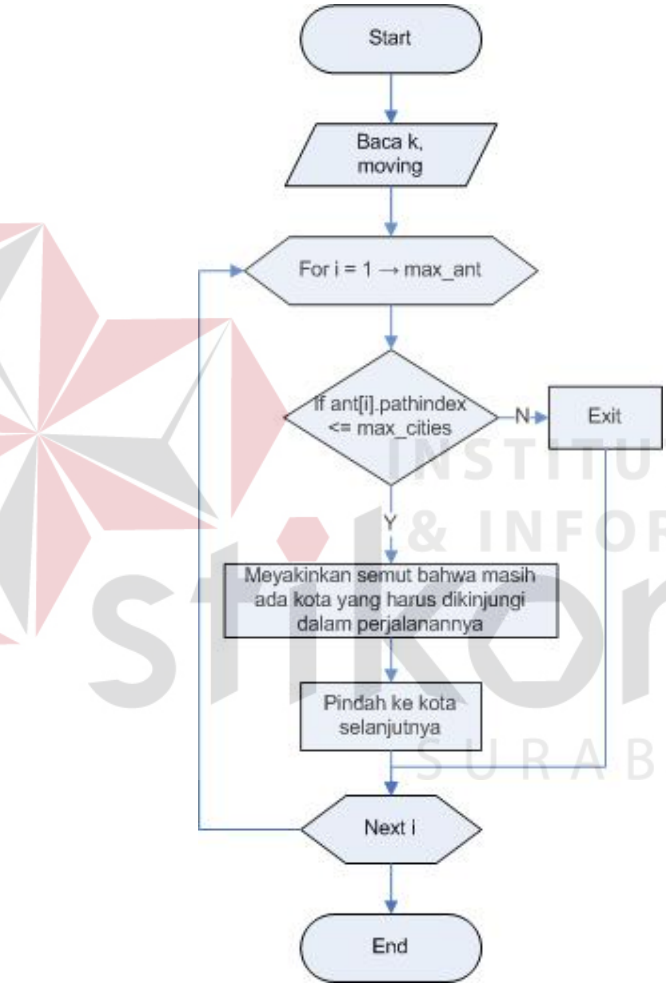
Keterangan:

Ganbar 3.1 menunjukkan algoritma ACO, dimana algoritma tersebut merupakan inti dari algoritma untuk menyelesaikan masalah transportasi algoritma ini menjelaskan alur perjalanan tiap semut dari lokasi asal (*source*) ke tempat tujuan (*destination*) dengan menempuh jalur yang telah tersedia. Dalam

algoritma tersebut terdapat beberapa proses yang saling berhubungan untuk mendapatkan *total cost least*, antara lain:

Proses simulate ant

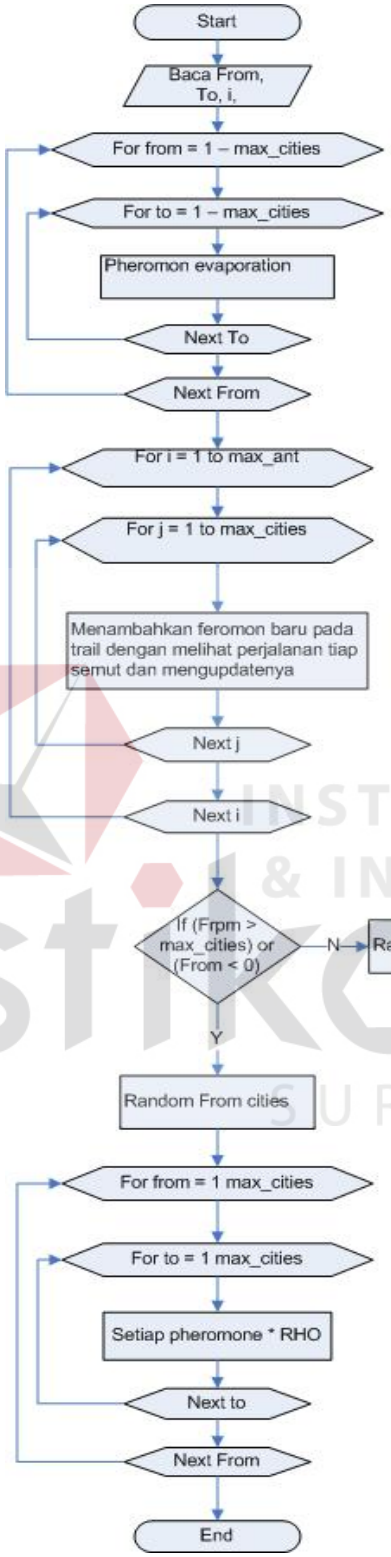
Proses ini ini lakukan untuk memastikan masih ada tujuan lagi yang harus didatangai oleh semut pada setiap melakukan suatu perjalanan.



