

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Pengumpulan Data

Pengumpulan data yang digunakan pada Terminal Petikemas Semarang didapatkan melalui data historis. Data yang dibutuhkan dalam penelitian ini diantaranya gambaran umum dan informasi perusahaan, struktur organisasi, proses bisnis serta data *container volume*.

4.1.1 Deskripsi Perusahaan

Terminal Petikemas Semarang (TPKS) adalah Pelabuhan kelas satu di lingkungan PT Pelabuhan Indonesia III (Persero) yang terletak di pantai utara Jawa Tengah di ibu kota Provinsi Jawa Tengah Semarang, yang beralamat di Jl. Coaster No. 10A, Semarang 50116. Sejarah berdirinya TPKS tidak lepas dari sejarah Pelabuhan Tanjung Emas. Bentuk pengelolaan pelabuhan telah mengalami beberapa kali perubahan, mulai dari perusahaan negara (PN) pelabuhan tahun 1960, Badan Pengusahaan Pelabuhan (BPP) tahun 1969, dan perum Pelabuhan tahun 1983.

Berdasarkan bagiannya adapun pelabuhan Semarang berada di bawah Perum Pelabuhan Indonesia III yang berkantor pusat di Surabaya. Pada periode ini dilaksanakan Proyek Pembangunan tahap I pelabuhan Semarang dan diresmikan oleh Presiden Soeharto pada tanggal 23 November 1985 yang kemudian diberi nama Pelabuhan Tanjung Emas. Bentuk Pengelolaan pelabuhan mengalami Perubahan terakhir kali pada tahun 1992 dengan pembagian yang masih sama yaitu PT.(Persero) Pelabuhan Indonesia I, II, III, dan IV. Awal kegiatan bongkar muat petikemas di pelabuhan Tanjung Emas dilakukan secara konvensional yaitu menjadi satu kesatuan bongkar muat barang umum

(General Cargo) yang berada di bawah kendali divisi usaha Terminal cabang Pelabuhan Tanjung Emas. Setelah selesainya pembangunan tahap II tahun 1997, penanganan Petikemas memasuki tahap pelayanan terminal sendiri yang dikendalikan divisi Terminal petikemas cabang Tanjung Emas (divisi TPK).

Sebagai langkah antisipasi terhadap pertumbuhan angkutan petikemas di Pelabuhan Tanjung Emas yang secara nyata memerlukan pengelolaan yang lebih profesional, manajemen pelabuhan Indonesia III melakukan pemekaran Organisasi Pelabuhan Tanjung Emas, menjadi 2 bagian yaitu pengelolaan TPKS secara mandiri dibawa tanggung jawab General Manager TPKS dan pengelolaan pelabuhan di bawah tanggung jawab General Manager Pelabuhan Tanjung Emas. TPKS berdiri berdasarkan Surat keputusan Direksi PT. Pelabuhan Indonesia III (Persero) Nomor : KEP.46/PP.1.08/P.III-2001 tanggal 29 Juni 2001 tentang pembentukan TPKS terhitung sejak tanggal 21 Juli 2001 dimana TPKS merupakan cabang yang berdiri sendiri terpisah dari Pelabuhan Tanjung Emas Semarang. Sehingga semua urusan handling Petikemas sepenuhnya dilakukan sendiri oleh manajemen TPKS.

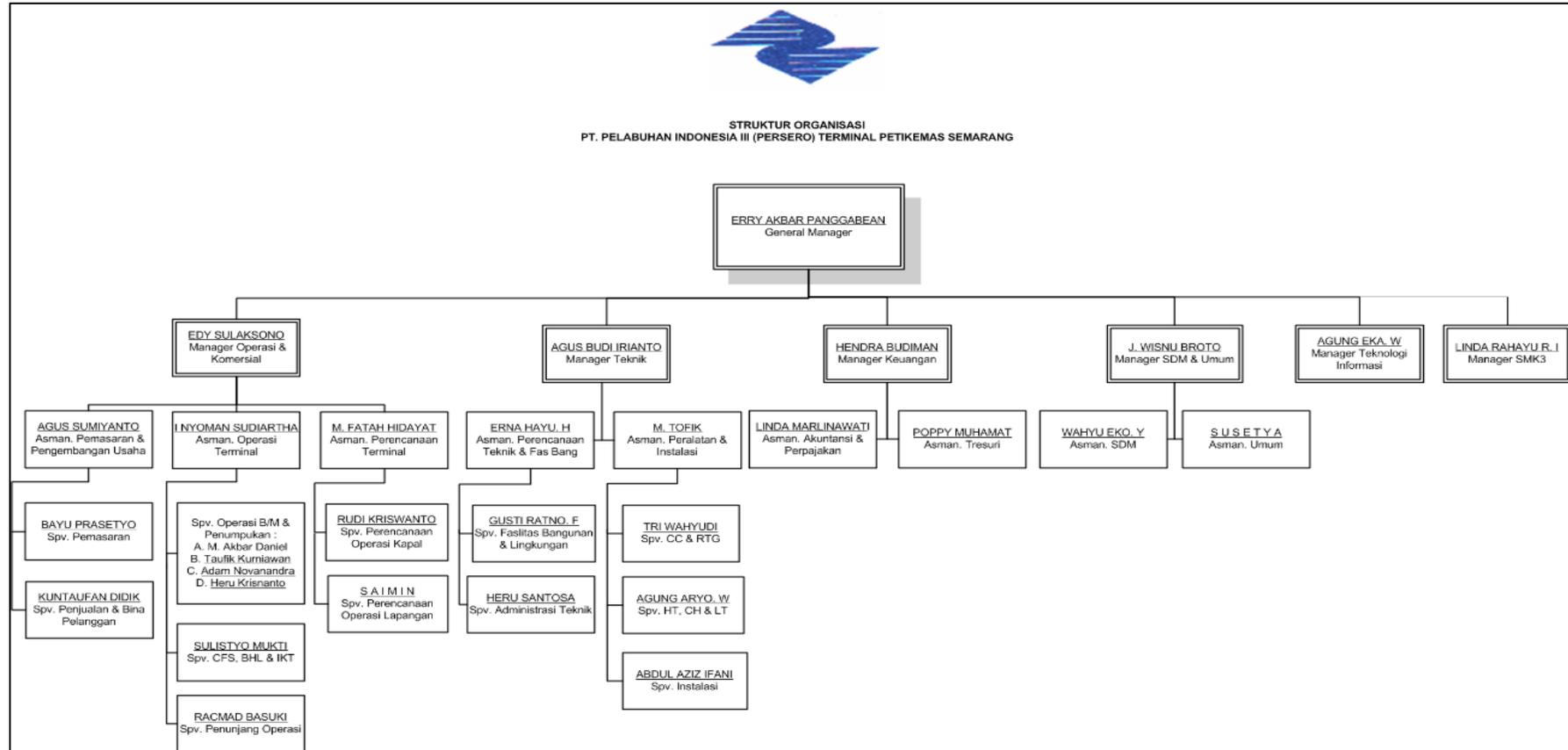
4.1.2 Visi dan Misi Perusahaan

Visi dari TPKS adalah untuk berkomitmen memacu integrasi logistik dengan layanan jasa pelabuhan yang prima, sedangkan misi yang ditetapkan oleh TPKS adalah sebagai berikut:

1. Menjamin penyediaan jasa pelayanan prima melampaui standar yang berlaku secara konsisten.
2. Memacu kesinambungan daya saing industri nasional melalui biaya logistik yang kompetitif.
3. Memenuhi harapan semua stakeholder melalui prinsip kesetaraan dan tata kelola perusahaan yang baik.
4. Menjadikan SDM yang berkompeten, berkinerja handal, dan berpekerja luhur.
5. Mendukung perolehan devisa negara dengan memperlancar arus perdagangan

4.1.3 Struktur Organisasi

Struktur organisasi dari TPKS ialah seperti yang terlihat pada Gambar 4.1 berikut:



Gambar 4.1 Struktur Organisasi Terimnal PetiKemas Semarang (TPKS)

Deskripsi dari Gambar 4.1 secara umum bagian pada struktur organisasi Terminal Petikemas Semarang adalah sebagai berikut:

- a. Tugas Manager Operasi
 1. Menyelenggarakan perusahaan jasa pelayanan kapal dan bongkar muat peti kemas di dermaga, gudang (CFS) dan lapangan (CY).
 2. Menyelenggarakan pengendalian kelancaran kegiatan operasi kapal, lapangan dan gudang, sarana dan prasana serta sumber daya manusianya.
 3. Menyelenggarakan pengamanan, keselamatan dan kesehatan kerja di lingkungan TPKS.

