

**PREDIKSI KEBANGKRUTAN MENGGUNAKAN METODE  
BACKPROPAGATION (STUDI KASUS: PERSEROAN TERBATAS  
TERDAFTAR PADA BURSA EFEK INDONESIA)**

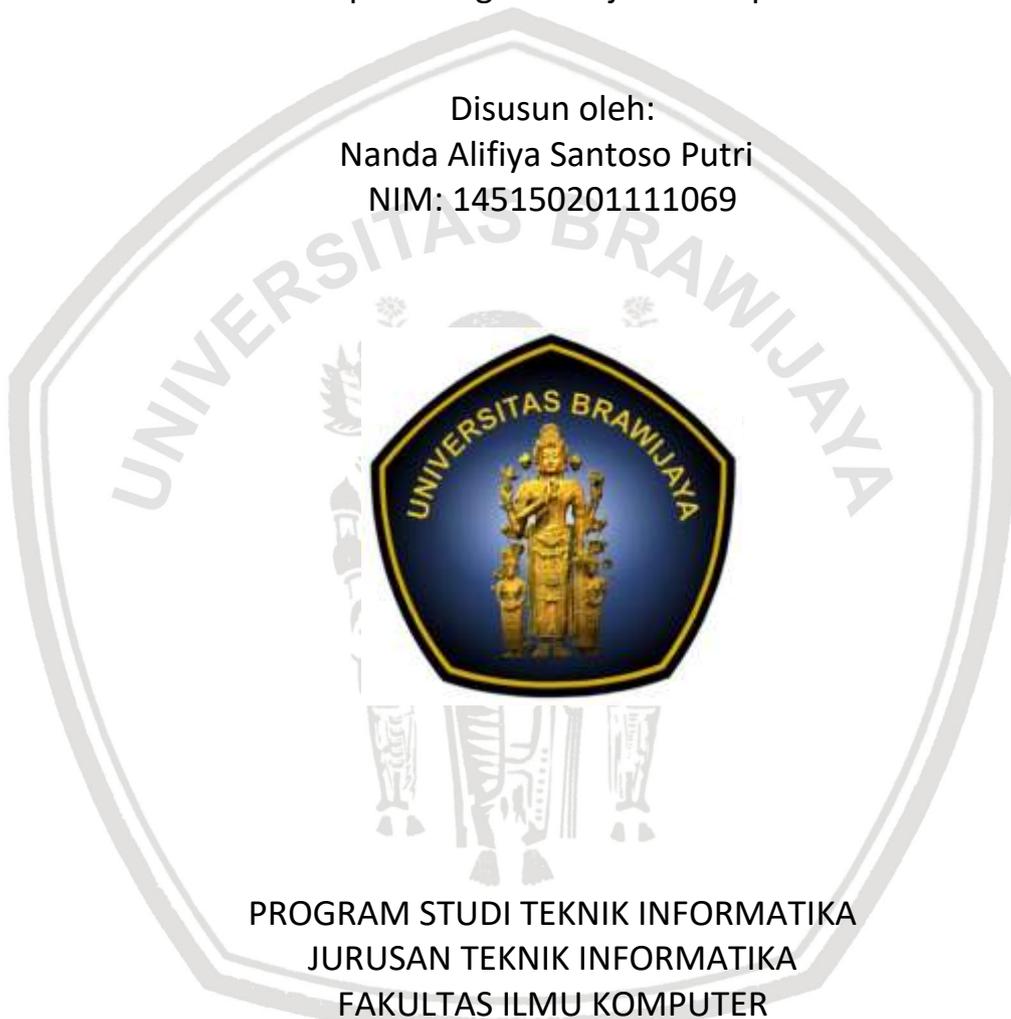
**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

Nanda Alifiya Santoso Putri

NIM: 145150201111069



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

# PENGESAHAN

PREDIKSI KEBANGKRUTAN MENGGUNAKAN METODE *BACKPROPAGATION*  
(STUDI KASUS: PERSEROAN TERBATAS TERDAFTAR PADA BURSA EFEK  
INDONESIA)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Nanda Alifiya Santoso Putri  
NIM: 145150201111069

Skripsi ini telah diuji dan dinyatakan lulus pada  
26 Desember 2018  
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II



Dian Eka Ratnawati, S.Si, M.Kom  
NIP. 19730619 200212 2 001



Bayu Rahayudi, S.T, M.T  
NIP. 19740712 200604 1 001

Mengetahui  
Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T, Ph.D  
NIP: 19710518 200312 1 001

## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 26 Desember 2018



Nanda Alifiya Santoso Putri  
NIM: 145150201111069

## KATA PENGANTAR

Puji syukur selalu penulis panjatkan ke hadirat Allah SWT karena atas limpahan rahmat dan kasih sayang-Nya, penulisan skripsi ini dapat penulis selesaikan. Salawat serta salam semoga tetap tercurahkan kepada Nabi Muhammad SAW hingga di akhir masa.

Penulisan skripsi ini tidak dapat penulis selesaikan tepat waktu tanpa dukungan morel dan materiel dari orang-orang terdekat penulis. Pada kata pengantar ini, penulis ingin menyampaikan rasa terima kasih kepada:

1. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
2. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
3. Bapak Agus Wahyu Widodo, S.T, M.Cs selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
4. Ibu Dian Eka Ratnawati, S.Si, M.Kom selaku Dosen Pembimbing I yang telah banyak membantu dan mengarahkan dalam proses penulisan skripsi ini.
5. Bapak Bayu Rahayudi, S.T, M.T selaku Dosen Pembimbing II yang telah meluangkan waktunya untuk memeriksa dan memberikan masukan berharga bagi penulisan skripsi ini.
6. Papa dan mama yang selalu memberikan dukungan morel serta materiel bagi penulis.
7. Kak Irfan yang selalu memberikan dukungan morel, nasihat, saran dan semangat bagi penulis.
8. Kiki yang selalu menemani dan memberikan semangat bagi penulis.
9. Dan teman-teman penulis yang tidak dapat penulis sebutkan satu-persatu.

Penulis menyadari bahwa dalam proses penulisan skripsi ini masih terdapat kesalahan dan kekurangan di beberapa bagian. Karena itu kritikan dan saran yang membangun sangat penulis harapkan demi perbaikan dan pengembangan skripsi ini di masa mendatang. Semoga skripsi ini bermanfaat bagi siapapun yang membaca dan menerapkannya dalam kehidupan sehari-hari.

Malang, 26 Desember 2018

Penulis  
nandaalifiya2@gmail.com

## ABSTRAK

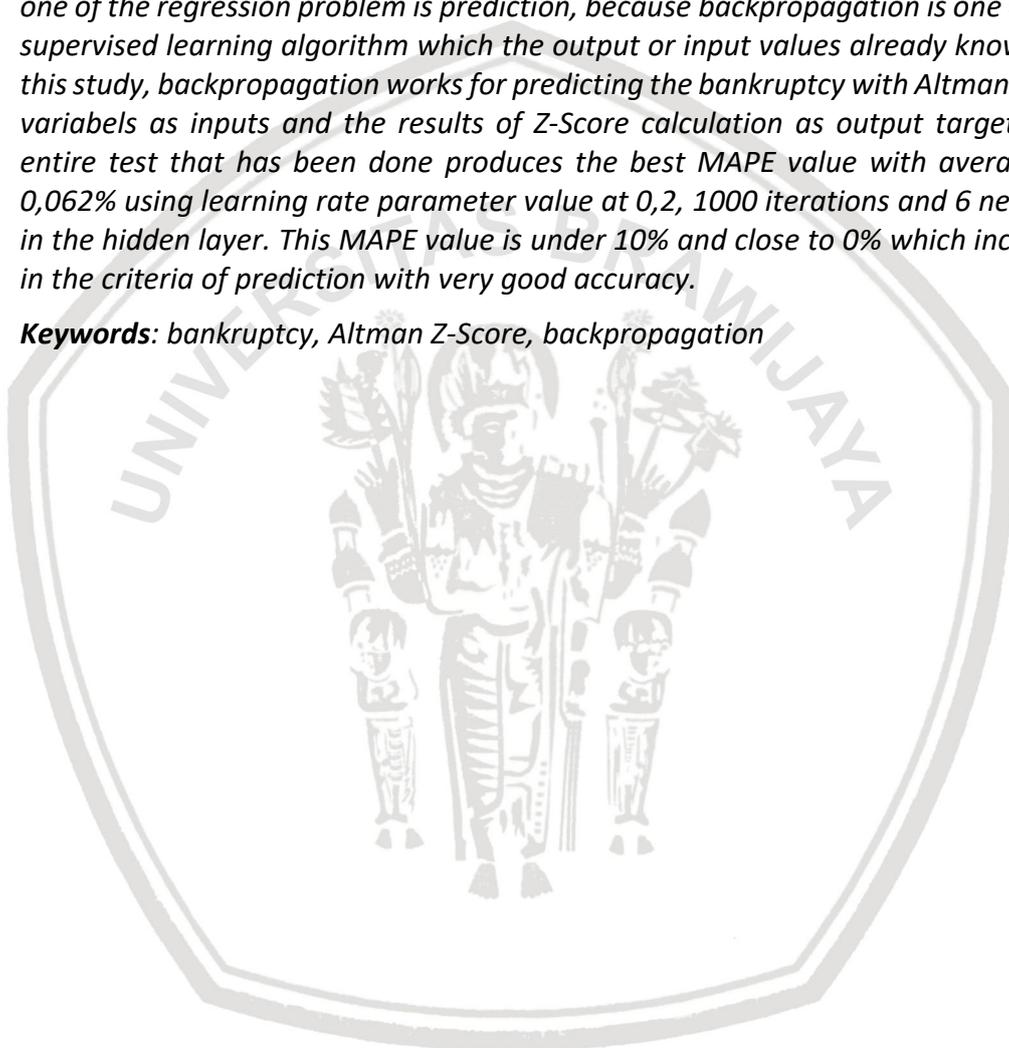
Kebangkrutan merupakan kondisi dimana sebuah perusahaan mengalami kegagalan baik itu kegagalan ekonomi atau kegagalan keuangan. Kebangkrutan menyebabkan adanya sita umum atas semua kekayaan Debitor Pailit (perusahaan) yang pengurusan dan pemberesannya dilakukan oleh Kurator (pengawas aktiva benda debitor). Karena dapat menimbulkan akibat yang berat, beberapa upaya dilakukan sebagai alternatif penanggulangan kebangkrutan. Salah satu upaya tersebut adalah upaya preventif dengan memprediksi kebangkrutan. *Backpropagation* adalah salah satu metode jaringan saraf tiruan yang banyak digunakan dalam konteks permasalahan dengan *dataset* yang bersifat klasifikasi dan regresi yang salah satunya adalah prediksi karena *backpropagation* merupakan salah satu algoritme *supervised learning* atau algoritme yang sudah diketahui nilai *output* dan inputnya. Dalam penelitian ini, *backpropagation* bertugas untuk memprediksi kebangkrutan dengan input berupa lima variabel *Altman* dan hasil dari perhitungan *Z-Score* menjadi target *output*. Keseluruhan pengujian yang telah dilakukan menghasilkan nilai MAPE terbaik dengan rata-rata 0,062% dengan parameter nilai *learning rate* sebesar 0,2, jumlah iterasi sebanyak 1000 dan jumlah neuron pada *hidden layer* sebanyak 6. Nilai MAPE tersebut <10% serta mendekati 0% yang termasuk dalam kriteria prediksi dengan akurasi sangat baik.

**Kata kunci:** kebangkrutan, *Altman Z-Score*, *backpropagation*

## ABSTRACT

*Bankruptcy is a condition while a company fails either economic failure or even financial failure. Bankruptcy causes a general seizure of all the assets of a bankrupt Debitor (company) that settled and managed by the Curator (supervisor of Debitor's asset). Because it can causes a severe consequences, several attempts were done as an alternative for bankruptcy prevention. One of those attempts is by predicting the bankruptcy itself. Backpropagation is a method of artificial neural network that widely used in the context of classification or regression datasets, one of the regression problem is prediction, because backpropagation is one of the supervised learning algorithm which the output or input values already known. In this study, backpropagation works for predicting the bankruptcy with Altman's five variabels as inputs and the results of Z-Score calculation as output target. The entire test that has been done produces the best MAPE value with average at 0,062% using learning rate parameter value at 0,2, 1000 iterations and 6 neurons in the hidden layer. This MAPE value is under 10% and close to 0% which included in the criteria of prediction with very good accuracy.*

**Keywords:** *bankruptcy, Altman Z-Score, backpropagation*



## DAFTAR ISI

PENGESAHAN .....	1
PERNYATAAN ORISINALITAS .....	2
KATA PENGANTAR.....	3
ABSTRAK.....	4
ABSTRACT.....	5
DAFTAR ISI.....	6
DAFTAR TABEL.....	9
DAFTAR GAMBAR.....	10
BAB 1 PENDAHULUAN .....	11
1.1 Latar Belakang .....	11
1.2 Rumusan Masalah .....	14
1.3 Tujuan.....	14
1.4 Manfaat .....	14
1.4.1 Bagi Penulis .....	14
1.4.2 Bagi Pembaca .....	14
1.5 Batasan Masalah .....	14
1.6 Sistematika Pembahasan .....	15
BAB 2 LANDASAN KEPUSTAKAAN.....	16
2.1 Kajian Pustaka .....	16
2.2 Kebangkrutan .....	18
2.2.1 Definisi Kebangkrutan .....	18
2.2.2 Manfaat Informasi Kebangkrutan .....	18
2.3 <i>Altman Z-Score</i> .....	19
2.4 <i>Backpropagation</i> .....	21
2.4.1 Fase Pelatihan <i>Backpropagation</i> .....	22
2.4.2 Algoritme Pelatihan <i>Backpropagation</i> .....	22
2.5 <i>Mean Absolute Percentage Error</i> .....	24
2.6 <i>K-Fold Cross Validation</i> .....	24
BAB 3 METODOLOGI .....	25
3.1 Tipe Penelitian.....	25
3.2 Strategi dan Rancangan Penelitian .....	25
3.2.1 Metode Penelitian.....	25
3.2.2 Obyek Penelitian .....	26
3.2.3 Lokasi Penelitian.....	26
3.2.4 Pengumpulan Data.....	26
3.2.5 Analisis Data .....	29
3.2.6 Peralatan Pendukung Penelitian .....	29

BAB 4	PERANCANGAN .....	30
4.1	Formulasi Permasalahan .....	30
4.2	Siklus Penyelesaian Masalah .....	31
4.2.1	Inisialisasi Bobot .....	32
4.2.2	Fase Propagasi Maju pada Data <i>Training</i> .....	33
4.2.3	Fase Propagasi Mundur pada Data <i>Training</i> .....	33
4.2.4	Fase Perubahan Bobot pada Data <i>Training</i> .....	35
4.2.5	Fase <i>Testing</i> dan Menghitung MAPE Data <i>Testing</i> .....	35
4.3	Perhitungan Manual .....	36
4.3.1	Inisialisasi Bobot .....	37
4.3.2	Fase Propagasi Maju pada Data <i>Training</i> .....	38
4.3.3	Fase Propagasi Mundur pada Data <i>Training</i> .....	39
4.3.4	Fase Perubahan Bobot pada Data <i>Training</i> .....	40
4.3.5	Fase Propagasi Maju dan Menghitung MAPE Data <i>Testing</i> .....	41
4.4	Perancangan Antarmuka .....	42
4.4.1	Halaman utama .....	42
4.1.1	Halaman Kedua .....	43
4.2	Perancangan Pengujian .....	43
4.2.1	Pengujian Nilai <i>Learning Rate</i> ( $\alpha$ ) .....	44
4.2.2	Pengujian Jumlah Iterasi .....	44
4.2.3	Pengujian Jumlah Neuron pada <i>Hidden Layer</i> .....	45
4.1.1	Pengujian <i>Cross Validation</i> .....	45
BAB 5	IMPLEMENTASI .....	47
5.1	Implementasi Algoritme <i>Backpropagation</i> .....	47
5.1.1	Implementasi Proses Inisialisasi Bobot .....	47
5.1.2	Implementasi Fase Propagasi Maju .....	48
5.1.3	Implementasi Fase Propagasi Mundur .....	49
5.1.4	Implementasi Fase Perubahan Bobot .....	50
5.1.5	Implementasi Proses Menghitung MAPE .....	51
5.2	Implementasi Antarmuka .....	52
5.2.1	Halaman Utama .....	52
5.2.2	Halaman Kedua .....	52
BAB 6	PENGUJIAN DAN ANALISIS .....	54
6.1	Pengujian Nilai <i>Learning Rate</i> .....	54
6.2	Pengujian Jumlah Iterasi .....	55
6.3	Pengujian Jumlah Neuron pada <i>Hidden Layer</i> .....	56
6.4	<i>K-fold Cross Validation</i> .....	57
6.5	Analisis Pengujian .....	59
BAB 7	KESIMPULAN .....	60
7.1	Kesimpulan .....	60

7.2 Saran.....60  
DAFTAR PUSTAKA.....61  
LAMPIRAN 1 .....63  
    Data Olah Laporan Keuangan (*Data Training*).....63  
LAMPIRAN 2 .....66  
    Data Olah Laporan Keuangan (*Data Testing*) .....66



## DAFTAR TABEL

Tabel 2.1 Kajian Pustaka .....	16
Tabel 2.2 Kriteria prediksi .....	24
Tabel 3.1 Fitur data .....	27
Tabel 3.2 Contoh data yang sudah didapatkan .....	28
Tabel 3.3 Contoh data yang sudah diolah.....	29
Tabel 4.1 Contoh data <i>training</i> .....	36
Tabel 4.2 Contoh data <i>testing</i> .....	37
Tabel 4.3 Contoh inisialiasi bobot.....	38
Tabel 4.4 Nilai keluaran di <i>hidden layer</i> ( $z_{in}$ ) .....	38
Tabel 4.5 Fungsi linier pada nilai keluaran di <i>hidden layer</i> $fz_{in}$ .....	38
Tabel 4.6 Nilai perubahan bobot $Wk0$ dan $Wkj$ .....	39
Tabel 4.7 Nilai faktor $\delta$ <i>hidden layer</i> .....	39
Tabel 4.8 Nilai koreksi galat $\delta_j$ .....	39
Tabel 4.9 Nilai perubahan bobot $Vij$ dan $Vj0$ .....	40
Tabel 4.10 Nilai $wbaru$ .....	40
Tabel 4.11 Nilai $vbaru$ .....	40
Tabel 4.12 Bobot terakhir <i>update</i> pada iterasi terakhir .....	41
Tabel 4.13 Nilai keluaran di <i>hidden layer</i> ( $z_{in}$ ) .....	41
Tabel 4.14 Fungsi linier pada nilai keluaran di <i>hidden layer</i> $fz_{in}$ .....	41
Tabel 4.15 Representasi hasil pengujian nilai learning rate .....	44
Tabel 4.16 Representasi hasil pengujian jumlah iterasi .....	44
Tabel 4.17 Representasi hasil pengujian jumlah neuron pada hidden layer.....	45
Tabel 4.18 Representasi hasil pengujian 5-fold cross validation.....	45
Tabel 4.19 Representasi hasil pengujian 10-fold cross validation .....	46
Tabel 6.1 Hasil pengujian nilai learning rate.....	54
Tabel 6.2 Hasil pengujian jumlah iterasi.....	55
Tabel 6.3 Hasil pengujian jumlah neuron pada hidden layer .....	56
Tabel 6.4 Hasil pengujian 5-fold cross validation .....	57
Tabel 6.5 Hasil pengujian 10-fold cross validation .....	58

## DAFTAR GAMBAR

Gambar 2.1	Arsitektur sederhana backpropagation satu hidden layer .....	21
Gambar 3.1	Flowchart penyelesaian permasalahan menggunakan metode backpropagation .....	25
Gambar 4.1	Arsitektur jaringan backpropagation dengan lima input dengan enam neuron pada hidden layer .....	30
Gambar 4.2	Flowchart siklus penyelesaian masalah .....	31
Gambar 4.3	Flowchart proses inisialisasi bobot .....	32
Gambar 4.4	Flowchart proses propagasi maju pada data training .....	33
Gambar 4.5	Flowchart proses propagasi mundur pada training .....	34
Gambar 4.6	Flowchart proses perubahan bobot pada data training .....	35
Gambar 4.7	Flowchart proses testing dan hitung MAPE data testing .....	36
Gambar 4.8	Ilustrasi antarmuka halaman utama .....	42
Gambar 4.9	Ilustrasi antarmuka halaman kedua .....	43
Gambar 5.1	Antarmuka halaman utama .....	53
Gambar 5.2	Antarmuka halaman kedua .....	53
Gambar 6.1	Grafik pengaruh <i>learning rate</i> terhadap nilai MAPE .....	54
Gambar 6.2	Grafik pengaruh jumlah iterasi terhadap nilai MAPE .....	55
Gambar 6.3	Grafik pengaruh jumlah neuron pada <i>hidden layer</i> terhadap nilai MAPE .....	56
Gambar 6.4	Grafik pengujian <i>5-fold cross validation</i> terhadap nilai MAPE .....	57
Gambar 6.5	Grafik pengujian <i>10-fold cross validation</i> terhadap nilai MAPE .....	58
Gambar 6.6	Grafik perbandingan rata-rata MAPE keseluruhan pengujian <i>5-fold</i> dan <i>10-fold cross validation</i> .....	59

## BAB 1 PENDAHULUAN

### 1.1 Latar Belakang

Perusahaan merupakan istilah perekonomian yang dikenal dalam Kitab Undang-Undang Hukum Perdagangan (KUHD) yang memiliki arti sebuah bentuk badan usaha yang bergerak dalam bidang perekonomian yang dijalankan secara terus-menerus dan bertujuan untuk mencari keuntungan yang juga menjalankan setiap jenis usaha, baik itu suatu persekutuan atau badan usaha lain yang didirikan, bekerja dan berkedudukan di Indonesia (Asyhadie & Sutrisno, 2012). Menurut Wasis (1997), Perusahaan merupakan sebuah bentuk organisasi yang bertujuan untuk mencari laba dengan memproduksi barang atau menyediakan jasa untuk keperluan masyarakat, bukanlah sebuah perusahaan kalau organisasi tersebut tidak mempunyai tujuan untuk menghasilkan laba. Untuk bisa dikatakan sebagai perusahaan, sebuah organisasi harus memenuhi beberapa syarat tertentu diantaranya: 1) harus bertujuan untuk mencari laba, jika tidak memiliki tujuan itu maka organisasi tersebut tentu saja bukanlah sebuah perusahaan; 2) tujuan mencari laba tersebut harus bersifat terus-menerus, bukan hanya bersifat insidental saja; 3) dalam mencari laba secara terus-menerus tersebut harus dijalankan dengan memproduksi barang atau menyediakan jasa; 4) memiliki tempat yang jelas, secara geografis maupun secara hukum yang berlaku.