Gambar 3.2 Algoritma Simulate Ant

Proses update trail

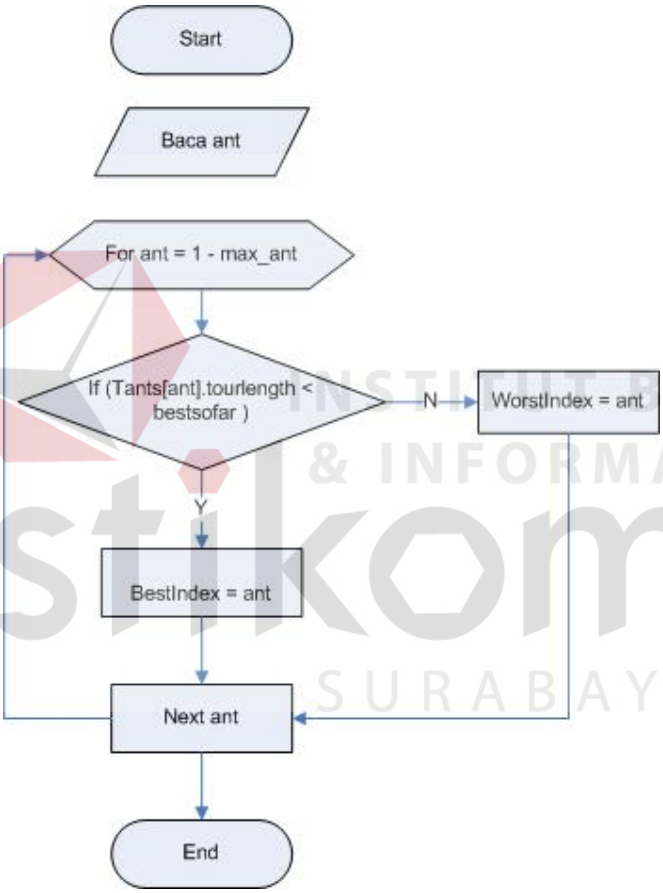
Proses ini dilakukan setelah pengalokasian pengiriman barang dari tiap asal ke tujuan terpenuhi, dimana feromon trails dijadikan landasan pencarian oleh tiap-tiap semut.



Gambar 3.3 Algoritma Update Trails

Proses restart ant system

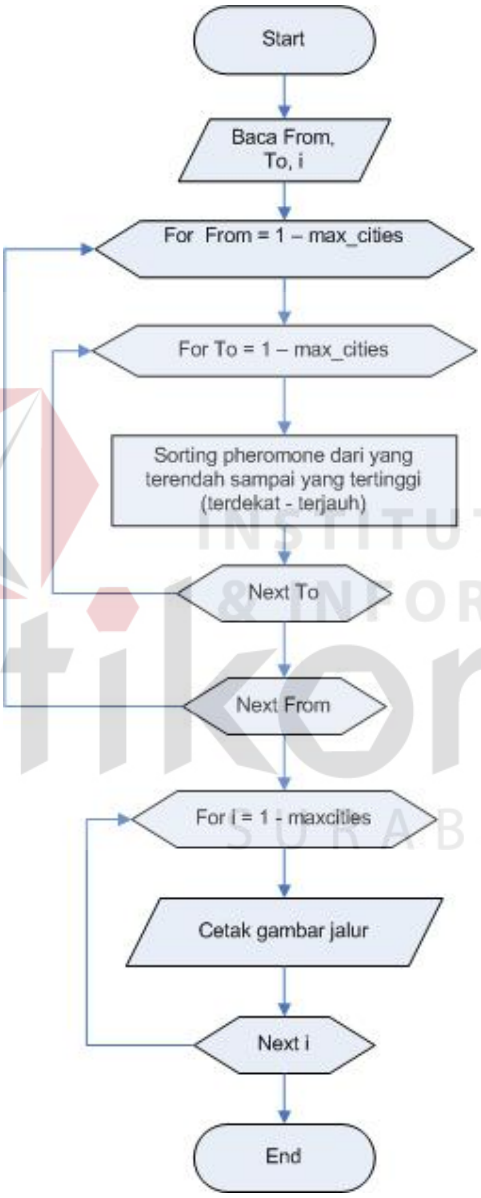
Proses ini dilakukan untuk mengeset ulang sistem tiap kali semut selesai melakukan perjalannya, apabila perjalanan semut adalah yang terjauh, maka akan disimpan di *WorstIndex.* Sedangkan apabila perjalanan semut adalah yang terpendek maka akan disimpan ke dalam BestIndex. Kemudian Index akan kembali keawal lagi menjadi 0 apabila semut akan memulai perjalanan barunya.