- b. Tugas Asisten Manager Operasi Terminal
 1. Mengkoordinasi dan mengawasi kegiatan perusahaan jasa pelayanan peti kemas di dermaga, gudang (CFS), dan lapangan (CY).
 2. Mengkoordinir pelaksanaan pengendalian kelancaran kegiatan operasi kapal, operasi lapangan dan gudang, sarana dan prasarana serta sumber daya manusianya.

- c. Tugas *Supervisor* Operasi Kapal
 1. Melaksanakan kegiatan perusahaan jasa pelayanan kapal yang meliputi jasa tambat, dermaga yang sesuai dengan rencana yang telah ditetapkan.
 2. Merencanakan, melaksanakan, mengatur kelancaran dan ketertiban kegiatan operasi bongkar muat peti kemas.

- d. Tugas Supervisor Lapangan, *Gate*, dan CFS

Melaksanakan dan mengatur pengendalian kelancaran, ketertiban kegiatan operasi gerakan, pengangkutan dan penumpukan peti kemas di lapangan, pengisian, dan pengopersian peti kemas serta operasi penumpukan barang di gudang (CFS).

- e. Tugas Asisten Manager Perencanaan dan Pemasaran
 1. Mengkoordinasi pelayanan permintaan fasilitas penyandaran kapal dan kegiatan bongkar muat peti kemas.

2. Mengkoordinasi dukungan administrasi kepegawaian, kerumah tanggaan dan ketatausahaan di lingkungan divisi operasi.
- f. Tugas Supervisor Perencanaan
1. Merencanakan pelayanan terhadap permintaan fasilitas penyandaran kapal dan kegiatan bongkar muat *container*.
 2. Menyusun rencana penetapan alokasi tambatan dan waktu tambat berdasarkan atas skala prioritas dan target produktivitas.
- g. Tugas Supervisor Pemasaran
1. Melaksanakan pembuatan realisasi produksi, pendapatan dan biaya jasa operasi kapal, lapangan dan gudang secara periodik.
 2. Melaksanakan pemasaran jasa terminal petikemas Semarang.
 3. Melaksanakan pembuatan dokumen pranota pendapatan jasa, operasi kapal, gudang, dan lapangan.
- h. Tugas Supervisor Administrasi Operasi
- 1) Melaksanakan dukungan administrasi kepegawaian, kerumah tanggaan dan ketatausahaan di divisi operasi.
 - 2) Melaksanakan penyiapan tenaga kerja sesuai permintaan sub dinas yang membutuhkan.

4.1.4 Variabel Input dan Output

Penentuan variabel input untuk model prediksi *container throughput* berdasarkan studi literatur dan data historis yang tersedia. Dalam penelitian ini terdapat 3 variabel input yang digunakan, yaitu PDB Jawa Tengah (x_1), populasi Jawa Tengah (x_2), dan tingkat inflasi (x_3), sedangkan variabel output pada penelitian ini adalah *container throughput volume* di TPKS (x_4).

Tabel 4.1 Data Variabel Input dan Output

NO	Bulan	Tahun	X₁	X₂	X₃	X₄
1	Januari	2010	47.871.117,51	2.692.937	3,72%	29.835
2	Februari	2010	49.535.510,67	2.694.404	3,81%	28.761
3	Maret	2010	51.199.903,83	2.697.499	3,43%	35.463
4	April	2010	48.636.070,33	2.698.172	3,91%	32.838
5	Mei	2010	51.554.028,33	2.700.082	4,16%	32.688
6	Juni	2010	54.471.986,34	2.702.667	5,05%	33.837
7	Juli	2010	50.968.124,81	2.704.556	6,22%	31.527
8	Agustus	2010	54.290.716,67	2.706.411	6,44%	28.352
9	September	2010	57.613.308,53	2.708.820	5,80%	30.068
10	Oktober	2010	49.209.135,64	2.710.501	5,67%	35.372
11	November	2010	52.361.285,00	2.713.192	6,96%	33.359
12	Desember	2010	55.513.434,36	2.714.646	7,02%	32.422
13	Januari	2011	52.871.384,82	2.716.301	6,84%	33.191
14	Februari	2011	54.709.628,33	2.717.781	6,65%	31.995
15	Maret	2011	56.547.871,85	2.720.904	6,16%	39.452
16	April	2011	54.193.962,69	2.721.582	5,96%	36.531
17	Mei	2011	57.445.370,67	2.723.508	5,54%	36.364
18	Juni	2011	60.696.778,65	2.726.116	4,61%	37.643
19	Juli	2011	57.148.136,93	2.728.021	4,79%	35.073
20	Agustus	2011	60.873.601,33	2.729.892	4,61%	31.541
21	September	2011	64.599.065,73	2.732.323	4,42%	33.450
22	Oktober	2011	54.344.193,76	2.734.018	3,79%	39.350
23	November	2011	57.825.275,33	2.736.732	6,33%	37.110
24	Desember	2011	61.306.356,91	2.738.199	4,15%	36.069
25	Januari	2012	56.447.819,88	2.738.987	3,65%	35.463
26	Februari	2012	60.514.386,67	2.740.479	3,56%	34.186
27	Maret	2012	64.580.953,45	2.743.628	3,97%	42.153
28	April	2012	59.780.809,67	2.744.312	4,50%	39.032
29	Mei	2012	63.468.319,00	2.746.254	4,45%	38.854
30	Juni	2012	67.155.828,33	2.748.884	4,53%	40.220
31	Juli	2012	62.174.090,33	2.750.805	4,56%	37.474
32	Agustus	2012	64.818.693,00	2.752.692	4,58%	33.700
33	September	2012	67.463.295,67	2.755.142	4,31%	35.740
34	Oktober	2012	59.660.784,76	2.756.852	4,61%	42.044
35	November	2012	62.708.413,67	2.759.589	4,32%	39.651
36	Desember	2012	65.756.042,57	2.761.068	4,30%	38.538
37	Januari	2013	62.339.658,17	2.761.037	4,57%	39.537
38	Februari	2013	65.038.767,00	2.762.541	5,31%	37.029
39	Maret	2013	67.737.875,83	2.765.715	5,90%	38.416
40	April	2013	65.007.112,54	2.766.404	5,57%	38.316