Perusahaan memiliki tiga jenis yang berbeda yakni: 1) perusahaan perseorangan yang badan usaha kepemilikannya dimiliki oleh satu orang; 2) perusahaan persekutuan badan hukum yang berbentuk perseroan terbatas (PT), koperasi atau Badan Usaha Milik Negara (BUMN); 3) perusahaan persekutuan bukan badan hukum contohnya persekutuan firma atau persekutuan komanditer (CV). Dari ketiga jenis perusahaan tersebut, perusahaan persekutuan badan hukum berbentuk perseroan terbatas memiliki pengaruh penting terhadap perekonomian di Indonesia (Manan, 1995). Pentingnya perseroan terbatas terhadap perekonomian di Indonesia dikarenakan adanya pengerahan dana masyarakat untuk pengembangan perusahaan melalui kepemilikan saham perseroan. Selain itu perseroan terbatas sendiri berada di bawah kontrol masyarakat melalui pemegang saham dan mekanisme pasar modal, sehingga masyarakat dapat ikut serta dalam kegiatan ekonomi yang menguntungkan.

Menurut Pasal 1 angka 1 UU No. 40 tahun 2007 tentang perseroan terbatas, yang dimaksud dengan perseroan terbatas adalah “sebuah badan hukum yang terdiri dari persekutuan modal dan didirikan berdasarkan perjanjian, melakukan kegiatan usaha dengan modal dasar yang seluruhnya terbagi dalam saham serta memenuhi persyaratan yang ditetapkan dalam undang-undang”. Lalu istilah “terbatas” dalam perseroan terbatas menuju pada suatu tanggung jawab para pemegang saham yang terbatas hanya pada nilai nominal dari semua saham yang dimilikinya. Dalam tujuannya untuk mencari laba secara terus-menerus dengan cara memproduksi barang atau menyediakan jasa, sebuah perseroan terbatas dihadapkan pada resiko kebangkrutan. Kebangkrutan merupakan kondisi dimana sebuah perusahaan mengalami kegagalan, baik itu kegagalan ekonomi atau kegagalan keuangan (Weston & Copeland, 1991).

Kegagalan ekonomi perusahaan diartikan sebagai tidak seimbangnya pendapatan dengan pengeluaran, yang berarti laba yang dihasilkan perusahaan tidak dapat menutupi kewajiban perusahaan karena jumlah laba yang dihasilkan ternyata tidak sebesar yang diharapkan. Selanjutnya, kegagalan keuangan diartikan pada kondisi total kewajiban perusahaan melebihi nilai wajar dari total aktivasnya, atau dapat juga perusahaan tersebut tidak dapat membayar kewajibannya pada masa jatuh tempo meski total aktivasnya melebihi kewajibannya. Kegagalan tersebut disebabkan oleh ketidakmampuan manajemen perusahaan dalam mengelola arus keuangan perusahaan (Weston & Copeland, 1991).

Menurut UU No. 37 tahun 2004 tentang kepailitan dan penundaan kewajiban pembayaran utang, kebangkrutan mengakibatkan adanya "sita umum atas semua kekayaan Debitor Pailit (perusahaan) yang pengurusan dan pemberesannya dilakukan oleh Kurator (pengawas aktiva benda debitor) di bawah pengawasan Hakim Pengawas sebagaimana diatur dalam undang-undang kepailitan". Kebangkrutan sebuah perseroan terbatas dapat mengakibatkan peningkatan jumlah pengangguran jika perseroan terbatas tersebut sampai diputuskan untuk dipailitkan dan ditutup sehingga setelahnya dapat mempengaruhi laju perekonomian di Indonesia.

Karena dapat menimbulkan beberapa akibat yang cukup berat, beberapa upaya dilakukan sebagai alternatif penanggulangan kebangkrutan diantaranya melakukan merger dengan perusahaan lain, upaya di luar pengadilan dengan terus beroperasi maupun menutup perusahaan serta upaya dalam prosedur legal dan formal sesuai aturan undang-undang yang berlaku. Namun upaya-upaya tersebut dilakukan sebagai tindakan represif ketika kebangkrutan sudah terjadi, bukan tindakan preventif sebagai upaya antisipasi. Memprediksi kebangkrutan memungkinkan sebuah tindakan preventif untuk mengantisipasi terjadinya kebangkrutan itu sendiri.

Prediksi merupakan suatu proses memperkirakan secara sistematis tentang suatu kemungkinan di masa mendatang, berdasarkan data yang direkam pada masa sekarang atau masa lalu. Dalam pelaksanaannya, prediksi memang tidak dapat memberikan hasil secara pasti dan tepat, melainkan untuk berusaha mengetahui nilai sedekat mungkin dengan yang akan terjadi dan dengan sekecil mungkin kesalahan (Hanke, Reitsch, & Wichern, 1999). Prediksi dapat dilakukan dengan banyak metode, salah satunya adalah pengembangan teknik kecerdasan buatan, dalam hal ini yang paling banyak digunakan adalah jaringan saraf tiruan (Lestari, 2017).

Jaringan saraf tiruan merepresentasikan bagaimana otak manusia berfungsi, sedangkan pemodelan datanya yakni dengan melatih hubungan antara input dan *output* secara berulang-ulang yang cukup mudah digunakan untuk menyelesaikan sebuah masalah (Hamid, Nawi, Ghazali, & Salleh, 2011). Jaringan saraf tiruan memiliki keunggulan dengan tidak memerlukan asumsi apapun mengenai distribusi populasi dan juga tidak memerlukan penjelasan tentang hubungan antar variabel terlebih dahulu karena metodenya yang melibatkan pembelajaran dalam hubungan antar variabel itu sendiri (Hanke, Reitsch, & Wichern, 1999). Beberapa

metode jaringan saraf tiruan yang biasa digunakan untuk prediksi adalah *backpropagation*, *Extreme Learning Machine*, *Perceptron*, dan masih banyak yang lainnya. Dari beberapa metode jaringan saraf tiruan tersebut, *backpropagation* adalah metode pelatihan yang paling banyak digunakan dalam beberapa konteks permasalahan dengan *dataset* yang bersifat regresi dan klasifikasi yang sudah diketahui nilai *output* dan inputnya. Hal tersebut dikarenakan metode *backpropagation* yang termasuk dalam jaringan saraf tiruan adalah satu dari beberapa algoritme *supervised learning* atau algoritme yang sudah diketahui nilai *output* dan inputnya.

Metode *backpropagation* memiliki arsitektur dengan tiga *layer* utama yakni input *layer*, *hidden layer* dan *output layer* (Fausett, 1994). Dalam memprediksi kebangkrutan dengan metode *backpropagation* dibutuhkan fitur data yang menjadi input jaringan pada input *layer*, fitur data ini dapat diperoleh dari laporan keuangan tahunan sebuah perusahaan. Namun data tersebut tidak dapat secara langsung digunakan karena dalam laporan keuangan setiap perseroan terbatas hanya disajikan dalam satuan keuangan dengan angka yang cukup besar, bukan dalam angka kecil ataupun rasio. Oleh karena itu rasio keuangan *Altman Z-Score* digunakan untuk mempersiapkan fitur-fitur input tersebut.

Analisis rasio keuangan *Z-Score* dikemukakan oleh Edward I. Altman pada tahun 1968. Analisis ini dirumuskan dalam lima rasio keuangan antara lain rasio antara modal kerja dengan total aktiva, laba ditahan dengan total aktiva, pendapatan sebelum pajak dan bunga terhadap total aktiva, nilai pasar ekuitas terhadap nilai buku dari hutang, dan penjualan terhadap total aktiva (Altman, 1968). Penelitian yang dilakukan oleh Tambunan, Dwiatmanto, dan Endang (2015) dengan menggunakan lima rasio keuangan tersebut dalam memprediksi kebangkrutan pada subsektor rokok menunjukkan bahwa analisis keuangan *Altman* memberikan hasil prediksi yang baik dan sesuai dengan realita adanya kebangkrutan pada beberapa pabrik rokok di lapangan. Lima rasio keuangan tersebut dapat menjadi fitur input jaringan, selanjutnya keadaan bangkrut sebuah perusahaan dapat menjadi target pada metode *backpropagation*.

Berdasarkan penelitian yang dilakukan oleh Werdiastu, Ratnawati dan Rahayudi (2018), dalam estimasi hasil produksi benih kenaf menggunakan metode *backpropagation* didapatkan nilai MAPE sebesar 0,938%. Sedangkan penelitian yang dilakukan oleh Ridhani, Putri, dan Sutopo (2017), untuk meramalkan peramalan dosis pupuk terhadap tanaman jeruk siam metode *backpropagation* juga memiliki nilai MAPE yang baik, sebesar 9,178%. Nilai MAPE dari suatu proses prediksi bisa dikatakan baik apabila nilai tersebut mendekati 0% atau <10%. Selain itu, penelitian lain yang dilakukan oleh Wahyuni, Adam, Mahmudy, dan Iriyani (2017), metode *backpropagation* memberikan nilai RMSE 8.14 dalam memprediksi curah hujan di Tengger, Jawa Timur.

Mengacu pada hal-hal yang tersebut, metode *backpropagation* diharapkan dapat menyelesaikan permasalahan dalam memprediksi kebangkrutan perseroan terbatas dengan memberikan hasil prediksi dengan hasil akurasi yang lebih baik dibandingkan dengan metode prediksi konvensional atau manual. Selain itu, diharapkan juga penelitian ini dapat membantu perusahaan dalam memberikan

tindakan preventif sebagai upaya antisipasi sekaligus pencegahan adanya kegagalan dalam perusahaan yang mengakibatkan adanya kebangkrutan.

## 1.2 Rumusan Masalah

Masalah yang ditemukan penulis dijabarkan sebagai berikut:

1. Bagaimana mengolah laporan keuangan perusahaan menjadi fitur input dalam memprediksi kebangkrutan?
2. Bagaimana menerapkan *Altman Z-Score* dalam memprediksi kebangkrutan menggunakan metode *backpropagation*?
3. Bagaimana nilai MAPE yang dihasilkan metode *backpropagation* dalam memprediksi kebangkrutan?

## 1.3 Tujuan

Tujuan umum dari tulisan ini adalah menerapkan metode *Backpropagation* dalam memprediksi kebangkrutan perseroan terbatas, sedangkan tujuan khusus yang penulis jelaskan dalam tulisan ini ialah sebagai berikut:

1. Menjelaskan pengolahan laporan keuangan perusahaan sehingga menjadi fitur input dalam memprediksi kebangkrutan.
2. Menjabarkan penerapan *Altman Z-score* dalam memprediksi kebangkrutan menggunakan metode *backpropagation*.
3. Menjabarkan nilai MAPE dalam prediksi kebangkrutan menggunakan metode *backpropagation*.

## 1.4 Manfaat

Manfaat secara umum dari tulisan ini adalah untuk membantu mencegah adanya kebangkrutan pada perseroan terbatas dengan memprediksi adanya kebangkrutan tersebut. Adapun manfaat-manfaat selanjutnya dari penelitian ini antara lain:

### 1.4.1 Bagi Penulis

1. Menerapkan ilmu yang dimiliki penulis untuk kepentingan masyarakat umum terutama bagi perusahaan/badan usaha untuk menjadikan hasil prediksi dalam penelitian ini sebagai tolak ukur yang berkaitan dengan pengambilan keputusan pada sebuah perusahaan.
2. Memahami implementasi metode *backpropagation* dalam kehidupan sehari-hari.

### 1.4.2 Bagi Pembaca

1. Sebagai sumber informasi terkait prediksi kebangkrutan dengan metode *backpropagation*.

## 1.5 Batasan Masalah

Batasan-batasan permasalahan yang digunakan antara lain:

1. Data yang digunakan berasal dari laporan keuangan per tahun perseroan terbatas dari tahun 2014 sampai 2017 yang terdaftar pada Bursa Efek Indonesia.

2. Data yang didapatkan diolah terlebih dahulu menjadi lima rasio keuangan *Altman* dan ditentukan keadaan bangkrutnya dengan persamaan *Z-Score*.
3. Data *training* adalah data laporan keuangan dari tahun 2014 sampai 2016 yang masing-masing berjumlah 100 laporan, sedangkan data *testing* adalah data laporan keuangan tahun 2017 yang berjumlah 130 laporan.

## 1.6 Sistematika Pembahasan

Sistematika pembahasan penulisan penelitian ini antara lain:

### BAB I : PENDAHULUAN

Bagian pendahuluan menjelaskan latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan dalam penulisan dokumen skripsi. Dalam bab ini akan dijelaskan tujuan penulis menggunakan judul tersebut dalam penelitiannya, metode yang digunakan dan hipotesis awal penulis dalam menentukan penyelesaian masalah yang ada.

### BAB II : KAJIAN PUSTAKA

Bagian kajian pustaka berisi penjelasan mengenai teori-teori yang menjadi dasar argumentasi untuk seluruh proses penelitian. Pada bab ini akan dijelaskan pula referensi yang berkaitan dengan judul yang digunakan, yakni teori tentang pengertian kebangkrutan, *Altman Z-Score*, *backpropagation*, *Mean Absolute Percentage Error (MAPE)* dan *K-Fold Cross Validation*.

### BAB III : METODOLOGI

Bab ini berisi tentang mekanisme alur metode dalam pelaksanaan teknik penelitian. Setiap alur dari metode tersebut secara mendetail akan dijelaskan pada bab ini.

### BAB IV : HASIL

Bab ini berisi tentang hasil pelaksanaan teknik penelitian menggunakan metode *backpropagation*, selain itu pada bab ini juga disajikan data yang mendukung hasil pelaksanaan teknik penelitian.

### BAB V : PEMBAHASAN

Bab ini berisi tentang penjelasan dan penerjemahan hasil pelaksanaan teknik penelitian untuk menjawab rumusan masalah dalam penelitian. Dalam bab ini juga akan dibahas pemahaman baru yang dihasilkan dalam pelaksanaan teknik penelitian.

### BAB VI : PENUTUP

Bab ini berisi kesimpulan dari seluruh kegiatan penelitian yang dilakukan. Bagian ini juga menjelaskan saran untuk penelitian yang serupa lebih lanjut di kemudian hari.

## BAB 2 LANDASAN KEPUSTAKAAN

Bab landasan kepastakaan berisi tentang dasar teori yang berkaitan dengan judul penelitian yang diajukan. Selain itu, dalam bab ini dibahas pula penelitian sebelumnya yang masih memiliki korelasi dengan judul penelitian.

### 2.1 Kajian Pustaka

Penelitian yang akan dilaksanakan ini menggunakan beberapa penelitian terdahulu yang memiliki korelasi sebagai acuan. Sebagai lingkup korelasinya, penelitian terdahulu yang digunakan sebagai acuan adalah penelitian yang menggunakan metode *backpropagation* dalam kegiatan peramalan atau prediksi.

Berdasarkan penelitian yang dilakukan oleh Wahyuni, Adam, Mahmudy, dan Iriany, (2017) dalam memprediksi curah hujan di daerah Tengger, Jawa Timur dengan menggunakan *backpropagation* menunjukkan nilai *Root Mean Squared Error* (RMSE) 8,28. Nilai tersebut didapatkan dari data curah hujan selama 10 tahun antara tahun 2005 hingga tahun 2014, dengan empat parameter yang berupa rata-rata curah hujan dalam beberapa kurun waktu. Selanjutnya penelitian yang dilakukan oleh Ridhani, Putri dan Sutopo (2017), metode *backpropagation* digunakan untuk meramalkan dosis pupuk berdasarkan karakteristik dan lingkungan tanaman jeruk siam terhadap data hasil penjualan dan stok produk jeruk siam menunjukkan jika penelitian ini menghasilkan nilai MAPE terbaik sebesar 9,178%. Nilai MAPE tersebut didapatkan dari rata-rata galat nilai dosis pupuk nitrogen, fosfor dan kalium.

Selanjutnya, penelitian yang dilakukan oleh Werdiastu, Ratnawati dan Rahayudi (2018), metode *backpropagation* digunakan untuk mengestimasi hasil produksi benih tanaman kenaf berdasarkan karakteristiknya dengan studi kasus pada Balai Penelitian Tanaman Pemanis dan Serat (BALITTAS) Malang, menghasilkan nilai MAPE 0,938%. Nilai MAPE tersebut didapatkan dari rata-rata galat nilai galat dari umur bunga 1, diameter bawah, jumlah kapsul masak dan berat benih dari 10 tanaman. Yang terakhir yakni penelitian yang dilakukan oleh Malaka dan Hartojo (2014), dengan menggunakan metode *backpropagation* dalam memprediksi kepailitan bank umum di Indonesia didapatkan hasil akurasi model jaringan yang baik sebesar 86,11%. Akurasi tersebut didapatkan dari 76 data laporan keuangan bank umum di Indonesia. Penelitian terdahulu yang berkaitan dengan judul penelitian yang akan dilakukan disimpulkan pada Tabel 2.1:

**Tabel 2.1 Kajian Pustaka**

No	Judul	Obyek	Metode	Hasil
1.	Model Prediksi Kepailitan Bank Umum Di Indonesia Menggunakan Algoritma <i>Backpropagation</i> (Malaka & Hartojo, 2014)	Data keuangan tahunan bank umum tahun 1997.	<i>Backpropagation</i>	Penelitian ini menghasilkan akurasi model jaringan sebesar 86,11%.

Tabel 2.1 Kajian Pustaka (Lanjutan)

No	Judul	Obyek	Metode	Hasil
2.	Peramalan Dosis Pupuk Berdasarkan Karakteristik dan Lingkungan Tanaman Jeruk Siam Menggunakan Metode <i>Backpropagation</i> (Ridhani, Putri, & Sutopo, 2017)	Data hasil penjualan dan stok produk.	<i>Backpropagation</i>	Penelitian ini menghasilkan nilai MAPE terbaik sebesar 9,178%.
3.	Estimasi Hasil Produksi Benih Berdasarkan Karakteristik Tanaman Kenaf Menggunakan Metode <i>Backpropagation</i> (Studi Kasus: Balai Tanaman Pemanis dan Serat Kota Malang) (Werdiastu, Ratnawati, & Rahayudi, 2018)	Data karakteristik kenaf.	<i>Backpropagation</i>	Penelitian ini menghasilkan nilai MAPE terbaik sebesar 0,938%.
4.	<i>Modeling Backpropagation Neural Network for Rainfall Prediction in Tengger East Java</i> (Wahyuni, Adam, Mahmudy, & Iriany, 2017)	Data curah hujan pada daerah Tengger, Jawa Timur dari tahun 2005-2014.	<i>Backpropagation</i>	Penelitian ini menghasilkan nilai RMSE terbaik sebesar 8,28.
5.	Prediksi Kebangkrutan Menggunakan Metode <i>Backpropagation</i> (Studi Kasus: Emiten Terdaftar Pada Bursa Efek Indonesia) <b>(Usulan)</b>	Data laporan keuangan per tahun perseroan terbatas dari tahun 2014-2017 yang terdaftar pada Bursa Efek Indonesia.	<i>Backpropagation</i>	Diharapkan prediksi kebangkrutan menggunakan metode <i>Backpropagation</i> memberikan hasil yang baik.

## 2.2 Kebangkrutan

### 2.2.1 Definisi Kebangkrutan

Kebangkrutan atau kepailitan sering kali disangkutpautkan dengan kegagalan perusahaan. Kebangkrutan merupakan kondisi dimana sebuah perusahaan mengalami kegagalan, baik itu kegagalan ekonomi atau kegagalan keuangan (Weston & Copeland, 1991). Menurut Brigham dan Gapenski (1993) ada beberapa definisi tentang kebangkrutan:

#### 2.2.1.1 Kegagalan Ekonomi

Kegagalan ekonomi menandakan jika pendapatan perusahaan tidak dapat menutupi total biaya, termasuk biaya modal. Perusahaan yang mengalami kegagalan ekonomi masih dapat beroperasi selama investor masih bersedia memberikan tambahan modal. Akhirnya, meski tidak ada modal baru yang disediakan atau aset akan habis, perusahaan harus melakukan berbagai opsi untuk mengembalikan pendapatan sehingga bisa menutupi seluruh biaya modal yang dikeluarkan.