Gambar 3.4 Algoritma Restart Ant System

Proses drawegde/returnpath

Proses ini untuk menggambarkan jalur yang telah dilewati. Semut dengan *BestIndex* feromon akan diurut dari yang terendah sampai yang tertinggi (terdekat - terjauh) kemudian jalurnya akan digambar sampai dengan jumlah kota terakhir.



Gambar 3.5 Algoritma DrawEgde

3.6 Rancangan Penyimpanan

Rancangan penyimpanan yang digunakan pada sistem ini menggunakan file text yang disimpan ke memori komputer, sehingga masalah transportasi yang telah dimasukkan dapat digunakan atau dipanggil kembali melalui file text tersebut kapan saja. Dimana file ini berisi matriks masalah transportasi itu sendiri dan feromon yang dihasilkan

3.7 Rancangan Input Output

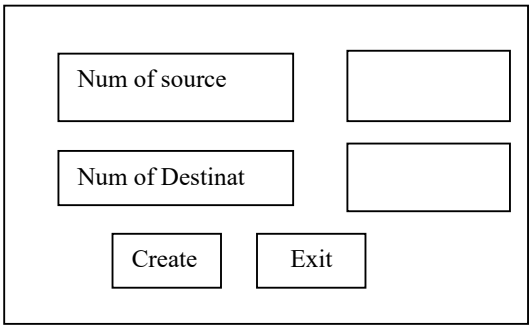
Rancangan user interface yang digunakan pada sistem ini dibuat agar dapat menggunakan mouse atau keyboard secara maksimal karena pada dasarnya aplikasi ini berbasis windows yang selalu menggunakan mouse dan keyboard dalam mempermudah dalam pemasukkan data.

Dalam menampilkan form, penulis merancangnya denggan menggunakan konsep interaksi manusia dengan komputer diman seorang user dengan hanya melihat form user akan mudah mengenali apa yang akan dilakukan selanjutnya.

Didalam form tersebut digunakan kontrol-kontrol untuk mengolah data atau menampilkan data. Adapun kontrol-kontrol yang digunakan antara lain:

1. Command Button, digunakan untuk mengeksekusi atau memproses data setelah pemakai melakukan masukkan atau melakukan suatu pilihan.
2. Text Box, digunakan sebagai tempat penginputan data yang ada dalam sistem, pada text box ini pemakai dapat mengubah tulisan maupun angka secara langsung.

Berikut ini adalah bentuk rangcangan input dari sistem yang nantinya akan diimplementasikan dalam bentuk aplikasi:



Gambar 3.6. Rancangan Input Source dan Destination

Form ini digunakan untuk melakukan pengisian banyaknya sumber (source) dan tujuan (destination).