NO	Bulan	Tahun	X₁	X₂	X₃	X₄
41	Mei	2013	68.987.703,00	2.768.362	5,47%	47.760
42	Juni	2013	72.968.293,46	2.771.013	5,90%	42.693
43	Juli	2013	68.736.366,10	2.772.949	8,61%	45.372
44	Agustus	2013	72.460.854,00	2.774.851	8,79%	30.799
45	September	2013	76.185.341,90	2.777.322	8,40%	45.401
46	Oktober	2013	66.436.819,67	2.779.045	8,32%	45.717
47	November	2013	70.184.681,67	2.781.804	8,37%	42.957
48	Desember	2013	73.932.543,67	2.783.295	8,38%	44.706
49	Januari	2014	69.554.492,65	2.782.478	8,22%	45.284
50	Februari	2014	73.750.920,00	2.783.994	7,75%	45.241
51	Maret	2014	77.947.347,35	2.787.193	7,32%	50.234
52	April	2014	73.404.214,80	2.787.887	7,35%	51.288
53	Mei	2014	76.367.264,67	2.789.861	7,32%	52.757
54	Juni	2014	79.330.314,54	2.792.532	6,70%	49.662
55	Juli	2014	75.319.780,47	2.794.484	4,53%	47.410
56	Agustus	2014	80.050.781,67	2.796.400	3,99%	42.991
57	September	2014	84.781.782,86	2.798.890	4,53%	50.274
58	Oktober	2014	72.975.963,12	2.800.627	4,83%	46.990
59	November	2014	77.321.427,33	2.803.407	6,23%	47.249
60	Desember	2014	81.666.891,55	2.804.910	8,36%	46.291
61	Januari	2015	76.409.943,48	2.803.352	6,96%	52.308
62	Februari	2015	80.925.591,49	2.804.879	6,29%	47.759
63	Maret	2015	85.441.239,50	2.808.101	6,38%	51.397
64	April	2015	79.276.849,30	2.808.801	6,79%	54.263
65	Mei	2015	83.678.329,42	2.810.790	7,15%	57.194
66	Juni	2015	88.079.809,55	2.813.481	7,26%	55.325
67	Juli	2015	82.964.568,07	2.815.447	7,26%	35.684
68	Agustus	2015	87.340.318,00	2.817.378	7,18%	53.332
69	September	2015	91.716.067,93	2.819.887	6,83%	49.520
70	Oktober	2015	79.922.713,01	2.821.636	6,25%	51.153
71	November	2015	85.051.306,81	2.824.437	4,89%	48.618
72	Desember	2015	90.179.900,61	2.825.952	3,35%	51.648
73	Januari	2016	83.003.409,07	2.820.203	4,14%	54.162
74	Februari	2016	87.436.436,40	2.837.509	4,42%	47.792
75	Maret	2016	91.869.463,72	2.820.203	4,45%	52.131
76	April	2016	85.483.064,97	2.820.205	3,60%	54.456
77	Mei	2016	90.650.121,92	2.820.207	3,33%	57.379
78	Juni	2016	95.817.178,87	2.828.698	3,45%	55.240
79	Juli	2016	89.499.797,41	2.828.698	3,21%	38.324
80	Agustus	2016	94.210.313,07	2.828.698	2,79%	48.586
81	September	2016	98.920.828,72	2.852.370	3,07%	47.815
82	Oktober	2016	87.169.098,67	2.857.562	3,31%	51.480

NO	Bulan	Tahun	X ₁	X ₂	X ₃	X ₄
83	November	2016	92.076.791,66	2.852.370	3,58%	51.144
84	Desember	2016	96.984.484,66	2.852.370	3,02%	56.623
85	Januari	2017	89.791.555,07	2.861.153	3,49%	53.765
86	Februari	2017	94.816.848,02	2.857.686	3,83%	47.344
87	Maret	2017	99.861.104,33	2.828.736	3,82%	53.354
88	April	2017	93.226.266,07	2.859.968	4,17%	56.685
89	Mei	2017	98.319.200,66	2.827.715	4,33%	58.043
90	Juni	2017	103.412.135,26	2.859.968	4,37%	44.479
91	Juli	2017	96.768.074,94	2.877.784	3,88%	54.191
92	Agustus	2017	102.043.736,10	2.773.685	3,82%	48.260
93	September	2017	107.319.397,26	2.877.793	3,72%	51.725
94	Oktober	2017	95.075.983,16	2.877.793	3,58%	55.129
95	November	2017	100.503.153,45	2.877.793	3,30%	54.892

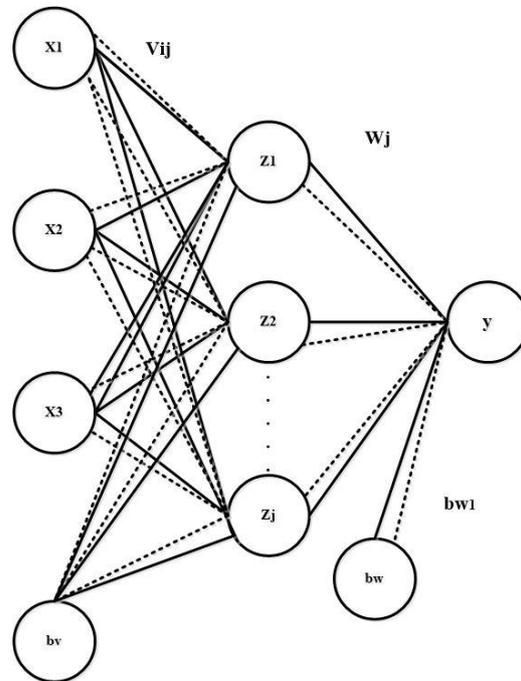
4.2 Pengolahan Data

Pada bagian ini dijelaskan pengolahan data pembuatan model prediksi *container throughput* berdasarkan variabel input dan output yang telah ditentukan. Metode yang digunakan adalah *Artificial Neural Network (ANN)* algoritma *backpropagation* di mana pengolahan data dilakukan dengan menggunakan software MATLAB.

4.2.1 Arsitektur Jaringan

Arsitektur jaringan pada algoritma *backpropagation* secara umum terdiri dari 3 lapisan yaitu, lapisan input, lapisan tersembunyi, dan lapisan output di mana masing-masing dari lapisan tersebut memiliki *neuron*. Pada lapisan input dan output, jumlah *neuron* yang digunakan disesuaikan dengan jumlah variabel input dan outputnya. Sementara itu, pada lapisan tersembunyi tidak terdapat aturan khusus mengenai jumlah *neuron*, sehingga teknik yang digunakan untuk menentukan jumlah *neuron* pada lapisan tersembunyi adalah percobaan. Jumlah lapisan tersembunyi yang digunakan dalam penelitian ini adalah 1 yang disesuaikan dengan penelitian sebelumnya. Hal yang perlu diperhatikan untuk lapisan tersembunyi adalah penambahan jumlah lapisan dapat menyebabkan peningkatan *overfitting* dan waktu eksekusi serta selalu memilih jaringan dengan performa terbaik dengan jumlah *neuron* sedikit (Anggraeni et al., 2015).

Adapun arsitektur jaringan yang digunakan dalam penelitian ini dapat dilihat pada Gambar 4.2 Arsitektur Jaringan berikut.