#### 2.2.1.2 Kegagalan Bisnis

Istilah kegagalan bisnis digunakan untuk menentukan kegiatan bisnis yang telah memberatkan operasi perusahaan dan menimbulkan kerugian yang diakibatkan oleh kreditor.

#### 2.2.1.3 Kebangkrutan Teknis

Perusahaan dianggap mengalami kebangkrutan secara teknis jika tidak dapat memenuhi kewajibannya karena telah jatuh tempo. Kebangkrutan teknis dapat menunjukkan kurangnya likuiditas sementara. Dalam waktu tertentu, perusahaan yang secara teknis bangkrut mungkin bisa mengumpulkan uang tunai, melunasi kewajibannya, dan bertahan. Di sisi lain, jika kebangkrutan teknis merupakan gejala awal kegagalan ekonomi, kemungkinan akan terjadinya kegagalan finansial menjadi lebih besar.

#### 2.2.1.4 Kebangkrutan dalam Kebangkrutan

Perusahaan dapat disebut mengalami kondisi kebangkrutan dalam kebangkrutan ketika nilai buku dari total kewajibannya melebihi nilai pasar sebenarnya dari asetnya. Ini adalah kondisi yang lebih serius daripada kebangkrutan teknis karena menjadi pertanda kegagalan ekonomi, dan menyebabkan adanya likuidasi bisnis.

#### 2.2.1.5 Kebangkrutan Hukum

Meskipun penggunaan istilah kebangkrutan merujuk pada perusahaan yang "gagal", perusahaan tidak bangkrut secara hukum kecuali telah mengajukan kebangkrutan berdasarkan undang-undang yang berlaku.

### 2.2.2 Manfaat Informasi Kebangkrutan

Menurut Hanafi (2004), informasi kebangkrutan bisa bermanfaat bagi beberapa pihak sebagai berikut ini:

### 2.2.2.1 Pemberi Pinjaman (seperti Pihak Bank)

Informasi kebangkrutan bermanfaat sebagai informasi awal pengambilan keputusan untuk memberi pinjaman terhadap peminjam dan bermanfaat juga untuk kebijakan monitor pinjaman yang ada.

### 2.2.2.2 Investor

Investor saham suatu perusahaan tentunya akan sangat berkepentingan melihat adanya kemungkinan bangkrut atau tidaknya perusahaan yang menjual surat berharga tersebut. Investor yang menganut strategi aktif akan mengembangkan model prediksi kebangkrutan untuk melihat tanda-tanda kebangkrutan seawal mungkin dan kemudian mengantisipasi kemungkinan tersebut.

### 2.2.2.3 Pihak Pemerintah

Pada beberapa sektor usaha, lembaga pemerintah mempunyai tanggung jawab untuk mengawasi jalannya usaha tersebut. Juga pemerintah mempunyai badan-badan usaha (BUMN) yang harus selalu diawasi. Lembaga pemerintah mempunyai kepentingan untuk melihat tanda-tanda kebangkrutan lebih awal supaya tindakan-tindakan yang perlu bisa dilakukan lebih awal.

### 2.2.2.4 Akuntan

Akuntan mempunyai kepentingan terhadap informasi kelangsungan suatu usaha karena akuntan akan menilai kemampuan *going concern* suatu perusahaan.

### 2.2.2.5 Manajemen

Kebangkrutan berarti munculnya biaya-biaya yang berkaitan dengan kebangkrutan dan biaya ini cukup besar. Apabila manajemen bisa mendeteksi kebangkrutan ini lebih awal, maka tindakan-tindakan penghematan bisa dilakukan, misalnya dengan melakukan merger atau restrukturisasi keuangan sehingga biaya kebangkrutan bisa dihindari.

## 2.3 Altman Z-Score

*Altman Z-Score* adalah salah satu dari beberapa teknik statistik yang digunakan untuk memprediksi kebangkrutan perusahaan. Metode ini dikembangkan oleh Edward I. Altman pada tahun 1968 yang menggunakan lima rasio keuangan. Hanafi (2004) menjelaskan bahwa pada tahun 1983 dan 1984, model prediksi kebangkrutan dikembangkan kembali oleh Altman dan dari pengembangan tersebut ditemukan nilai Z yang baru yang menghasilkan metode *Altman Z-Score* baru dengan tingkat kevalidan hingga 95%. Metode *Altman* ditunjukkan dengan Persamaan 2.1:

$$Z = 1,2X_1 + 1,4X_2 + 3,3X_3 + 0,6X_4 + 1,0X_5 \quad (2.1)$$

Keterangan:

$$X_1 = \frac{\text{modal kerja}}{\text{total aktiva}}$$

$$X_2 = \frac{\text{laba ditahan}}{\text{total aktiva}}$$

$$X3 = \frac{\text{pendapatan sebelum pajak dan bunga}}{\text{total aktiva}}$$

$$X4 = \frac{\text{nilai pasar ekuitas}}{\text{nilai buku dari hutang}}$$

$$X5 = \frac{\text{penjualan}}{\text{total aktiva}}$$

Kriteria dalam prediksi kebangkrutan rumus *Z-Score* ini diklasifikasikan menjadi 3 skor. Untuk nilai *Z-Score* >2,99 diklasifikasikan sebagai perusahaan yang sehat. Jika nilai *Z-Score* yang dihasilkan berada di antara 1,81 hingga 2,99 maka perusahaan dinyatakan di dalam dalam *grey area* atau daerah rawan dan memiliki sedikit potensi bangkrut. Sedangkan jika nilai *Z-Score* < 1,81 maka perusahaan tersebut memiliki potensi bangkrut yang besar. Lima rasio dalam model ini saling berhubungan dan dapat dijelaskan sebagai berikut (Altman, 1968):

**a. X1 (Modal Kerja Terhadap Total aset)**

Rasio ini membandingkan antara modal kerja (*working capital*) dengan total aset perusahaan (*total assets*). Working capital merupakan selisih antara aktiva lancar (*current assets*) dengan hutang lancar (*current liabilities*). Rasio ini memberikan keterangan bahwa likuid tidaknya suatu perusahaan tergantung kepada modal kerja perusahaan tersebut. Kesulitan perusahaan dapat tercermin jika dalam perusahaan terjadi penurunan modal kerja. Modal kerja dalam sebuah perusahaan dapat bernilai negatif karena hutang perusahaan yang lebih besar dari total aset perusahaan.

**b. X2 (Laba Ditahan Terhadap Total Aset)**

Rasio ini digunakan untuk mengetahui perbandingan dari laba ditahan (*retained earning*) dengan total aset perusahaan (*total assets*). Laba perusahaan menunjukkan bahwa perusahaan tersebut adalah perusahaan yang berhasil. Jika semakin positif nilai dari laba perusahaan yang ditahan maka perusahaan dianggap mampu menutupi kekurangan finansial di masa mendatang. Namun jika nilai laba perusahaan ini semakin menurun dan negatif dapat memberikan peringatan terhadap perusahaan untuk berhenti beroperasi.

**c. X3 (Laba Sebelum Pajak dan Bunga Terhadap Total Aset)**

Rasio ini membandingkan antara pendapatan atau laba sebelum pajak dan bunga (*earning before interest and taxes*) terhadap total aset perusahaan (*total assets*). Rasio ini mencerminkan penggunaan aset perusahaan untuk menghasilkan laba bruto perusahaan. Hal ini cukup relevan dengan prediksi kebangkrutan karena kebangkrutan terjadi jika total hutang atau kewajiban melebihi total nilai dari aset perusahaan dengan nilai yang ditentukan dari kemampuan aset dalam menghasilkan laba.

**d. X4 (Nilai Pasar Ekuitas Terhadap Nilai Buku dari Total Hutang)**

Rasio yang membandingkan antara nilai pasar ekuitas (*market value equity*) dengan nilai buku dari total hutang (*book value of total debt*) ini mengukur seberapa banyak aset perusahaan dapat menurun nilainya sebelum kewajiban (hutang) melebihi aset dan perusahaan menjadi bangkrut. Nilai pasar ekuitas

didapat dari nilai pasar saham dengan total saham beredar. Sedangkan nilai buku dari total hutang adalah nilai yang didapatkan dari hasil hutang jangka pendek ditambah hutang jangka panjang.

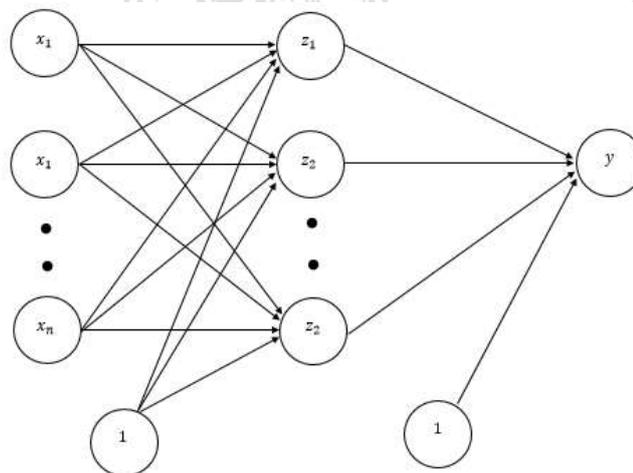
**e. X5 (Penjualan Terhadap Total Aset)**

Rasio ini merupakan rasio yang mengukur besarnya pengaruh dari aset perusahaan untuk menunjang proses penjualan barang perusahaan. Hal ini juga dapat digunakan untuk mengukur tingkat kemampuan dari manajemen dalam menghadapi persaingan usaha. Rasio ini dirumuskan dengan penjualan (*sales*) dibagi dengan total aset (*total assets*).

**2.4 Backpropagation**

Metode *backpropagation* merupakan salah satu algoritme *supervised learning* yang menerapkan sistematika pelatihan *multilayer* yang merupakan metode pembelajaran/pelatihan yang bersifat terkontrol dan menggunakan aturan pengoreksian galat (Fausett, 1994). Algoritme *supervised learning* sendiri merupakan pendekatan algoritme dimana terdapat data yang sudah dilatih dan variabel yang ditargetkan, tetapi algoritme tersebut melakukan pengulangan prediksi pada data latih tersebut dan berhenti ketika sudah memenuhi *stopping condition*. Data-data yang sesuai untuk digunakan pada algoritme *supervised learning* adalah data yang meliputi kriteria regresi (data dengan variabel *output* bernilai real, seperti jumlah produksi) dan klasifikasi (data dengan variabel *output* yang bernilai kategori, seperti “jeruk” atau “mangga”). Dengan demikian, diharapkan hasil pelatihan metode *backpropagation* ini cukup baik untuk keperluan prediksi kebangkrutan pada penelitian ini karena variabel *output* memiliki kesesuaian dengan kriteria tersebut.

Proses pelatihan metode *backpropagation* tergolong metode yang sederhana yaitu jika hasil keluaran tidak sesuai atau salah, maka bobot dikoreksi dan memperkecil nilai galat dengan anggapan mendekati nilai yang benar. *Backpropagation* memiliki beberapa *layer* yakni: 1) *Input Layer*, 2) *Hidden Layer* dan 3) *Output Layer* yang arsitektur sederhananya diilustrasikan pada Gambar 2.1.



**Gambar 2.1** Arsitektur sederhana *backpropagation* satu *hidden layer*



### 2.4.1 Fase Pelatihan *Backpropagation*

Pelatihan *backpropagation* meliputi 3 (tiga) fase, yakni:

#### 2.4.1.1 Fase propagasi maju

Dalam fase ini pola input dihitung maju mulai input *layer* hingga *output layer* menggunakan fungsi aktivasi yang ditentukan. Fungsi aktivasi adalah fungsi yang digunakan untuk mengaktifkan atau tidak mengaktifkan neuron. Fungsi aktivasi seharusnya memiliki karakteristik yang kontinu, terdiferensiasi dan tidak menurun secara monoton (Fausett, 1994). Dalam penelitian ini digunakan fungsi aktivasi linier, fungsi aktivasi ini memiliki nilai *output* yang sama dengan nilai inputnya. Persamaan 2.2 merupakan fungsi aktivasi linier dalam sifat ekuatif dan Persamaan 2.3 merupakan fungsi aktivasi linier dalam sifat turunan. Fungsi aktivasi linier digunakan karena target jaringan bersifat bilangan desimal, bukan bilangan biner dan tidak memiliki nilai minus.

$$f(x) = x \quad (2.2)$$

$$f'(x) = 1 \quad (2.3)$$

#### 2.4.1.2 Fase propagasi mundur

Dalam fase ini, selisih antara *output* jaringan dengan target yang diinginkan merupakan galat yang terjadi. Galat tersebut dipropagasikan mundur mulai dari garis yang berhubungan langsung dengan unit-unit di *output layer*.

#### 2.4.1.3 Fase perubahan bobot

Fase ini berfungsi untuk memodifikasi bobot sehingga menurunkan galat yang terjadi. Bobot merupakan nilai yang mendefinisikan tingkat hubungan antara satu node dengan node yang lain. Semakin besar bobot suatu hubungan maka dapat dikatakan hubungan antar node tersebut lebih penting.

### 2.4.2 Algoritme Pelatihan *Backpropagation*

Algoritme pelatihan *backpropagation* mencakup tiga fase utama yakni fase propagasi maju, propagasi mundur dan perubahan bobot, menurut Fausett (1994) Algoritme *backpropagation* dibahas lebih jelas sebagai berikut:

#### 2.4.2 Fase 1: propagasi maju

Algoritme pelatihan *backpropagation* dengan satu *hidden layer* adalah sebagai berikut:

**Langkah 0:** Inisialisasi semua bobot dengan bilangan *random* kecil.

**Langkah 1:** Jika *stopping condition* belum terpenuhi, lakukan langkah 2-9.

**Langkah 2:** Untuk setiap pasang data *training*, lakukan langkah 3-8.

**Langkah 3:** Tiap input *unit*  $x_i (i = 1, 2, \dots, n)$  menerima sinyal dan meneruskannya ke setiap unit dalam *hidden layer*.

**Langkah 4:** Hitung semua keluaran di *hidden layer*  $z_j = (j = 1, 2, \dots, p)$  dengan Persamaan 2.4 dan menghitung fungsi linier keluarannya dengan Persamaan 2.5.

$$z\_in_j = v_{j0} + \sum_{i=1}^n x_i v_{ji} \quad (2.4)$$

$$z_j = f(z\_in_j) = z\_in \quad (2.5)$$

**Langkah 5:** Hitung semua keluaran jaringan di *output unit*  $y_k$  ( $k = 1, 2, \dots, m$ ) dengan Persamaan 2.6 dan dan menghitung fungsi linier keluarannya dengan Persamaan 2.7.

$$y\_in_k = w_{j0} + \sum_{j=1}^p z_j w_{kj} \quad (2.6)$$

$$y_k = f(y\_in_k) = y\_in \quad (2.7)$$

### 2.4.3 Fase 2: propagasi mundur

**Langkah 6:** Hitung faktor  $\delta$  *output layer* berdasarkan galat di setiap *output unit*  $y_k$  dengan Persamaan 2.8.

$$\delta_k = (t_k - y_k) f'(y\_in_k) \quad (2.8)$$

$\delta_k$  merupakan unit galat yang dipakai dalam perubahan bobot pada *layer* berikutnya (langkah 7), lalu menghitung perubahan bobot  $W_{kj}$  dengan *learning rate*  $\alpha$  menggunakan Pesamaan 2.9.

$$\Delta w_k = \alpha \delta_k z_j \quad (2.9)$$

Selain itu dihitung juga perubahan bias dengan Persamaan 2.10.

$$\Delta w_{k0} = \alpha \delta_k \quad (2.10)$$

**Langkah 7:** Hitung faktor  $\delta$  *hidden layer* berdasarkan galat pada *hidden unit*  $Z_j$  dengan Persamaan 2.11.

$$\delta\_in_j = \sum_{k=1}^m \delta_k w_{kj} \quad (2.11)$$

Lalu hitung faktor koreksi galat  $\delta_j$  dengan Persamaan 2.12.

$$\delta_j = \delta\_in_j f'(z\_in_j) \quad (2.12)$$

Hitung perubahan bobot  $V_{ji}$  (yang akan dipakai untuk merubah bobot  $V_{ji}$ ) dengan Persamaan 2.13.

$$\Delta v_{ji} = \alpha \delta_j x_i \quad (2.13)$$

Selain itu hitung juga perubahan bias untuk memperbarui  $v_{j0}$  dengan Persamaan 2.14.

$$\Delta v_{j0} = \alpha \delta_j \quad (2.14)$$

### 2.4.4 Fase 3: perubahan bobot

**Langkah 8:** Hitung semua perubahan bobot.

Perubahan bobot yang menuju ke *output unit* seperti Persamaan 2.15.

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (2.15)$$

Perubahan bobot yang menuju ke *hidden layer* seperti Persamaan 2.16.

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} \quad (2.16)$$

**Langkah 9:** Menguji apakah kondisi berhenti sudah terpenuhi.

Kondisi berhenti ini terpenuhi jika nilai galat yang dihasilkan lebih kecil dari nilai galat referensi.

## 2.5 Mean Absolute Percentage Error

*Mean Absolute Percentage Error* adalah rata-rata dari keseluruhan persentase selisih galat antara data aktual dengan data hasil prediksi. MAPE merupakan salah satu metode perhitungan galat yang banyak digunakan karena keuntungannya dari skala independensi dan interpretasinya (Kim & Kim, 2016). Semakin kecil nilai persentase MAPE dan semakin dekat dengan 0% maka dapat dikatakan hasil tersebut adalah hasil yang baik. Selain itu, MAPE yang semakin mendekati 0% menandakan jika nilai prediksi tersebut semakin mendekati nilai aktual. Persamaan 2.17 merupakan persamaan yang digunakan untuk menghitung MAPE:

$$MAPE = \left( \frac{1}{N} \sum \frac{|A_i - F_i|}{|A_i|} \right) \times 100\% \quad (2.17)$$

Ket:

$N$  = jumlah data

$A_i$  = nilai aktual

$F_i$  = nilai prediksi

Menurut Hsu & Wang (2008), nilai MAPE dikelompokkan dalam empat kriteria untuk mempresentasikan baik atau tidaknya prediksi tersebut yang ditunjukkan pada Tabel 2.2.

**Tabel 2.2 Kriteria prediksi**

Nilai MAPE	Kriteria Prediksi
<10%	Prediksi dengan akurasi sangat baik
10% - 20%	Prediksi dengan akurasi baik
20% - 50%	Prediksi dengan akurasi cukup
>50%	Prediksi dengan akurasi lemah atau tidak akurat

## 2.6 K-Fold Cross Validation

*K-fold cross validation* merupakan salah satu metode evaluasi yang dilakukan dengan membagi data dalam *fold* (segmen) sejumlah  $k$  dan dilatih sebanyak  $n$  kali serta validasi sebanyak  $k$  kali (Werdiastu, Ratnawati, & Rahayudi, 2018). Metode ini berguna untuk mengevaluasi performa dari metode yang digunakan, hal tersebut dapat diketahui dari rentang nilai akurasi yang dihasilkan. Dalam *k-fold cross validation*, data dibagi sebanyak  $k$  dan pada setiap  $k$  dibagi menjadi set data *training* dan set data *testing* dalam rasio yang dikehendaki. Pada umumnya, jumlah  $k$  yang digunakan dalam validasi ini adalah 5 *fold* atau 10 *fold*.

## BAB 3 METODOLOGI

Dalam bab ini akan dibahas secara detail tipe penelitian dan strategi yang digunakan dalam penelitian. Tipe penelitian sendiri menjelaskan termasuk dalam tipe yang manakah penelitian yang dilakukan, implementatif atau non-implementatif dan pendekatan penelitian yang digunakan pada tipe tersebut. Selanjutnya, strategi yang digunakan dalam penelitian membahas lebih spesifik tentang metode penelitian, subyek penelitian, lokasi penelitian, teknik pengumpulan dan analisis data, peralatan-peralatan yang mendukung saat penelitian dan yang terakhir adalah jadwal penelitian.

### 3.1 Tipe Penelitian

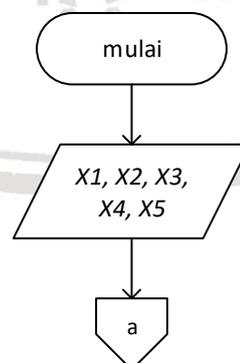
Berdasarkan judul yang digunakan, tipe penelitian ini termasuk dalam penelitian non-implementatif karena hasil akhir penelitian ini bukan prototipe atau perangkat lunak melainkan hasil analisis dari sekumpulan data. Lalu tipe pendekatan penelitian ini adalah pendekatan analitik, ini dijelaskan dengan penggunaan data berdasarkan lima variabel *Altman* yang diamati dalam waktu tertentu dengan tujuan menjelaskan kelima variabel tersebut terhadap adanya kebangkrutan pada sebuah perseroan terbatas yang bangkrut maupun tidak bangkrut.

### 3.2 Strategi dan Rancangan Penelitian

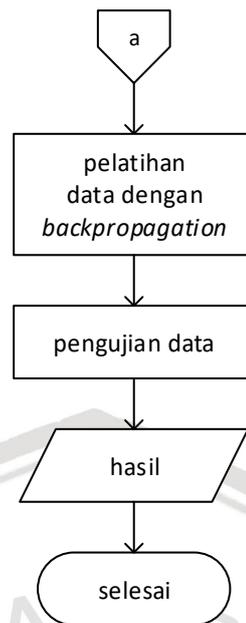
Strategi dan rancangan penelitian menjelaskan secara detail bagaimana metode penelitian yang akan dilakukan, hal yang menjadi obyek penelitian, dimanakah lokasi penelitian ini, bagaimana pengumpulan data dilaksanakan, teknik analisis data yang digunakan dan pembahasan hasilnya, peralatan pendukung dalam melakukan penelitian dan jadwal penelitian secara rinci.