Supply	Amounts
Supply1	
Supply2	
Supply3	

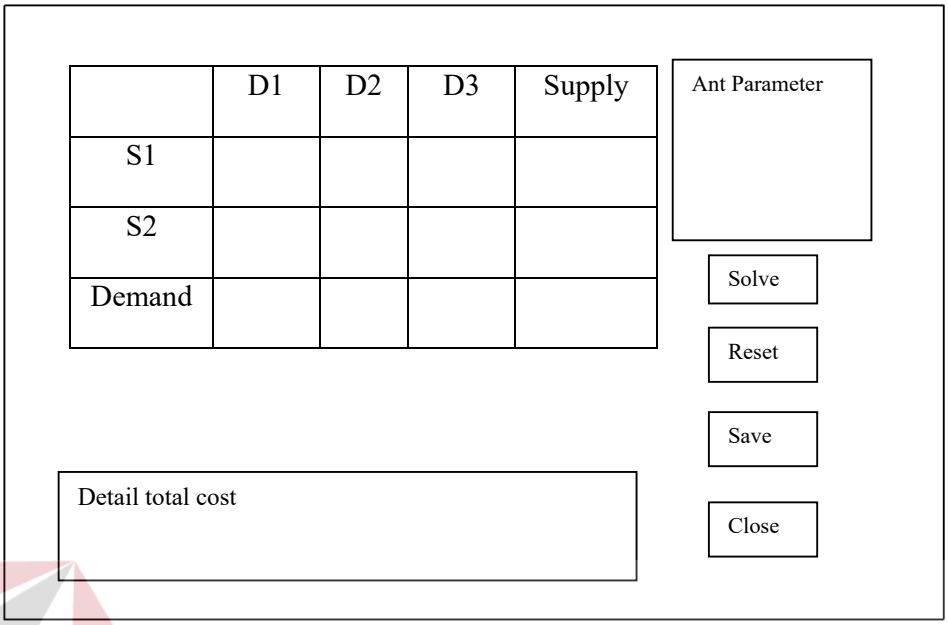
Gambar 3.7. Rancangan Input Besar Supply

Form ini digunakan untuk melakukan pengisian banyaknya barang yang akan diangkut (supply).

Demand	Amounts
Demand 1	
Demand 2	
Demand 3	

Gambar 3.8. Rancangan Input Besar Demand

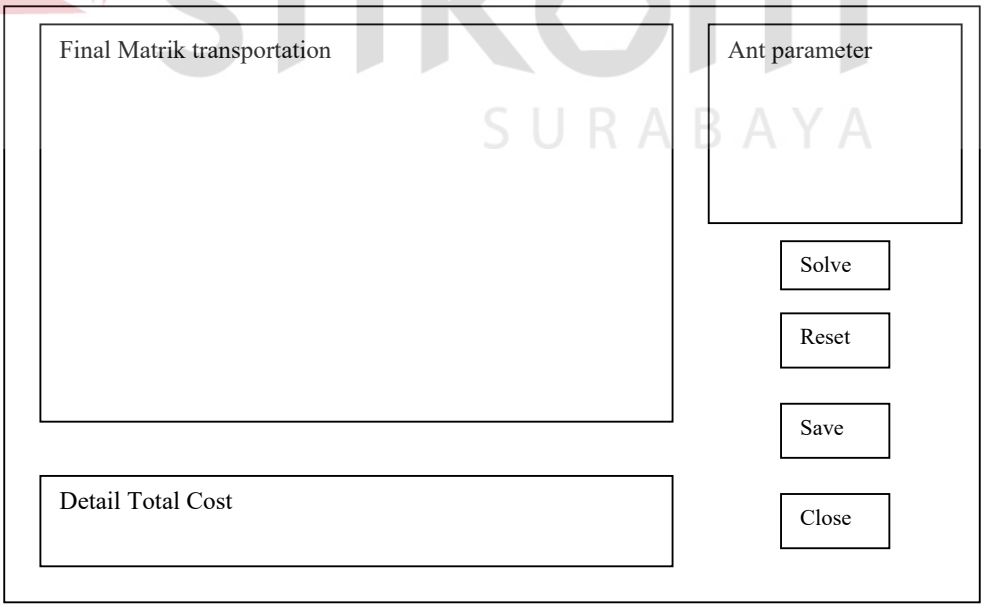
Form ini digunakan untuk memasukkan besarnya batasan demand.



Gambar 3.9. Rancangan Input Nilai dari Unit Cost dan Ant Parameter

Form ini digunakan untuk melakukan pengisian tiap unit cost dari masalah transportasi dan ant parameter untuk pencarian solusi.

Sedangkan untuk rancangan output dapat dilihat pada Gambar 3.4. dibawah ini:



Gambar 3.10. Rancangan Output Masalah Transportasi.

Form ini digunakan untuk menampilkan hasil dari pencarian total cost least, yang berupa matrik akhir dari masalah transportasi dan detail dari total cost least.