Gambar 4.2 Arsitektur Jaringan

Arsitektur jaringan pada Gambar 4.2 menunjukkan lapisan output yang terdiri dari 1 *neuron* yaitu nilai y yang merupakan *container throughput volume*. Jumlah *neuron* pada lapisan tersembunyi adalah 15 *neuron*. Penentuan jumlah *neuron* pada lapisan tersembunyi dilakukan dengan teknik percobaan yang disesuaikan dengan nilai *error* yang dihasilkan. Selain lapisan dan *neuron*, dalam arsitektur jaringan *backpropagation* terdapat bobot-bobot terhubung pada *neuron*. Bobot yang menghubungkan antara lapisan input dengan lapisan tersembunyi adalah bobot v_{ij} (dari neuron input ke- i dengan neuron ke- j pada lapisan tersembunyi). Misalnya, bobot yang menghubungkan lapisan input dengan neuron pertama pada lapisan tersembunyi adalah v_{11} , v_{21} , dan v_{31} . Bobot yang menghubungkan lapisan tersembunyi dengan lapisan output adalah w_j (lapisan tersembunyi ke- j). Berdasarkan arsitektur jaringan pada gambar sebelumnya, bobot-bobot yang menghubungkan lapisan tersembunyi dengan lapisan output w_1 - w_{15} . Terdapat 2 bobot bias dalam arsitektur jaringan ini yaitu bobot bias input (b_{v_j} : bobot bias yang

menuju *neuron* ke- j pada lapisan tersembunyi) dan bobot bias lapisan (bw_1) yang menuju output y .

4.2.2 Pembagian Data Pelatihan dan Data Pengujian

Pada penelitian ini data yang digunakan adalah data *container throughput* pada bulan Januari 2010 sampai November 2017 atau sebanyak 95 bulan dimana data tersebut dibagi menjadi dua data, yaitu data pelatihan dan data pengujian. Pada penelitian ini data pelatihan yang digunakan adalah data *container throughput* selama 80 bulan. Sementara itu data pengujian yang digunakan adalah data *container throughput* selama 15 bulan. Pembagian kedua data tersebut dilakukan secara random dengan mempertimbangkan jumlah data, nilai *Mean Square Error* (MSE) pada jaringan, dan nilai *Mean Absolute Percentage Error* (MAPE) yang dihasilkan pada output pelatihan dan pengujian. Dalam hal ini data pelatihan digunakan untuk membangun jaringan dengan proses pembelajaran pola data input terhadap outputnya. Data pengujian digunakan untuk mengevaluasi output dari jaringan yang telah terbentuk. Adapun jumlah data yang digunakan untuk data pelatihan adalah sebesar 85%, sedangkan untuk data pengujian adalah sebesar 15%.

4.2.3 Input Data Pelatihan

Input data pelatihan dilakukan dengan import data *excel* ke MATLAB. Data yang di *import* hanya data pelatihan yang akan diolah dan terdiri dari 3 variabel input dan 1 variabel output. *Import* data dengan *syntax* sebagai berikut.

filename= 'Prediksi Container.xls';.....(4.1)

sheet = 3;.....(4.2)

xlRange = 'A1:D80';.....(4.3)

Data = xlsread(filename, sheet, xlRange);(4.4)

Setelah data *excel* dibaca oleh MATLAB, selanjutnya dilakukan pendefinisian variabel untuk matriks input dan output. Pendefinisian ini dilakukan dengan *syntax* sebagai berikut.

P = Data(:,1:3);(4.5)

$$T = \text{Data}(:,4)'; \dots\dots\dots(4.6)$$

Dengan *syntax* tersebut, data yang telah di-import didefinisikan sebagai matriks P dan matriks T. Matriks P yaitu kolom 1-3 merupakan matriks input dan matriks T yaitu kolom 4 merupakan matriks output untuk data pelatihan.

4.2.4 Pre-processing Data

Sebelum dilakukan pelatihan jaringan, data yang akan diolah dinormalisasi terlebih dahulu baik data input maupun target sehingga data tersebut terdapat pada skala tertentu. Dalam penelitian ini metode normalisasi yang digunakan adalah mean dan standar deviasi dan dilakukan dengan *syntax* pada MATLAB sebagai berikut.

$$[\text{pn}, \text{meanp}, \text{stdp}, \text{tn}, \text{meant}, \text{stdt}] = \text{prestd}(\text{P}, \text{T}); \dots\dots\dots(4.7)$$

Keterangan	:
pn	: matriks input ternormalisasi
tn	: matriks target ternormalisasi
meanp	: mean pada matriks input asli (P)
stdp	: standar deviasi pada matriks input asli (P)
meant	: mean pada matriks output asli (T)
stdt	: standar deviasi pada matriks output asli (T)

Output yang dihasilkan dari *syntax* ini adalah matriks input dan output target yang telah ternormalisasi. Hasil normalisasi ini kemudian digunakan untuk membentuk jaringan.

4.2.5 Pembentukan Jaringan Syaraf *Feedforward*

Pembentukan jaringan syaraf *feedforward* dilakukan untuk mendapatkan nilai *error* output dengan *syntax* sebagai berikut.

$$\text{net} = \text{newff}(\text{minmax}(\text{pn}), [15, 1], \{\text{'tansig' 'purelin'}\}, \text{'traingdm'}); \dots\dots\dots(4.8)$$

Syntax yang digunakan pada arsitektur jaringan ini adalah [15 1] yang berarti bahwa lapisan tersembunyi yang digunakan berjumlah 1 dengan 15 *neuron* yang ada di dalamnya dan 1 *neuron* pada lapisan output. Fungsi aktivasi yang digunakan sigmoid bipolar (tansig) dan purelin untuk fungsi aktivasi pada lapisan tersembunyi. Sementara itu, metode pembelajaran yang digunakan adalah *Gradient Descent* dengan Momentum (traingdm). Pemilihan fungsi aktivasi dan metode pembelajaran tersebut dilakukan dengan teknik percobaan dan berdasarkan studi literatur. Kriteria yang digunakan untuk menentukan fungsi aktivasi adalah kemampuannya untuk mempercepat fase pembelajaran dan meningkatkan keakuratan jaringan syaraf (Anggraeni et al., 2015).

4.2.6 Inisialisasi Bobot Awal

Dalam algoritma backpropagation terdapat beberapa bobot awal yang akan dibangkitkan secara random. Inisialisasi bobot awal disesuaikan dengan arsitektur jaringan yang telah dibentuk di awal. Inisialisasi bobot awal dilakukan dengan *syntax* sebagai berikut.

$$\text{BobotAwal_Input} = \text{net.IW}\{1,1\}; \dots \dots \dots (4.9)$$

$$\text{BobotAwal_Bias_Input} = \text{net.b}\{1,1\}; \dots \dots \dots (4.10)$$

$$\text{BobotAwal_Lapisan} = \text{net.LW}\{2,1\}; \dots \dots \dots (4.11)$$

$$\text{BobotAwal_Bias_Lapisan} = \text{net.b}\{2,1\}; \dots \dots \dots (4.12)$$

Output dari *syntax* tersebut adalah bobot awal input, bobot awal bias input, bobot awal lapisan, dan bobot awal bias lapisan.

4.2.7 Training Parameter

Sebelum dilakukan proses pelatihan dan simulasi jaringan, terdapat beberapa parameter yang harus ditetapkan terlebih dahulu di mana parameter ini berpengaruh terhadap performansi jaringan. *Training* parameter yang digunakan dalam penelitian ini diuraikan sebagai berikut (Kusumadewi, 2004).