#### 3.2.1 Metode Penelitian

Dalam menyelesaikan permasalahan dan mengacu pada tipe penelitian non-implementatif, metode penelitian prediksi kebangkrutan menggunakan metode *backpropagation* ini dilakukan beberapa tahap seperti pada Gambar 3.1:



Gambar 3.1 *Flowchart* penyelesaian permasalahan menggunakan metode *backpropagation*



**Gambar 3.1 Flowchart penyelesaian permasalahan menggunakan metode *backpropagation* (Lanjutan)**

Berdasarkan Gambar 3.1, penyelesaian prediksi kebangkrutan menggunakan metode *backpropagation* diawali dengan memasukkan fitur ( $x_1, x_2, x_3, x_4, x_5$ ) data kemudian dilanjutkan dengan pelatihan data dengan *backpropagation*. Lalu dilakukan pengujian terhadap data yang digunakan dan ditampilkan *output* hasil pengujian data tersebut.

### 3.2.2 Obyek Penelitian

Obyek yang menjadi sasaran penelitian ini adalah laporan keuangan tahunan tahun 2014 sampai 2017 perseroan terbatas yang terdaftar pada Bursa Efek Indonesia yang memenuhi lima variabel *Altman*.

### 3.2.3 Lokasi Penelitian

Lokasi penelitian berada di Lab. Riset Komputasi Cerdas dan Visualisasi, Fakultas Ilmu Komputer, Universitas Brawijaya.

### 3.2.4 Pengumpulan Data

Data dikumpulkan dengan mengunduh laporan keuangan tahunan pada *website* resmi Bursa Efek Indonesia pada <http://www.idx.co.id/>. Laporan keuangan tahun 2014 sampai 2016 digunakan sebagai data *training* dengan masing-masing 100 laporan tiap tahunnya dan 130 laporan tahunan pada tahun 2017 digunakan sebagai data *testing*. Seluruh laporan keuangan tahunan perseroan terbatas yang sudah dikumpulkan diolah terlebih dahulu berdasarkan lima rasio keuangan *Altman* seperti subbab 2.1 dan *Z-Score* digunakan sebagai referensi kebangkrutan perseroan terbatas seperti Persamaan 2.1. Tabel 3.1 menjelaskan fitur input lima variabel *Altman*. Contoh data yang dikumpulkan ditunjukkan pada Tabel 3.2, lalu data yang sudah diolah ditunjukkan pada Tabel 3.3, untuk data yang lebih lengkap dapat dilihat pada Lampiran 1 dan Lampiran 2.

**Tabel 3.1** Fitur data

Sifat	Fitur	Keterangan
input	$x_1$	$\frac{\text{modal kerja}}{\text{total aktiva}}$
	$x_2$	$\frac{\text{laba ditahan}}{\text{total aktiva}}$
	$x_3$	$\frac{\text{pendapatan sebelum pajak dan bunga}}{\text{total aktiva}}$
	$x_4$	$\frac{\text{nilai pasar ekuitas}}{\text{nilai buku dari hutang}}$
	$x_5$	$\frac{\text{penjualan}}{\text{total aktiva}}$
target	$y$	Nilai <i>Z-Score</i> masing-masing perusahaan

Dari Tabel 3.1 dapat dijelaskan bahwa:

1. Fitur input  $x_1$  didapatkan dari pembagian modal kerja dengan total aktiva.
2. Fitur input  $x_2$  didapatkan dari pembagian laba ditahan dengan total aktiva.
3. Fitur input  $x_3$  didapatkan dari pembagian pendapatan sebelum pajak dan bunga dengan total aktiva.
4. Fitur input  $x_4$  didapatkan dari pembagian nilai pasar ekuitas dengan nilai buku dari hutang.
5. Fitur input  $x_5$  didapatkan dari pembagian penjualan dibagi dengan total aktiva.
6. Nilai *Z-Score* masing-masing perusahaan diketahui dengan menggunakan persamaan 2.1.

Tabel 3.2 Contoh data yang sudah didapatkan

Kode Saham	Total aktiva	Modal kerja	Laba ditahan	Pendapatan sebelum pajak dan bunga	Nilai pasar ekuitas	Nilai buku dari hutang	Penjualan
DNET	10,899,945,000,000	1,761,426,000,000	2,024,403,065,335	1,764,461,447,372	31,914,000,000,000	39,425,110,339,547	3,656,369,000,000
MAMI	883,810,096,621	207,255,927,537	258,580,088,790	90,163,125,868	385,756,000,000	636,254,044,687	152,615,094,145
BMTR	27,694,734,000,000	6,091,632,000,000	6,588,210,000,000	4,026,069,000,000	8,168,748,016,980	13,568,375,000,000	10,229,450,000,000
HDTX	4,035,086,385,000	835,800,622,000	622,378,284,583	700,884,530,000	1,786,325,548,800	3,701,551,196,000	1,293,363,942,000
JIHD	6,655,376,027,000	3,132,221,684,000	1,404,642,633,000	829,824,266,000	1,071,358,621,720	2,307,230,792,000	1,371,672,386,000
ABBA	427,808,472,537	90,535,163,406	56,433,262,374	58,206,514,171	137,756,250,000	309,462,989,695	267,932,542,391
LMPI	834,548,374,286	211,768,317,674	125,439,556,521	94,598,578,079	168,422,450,723	458,292,046,535	411,144,165,006
DYAN	1,452,680,785,154	600,000,000,000	186,596,754,804	259,145,684,540	235,792,075,624	661,368,644,600	461,193,680,750
KOBX	1,201,175,044,680	563,512,108,480	148,072,173,800	134,647,846,200	198,792,500,000	908,576,555,520	609,214,857,080
LMAS	446,339,446,199	157,371,381,771	53,812,623,477	88,165,510,848	48,846,794,550	318,464,006,522	189,230,788,762
CNKO	3,705,091,478,000	1,019,861,127,000	897,338,308,000	679,376,235,000	456,774,421,506	3,285,230,351,000	938,822,975,000
INDR	10,785,462,189,080	3,050,713,328,720	1,612,910,604,560	1,260,140,984,600	817,939,633,750	6,953,018,171,800	6,486,429,741,400
MAPI	11,425,390,076,000	2,233,827,684,000	2,198,886,869,000	1,597,451,062,000	10,252,889,160,000	10,282,975,931,000	6,565,635,149,000
TELE	8,749,797,000,000	5,965,781,000,000	1,519,517,000,000	1,261,256,000,000	7,055,046,860,385	7,206,421,000,000	7,914,330,000,000
INDF	87,939,488,000,000	10,877,636,000,000	19,506,084,000,000	17,658,554,000,000	66,950,752,062,500	69,282,764,000,000	70,186,618,000,000

Tabel 3.3 Contoh data yang sudah diolah

tahun	x1	x2	x3	x4	x5	y
2014	0.200934	0.369033	0.104748	0.662185	0.378864	1.879609
2014	0.325374	0.311891	0.15822	0.277182	0.365822	1.881351
2014	0.211382	0.500894	0.146983	0.380372	0.21626	1.884436
2014	0.203515	0.449121	0.10222	0.173431	0.595745	1.910118
2014	0.152084	0.190191	0.367373	0.146828	0.185033	1.934229
2014	0.58711	0.138302	0.168774	0.473759	0.198972	1.938335
2014	0.242608	0.484395	0.117186	0.28471	0.423245	1.950068
2014	0.200934	0.369033	0.104748	0.662185	0.378864	1.879609
	...	...	...	...	...	...

### 3.2.5 Analisis Data

Analisis yang dilakukan terhadap data adalah dengan perhitungan MAPE. MAPE digunakan untuk menghitung persentase nilai galat prediksi pada akhir pelatihan *backpropagation*.

### 3.2.6 Peralatan Pendukung Penelitian

Berikut ini adalah daftar spesifikasi peralatan pendukung yang digunakan dalam penelitian ini:

- Perangkat Keras
  1. Prosesor AMD RYZEN 5-2500U
  2. RAM 8 GB
- Perangkat Lunak
  1. Sistem operasi *Windows 10* (atau yang lebih rendah)
  2. *Netbeans Integrated Development Environment (IDE)* versi 8.0 (atau yang lebih tinggi) sebagai perangkat lunak dalam implementasi sistem

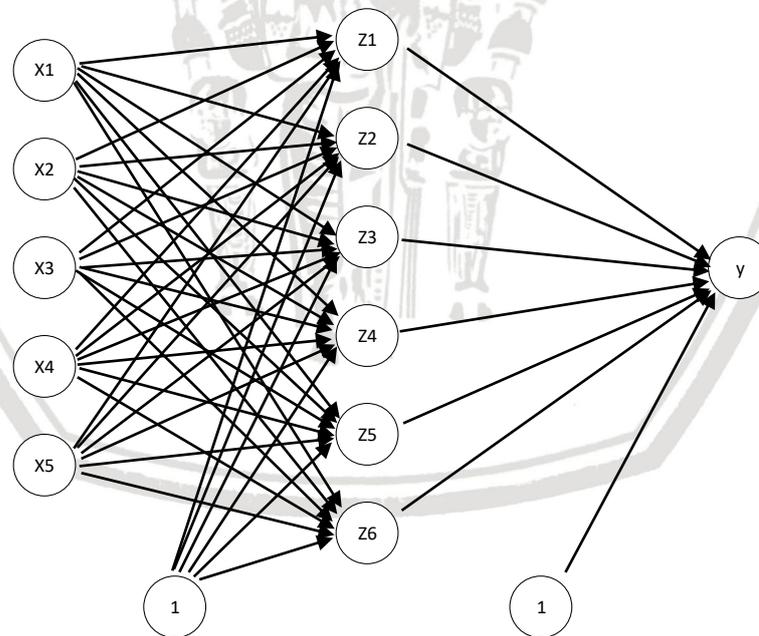
## BAB 4 PERANCANGAN

Pada bab ini akan dijelaskan hasil yang relevan dengan pelaksanaan metode penelitian yang digunakan. Penjelasan hasil tersebut didukung dengan penyajian data, gambar atau tabel yang berkesesuaian.

### 4.1 Formulasi Permasalahan

Kasus yang akan diselesaikan dalam penelitian ini adalah memprediksi adanya kebangkrutan pada perseroan terbatas yang terdaftar pada Bursa Efek Indonesia. Data yang digunakan adalah laporan keuangan perseroan terbatas pada tahun 2014 sampai tahun 2017 dengan total keseluruhan 430 laporan. Laporan keuangan yang sudah dikumpulkan diolah terlebih dahulu berdasarkan lima variabel *Altman* seperti pada subbab 2.1. Selanjutnya untuk mengetahui keadaan kebangkrutannya digunakan persamaan *Z-Score* seperti Persamaan 2.1. Data laporan keuangan tahun 2017 digunakan sebagai data uji karena menjadi tahun *target* prediksi, selanjutnya data laporan keuangan tahun 2014-2016 digunakan sebagai data latih.

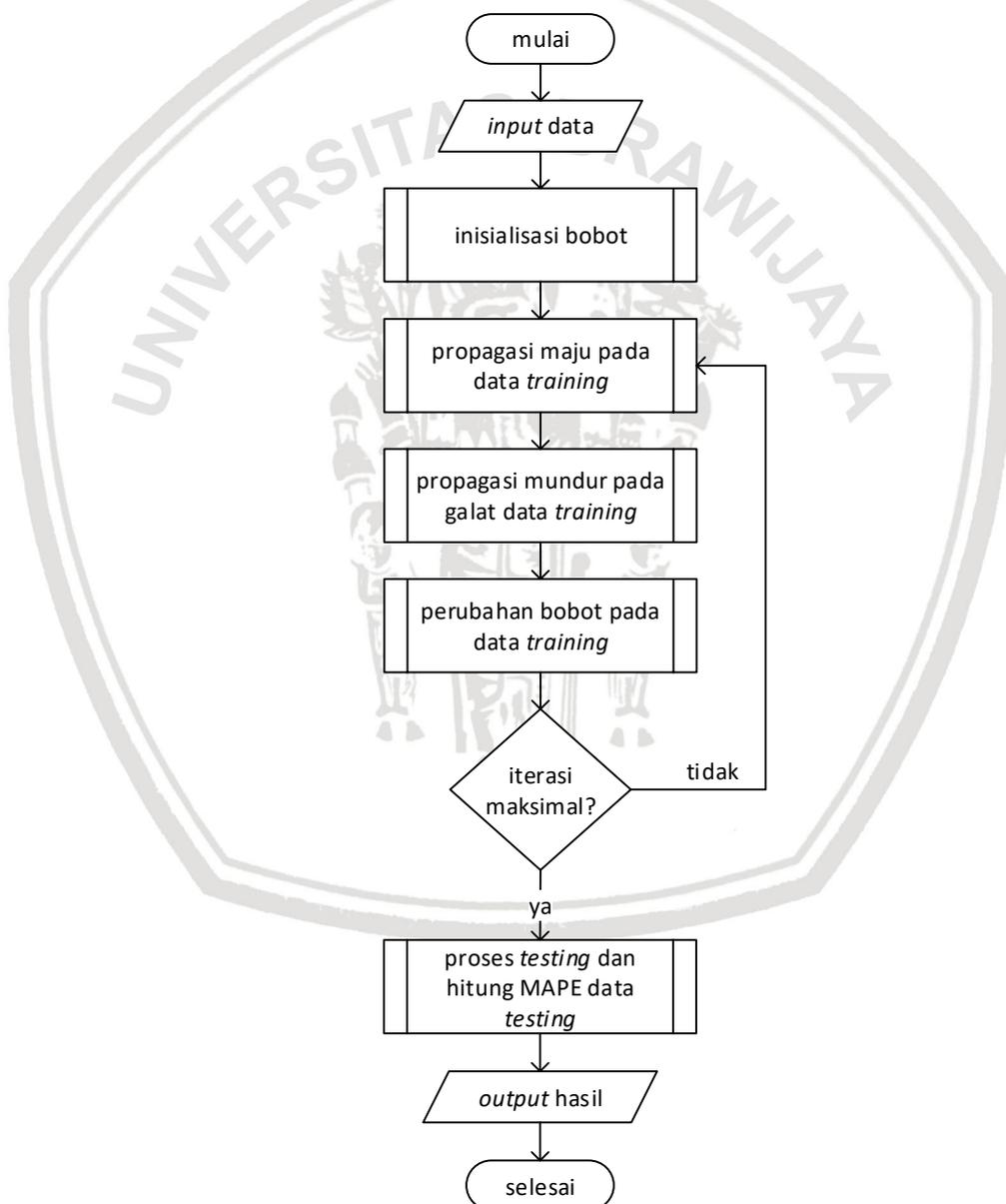
Penyelesaian masalah utama dalam penelitian ini adalah *backpropagation* digunakan untuk memprediksi kebangkrutan perseroan terbatas dengan lima variabel *Altman* sebagai input dan hasil dari perhitungan *Z-Score* sebagai target *output*. Berdasarkan penggunaan lima variabel *Altman* sebagai input, Gambar 4.1 merupakan arsitektur jaringan *backpropagation* yang digunakan dalam penelitian ini dengan contoh jumlah neuron pada *hidden layer* sebanyak enam.



Gambar 4.1 Arsitektur jaringan *backpropagation* dengan lima input dengan enam neuron pada *hidden layer*

## 4.2 Siklus Penyelesaian Masalah

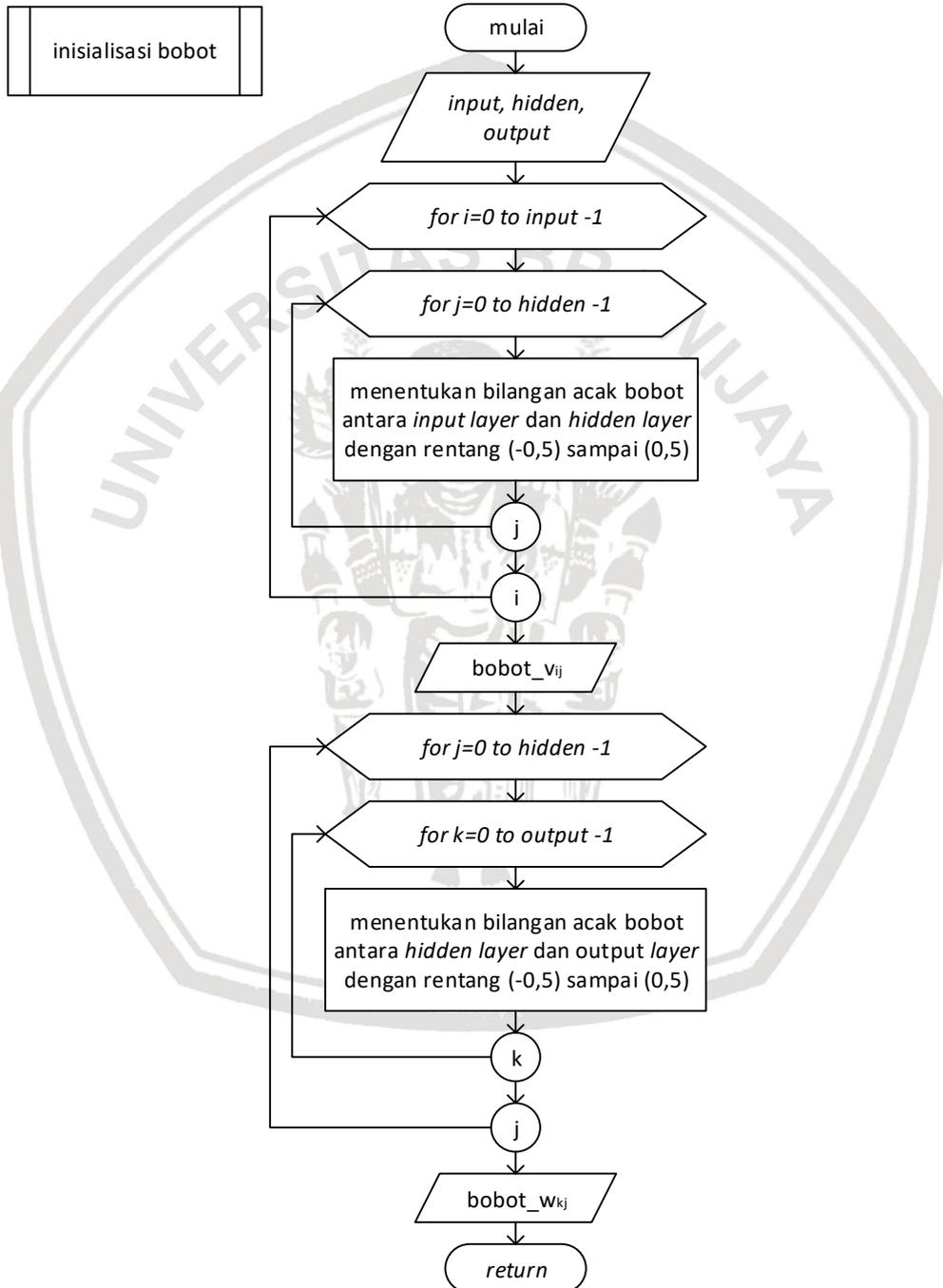
Penyelesaian masalah prediksi *business failure* pada perseroan terbatas yang terdaftar pada Bursa Efek Indonesia berdasarkan Gambar 4.2 dimulai dengan input data variabel *Altman* yang didapatkan dari laporan keuangan tahunan. Langkah selanjutnya dilakukan inialisasi bobot awal jaringan dan dilanjutkan dengan pelatihan *backpropagation* pada data *training* yang terdiri dari fase propagasi maju, fase propagasi mundur dan fase perubahan bobot. Setelah dilakukan ketiga fase *backpropagation* tersebut, selanjutnya dilakukan cek kondisi apakah memenuhi kondisi iterasi maksimal yang telah ditentukan. Jika sudah memenuhi kondisi tersebut, maka dilanjutkan dengan fase propagasi maju dengan bobot awal yang sudah diinisialisasi pada data *testing* dan menghitung nilai MAPE data *testing*.