1. Maksimum epoh

Maksimum epoh merupakan jumlah epoh maksimum yang diperbolehkan untuk dilakukan selama proses pelatihan. Apabila nilai epoh telah melebihi epoh maksimum maka iterasi akan dihentikan. Untuk menetapkan nilai pada parameter ini dilakukan dengan *syntax* sebagai berikut.

$$\text{net.trainParam.epochs} = \text{MaxEpoh} \dots\dots\dots(4.13)$$

2. Kinerja tujuan

Kinerja tujuan adalah target nilai fungsi kinerja atau target *error*. Apabila fungsi kinerja kurang dari atau sama dengan kinerja tujuan maka iterasi akan dihentikan. Penetapan nilai pada parameter ini dilakukan dengan *syntax* sebagai berikut.

$$\text{net.trainParam.goal} = \text{TargetError} \dots\dots\dots(4.14)$$

3. *Learning rate*

Learning rate adalah kecepatan pembelajaran. *Learning rate* yang semakin besar berdampak pada semakin besarnya langkah pembelajaran. Namun, *learning rate* yang diset terlalu besar dapat mengakibatkan algoritma tidak stabil. Sebaliknya, jika *learning rate* diset terlalu kecil, maka algoritma konvergen dalam jangka waktu yang sangat lama. Penetapan nilai pada parameter ini dilakukan dengan *syntax* sebagai berikut.

$$\text{net.trainParam.lr} = \text{LearningRate} \dots\dots\dots(4.15)$$

4. Jumlah epoh yang ditunjukkan kemajuannya

Nilai ini menunjukkan jumlah epoh berselang yang akan ditunjukkan kemajuannya. Untuk menetapkan nilai pada parameter ini dilakukan dengan *syntax* sebagai berikut.

$$\text{net.trainParam.show} = \text{EpohShow} \dots\dots\dots(4.16)$$

5. Nilai momentum

Untuk menetapkan nilai pada parameter ini dilakukan dengan *syntax* sebagai berikut.

$$\text{net.trainParam.mc} = \text{Momentum} \dots\dots\dots(4.17)$$

Setelah dilakukan percobaan terhadap beberapa nilai-nilai tersebut, *training* parameter yang ditentukan dalam penelitian ini dapat dilihat pada Tabel 4.2 berikut.

Tabel 4.2 *Training* Parameter

Training Parameter	Parameter
Max. Epoch	300
Target <i>Error</i>	0
<i>Learning Rate</i>	0,2
Epoch <i>Show</i>	25
Momentum	0,75

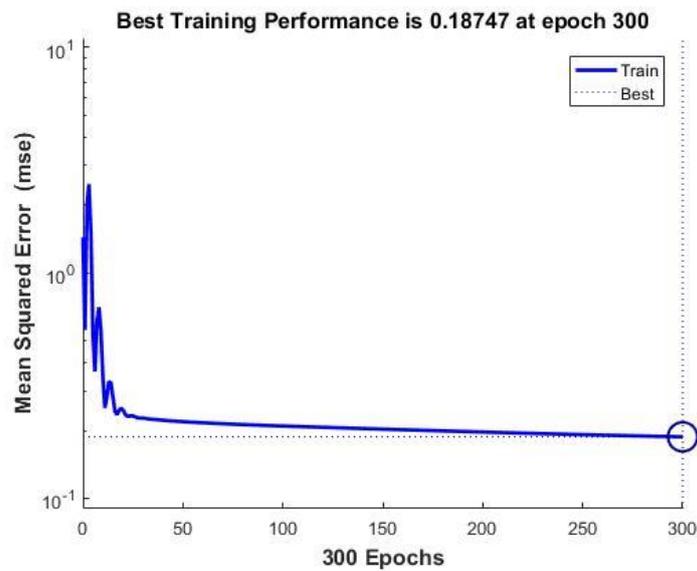
Penentuan *training* parameter tersebut dilakukan dengan cara percobaan. Hasil percobaan ini memberikan perubahan terhadap nilai *error* pada proses pelatihan maupun pengujian. *Training* parameter yang dipilih untuk digunakan adalah yang menghasilkan akurasi terbaik.

4.2.8 Pelatihan Jaringan

Pelatihan jaringan bertujuan untuk mengetahui performansi dan nilai *error* pada jaringan. Nilai *error default* yang digunakan adalah *Mean Square Error* (MSE). Pelatihan jaringan dilakukan dengan *syntax* sebagai berikut.

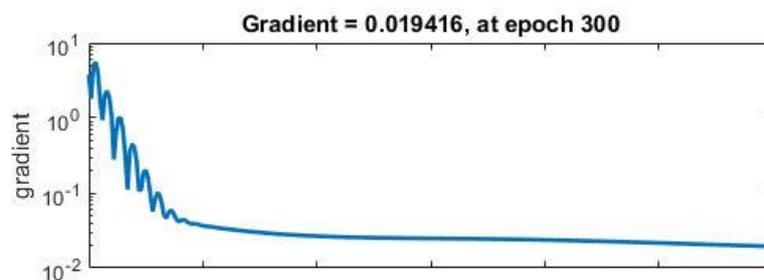
$$\text{net} = \text{train}(\text{net}, \text{pn}, \text{tn}); \dots\dots\dots (4.18)$$

Apabila hasil pelatihan memiliki nilai *error* yang besar maka perubahan terhadap *training* parameter perlu dilakukan untuk mendapatkan performansi yang baik. Hasil output pelatihan jaringan untuk model prediksi *container throughput* dapat dilihat pada Gambar 4.3 berikut.



Gambar 4.3 Pelatihan Jaringan

Pada proses pelatihan, iterasi berhenti ketika telah mencapai epoch ke-300 yaitu epoch maksimum yang telah ditentukan sebelumnya, walaupun kinerja tujuannya belum mencapai batas yang ditentukan. Performansi jaringan ini dapat diukur dari nilai MSE yang dihasilkan yaitu sebesar 0,18747. Adapun nilai *gradient* dapat dilihat pada Gambar 4.4 berikut.



Gambar 4.4 *Gradient* Data Pelatihan

Berdasarkan Gambar 4.4, nilai *gradient* yang dihasilkan pada epoch 300 pada data pelatihan adalah sebesar 0,019416.

4.2.9 Bobot-Bobot Akhir

Selain bobot awal, dalam algoritma *backpropagation* terdapat bobot-bobot akhir yang didapatkan dari proses pelatihan jaringan. Algoritma *backpropagation* menggunakan nilai *error* dari output untuk mengubah nilai bobot-bobotnya dalam arah mundur atau *backward* (Kusumadewi, 2004). Untuk melihat bobot akhir dilakukan dengan *syntax* sebagai berikut.

$$\text{BobotAkhir_Input} = \text{net.IW}\{1,1\}; \dots\dots\dots(4.19)$$

$$\text{BobotAkhir_Bias_Input} = \text{net.b}\{1,1\}; \dots\dots\dots(4.20)$$

$$\text{BobotAkhir_Lapisan} = \text{net.LW}\{2,1\}; \dots\dots\dots(4.21)$$

$$\text{BobotAkhir_Bias_Lapisan} = \text{net.b}\{2,1\}; \dots\dots\dots(4.22)$$

Output dari proses ini adalah bobot-bobot akhir pada bobot input, bias input, bobot lapisan, dan bias lapisan.

4.2.10 Simulasi Jaringan

Simulasi jaringan akan menghasilkan output jaringan syaraf dari hasil proses pelatihan sebelumnya, sehingga dapat diketahui perbedaan antara output target pada data asli dengan output jaringan syaraf. Simulasi jaringan dilakukan dengan *syntax* sebagai berikut.

$$\text{an} = \text{sim}(\text{net}, \text{pn}); \dots\dots\dots(4.23)$$

Pada bagian *pre-processing* data dilakukan normalisasi untuk mengubah nilai input dan output dalam skala tertentu yang di mana hal ini dapat menghasilkan *range* nilai yang sama pada output jaringan syarafnya. Oleh karena itu, perlu dilakukan *post-processing* data untuk mengubah output jaringan tersebut sesuai dengan kondisi semula. *Syntax* yang digunakan untuk *post-processing* data adalah sebagai berikut.

$$\text{a} = \text{poststd}(\text{an}, \text{meant}, \text{stdt}); \dots\dots\dots(4.24)$$

Selanjutnya, untuk menampilkan hasil output jaringan dan nilai errornya terhadap output asli, menggunakan *syntax* berikut.

$$H = [(1:\text{size}(P,2))' T' a' (T'-a)]; \dots \dots \dots (4.25)$$

Pada *syntax* tersebut, nilai T merupakan target output pada data asli. Di samping itu, nilai a merupakan output yang dihasilkan oleh simulasi jaringan dan nilai T-a adalah nilai *error* yang merupakan selisih antara data target dengan output jaringan. Dengan *syntax* tersebut, dapat diketahui hasil output jaringan dan nilai *errornya* terhadap output asli.

Tabel 4.3 Output Jaringan Data Pelatihan

No	Container Throughput (TEUs)	Output ANN	Error
1	29.835	31.324	-1.489
2	28.761	31.510	-2.749
3	35.463	33.544	1.919
4	32.838	32.150	688
5	32.688	32.077	611
6	33.837	31.525	2.313
7	31.527	31.396	131
8	28.352	32.404	-4.052
9	30.068	33.247	-3.179
10	35.372	32.984	2.388
11	33.359	31.558	1.800
12	32.422	32.639	-217
13	33.191	32.938	253
14	31.995	33.944	-1.948
15	39.452	35.509	3.943
16	36.531	35.900	631
17	36.364	36.196	168
18	37.643	35.218	2.425
19	35.073	35.795	-722
20	31.541	34.672	-3.131
21	33.700	36.987	-3.287
22	35.740	37.287	-1.547
23	42.044	38.306	3.738
24	39.651	37.728	1.923
25	38.538	37.917	621
26	39.537	38.294	1.243
27	37.029	39.375	-2.346
28	38.416	41.030	-2.614

No	Container Throughput (TEUs)	Output ANN	Error
29	38.316	40.786	-2.470
30	47.760	41.836	5.924
31	42.693	43.988	-1.295
32	45.372	41.526	3.846
33	30.799	40.944	-10.145
34	45.401	43.012	2.389
35	45.717	43.532	2.185
36	42.957	43.861	-904
37	44.706	44.170	536
38	45.284	44.569	715
39	45.241	46.775	-1.534
40	50.234	48.928	1.306
41	51.288	48.725	2.563
42	52.757	49.449	3.308
43	49.662	50.262	-600
44	47.410	47.458	-48
45	42.991	47.181	-4.190
46	50.274	48.759	1.515
47	46.990	47.632	-642
48	47.249	50.048	-2.799
49	46.291	47.845	-1.554
50	52.308	51.441	867
51	47.759	50.672	-2.913
52	51.397	50.934	463
53	54.263	51.624	2.639
54	57.194	51.169	6.025
55	55.325	50.050	5.275
56	35.684	45.613	-9.929
57	53.332	49.593	3.739
58	49.520	49.085	435
59	47.792	50.811	-3.019
60	52.131	51.082	1.049
61	54.456	49.566	4.890
62	57.379	49.754	7.625
63	55.240	51.292	3.948
64	38.324	50.351	-12.027
65	48.586	50.413	-1.827
66	47.815	52.636	-4.821
67	51.480	52.079	-599
68	51.144	52.240	-1.096
69	56.623	52.178	4.445
70	53.765	52.648	1.117
71	47.344	52.765	-5.421

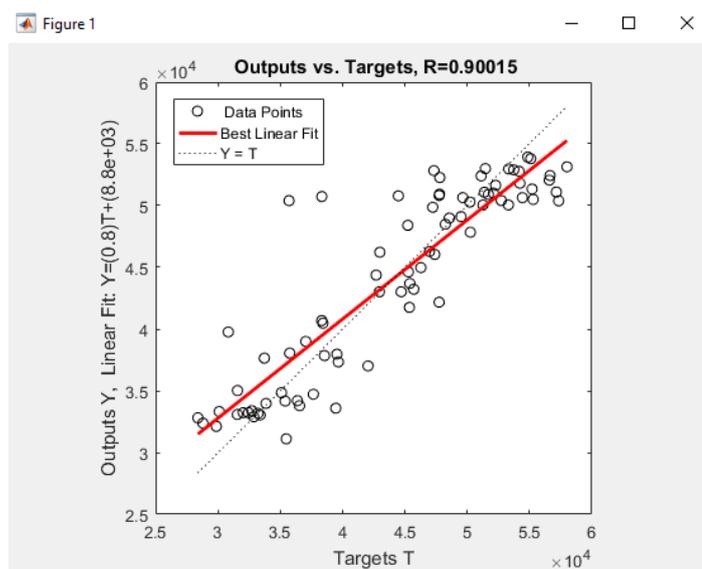
No	Container Throughput (TEUs)	Output ANN	Error
72	53.354	53.320	34
73	56.685	52.518	4.167
74	58.043	53.400	4.643
75	44.479	51.345	-6.866
76	54.191	52.300	1.891
77	48.260	48.393	-133
78	51.725	51.301	424
79	55.129	53.037	2.092
80	54.892	53.137	1.755

4.2.11 Evaluasi Simulasi Jaringan

Pada penelitian ini, simulasi jaringan dievaluasi dengan membandingkan nilai target pada output asli dengan output yang dihasilkan oleh jaringan syaraf. Evaluasi dilakukan dengan mengidentifikasi hubungan linier dan plot data keduanya. *Syntax* yang digunakan untuk identifikasi hubungan linier pada hasil simulasi jaringan adalah sebagai berikut.

$$[m1,a1,r1] = \text{postreg}(a,T); \dots\dots\dots(4.26)$$

Hasil output simulasi jaringan dapat dilihat pada Gambar 4.5 Output Regresi Linier Simulasi Jaringan berikut.



Gambar 4.5 Output Regresi Linier Simulasi Jaringan

Dari gambar tersebut terlihat bahwa nilai R yang dihasilkan antara target asli data pelatihan dengan output jaringan syaraf yaitu sebesar 0,90015. Adapun output lainnya yaitu nilai $m1 = 0,8008$, nilai $a1 = 8780,4570$, dan nilai $r1 = 0,90015$. Nilai $m1$ merupakan gradien garis terbaik dengan nilai $a1$ adalah konstanta. Nilai R dan $r1$ merupakan koefisien korelasi yang di mana hasil yang didapat mendekati 1 sehingga menunjukkan kecocokan yang baik antara target output pada data asli dengan output jaringan syaraf. Persamaan garis untuk kecocokan terbaik adalah sebagai berikut.