Gambar 4.2 Flowchart siklus penyelesaian masalah

### 4.2.1 Inisialisasi Bobot

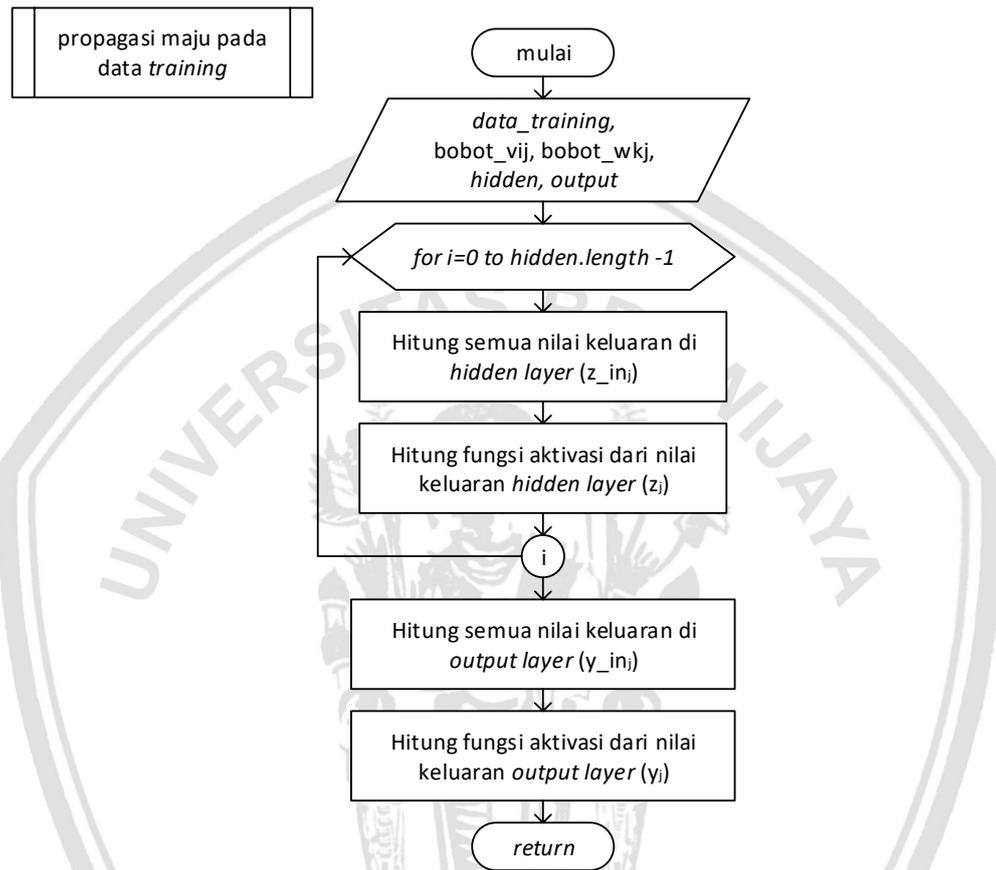
Proses inisialisasi bobot berdasarkan Gambar 4.3 proses inisialisasi dimulai dengan membangkitkan nilai bobot antara input layer dengan hidden layer ( $v_{ij}$ ) sebanyak jumlah neuron pada input layer dan bias dikalikan dengan jumlah neuron pada hidden layer. Selanjutnya membangkitkan nilai bobot antara hidden layer dengan output layer ( $w_{kj}$ ) sebanyak jumlah neuron pada hidden layer dan bias dikalikan dengan jumlah neuron pada output layer. Nilai bobot masing-masing neuron diacak dengan rentang antara (-0,5) hingga (0,5).



Gambar 4.3 Flowchart proses inisialisasi bobot

### 4.2.2 Fase Propagasi Maju pada Data Training

Dalam proses propagasi maju pada data *training* seperti Gambar 4.4 menghitung nilai keluaran pada *hidden layer* ( $z_{in_j}$ ) menggunakan Persamaan 2.4 dan menghitung fungsi aktivasi dari nilai keluaran pada *hidden layer* ( $z_j$ ) menggunakan Persamaan 2.5. Selanjutnya dihitung nilai keluaran pada output layer ( $y_{in_k}$ ) menggunakan Persamaan 2.6 dan menghitung fungsi aktivasi dari nilai keluaran pada *output layer* ( $y_k$ ) menggunakan Persamaan 2.7.

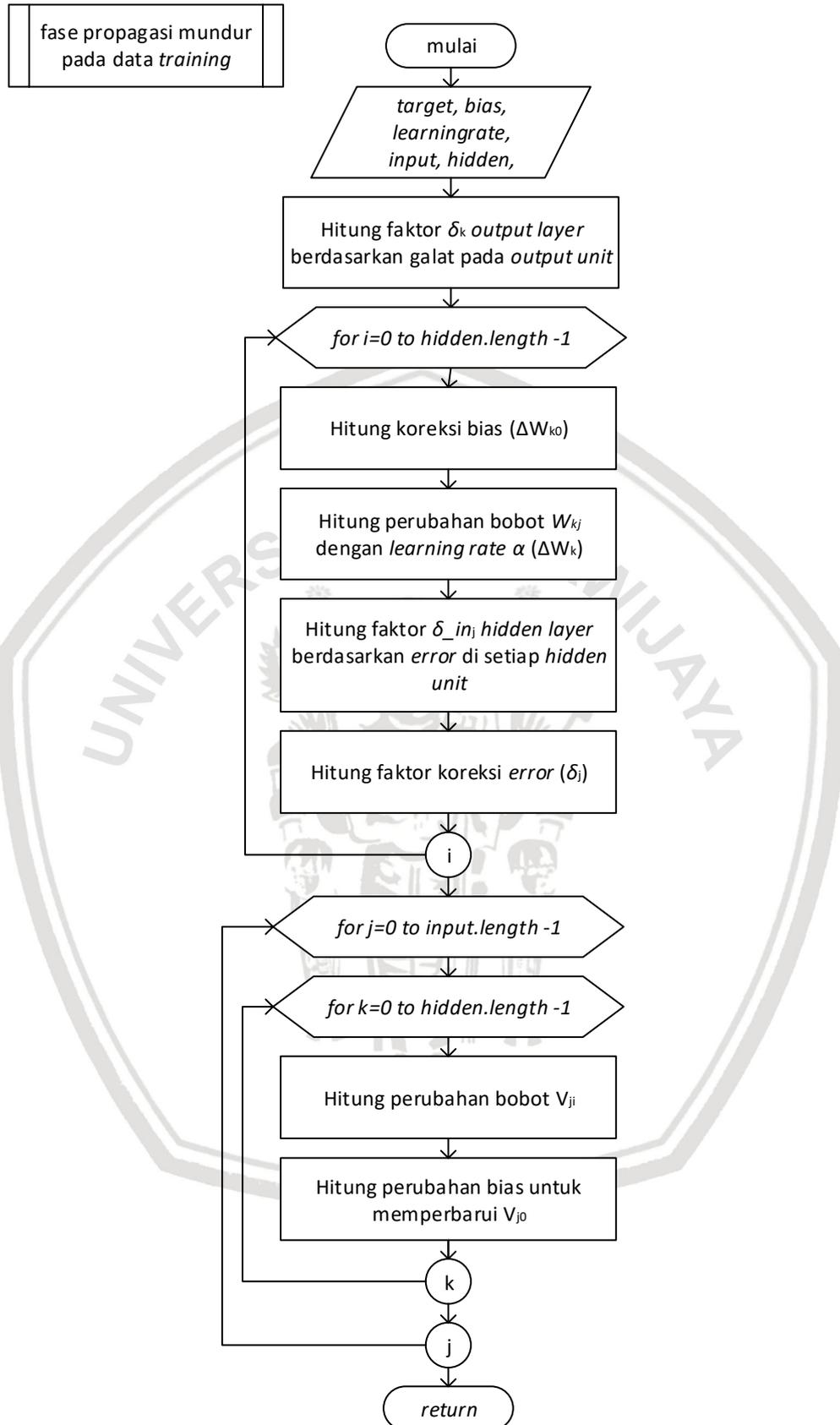


Gambar 4.4 Flowchart proses propagasi maju pada data *training*

### 4.2.3 Fase Propagasi Mundur pada Data Training

Dalam proses propagasi mundur pada data *training* yang ditunjukkan dengan Gambar 4.5, dilakukan perhitungan faktor galat ( $\delta$ ) *output layer* berdasarkan galat pada *output unit* menggunakan Persamaan 2.8. Lalu menghitung perubahan bobot  $W_{kj}$  ( $\Delta W_{kj}$ ) dengan *learning rate*  $\alpha$  menggunakan Persamaan 2.9 dan menghitung koreksi bias ( $\Delta W_{k0}$ ) menggunakan Persamaan 2.10. Proses selanjutnya yakni menghitung faktor galat *hidden layer* berdasarkan galat pada *hidden unit* ( $\delta_{in_j}$ ) menggunakan Persamaan 2.11 yang kemudian dilanjutkan dengan menghitung faktor koreksi galat ( $\delta_j$ ) menggunakan Persamaan 2.12. Proses terakhir dalam proses propagasi mundur yakni dengan menghitung perubahan bobot  $V_{ji}$  ( $\Delta v_{ji}$ ) menggunakan Persamaan 2.13 dan menghitung perubahan bias ( $\Delta v_{j0}$ ) menggunakan Persamaan 2.14.

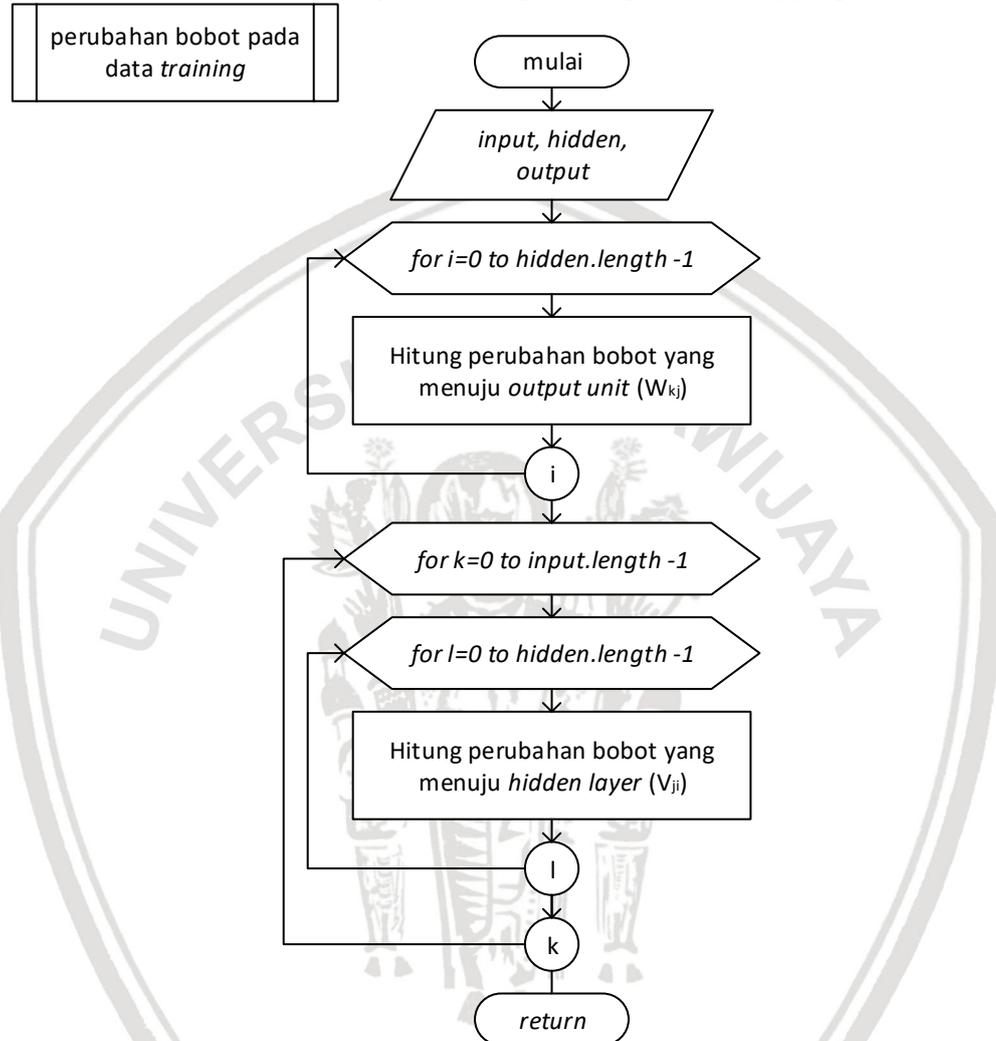




Gambar 4.5 Flowchart proses propagasi mundur pada *training*

#### 4.2.4 Fase Perubahan Bobot pada Data Training

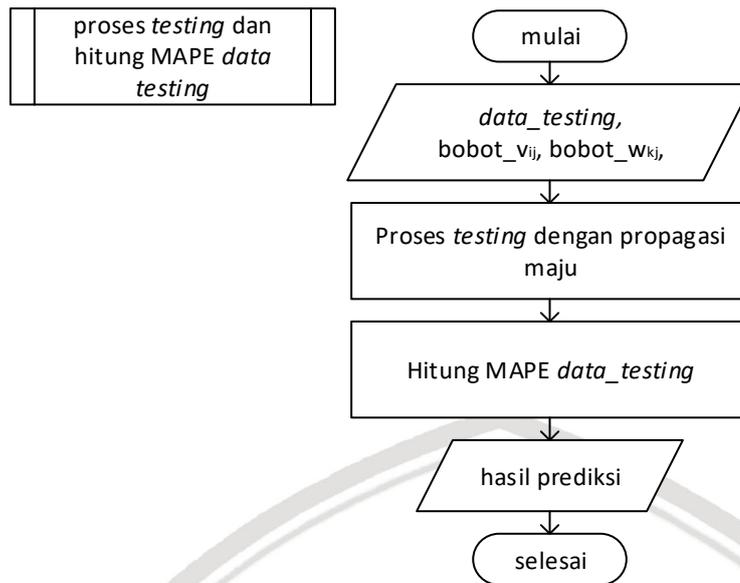
Berdasarkan Gambar 4.6, proses perubahan bobot diawali dengan menghitung bobot  $w_{kj}$  (*baru*) yang menuju *output unit* menggunakan Persamaan 2.15 lalu dilanjutkan dengan menghitung bobot  $v_{ji}$  (*baru*) yang menuju pada *hidden layer* menggunakan Persamaan 2.16. Setelah proses perubahan bobot selesai dilakukan maka dilanjutkan dengan pengecekan *stopping condition*.



Gambar 4.6 Flowchart proses perubahan bobot pada data training

#### 4.2.5 Fase Testing dan Menghitung MAPE Data Testing

Dalam proses *testing* seperti Gambar 4.7 dilakukan dengan propagasi maju pada data *testing* yang sama seperti pada data *training* sebelumnya. Bobot yang digunakan pada proses propagasi maju pada data *testing* merupakan bobot terbaik yang didapat dalam proses *training*. Setelah dilakukan *testing* maka dilanjutkan dengan menghitung MAPE data *testing* sehingga dapat diketahui akurasi hasil prediksinya.



Gambar 4.7 Flowchart proses testing dan hitung MAPE data testing

### 4.3 Perhitungan Manual

Penyelesaian masalah seperti yang dijelaskan pada Subbab 4.1 dilakukan dengan metode *backpropagation*. Sebelum memulai perhitungan manual dengan *backpropagation*, ditentukan terlebih dahulu contoh parameter jumlah input neuron, jumlah *hidden neuron*, jumlah *output neuron*, jumlah iterasi dan *learning rate* ( $\alpha$ ) sebagai berikut:

- Jumlah fitur input *neuron* = 5 (+ 1 bias)
- jumlah *hidden neuron* = 6 (+ 1 bias)
- jumlah *output neuron* = 1
- jumlah iterasi = 3
- *learning rate* = 0,2
- data *training* = 20
- data *testing* = 10

Inisialisasi bobot ditentukan secara acak dalam rentang nilai [-0,5; 0,5], bobot diacak sesuai dengan jumlah input *neuron*, jumlah *hidden neuron* dan jumlah *output neuron* pada arsitektur *backpropagation*. Selain itu, ketiga parameter tersebut juga berfungsi untuk menjadi acuan inisialisasi bobot. Tabel 4.1 adalah contoh data *training training* yang digunakan, lalu Tabel 4.2 adalah contoh data *testing* yang digunakan.

Tabel 4.1 Contoh data *training*

X1	X2	X3	X4	X5	Y
0.119604075	0.277198298	0.183076795	0.120747608	0.493137323	1.7013418
0.504619758	0.153281213	0.156124141	0.162184641	0.278370054	1.7110279
0.168538609	0.177865642	0.206134253	0.129804207	0.521686242	1.73107
0.158479956	0.125761513	0.138456951	0.644660658	0.530737443	1.7406838
0.280321418	0.226193614	0.116178285	0.952588834	0.173895028	1.7818934
0.389751088	0.192866555	0.135759282	0.314996305	0.461766033	1.8364839



**Tabel 4.1 Contoh data *training* (Lanjutan)**

X1	X2	X3	X4	X5	Y
0.402071563	0.177966102	0.219397363	0.263748597	0.24086629	1.8547652
0.213107124	0.188682539	0.188845447	0.560834686	0.399986036	1.8795609
0.200934139	0.369032665	0.104747834	0.662184654	0.378863974	1.8796093
0.325373619	0.311890838	0.158219623	0.277181792	0.365821962	1.8813513
0.211382114	0.500894309	0.146982517	0.380371705	0.216260163	1.8844361
0.203515264	0.449121184	0.102220167	0.173431275	0.595744681	1.910118
0.152083813	0.190191112	0.367372784	0.146828358	0.185033387	1.9342287
0.587110445	0.138301721	0.16877358	0.473759047	0.198971523	1.9383347
0.242608236	0.484395102	0.117185922	0.284709629	0.423245229	1.9500676
0.281362419	0.196983804	0.22631971	0.20854532	0.470806723	1.9562012
0.143824407	0.195835109	0.254980133	0.613230364	0.308744634	1.9648757
0.589388987	0.11169223	0.109479507	0.687138183	0.327985919	1.9651871
0.363180184	0.146480799	0.164391115	0.797084982	0.322705516	1.9843365
0.127490651	0.259469945	0.317922398	0.365791172	0.211929444	1.9967948
0.168026168	0.35302207	0.218269026	0.486620664	0.310099876	2.0182224
0.225688892	0.210598534	0.21185797	0.607739459	0.404928134	2.0343677
0.147909924	0.168949797	0.319741065	0.34997358	0.355529234	2.0346805
0.198108076	0.109780148	0.42541636	0.165912518	0.141358943	2.0362023
0.162338741	0.206545757	0.277485423	0.857175684	0.129205513	2.0431834
0.126400363	0.342462243	0.232293773	0.183114288	0.54401736	2.051583
0.260903427	0.300623053	0.277258567	0.386155606	0.171339564	2.0519426
0.393661551	0.267889177	0.283037672	0.203635527	0.153677497	2.0573218
0.102382486	0.256278171	0.200901481	0.612716763	0.555698648	2.067952
0.17409932	0.450619719	0.230273117	0.355043474	0.258452675	2.0711668

**Tabel 4.2 Contoh data *testing***

X1	X2	X3	X4	X5	Y
0.207133266	0.154241626	0.173697528	0.182588367	0.320529431	1.467782
0.234502783	0.292574264	0.102016402	0.262924129	0.172678605	1.358095
0.154050567	0.397578956	0.205400149	0.22659049	0.661437516	2.216684
0.511206796	0.100647609	0.298171097	0.311211203	0.358904047	2.28395
0.180066649	0.197396238	0.320240593	0.75845348	0.171583684	2.175884
0.58384458	0.41045937	0.208161672	0.185405773	0.297465876	2.370899
0.125473825	0.253158805	0.397826589	0.293080112	0.126529933	2.120197
0.36203165	0.465916516	0.144882896	0.745129668	0.124393331	2.136306
0.521925149	0.137634005	0.175602361	0.884889584	0.420776521	2.350196
0.575258628	0.353144864	0.241905033	0.166819636	0.136899549	2.219991

### 4.3.1 Inisialisasi Bobot

Jumlah bobot yang akan diinisialisasi mengacu pada jumlah input *neuron*, jumlah *hidden neuron* dan jumlah *output neuron*. Bobot dan bias awal yang digunakan antara *hidden layer* dan *output layer* berjumlah 7 bobot. Sedangkan antara input *layer* dan *hidden layer* berjumlah 36, sehingga ada 43 bobot. Nilai bobot yang akan diacak berada dalam rentang (-0,5) sampai (0,5). Indeks bobot W mewakili bobot dan bias yang berada antara *hidden layer* dan *output layer*,



sedangkan indeks bobot  $V$  mewakili bobot dan bias yang berada antara input *layer* dan *hidden layer*. Tabel 4.3 merupakan contoh inisialisasi bobot acak.

**Tabel 4.3 Contoh inisialisasi bobot**

w10	w11	w12	w13	w14	w15	w16	v10	v20	...	v45	v55	v65
0,2	0,3	0,4	0,1	0,49	-0,1	0,28	-0,3	-0,2	...	0,1	-0,23	0,14

### 4.3.2 Fase Propagasi Maju pada Data *Training*

Fase propagasi maju pada data *training* dimulai dengan menghitung seluruh keluaran pada *hidden layer* dan menghitung fungsi aktivasi setiap keluaran. Data *training* yang digunakan adalah seperti pada Tabel 4.1. Selanjutnya menghitung keluaran jaringan pada *output layer* dan menghitung fungsi aktivasi keluaran jaringan pada *output layer* tersebut. Fungsi aktivasi yang digunakan dalam penelitian ini adalah fungsi linier. Untuk menjelaskan prosesnya diberikan contoh penghitungan manual proses propagasi maju pada data *training*:

- Menghitung semua keluaran di *hidden layer* dengan Persamaan 2.4.  

$$z_{in_1} = -0,3 + 0,1196 \times 0,13 + 0,27719 \times (-0,19) + 0,18307 \times (-0,4) + 0,12074 \times 0,2 + 0,49313 \times 0,32 = (-0,2284)$$

Dengan cara yang sama didapatkan nilai-nilai keluaran di *hidden layer* ( $z_{in}$ ) yang dijelaskan pada Tabel 4.4.

**Tabel 4.4 Nilai keluaran di *hidden layer* ( $z_{in}$ ).**

$z_{in_1}$	$z_{in_2}$	$z_{in_3}$	$z_{in_4}$	$z_{in_5}$	$z_{in_6}$
-0,2284	0,15474	0,42618	-0,25187	0,40969	0,00652

- Menghitung fungsi aktivasi linier setiap keluaran dengan Persamaan 2.2.  

$$z = f(z_{in_1}) = z_{in_1} = -0,2284$$

Dengan cara yang sama didapatkan nilai-nilai fungsi aktivasi linier dari setiap keluaran  $f(z_{in})$  yang dijelaskan pada Tabel 4.5.

**Tabel 4.5 Fungsi linier pada nilai keluaran di *hidden layer*  $f(z_{in})$**

$f(z_{in_1})$	$f(z_{in_2})$	$f(z_{in_3})$	$f(z_{in_4})$	$f(z_{in_5})$	$f(z_{in_6})$
-0,2284	0,15474	0,42618	-0,25187	0,40969	0,00652

- Menghitung semua keluaran jaringan dengan Persamaan 2.6 lalu menghitung fungsi aktivasi linier dengan Persamaan 2.7  

$$y_{in_1} = 0,2 + (-0,2284) \times 0,3 + 0,15474 \times 0,4 + 0,42618 \times 0,1 + (-0,25187) \times 0,49 + 0,40969 \times (-0,1) + 0,00652 \times 0,28 = 0,07343$$

$$y = f(y_{in_1}) = y_{in_1} = 0,07343$$



### 4.3.3 Fase Propagasi Mundur pada Data Training

Fase propagasi maju pada data *training* diawali dengan menghitung faktor galat pada *output layer* kemudian dilanjutkan dengan menghitung perubahan bobot  $W_{kj}$  dan perubahan bias  $W_{k0}$ . Langkah selanjutnya yakni menghitung faktor galat pada setiap *hidden layer* dan menghitung koreksi galat. Langkah terakhir dalam fase propagasi mundur pada data *training* yakni menghitung perubahan bobot  $V_{ij}$  dan perubahan bias  $V_{j0}$ . Untuk menjelaskan prosesnya diberikan contoh penghitungan manual proses propagasi mundur pada data *training*:

1. Menghitung faktor  $\delta$  *output layer* berdasarkan galat di setiap *output unit* jaringan dengan Persamaan 2.8  

$$\delta_1 = ((-1) - 0,71168) \times 1 = 1,62719$$
2. Menghitung perubahan bobot  $W_{kj}$  dengan Persamaan 2.9 dan perubahan bias  $W_{k0}$  dengan Persamaan 2.10.  

$$\Delta w_0 = 0,2 \times 1,62719 \times 1 = 0,32558$$

$$\Delta w_1 = 0,2 \times 1,62719 \times -0,2284 = -0,07436$$

Dengan cara yang sama didapatkan nilai-nilai hasil hitung perubahan bobot  $W_{kj}$  dan  $W_{k0}$  yang ditunjukkan pada Tabel 4.6.

**Tabel 4.6 Nilai perubahan bobot  $W_{k0}$  dan  $W_{kj}$ .**

$\Delta w_0$	$\Delta w_1$	$\Delta w_2$	$\Delta w_3$	$\Delta w_4$	$\Delta w_5$	$\Delta w_6$
0,32558	-0,07436	0,05038	0,13876	-0,082	0,13339	0,00212

3. Menghitung faktor  $\delta$  *hidden layer* berdasarkan galat pada *hidden unit* dengan Persamaan 2.11.  

$$\delta_{in_1} = 1,62791 \times 0,3 = 0,48837$$

Dengan cara yang sama didapatkan nilai-nilai hasil hitung faktor  $\delta$  *hidden layer* yang ditunjukkan pada Tabel 4.7.

**Tabel 4.7 Nilai faktor  $\delta$  *hidden layer*.**

$\delta_{in_1}$	$\delta_{in_2}$	$\delta_{in_3}$	$\delta_{in_4}$	$\delta_{in_5}$	$\delta_{in_6}$
0,48837	0,65116	0,16279	0,79767	-0,16279	0,45581

4. Menghitung koreksi galat  $\delta_j$  dengan Persamaan 2.12.  

$$\delta_1 = 0,48837 \times 1 = 0,48837$$

Dengan cara yang sama didapatkan nilai-nilai hasil hitung koreksi galat  $\delta_j$  yang ditunjukkan pada Tabel 4.8.

**Tabel 4.8 Nilai koreksi galat  $\delta_j$**

$\delta_1$	$\delta_2$	$\delta_3$	$\delta_4$	$\delta_5$	$\delta_6$
0,48837	0,65116	0,16279	0,79768	-0,16279	0,45881

- Menghitung perubahan bobot  $V_{ij}$  dengan Persamaan 2.13 dan perubahan bias  $V_{j0}$  dengan Persamaan dan 2.14.

$$\Delta v_{10} = 0,2 \times 0,48837 \times 1 = 0,09767$$

$$\Delta v_{11} = 0,2 \times 0,65116 \times 0,1196 = 0,01168$$

...

Dengan cara yang sama didapatkan nilai-nilai hasil hitung perubahan bobot dan bias yang ditunjukkan pada Tabel 4.9.

**Tabel 4.9 Nilai perubahan bobot  $V_{ij}$  dan  $V_{j0}$ .**

$\Delta v_{10}$	$\Delta v_{20}$	$\Delta v_{30}$	$\Delta v_{40}$	$\Delta v_{50}$	$\Delta v_{60}$	$\Delta v_{11}$	...
0,0976	0,13023	0,03255	0,15954	-0,03255	0,09116	0,01168	...

#### 4.3.4 Fase Perubahan Bobot pada Data *Training*

Fase perubahan bobot pada data *training* berisi proses menghitung perubahan bobot yang menuju *output unit* dan proses menghitung perubahan bobot yang menuju *hidden layer*. Untuk menjelaskan prosesnya diberikan contoh penghitungan manual proses perubahan bobot pada data *training*:

- Menghitung semua perubahan yang menuju ke *output unit* dengan Persamaan 2.15.

$$w_{10}(\text{baru}) = 0,2 + 0,32558 = 0,52558$$

Dengan cara yang sama didapatkan nilai-nilai hasil perubahan bobot  $w(\text{baru})$  yang ditunjukkan pada Tabel 4.10.

**Tabel 4.10 Nilai  $w(\text{baru})$ .**

$w_{10}(\text{baru})$	$w_{11}(\text{baru})$	$w_{12}(\text{baru})$	$w_{13}(\text{baru})$	$w_{14}(\text{baru})$	$w_{15}(\text{baru})$	$w_{16}(\text{baru})$
0,52558	0,22564	0,45038	0,23876	0,408	0,03339	0,28212

- Menghitung semua perubahan yang menuju ke *hidden layer* dengan Persamaan 2.16.

$$v_{10}(\text{baru}) = -0,3 + 0,09767 = -0,20233$$

Dengan cara yang sama didapatkan nilai-nilai hasil perubahan bobot  $v(\text{baru})$  yang ditunjukkan pada Tabel 4.11.

**Tabel 4.11 Nilai  $v(\text{baru})$ .**

$v_{10}(\text{baru})$	$v_{20}(\text{baru})$	$v_{30}(\text{baru})$	$v_{40}(\text{baru})$	$v_{50}(\text{baru})$	$v_{60}(\text{baru})$	$v_{11}(\text{baru})$	...
-0,20233	-0,06977	0,23256	-0,27046	0,26744	-0,02088	0,14168	...

Cek *stopping condition* pada iterasi ke-3, bobot terakhir *update* pada keseluruhan tahap *training backpropagation* ditunjukkan pada Tabel 4.12.

**Tabel 4.12** Bobot terakhir *update* pada iterasi terakhir

w10	w11	w12	w13	...	v45	v55	v65
1,04324	0,08635	0,55617	0,36833	...	0,20957	-0,25061	0,20529

### 4.3.5 Fase Propagasi Maju dan Menghitung MAPE Data *Testing*

Setelah melaksanakan seluruh proses *training* dan cek *stopping condition* maka dilanjutkan dengan proses *testing*. Proses *testing* dilaksanakan dengan menggunakan data *testing* seperti Tabel 4.2 sebagai input sedangkan bobot yang digunakan dalam proses *testing* adalah bobot terbaik pada iterasi terakhir seperti yang ditunjukkan pada Tabel 4.12. Proses *testing* dilaksanakan dengan fase propagasi maju pada data *testing*. Setiap detail proses propagasi maju pada data *testing* sama seperti proses propagasi maju pada data *training*. Fase propagasi maju pada data *testing* dimulai dengan menghitung seluruh keluaran pada *hidden layer* yang dan menghitung fungsi aktivasi setiap keluaran. Selanjutnya menghitung keluaran jaringan pada *output layer* dan menghitung fungsi aktivasi keluaran. Fungsi aktivasi yang digunakan dalam penelitian ini adalah fungsi linier. Setelah dilakukan proses propagasi maju maka dilanjutkan dengan proses menghitung MAPE data *testing*. Untuk menjelaskan prosesnya diberikan contoh penghitungan manual proses propagasi maju dan hitung MAPE data *testing*:

- Menghitung semua keluaran di *hidden layer* dengan Persamaan 2.4.  

$$z_{in_1} = -0,09706 + 0,20713 \times 0,1804 + 0,15424 \times (-0,14835) + 0,17369 \times (-0,34761) + 0,18258 \times 0,19515 + 0,32052 \times 0,339033 = 0,0178$$

Dengan cara yang sama didapatkan nilai-nilai keluaran di *hidden layer* ( $z_{in}$ ) yang ditunjukkan pada Tabel 4.13.

**Tabel 4.13** Nilai keluaran di *hidden layer* ( $z_{in}$ )

$z_{in_1}$	$z_{in_2}$	$z_{in_3}$	$z_{in_4}$	$z_{in_5}$	$z_{in_6}$
0,0178	0,53175	0,52042	0,19312	0,44241	0,25802

- Menghitung fungsi aktivasi linier setiap keluaran dengan Persamaan 2.2.  

$$z = f(z_{in_1}) = z_{in_1} = 0,0178$$

Dengan cara yang sama didapatkan nilai-nilai fungsi aktivasi linier dari setiap keluaran  $f(z_{in})$  yang ditunjukkan pada Tabel 4.14.

**Tabel 4.14** Fungsi linier pada nilai keluaran di *hidden layer*  $f(z_{in})$

$f(z_{in_1})$	$f(z_{in_2})$	$f(z_{in_3})$	$f(z_{in_4})$	$f(z_{in_5})$	$f(z_{in_6})$
0,0178	0,53175	0,52042	0,19312	0,44241	0,25802

- Menghitung semua keluaran jaringan dengan Persamaan 2.6 dan menghitung fungsi aktivasi linier dengan Persamaan 2.7

$$y_{in_1} = 0,04324 + 0,0178 \times 0,08635 + 0,53175 \times 0,55617 + 0,52042 \times 0,36833 + 0,019312 + 0,42036 + 0,44241 \times 0,2534 + 0,25802 \times 0,31742 = 1,80734$$

$$y = f(y_{in_1}) = y_{in_1} = 1,80734$$

4. Melakukan perhitungan Nilai MAPE dengan Persamaan 2.17

$$MAPE = \left( \frac{1}{N} \sum \frac{|A_i - F_i|}{|A_i|} \right) \times 100\% \\ = \left( \frac{1}{10} \sum \left( \frac{|1,46778 - 1,80734|}{|1,46778|} + \frac{|1,35809 - 1,63634|}{|1,35809|} + \dots + \frac{|2,21999 - 0,94593|}{|2,21999|} \right) \right) \times 100\% = 2,31342\%$$

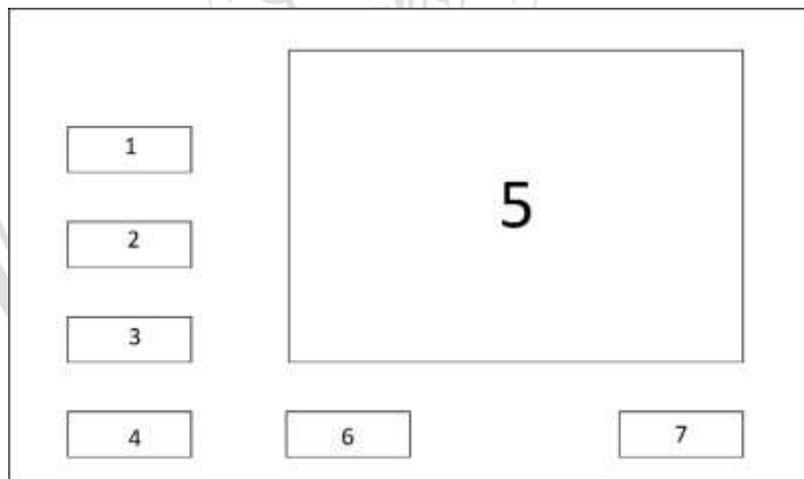
MAPE dari proses *testing* yang juga mewakili proses perhitungan manual adalah 2,31342%.

#### 4.4 Perancangan Antarmuka

Perancangan antarmuka dilakukan untuk membuat gambaran pertama antarmuka sistem yang akan dibuat. Perancangan antarmuka sistem pada penelitian ini terdiri halaman utama dan halaman kedua.

##### 4.4.1 Halaman utama

Perancangan halaman utama berisi kolom-kolom input parameter, tabel yang menampilkan fitur data variabel *Altman* dan juga berisi kolom *output* nilai MAPE hasil prediksi. Antarmuka halaman utama diilustrasikan pada Gambar 4.8.



Gambar 4.8 Ilustrasi antarmuka halaman utama

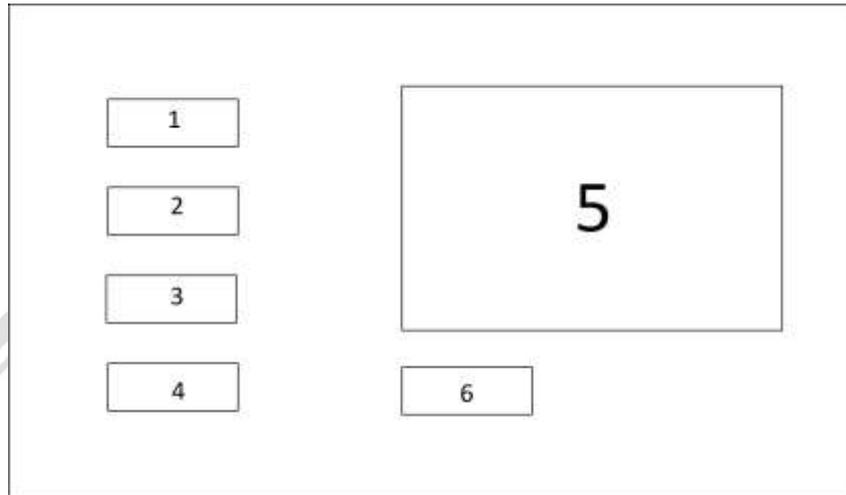
Keterangan untuk Gambar 4.8:

1. Kolom input nilai *learning rate* ( $\alpha$ ).
2. Kolom input jumlah iterasi.
3. Kolom input jumlah neuron pada *hidden layer*.
4. Tombol proses.
5. Tabel data.

6. Tombol *load data*.
7. Kolom nilai MAPE prediksi.

#### 4.1.1 Halaman Kedua

Perancangan halaman kedua berisi pengujian *k-fold cross validation* yang berisi kolom-kolom input parameter dan tabel untuk menampilkan MAPE pengujian *k-fold cross validation*. Antarmuka halaman kedua diilustrasikan pada Gambar 4.9.



Gambar 4.9 Ilustrasi antarmuka halaman kedua

Keterangan untuk Gambar 4.9:

1. Kolom input jumlah  $k$ .
2. Kolom input nilai *learning rate*.
3. Kolom input jumlah iterasi.
4. Kolom input jumlah neuron pada *hidden layer*.
5. Tombol proses.
6. Tabel *output* MAPE per *fold*.

#### 4.2 Perancangan Pengujian

Untuk mengetahui performa sistem yang akan dibuat maka dilakukan pengujian terhadap sistem, pengujian yang akan dilakukan terlebih dahulu dirancang untuk merencanakan parameter-parameter apa saja yang akan diuji pada sistem tersebut. Penggunaan metode penelitian mempengaruhi parameter yang akan diuji, dalam penelitian ini metode yang digunakan adalah *backpropagation*. Perancangan pengujian pada penelitian ini meliputi pengujian nilai *learning rate* ( $\alpha$ ), pengujian jumlah iterasi, pengujian jumlah neuron pada *hidden layer*. Setiap pengujian dilakukan dalam 10 kali percobaan, selanjutnya dihitung rata-rata pada masing-masing nilai parameter yang diujikan. Selanjutnya setiap parameter yang menghasilkan nilai MAPE terbaik digunakan sebagai parameter untuk melakukan *k-fold cross validation*. Berbeda dengan pengujian tiga parameter sebelumnya, *k-fold cross validation* dilakukan dalam 5 kali percobaan.

### 4.2.1 Pengujian Nilai *Learning Rate* ( $\alpha$ )

Pengujian jumlah nilai  $\alpha$  berfungsi untuk mengetahui fluktuasi nilai MAPE yang dihasilkan karena adanya perubahan nilai *learning rate*. Nilai *learning rate* yang digunakan berada pada rentang nilai 0,1 sampai 0,9, tetapi pada parameter jumlah iterasi dan jumlah neuron pada *hidden layer* ditetapkan pada nilai yang sama. Tabel 4.15 merepresentasikan hasil pengujian nilai *learning rate*.

**Tabel 4.15 Representasi hasil pengujian nilai *learning rate***

No	Nilai <i>learning rate</i> ( $\alpha$ )	Nilai MAPE percobaan ke-i										Rata-rata
		1	2	3	4	5	6	7	8	9	10	
1.	0,1											
2.	0,2											
3.	0,3											
4.	0,4											
5.	0,5											
6.	0,6											
7.	0,7											
8.	0,8											
9.	0,9											

### 4.2.2 Pengujian Jumlah Iterasi

Pengujian jumlah iterasi berfungsi untuk mengetahui fluktuasi nilai MAPE yang dihasilkan karena adanya perubahan jumlah iterasi. Nilai iterasi yang digunakan adalah nilai dengan kelipatan 100, yakni 100, 200, 300 hingga 1000, tetapi pada parameter nilai *learning rate* dan jumlah neuron pada *hidden layer* ditetapkan pada nilai yang sama. Tabel 4.16 merepresentasikan hasil pengujian jumlah iterasi.

**Tabel 4.16 Representasi hasil pengujian jumlah iterasi**

No	Jumlah iterasi	Nilai MAPE percobaan ke-i										Rata-rata
		1	2	3	4	5	6	7	8	9	10	
1.	100											
2.	200											
3.	300											
4.	400											
5.	500											
6.	600											
7.	700											
8.	800											
9.	900											
10.	1000											



### 4.2.3 Pengujian Jumlah Neuron pada *Hidden Layer*

Pengujian jumlah neuron pada *hidden layer* berfungsi untuk mengetahui fluktuasi nilai MAPE yang dihasilkan karena adanya perubahan jumlah neuron pada *hidden layer*. Akan tetapi, parameter nilai *learning rate* dan jumlah iterasi ditetapkan pada nilai yang sama. Tabel 4.17 merepresentasikan hasil pengujian jumlah neuron pada *hidden layer*.

**Tabel 4.17** Representasi hasil pengujian jumlah neuron pada *hidden layer*

No	Neuron pada <i>hidden layer</i>	Nilai MAPE percobaan ke-i										Rata-rata
		1	2	3	4	5	6	7	8	9	10	
1.	5											
2.	6											
3.	7											
4.	8											
5.	9											
6.	10											
7.	11											
8.	12											

### 4.1.1 Pengujian *Cross Validation*

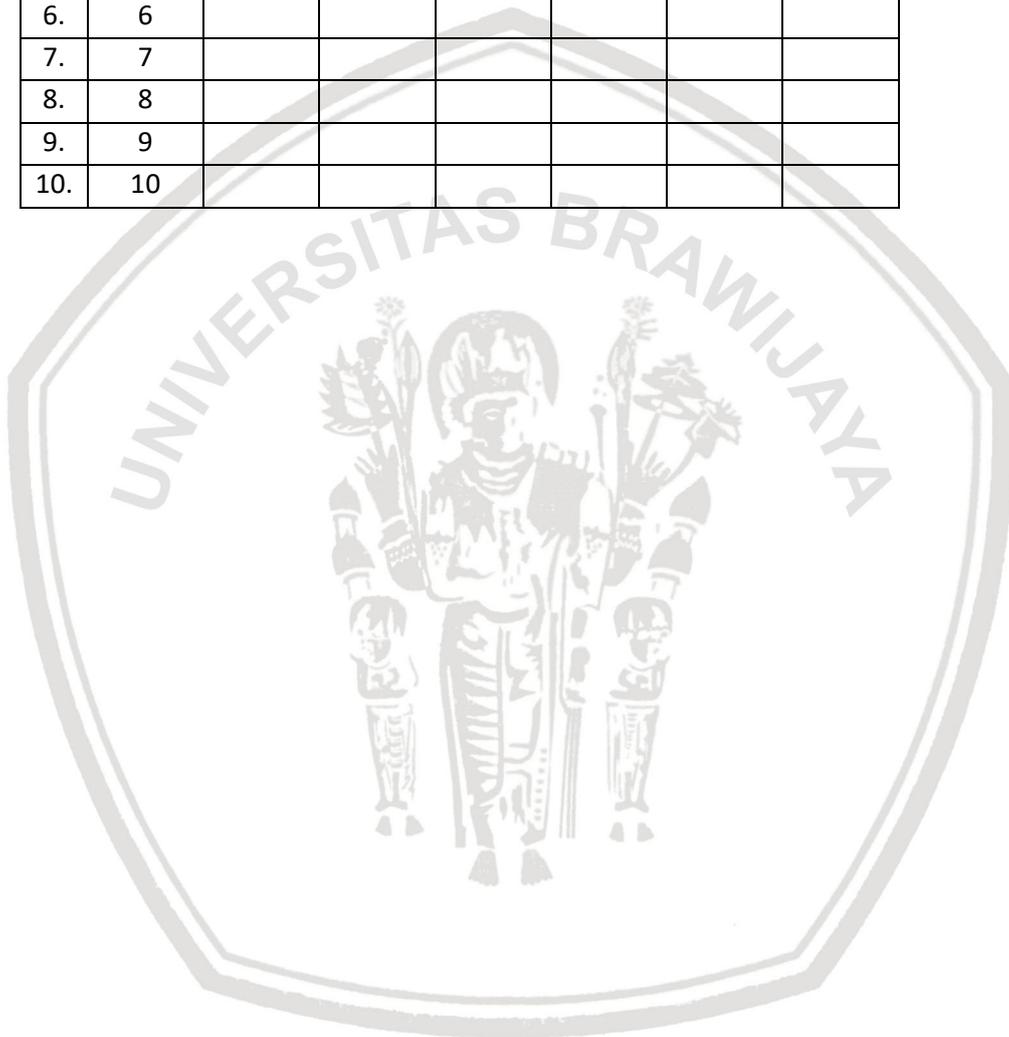
Pengujian *cross validation* atau validasi silang berfungsi untuk mengetahui performa jaringan saraf tiruan. Parameter-parameter yang digunakan adalah parameter terbaik dari pengujian sebelumnya dengan nilai  $k$  yang digunakan adalah  $k = 5$  dan  $k = 10$ . Selain itu, pada setiap *fold* data *training* dan data *testing* dibagi dalam rasio 90%:10%. Tabel 4.18 merepresentasikan hasil pengujian *5-fold cross validation* dan Tabel 4.19 merepresentasikan hasil pengujian *10-fold cross validation*.