$$Y = (0,8008) T + 8780,4570 \dots \dots \dots (4.27)$$

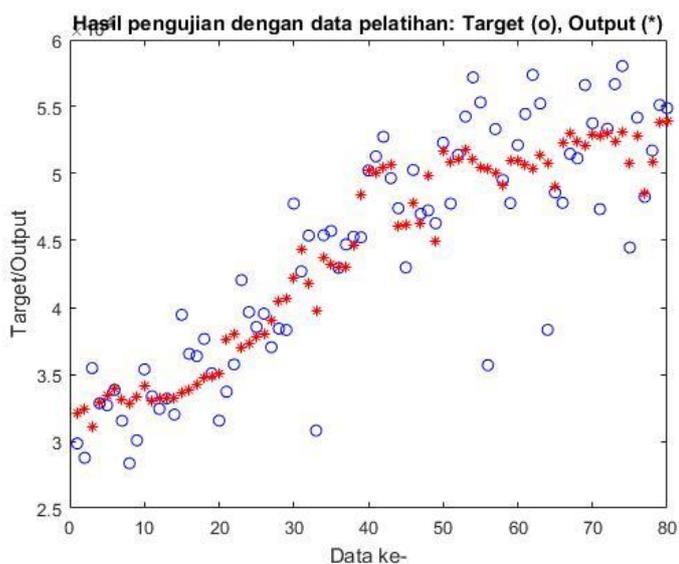
Di samping itu, output selanjutnya adalah untuk menampilkan perbandingan antara target pada data asli dengan output jaringan syaraf berupa plotting data. Dengan demikian posisi masing-masing data dapat dilihat dengan jelas. *Syntax* yang digunakan dalam hal ini adalah sebagai berikut.

$$\text{plot}([1:\text{size}(P,2)]', T, 'bo', [1:\text{size}(P,2)]', a, 'r*'); \dots \dots \dots (4.28)$$

$$\text{title}('Hasil Pengujian dengan Data Pelatihan: Target (o), Output (*)'); \dots \dots (4.29)$$

$$\text{xlabel}('Data ke'); \text{ylabel}('Target/Output'); \dots \dots \dots (4.30)$$

Output yang dihasilkan oleh syntax tersebut dapat dilihat pada Gambar 4.6 berikut.



Gambar 4.6 *Plot* Data Hasil Simulasi Jaringan Data Pelatihan

Perbandingan antara target output pada data asli dengan output jaringan dapat dilihat pada plot data pada gambar-gambar sebelumnya. Dari gambar tersebut output jaringan ditunjukkan dengan (*) atau yang berwarna merah dan target asli ditunjukkan dengan (o) dengan warna biru. Sebagian besar, posisi antara data tersebut saling berdekatan dan hampir menempati posisi yang sama. Di samping itu, sebagian kecil di antaranya juga sudah menempati posisi yang sama antara target dan output jaringan, sehingga ini disebut sebagai hasil terbaik karena antara target dan output jaringan memiliki nilai yang sama. Nilai yang dihasilkan output jaringan tidak ada yang melebihi batas minimum pada target.

Untuk mengetahui perbandingan dan mengevaluasi hasil simulasi jaringan, dilakukan perhitungan *error* untuk data pelatihan. Perhitungan *error* yang digunakan adalah *Mean Absolute Percentage Error* (MAPE) dengan rumus sebagai berikut.

$$\text{MAPE} = \frac{\sum \frac{|x_i - f_i|}{x_i}}{n} \times 100\% \quad \dots\dots\dots(4.31)$$

Keterangan :

- x_i : data target asli
- f_i : data output jaringan syaraf
- n : Jumlah data keseluruhan

Tabel 4.4 Perbandingan Target Output Data Pelatihan

	<i>Container Volume</i>	<i>Output ANN</i>	<i>Error</i>	<i>Error Absolute</i>	<i>MAPE</i>
Rata-rata	44.075	44.019	56	2.609	6,15%
Maximum	58.043	53.400	7.625	12.027	32,94%
Minimum	28.352	31.324	-12.027	34	0,06%

4.2.12 Input Data Pengujian

Selain data pelatihan, data pengujian juga perlu diinputkan ke MATLAB. Cara input data sama dengan input data pelatihan dan terdiri dari 3 variabel input dan 1 variabel output.

Import data dilakukan dengan *syntax* sebagai berikut.

filename = 'Prediksi Container.xls';(4.32)

sheet = 3; (4.33)

xlRange = 'A81:D95';(4.34)

Data = xlsread(filename, sheet, xlRange);(4.35)

Setelah data excel dibaca oleh MATLAB, juga dilakukan pendefinisian variabel untuk matriks input dan output. Pendefinisian untuk data pengujian dilakukan dengan *syntax* sebagai berikut.

Q = Data(:,1:3)';(4.36)

TQ = Data(:,4)';(4.37)

Dengan *syntax* tersebut, data yang telah di-*import* didefinisikan sebagai matriks Q dan matriks TQ. Matriks Q yaitu kolom 1-3 merupakan matriks input dan matriks TQ yaitu kolom 4 merupakan matriks output untuk data pengujian.

4.2.13 Pengujian Jaringan

Pengujian jaringan dilakukan dengan data-data pengujian yang tidak diikuti dalam proses pelatihan jaringan. Data pengujian yang telah diinputkan dinormalisasi terlebih dahulu dengan metode mean dan standar deviasi menggunakan *syntax* sebagai berikut.

Qn = trstd(Q,meanp,stdp);(4.38)

Setelah data baru tersebut dinormalisasi, proses pengujian jaringan dilakukan dengan *syntax* sebagai berikut.

bn = sim(net,Qn);(4.39)

Dikarenakan pada langkah sebelumnya dilakukan proses normalisasi pada data pengujian, selanjutnya dilakukan post-processing data yaitu denormalisasi untuk mengubah output jaringan tersebut sesuai dengan kondisi semula. *Syntax* yang digunakan untuk *post-processing* data adalah sebagai berikut.

b = poststd(bn,meant,stdt);(4.40)

Selanjutnya, untuk menampilkan hasil output pengujian jaringan dan nilai errornya terhadap output asli, menggunakan *syntax* berikut.

$$L = [(1:\text{size}(Q,2))' \text{TQ}' \text{b}' (\text{TQ}'\text{-b}')]; \dots\dots\dots(4.41)$$

Pada *syntax* tersebut, nilai TQ merupakan target output pada data pengujian. Nilai b merupakan output yang dihasilkan oleh pengujian jaringan dan nilai TQ-b adalah nilai *error* yang merupakan selisih antara data target dengan output jaringan. Dengan *syntax* tersebut, hasil output jaringan dan nilai *error*nya terhadap output asli dapat dilihat pada Tabel 4.5

Tabel 4.5 Output Pengujian Jaringan Data Pengujian

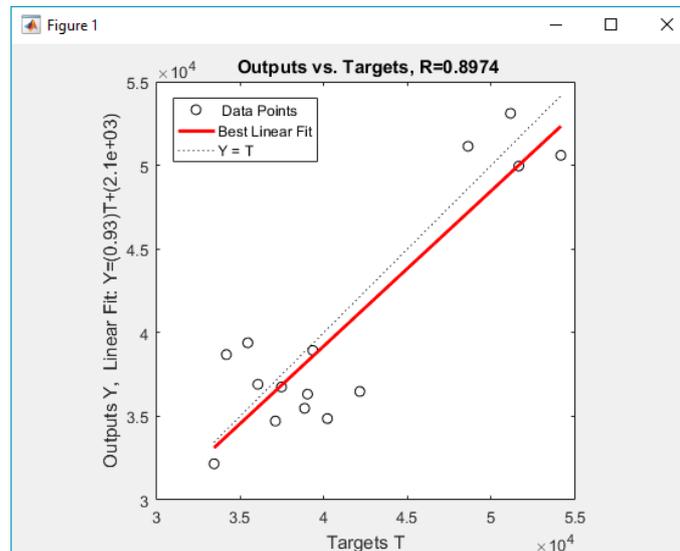
No	Container Throughput (TEUs)	Output ANN	Error
1	33.450	33.927	-478
2	39.350	37.134	2.216
3	37.110	38.411	-1.301
4	36.069	36.017	52
5	35.463	35.771	-308
6	34.186	34.953	-767
7	42.153	35.156	6.997
8	39.032	36.312	2.720
9	38.854	35.902	2.952
10	40.220	36.269	3.951
11	37.474	36.762	712
12	51.153	51.110	43
13	48.618	49.925	-1.307
14	51.648	49.348	2.300
15	54.162	49.706	4.456

4.2.14 Evaluasi Pengujian Jaringan

Pengujian jaringan dievaluasi dengan membandingkan nilai target pada output asli dengan output yang dihasilkan oleh jaringan syaraf. Evaluasi dilakukan dengan mengidentifikasi hubungan linier dan plot data keduanya. *Syntax* yang digunakan untuk mengidentifikasi hubungan linier pada hasil pengujian jaringan adalah sebagai berikut.

$$[m2,b1,r2] = \text{postreg}(b,\text{TQ}); \dots\dots\dots(4.42)$$

Hasil output pengujian jaringan dapat dilihat pada Gambar 4.7 berikut.



Gambar 4.7 Output Regresi Linear Pengujian Jaringan

Berdasarkan gambar tersebut terlihat bahwa nilai R yang dihasilkan antara target asli data pengujian dengan output jaringan syaraf yaitu sebesar 0,8974. Adapun output lainnya yaitu nilai $m_2 = 0,9280$, nilai $b_1 = 2075,0813$, dan nilai $r_2 = 0.8947$. Nilai m_2 merupakan gradien garis terbaik dengan nilai b_1 adalah konstanta pada data pengujian. Nilai R dan r_2 merupakan koefisien korelasi data pengujian yang di mana hasil yang didapat mendekati kurang dari 0,5 sehingga menunjukkan bahwa terdapat perbedaan yang cukup besar antara target output pada data asli dengan output jaringan syaraf. Hal ini dipengaruhi oleh sample data, sample hanya dilakukan dalam beberapa bulan sekali sehingga tidak sepenuhnya representatif. Adapun persamaan garis dari hasil pengujian di atas adalah sebagai berikut.

$$Y = (0,9280) T + 2075,0813 \dots \dots \dots (4.43)$$

Output selanjutnya adalah untuk menampilkan perbandingan antara target pada data asli dengan output jaringan syaraf berupa *plotting* data. Dengan demikian posisi masing-masing data dapat dilihat dengan jelas. *Syntax* yang digunakan dalam hal ini adalah sebagai berikut.

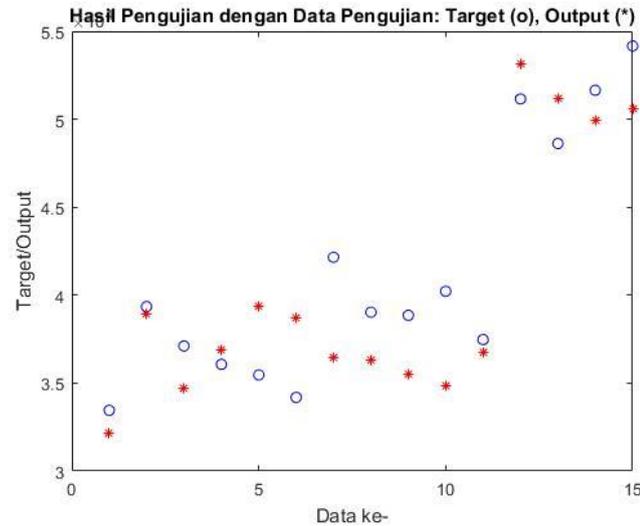
$$k = [1:\text{size}(Q,2)]'; \dots \dots \dots (4.44)$$

```

plot(k,TQ,'bo','k,b','r*'); ..... (4.45)
title ('Hasil Pengujian dengan Data Pengujian: Target (o), Output (*)'); ..... (4.46)
xlabel ('Data ke-'); ylabel ('Target/Output'); ..... (4.47)

```

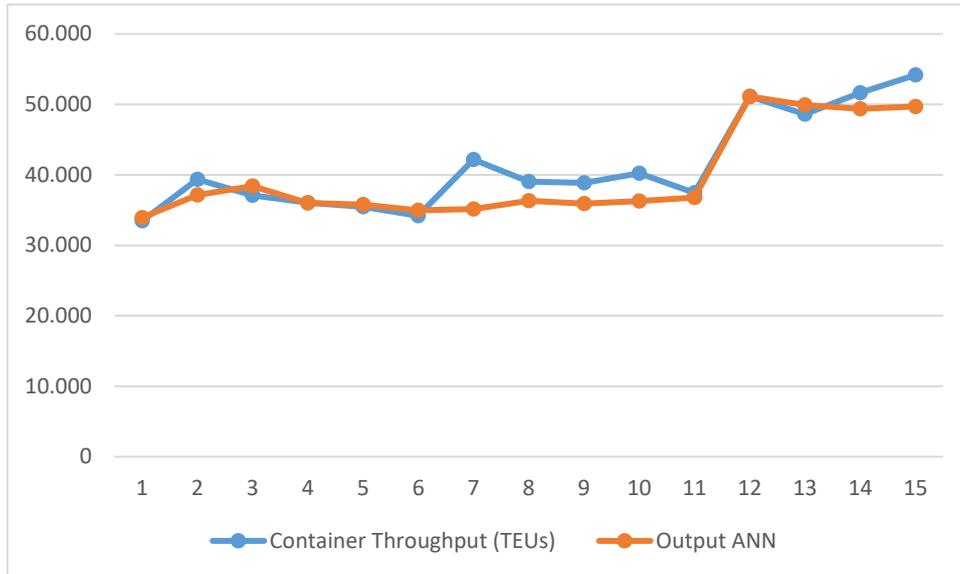
Output yang dihasilkan oleh *syntax* tersebut dapat dilihat pada Gambar 4.8



Gambar 4.8 *Plot Data Hasil Pengujian Jaringan Data Pengujian*

Perbandingan antara target output pada data asli dengan output jaringan untuk data pengujian dapat dilihat pada *plot* data pada gambar-gambar sebelumnya untuk masing-masing data tipe kamar hotel. Dari gambar tersebut output jaringan ditunjukkan dengan (*) berwarna merah dan target asli ditunjukkan dengan (o) dengan warna biru. Sebagian besar, posisi antara data target dan output jaringan berdekatan dan hampir menempati posisi yang sama. Apabila menempati posisi yang sama antara target dan output jaringan, maka disebut sebagai hasil terbaik karena antara target dan output jaringan memiliki nilai yang sama. Namun, berdasarkan hasil pengujian tidak didapatkan data target dan output yang *fit* dan menempati posisi yang sama.

Gambar 4.9 berikut merupakan representasi dari hasil pengujian jaringan dengan menggunakan grafik untuk memvisualisasikan perbandingan dan *error* yang dihasilkan.



Gambar 4.9 Perbandingan Target Aktual dan Output ANN

Untuk mengetahui perbandingan dan mengevaluasi hasil pengujian jaringan, dilakukan perhitungan error untuk data pelatihan. Perhitungan error yang digunakan adalah *Mean Absolute Percentage Error* (MAPE) yang juga digunakan pada evaluasi simulasi jaringan. Hasil perbandingan dan perhitungan nilai *error* secara keseluruhan dapat dilihat pada Tabel 4.6 berikut.

Tabel 4.6 Perbandingan Target Output Data Pengujian

	<i>Container Volume</i>	Output ANN	<i>Error</i>	<i>Error Absolute</i>	MAPE
Rata-rata	41.263	39.780	1.483	2.037	4,81%
Maximum	54.162	51.110	6.997	6.997	16,60%
Minimum	33.450	33.927	-1.307	43	0,08%