**Tabel 4.18** Representasi hasil pengujian *5-fold cross validation*

No	<i>Fold</i> ke-	Nilai MAPE percobaan ke-i					Rata-rata
		1	2	3	4	5	
1.	1						
2.	2						
3.	3						
4.	4						
5.	5						

**Tabel 4.19** Representasi hasil pengujian *10-fold cross validation*

No	Fold ke-	Nilai MAPE percobaan ke-i					Rata-rata
		1	2	3	4	5	
1.	1						
2.	2						
3.	3						
4.	4						
5.	5						
6.	6						
7.	7						
8.	8						
9.	9						
10.	10						



## BAB 5 IMPLEMENTASI

Pada bab ini akan dijelaskan tentang proses implementasi berdasarkan perancangan yang sudah dibuat sebelumnya. Dalam bab ini terdiri dari dua implementasi, implementasi algoritme *backpropagation* dan implementasi antarmuka.

### 5.1 Implementasi Algoritme *Backpropagation*

Implementasi algoritme *backpropagation* dilakukan dalam beberapa proses yang diulang sejumlah iterasi untuk mendapatkan hasil konvergensi. Implementasi algoritme *backpropagation* terdiri dari proses inialisasi bobot dan bias, fase propagasi maju, fase propagasi mundur, fase perubahan bobot dan menghitung MAPE pada data uji.

#### 5.1.1 Implementasi Proses Inialisasi Bobot

Proses inialisasi bobot dilakukan pada masing-masing neuron dengan membangkitkan nilai antara (-0,5) sampai (0,5). Inialisasi dilakukan pada masing-masing lapisan dan bias. Parameter yang dibutuhkan untuk melakukan inialisasi bobot adalah banyaknya fitur input, jumlah neuron pada *hidden layer*, serta nilai maksimal dan nilai minimal pengacakan. Proses inialisasi bobot ditunjukkan pada Kode Program 5.1.

```
1 double[][] weightv(int input, int hidden) {
2     double weightv[][] = new double[input][hidden];
3     for (int i = 0; i < input; i++) {
4         for (int j = 0; j < hidden; j++) {
5             double weight = (Math.random() * 1) + (-0.5);
6             weightv[i][j] = weight;
7         }
8     }
9     return weightv;
10 }

11 double[] weightbiasv(int hidden) {
12     double weightbiasv[] = new double[hidden];
13     for (int i = 0; i < hidden; i++) {
14         double weight = (Math.random() * 1) + (-0.5);
15         weightbiasv[i] = weight;
16     }
17     return weightbiasv;
18 }

19 double[] weightw(int hidden) {
20     double weightw[] = new double[hidden];
21     for (int i = 0; i < hidden; i++) {
22         double weight = (Math.random() * 1) + (-0.5);
23         weightw[i] = weight;
24     }
25     return weightw;
26 }
```

```

27 double weightbiasw() {
28     double weightbiasw = 0;
29     for (int i = 0; i < 1; i++) {
30         double weight = (Math.random() * 1) + (-0.5);
31         weightbiasw = weight;
32     }
33     return weightbiasw;
34 }

```

**Kode Program 5.1 Implementasi inialisasi bobot**

Penjelasan dari Kode Program 5.1:

1. Baris 1-10 adalah proses inialisasi bobot antara input *layer* dan *hidden layer*.
2. Baris 11-18 adalah proses inialisasi bobot bias antara input *layer* dan *hidden layer*.
3. Baris 19-26 adalah proses inialisasi bobot antara *hidden layer* dan *output layer*.
4. Baris 27-34 adalah proses inialisasi bobot bias *hidden layer* dan *output layer*.

### 5.1.2 Implementasi Fase Propagasi Maju

Fase propagasi maju dilakukan untuk menghitung nilai keluaran pada *hidden layer* dan nilai keluaran pada *output layer*. Parameter yang digunakan dalam fase propagasi maju adalah data input, bobot, dan bobot bias. Proses implementasi fase propagasi maju ditunjukkan pada Kode Program 5.2.

```

1 double[] z(double[][] weightv, double[] biasv, double[]
  data) {
2     double[] z_in = new double[weightv[0].length];
3     double[] fz_in = new double[z_in.length];
4     double sigma;
5     for (int i = 0; i < weightv[0].length; i++) {
6         sigma = 0;
7         for (int j = 0; j < data.length; j++) {
8             sigma += data[j] * weightv[j][i];
9         }
10        z_in[i] = sigma + biasv[i];
11        fz_in[i] = z_in[i];
12    }
13    return fz_in;
14 }

15 double y(double[] weightw, double biasw, double[] z) {
16     double sigma = 0;
17     for (int i = 0; i < z.length; i++) {
18         sigma += z[i] * weightw[i];
19     }
20     sigma += biasw;
21     double y = sigma;
22     return y;
23 }

```

**Kode Program 5.2 Implementasi propagasi maju**

Penjelasan dari Kode Program 5.2:

1. Baris 2 dan 3 adalah inialisasi variabel untuk menampung nilai keluaran pada *hidden layer* ( $z_{in}$ ) dan nilai fungsi aktivasi  $f(z_{in})$ .
2. Baris 7-10 adalah proses menghitung nilai keluaran pada *hidden layer* ( $z_{in}$ ).
3. Baris 11 adalah proses menghitung nilai fungsi aktivasi keluaran pada *hidden layer*  $f(z_{in})$ .
4. Baris 17-20 adalah proses menghitung nilai keluaran pada *output layer* ( $y_{in}$ ).
5. Baris 21 adalah proses menghitung nilai fungsi aktivasi keluaran pada *output layer* ( $f(y_{in})$ ).

### 5.1.3 Implementasi Fase Propagasi Mundur

Implementasi fase propagasi mundur dilakukan untuk menghitung galat nilai keluaran *output layer*, menghitung perubahan bobot pada *output layer*, menghitung galat nilai keluaran *hidden layer*, dan menghitung perubahan bobot pada *hidden layer*. Proses implementasi fase propagasi mundur ditunjukkan pada Kode Program 5.3

```

1  double δk(double data_y, double t) {
2      double delta = (t - data_y) * 1;
3      return delta;
4  }
5  double[] delta_wkj(double[] z, double δk, double alfa)
6  {
7      double[] delta_wkj = new double[z.length];
8      for (int i = 0; i < delta_wkj.length; i++) {
9          delta_wkj[i] = δk * alfa * z[i];
10     }
11     return delta_wkj;
12 }
13 double delta_wk0(double δk, double alpha) {
14     double deltabias_w = δk * alpha;
15     return deltabias_w;
16 }
17 double[] δ_in(double w[], double z[], double δk) {
18     double δ_in[] = new double[z.length];
19     double δ[] = new double[z.length];
20     for (int i = 0; i < δ_in.length; i++) {
21         δ_in[i] = δk * w[i];
22     }
23     for (int i = 0; i < z.length; i++) {
24         δ[i] = δ_in[i] * 1;
25     }
26     return δ;
27 }
28 double[][] delta_vji(int input, double[] z, double[] δ,
29     double[] data, double alfa) {

```

```

28     double delta_vji[][] = new double[input][z.length];
29     for (int i = 0; i < input; i++) {
30         for (int j = 0; j < z.length; j++) {
31             delta_vji[i][j] = alfa * δ[j] * data[i];
32         }
33     }
34     return delta_vji;
35 }

36 double[] delta_vj0(double[] z, double alpha, double δ[])
37 {
38     double[] delta_vj0 = new double[z.length];
39     for (int i = 0; i < delta_vj0.length; i++) {
40         delta_vj0[i] = alpha * δ[i];
41     }
42     return delta_vj0;
43 }

```

### Kode Program 5.3 Implementasi propagasi mundur

Penjelasan dari Kode Program 5.3:

1. Baris 1-4 adalah proses menghitung galat nilai keluaran pada *output layer* ( $\delta_k$ ).
2. Baris 6 adalah inialisasi variabel *array* untuk menampung nilai perubahan bobot pada *output layer*.
3. Baris 5-11 adalah proses menghitung perubahan bobot pada *output layer* ( $\Delta w_k$ ).
4. Baris 12-15 adalah proses menghitung perubahan bobot bias pada *output layer* ( $\Delta w_{k0}$ ).
5. Baris 17 dan 18 adalah inialisasi variabel *array* untuk menampung nilai galat keluaran pada *hidden layer* dan nilai fungsi aktivasinya.
6. Baris 16-21 adalah proses menghitung galat nilai keluaran pada *hidden layer* ( $\delta_{in}$ ).
7. Baris 22-26 adalah proses menghitung nilai fungsi aktivasi galat nilai keluaran pada *hidden layer* ( $f(\delta_{in})$ ).
8. Baris 28 adalah inialisasi variabel *array* untuk menampung nilai perubahan bobot pada *hidden layer*.
9. Baris 27-35 adalah proses menghitung perubahan bobot pada *hidden layer* ( $\Delta v_{ji}$ ).
10. Baris 37 adalah inialisasi variabel *array* untuk menampung nilai perubahan bobot bias pada *hidden layer*.
11. Baris 36-42 adalah proses menghitung perubahan bobot bias pada *hidden layer* ( $\Delta v_{j0}$ ).

#### 5.1.4 Implementasi Fase Perubahan Bobot

Implementasi perubahan bobot dilakukan untuk mengubah bobot pada setiap neuron yang menuju *hidden layer* atau *output layer*. Proses implementasi fase perubahan bobot ditunjukkan pada Kode Program 5.4.

```

1  double[] update_wkj(double w[], double delta_w[]) {
2      double update_wkj[] = new double[w.length];
3      for (int i = 0; i < update_wkj.length; i++) {
4          update_wkj[i] = w[i] + delta_w[i];
5      }
6      return update_wkj;
7  }

8  double update_bias_w(double bias_w, double
9      delta_bias_w) {
10     double update_bias_w = bias_w + delta_bias_w;
11     return update_bias_w;
12 }

12 double[][] update_vji(double v[][], double delta_v[][])
13 {
14     double[][] update_vji = new double[v.length]
15     [v[0].length];
16     for (int i = 0; i < update_vji.length; i++) {
17         for (int j = 0; j < update_vji[0].length; j++) {
18             update_vji[i][j] = v[i][j] + delta_v[i][j];
19         }
20     }
21     return update_vji;
22 }

21 double[] update_bias_v(double bias_v[], double
22     delta_bias_v[]) {
23     double update_bias_v[] = new double[bias_v.length];
24     for (int i = 0; i < update_bias_v.length; i++) {
25         update_bias_v[i] = bias_v[i] + delta_bias_v[i];
26     }
27     return update_bias_v;
28 }

```

**Kode Program 5.4 Implementasi perubahan bobot**

Penjelasan dari Kode Program 5.4:

1. Baris 1-7 adalah proses mengubah bobot yang menuju *output layer* ( $w_{kj}$ (baru)).
2. Baris 8-11 adalah proses mengubah bobot bias yang menuju *output layer* ( $w_{k0}$ (baru)).
3. Baris 12-20 adalah proses mengubah bobot yang menuju *hidden layer* ( $v_{ji}$ (baru)).
4. Baris 21-27 adalah proses mengubah bobot bias yang menuju *hidden layer* ( $v_{j0}$ (baru)).

### 5.1.5 Implementasi Proses Menghitung MAPE

Implementasi proses menghitung MAPE dilakukan untuk mengetahui nilai rata-rata dari keseluruhan persentase selisih galat antara data aktual dengan data hasil prediksi. Proses implementasi menghitung MAPE ditunjukkan pada Kode Program 5.5.

```
1 double mape(double[] t, double[] y) {
2     double total = 0;
3     for (int i = 0; i < t.length; i++) {
4         double temp = Math.abs((t[i] - y[i]) / t[i]);
5         total += temp;
6     }
7     double mape = (total / t.length) * 100;
8     return mape;
9 }
```

### Kode Program 5.5 Implementasi proses menghitung MAPE

Penjelasan dari Kode Program 5.4:

1. Baris 2 adalah inialisasi variabel untuk menampung absolut selisih nilai prediksi dan nilai aktual pada setiap iterasi.
2. Baris 3-6 adalah proses iterasi penghitungan absolut selisih nilai prediksi dan nilai aktual.
3. Baris 7 adalah proses penghitungan MAPE.

## 5.2 Implementasi Antarmuka

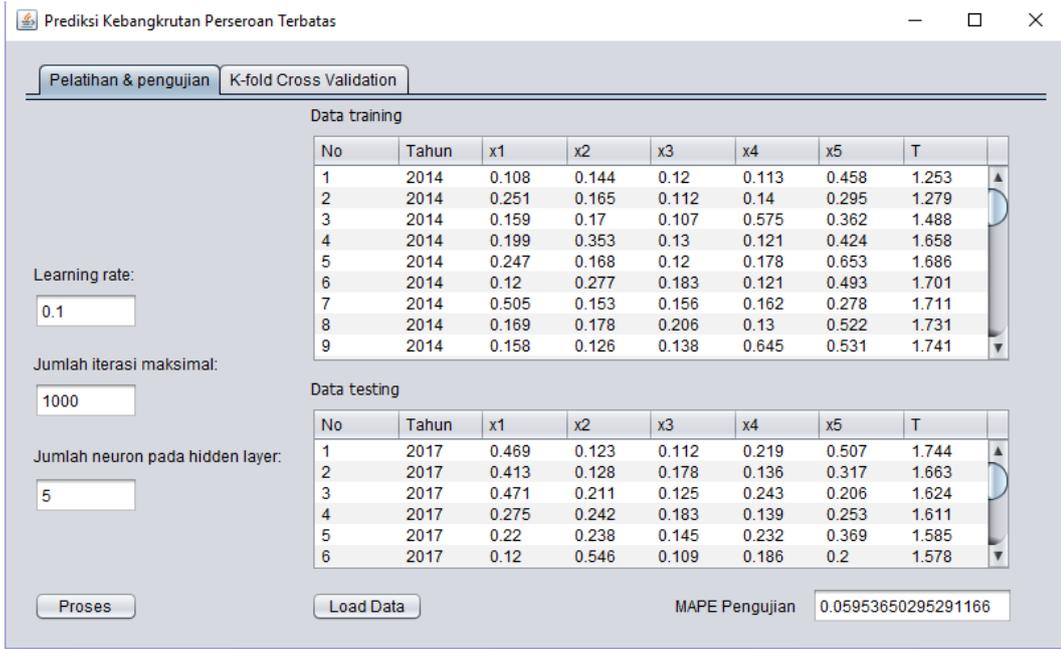
Implementasi antarmuka adalah hasil dari implementasi perancangan antarmuka yang sudah dilakukan pada Bab sebelumnya. Antarmuka merupakan media perantara interaksi antara pengguna dan sistem. Implementasi antarmuka dilakukan menggunakan *JFrame* dalam bahasa pemrograman *java*. Antarmuka yang sudah diimplementasikan terdiri dari dua halaman yakni halaman utama dan halaman pengujian *K-Fold Cross Validation*.

### 5.2.1 Halaman Utama

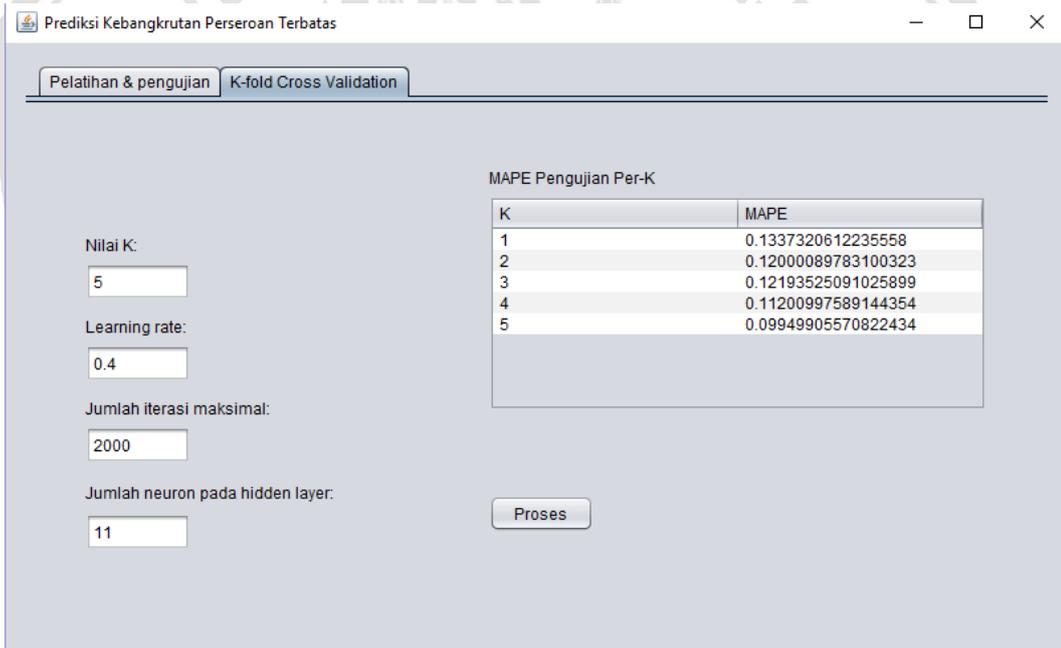
Implementasi halaman utama berisi kolom input parameter-parameter yang digunakan dalam *backpropagation*. Parameter-parameter tersebut adalah nilai *learning rate*, jumlah iterasi dan jumlah neuron pada *hidden layer*. Selain itu, halaman utama juga menampilkan data latih dan data uji yang akan digunakan serta MAPE pengujian. Gambar 5.1 merupakan hasil implementasi antarmuka halaman utama.

### 5.2.2 Halaman Kedua

Implementasi halaman kedua berisi hasil pengujian *cross validation* dan kolom input parameter yang digunakan untuk pelatihan *backpropagation*. Gambar 5.2 merupakan hasil implementasi halaman kedua.



Gambar 5.1 Antarmuka halaman utama



Gambar 5.2 Antarmuka halaman kedua



## BAB 6 PENGUJIAN DAN ANALISIS

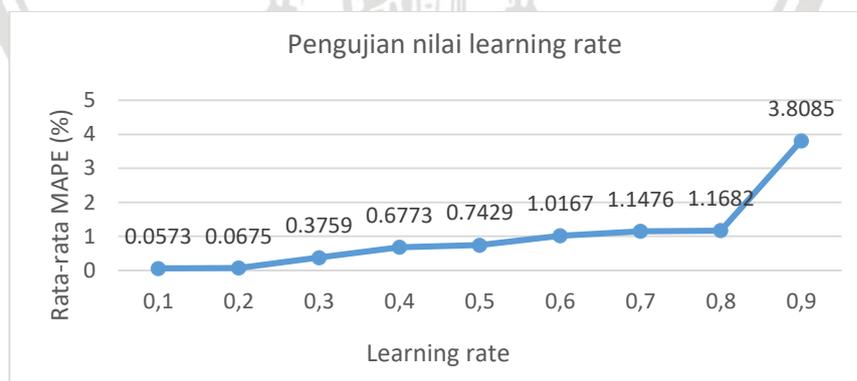
Pada bab ini dijelaskan pengujian sistem yang telah dilaksanakan berdasarkan perancangan pengujian yang sudah disusun sebelumnya. Pengujian yang dilakukan terdiri dari pengujian nilai *learning rate*, pengujian jumlah iterasi, pengujian jumlah neuron pada *hidden layer* dan pengujian cross validation. Selain itu, dijelaskan juga analisis dari setiap pengujian yang sudah dilakukan.

### 6.1 Pengujian Nilai *Learning Rate*

Pengujian nilai *learning rate* bertujuan untuk mengetahui fluktuasi nilai MAPE yang dihasilkan karena adanya perubahan nilai *learning rate*. Dalam pengujian ini digunakan parameter *learning rate* antara 0,1 sampai 0,9. Selanjutnya parameter lain yang digunakan 100 iterasi dan 5 neuron pada *hidden layer*. Tabel 6.1 merupakan tabulasi hasil pengujian nilai *learning rate* dan hasil dari pengujian *learning rate* disimpulkan dalam Gambar 6.1.

**Tabel 6.1 Hasil pengujian nilai *learning rate***

No	Nilai <i>learning rate</i> ( $\alpha$ )	Nilai MAPE percobaan ke-i										Rata-rata
		1	2	3	4	5	6	7	8	9	10	
1.	0,1	0.056	0.055	0.059	0.059	0.057	0.058	0.058	0.057	0.057	0.058	0.057
2.	0,2	0.071	0.073	0.065	0.069	0.059	0.065	0.067	0.079	0.065	0.062	0.068
3.	0,3	0.327	0.303	0.306	0.558	0.439	0.269	0.674	0.446	0.262	0.174	0.376
4.	0,4	0.753	0.76	0.691	0.884	0.687	0.542	0.638	0.635	0.776	0.407	0.677
5.	0,5	0.441	1.078	0.887	1.078	1.037	0.365	0.786	0.349	0.865	0.544	0.743
6.	0,6	1.219	1.151	1.079	1.114	0.971	0.844	0.952	0.885	0.935	1.018	1.017
7.	0,7	1.355	1.051	1.129	0.999	1.031	1.187	1.237	1.183	0.958	1.346	1.148
8.	0,8	1.327	0.798	1.218	0.853	0.946	1.031	1.203	1.132	1.861	1.313	1.168
9.	0,9	2.892	2.148	1.898	4.746	3.923	4.348	5.247	1.882	5.779	5.222	3.809



**Gambar 6.1 Grafik pengaruh *learning rate* terhadap nilai MAPE**

Berdasarkan Tabel 6.1, dapat diketahui jika nilai *learning rate* dapat mempengaruhi nilai MAPE yang dihasilkan jaringan. Hal tersebut dibuktikan dengan adanya fluktuasi secara signifikan terhadap nilai MAPE seperti pada

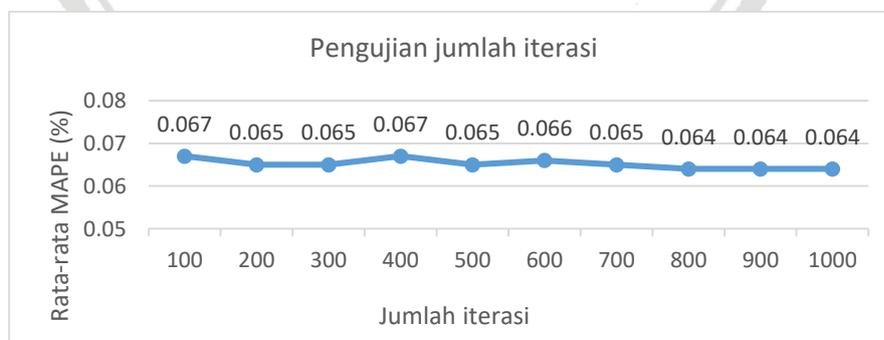
Gambar 6.1 pada setiap perubahan nilai *learning rate* yang digunakan. Dalam pengujian ini, MAPE terbaik didapatkan pada nilai *learning rate* 0,2 dengan rata-rata 0,062%. Sebaliknya, MAPE terbesar didapatkan pada penggunaan nilai *learning rate* sebesar 0,9 dengan rata-rata nilai MAPE sebesar 3,7895%. Nilai MAPE akan meningkat saat nilai *learning rate* juga meningkat karena nilai *learning rate* yang besar akan mempercepat proses pembelajaran namun membuat bobot mengalami perubahan secara drastis yang menyebabkan selisih galat jaringan tidak terkontrol.

## 6.2 Pengujian Jumlah Iterasi

Pengujian jumlah iterasi bertujuan untuk mengetahui fluktuasi nilai MAPE yang dihasilkan karena adanya perubahan jumlah iterasi. Jumlah iterasi yang digunakan adalah bilangan dengan kelipatan 100, dari 100 hingga 1000. Parameter lain yang digunakan adalah nilai *learning rate* sebesar 0,2 sebagai asumsi nilai terbaik yang menghasilkan rata-rata MAPE terbaik pada pengujian *learning rate* sebelumnya dan jumlah neuron pada *hidden layer* sebanyak 5. Tabel 6.2 merupakan tabulasi hasil pengujian jumlah iterasi dan hasil dari pengujian jumlah iterasi disimpulkan dalam Gambar 6.2.

**Tabel 6.2 Hasil pengujian jumlah iterasi**

No	Jumlah iterasi	Nilai MAPE percobaan ke-i										Rata-rata
		1	2	3	4	5	6	7	8	9	10	
1.	100	0.058	0.066	0.066	0.067	0.06	0.066	0.071	0.073	0.071	0.074	0.067
2.	200	0.061	0.055	0.071	0.062	0.069	0.066	0.064	0.057	0.067	0.068	0.064
3.	300	0.079	0.066	0.06	0.062	0.068	0.059	0.069	0.063	0.06	0.059	0.065
4.	400	0.063	0.064	0.061	0.08	0.077	0.076	0.062	0.056	0.067	0.069	0.067
5.	500	0.064	0.063	0.058	0.065	0.057	0.064	0.07	0.069	0.075	0.054	0.064
6.	600	0.071	0.056	0.055	0.082	0.066	0.06	0.078	0.066	0.063	0.064	0.066
7.	700	0.067	0.065	0.07	0.058	0.06	0.056	0.081	0.056	0.071	0.068	0.065
8.	800	0.064	0.065	0.07	0.057	0.067	0.078	0.06	0.059	0.065	0.056	0.064
9.	900	0.061	0.078	0.064	0.06	0.055	0.062	0.064	0.07	0.066	0.062	0.064
10.	1000	0.057	0.073	0.064	0.062	0.065	0.059	0.063	0.078	0.059	0.06	0.064



**Gambar 6.2 Grafik pengaruh jumlah iterasi terhadap nilai MAPE**

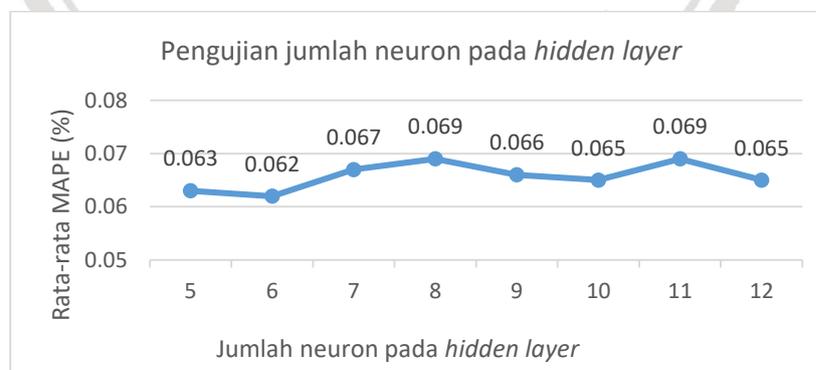
Berdasarkan Tabel 6.2, dapat diketahui jika jumlah iterasi dapat mempengaruhi nilai MAPE yang dihasilkan oleh jaringan tetapi tidak signifikan. Nilai MAPE yang dihasilkan dalam pengujian ini masih tetap dalam rentang nilai 0,06% meski jumlah iterasi berubah-ubah seperti ditunjukkan pada Gambar 6.2. Pada pengujian ini, nilai MAPE terbaik didapatkan dari 800 sampai 1000 iterasi dengan rata-rata 0,064%. Nilai MAPE terbesar dihasilkan saat menggunakan 100 dan 400 iterasi dengan rata-rata 0,067%. Pengujian jumlah iterasi ini mencapai konvergensi pada 800 iterasi, sehingga saat menggunakan 900 dan 1000 iterasi rata-rata MAPE yang dihasilkan tetap konstan.

### 6.3 Pengujian Jumlah Neuron pada *Hidden Layer*

Pengujian jumlah neuron pada *hidden layer* bertujuan untuk mengetahui fluktuasi nilai MAPE yang dihasilkan karena adanya perubahan jumlah neuron pada *hidden layer*. Jumlah neuron yang digunakan dalam pengujian ini adalah 5 sampai 12, dengan parameter nilai *learning rate* sebesar 0,2 sebagai asumsi nilai terbaik yang menghasilkan rata-rata MAPE terbaik pengujian nilai *learning rate*. Selanjutnya parameter jumlah iterasi sebanyak 1000 sebagai asumsi jumlah iterasi terbesar yang menghasilkan rata-rata MAPE terbaik pada pengujian jumlah iterasi. Tabel 6.3 merupakan hasil pengujian jumlah neuron pada *hidden layer* dan hasil pengujian jumlah neuron pada *hidden layer* disimpulkan pada Gambar 6.3.

Tabel 6.3 Hasil pengujian jumlah neuron pada *hidden layer*

No	Neuron pada <i>hidden layer</i>	Nilai MAPE percobaan ke-i										Rata-rata
		1	2	3	4	5	6	7	8	9	10	
1.	5	0.061	0.078	0.052	0.06	0.068	0.056	0.07	0.061	0.058	0.064	0.063
2.	6	0.057	0.06	0.074	0.077	0.058	0.058	0.053	0.068	0.061	0.054	0.062
3.	7	0.066	0.066	0.062	0.082	0.066	0.064	0.054	0.071	0.065	0.07	0.067
4.	8	0.058	0.079	0.071	0.06	0.076	0.067	0.087	0.054	0.077	0.064	0.069
5.	9	0.068	0.08	0.064	0.062	0.072	0.073	0.063	0.059	0.062	0.059	0.066
6.	10	0.076	0.057	0.065	0.082	0.066	0.056	0.06	0.058	0.069	0.06	0.065
7.	11	0.061	0.069	0.07	0.068	0.063	0.075	0.086	0.059	0.066	0.078	0.069
8.	12	0.078	0.055	0.062	0.067	0.063	0.071	0.06	0.059	0.067	0.069	0.065



Gambar 6.3 Grafik pengaruh jumlah neuron pada *hidden layer* terhadap nilai MAPE

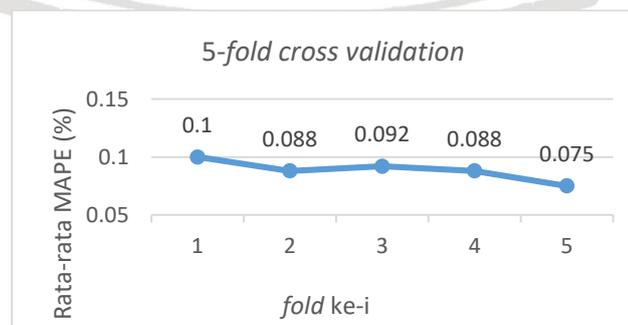
Berdasarkan Tabel 6.3, dapat diketahui bahwa jumlah neuron pada *hidden layer* dapat mempengaruhi nilai MAPE yang dihasilkan jaringan, tetapi tidak terlalu signifikan. Ini dibuktikan dengan nilai MAPE yang dihasilkan jaringan tetap dalam rentang nilai 0,06% seperti ditunjukkan pada Gambar 6.3. Pada pengujian ini, nilai MAPE terbaik didapatkan dari 6 jumlah neuron pada *hidden layer* dengan rata-rata 0,062%. Sebaliknya, nilai MAPE terbesar didapatkan pada 8 dan 11 jumlah neuron pada *hidden layer* dengan rata-rata 0,069%. Jumlah neuron pada *hidden layer* mempengaruhi nilai  $y_{in_1}$  yang dihasilkan oleh jaringan. Semakin banyak neuron pada *hidden layer* maka nilai  $y_{in_1}$  yang dihasilkan juga akan semakin besar, hal ini tentu berbanding lurus dengan persentase galat atau MAPE yang dihasilkan jaringan.

### 6.4 K-fold Cross Validation

Pengujian *k-fold cross validation* dilakukan untuk mengevaluasi performa metode yang digunakan dan pengujiannya dilakukan terhadap setiap *fold*. Jumlah *fold* yang digunakan dalam pengujian ini adalah  $k = 5$  dan  $k = 10$ , sedangkan parameter-parameter yang digunakan adalah parameter terbaik pada pengujian-pengujian sebelumnya. Parameter nilai *learning rate* yang digunakan sebesar 0,2 dengan asumsi nilai terbaik yang dihasilkan oleh pengujian nilai *learning rate*, asumsi jumlah iterasi terbaik sebanyak 1000 yang dihasilkan oleh pengujian jumlah iterasi dan asumsi jumlah neuron terbaik sebanyak 6 yang dihasilkan oleh pengujian jumlah neuron pada *hidden layer*. Tabel 6.4 merupakan hasil *5-fold cross validation* dan Tabel 6.5 merupakan hasil *10-fold cross validation*. Selanjutnya hasil dari masing-masing pengujian disimpulkan dalam Gambar 6.4 dan Gambar 6.5.

**Tabel 6.4 Hasil pengujian 5-fold cross validation**

No	Fold ke-	Nilai MAPE percobaan ke-i					Rata-rata per fold
		1	2	3	4	5	
1.	1	0.098	0.103	0.103	0.094	0.104	0.1
2.	2	0.093	0.094	0.086	0.088	0.077	0.088
3.	3	0.087	0.097	0.106	0.077	0.092	0.092
4.	4	0.086	0.082	0.093	0.095	0.083	0.088
5.	5	0.079	0.071	0.081	0.074	0.072	0.075
Rata-rata hasil pengujian 5-fold cross validation							0.089

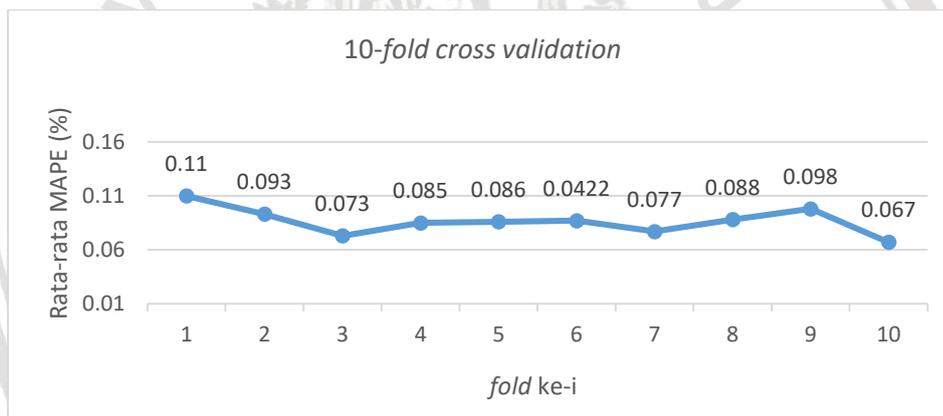


**Gambar 6.4 Grafik pengujian 5-fold cross validation terhadap nilai MAPE**

Seperti ditunjukkan pada Gambar 6.4, MAPE terbaik pada *5-fold cross validation* didapatkan dari *fold* ke-5 dengan rata-rata 0,075%. Lalu berdasarkan Tabel 6.4, dari keseluruhan hasil pengujian *5-fold cross validation* menghasilkan rata-rata MAPE sebesar 0,089%.

**Tabel 6.5 Hasil pengujian *10-fold cross validation***

No	Fold ke-	Nilai MAPE percobaan ke-i					Rata-rata per fold
		1	2	3	4	5	
1.	1	0.103	0.117	0.108	0.114	0.107	0.11
2.	2	0.088	0.093	0.088	0.095	0.099	0.093
3.	3	0.069	0.079	0.075	0.069	0.072	0.073
4.	4	0.082	0.089	0.089	0.087	0.079	0.085
5.	5	0.085	0.087	0.08	0.091	0.087	0.086
6.	6	0.093	0.086	0.083	0.084	0.09	0.087
7.	7	0.076	0.076	0.078	0.077	0.077	0.077
8.	8	0.095	0.083	0.088	0.091	0.084	0.088
9.	9	0.103	0.091	0.102	0.101	0.095	0.098
10.	10	0.066	0.074	0.067	0.059	0.068	0.067
Rata-rata hasil pengujian <i>10-fold cross validation</i>							0.086

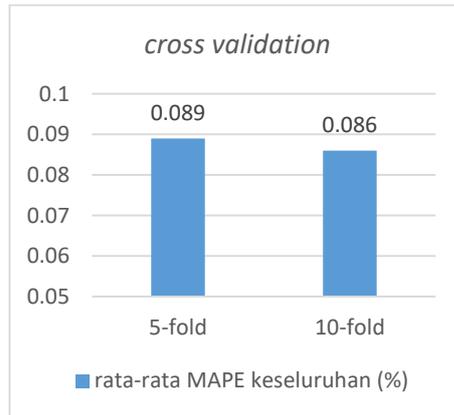


**Gambar 6.5 Grafik pengujian *10-fold cross validation* terhadap nilai MAPE**

Sementara itu seperti ditunjukkan pada Gambar 6.5, MAPE terbaik pada *10-fold cross validation* didapatkan dari *fold* ke-3 dengan rata-rata 0,073%. Lalu berdasarkan Tabel 6.5, dari keseluruhan hasil pengujian *10-fold cross validation* menghasilkan rata-rata MAPE sebesar 0,086%. Gambar 6.6 adalah grafik perbandingan rata-rata MAPE yang dihasilkan oleh pengujian *5-fold cross validation* dan *10-fold cross validation*.

Dari gambar 6.6, dapat disimpulkan bahwa antara *5-fold* dan *10-fold cross validation* memiliki rata-rata MAPE keseluruhan yang tidak jauh berbeda. Hasil tersebut menunjukkan bahwa tidak ada data dengan rentang nilai yang terlalu jauh yang dianggap sebagai pencilan data.





**Gambar 6.6** Grafik perbandingan rata-rata MAPE keseluruhan pengujian *5-fold* dan *10-fold cross validation*

## 6.5 Analisis Pengujian

Analisis dari seluruh pengujian yang sudah dilakukan, prediksi kebangkrutan perseroan terbatas menggunakan metode *backpropagation* mendapatkan MAPE terbaik dengan rata-rata 0,062% pada pengujian dengan parameter *learning rate* sebesar 0,2, jumlah iterasi sebanyak 1000 dan jumlah neuron pada *hidden layer* sebanyak 6. Karena nilai MAPE tersebut <10% dan mendekati nilai 0%, maka dapat disimpulkan jika persentase galat yang dihasilkan jaringan sangat kecil. Hasil tersebut menunjukkan bahwa nilai prediksi kebangkrutan sangat mendekati nilai aktual dan kebangkrutan dapat diklasifikasikan mendekati kebenaran.

Selanjutnya pada pengujian *5-fold cross validation* menghasilkan rata-rata nilai MAPE keseluruhan pengujian sebesar 0,089%, sedangkan pada pengujian *10-fold cross validation* menghasilkan rata-rata nilai MAPE keseluruhan pengujian sebesar 0,086%. Hasil dari pengujian-pengujian yang telah dilakukan menunjukkan bahwa prediksi kebangkrutan perseroan terbatas menggunakan metode *backpropagation* termasuk dalam kriteria prediksi dengan akurasi yang sangat baik.

## BAB 7 KESIMPULAN

### 7.1 Kesimpulan

Kesimpulan yang dapat diambil dalam keseluruhan proses penelitian ini antara lain:

1. Laporan keuangan perusahaan yang didapatkan berupa data dalam satuan rupiah yang tidak mungkin digunakan secara langsung, agar dapat digunakan sebagai fitur input maka data keuangan tersebut diubah menjadi variabel rasio keuangan *Altman*. Data keuangan yg digunakan adalah total aktiva; modal kerja; laba ditahan; pendapatan sebelum pajak dan bunga; nilai pasar ekuitas; dan nilai buku dari hutang; dan penjualan. Kemudian data keuangan tersebut diubah menjadi fitur input berupa lima variabel *Altman*.
2. Dalam penelitian ini, metode *backpropagation* digunakan untuk memprediksi kebangkrutan dengan menggunakan lima variabel *Altman* sebagai fitur input dan hasil dari perhitungan *Z-Score* sebagai target *output* jaringan.
3. Dari keseluruhan pengujian, didapatkan nilai MAPE terbaik dengan rata-rata 0,062% pada pengujian dengan parameter *learning rate* sebesar 0,2, jumlah iterasi sebanyak 1000 dan jumlah neuron pada *hidden layer* sebanyak 6. Hasil tersebut menunjukkan bahwa nilai prediksi kebangkrutan mendekati nilai target asli dan kebangkrutan dapat diklasifikasikan mendekati kebenaran. Hal tersebut dikarenakan galat jaringan yang dihasilkan sangat kecil sehingga nilai MAPE yang dihasilkan mendekati nilai 0%. Selanjutnya pengujian *5-fold cross validation* dan pengujian *10-fold cross validation* menghasilkan rata-rata nilai MAPE keseluruhan pengujian sebesar 0,089% dan 0,086%. Hasil dari pengujian-pengujian yang telah dilakukan menunjukkan bahwa prediksi kebangkrutan perseroan terbatas menggunakan metode *backpropagation* termasuk dalam kriteria prediksi dengan akurasi yang sangat baik.

### 7.2 Saran

Saran yang dapat diberikan dalam proses pengembangan penelitian ini pada masa mendatang antara lain:

1. Penambahan jumlah perusahaan yang laporan tahunannya digunakan. Sehingga nanti dapat juga dilakukan pengelompokan *testing* terhadap perseroan terbatas pada masing-masing subsektor yang digunakan.

## DAFTAR PUSTAKA

- Altman, E. I. (1968). Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. *Journal of Finance Vol. 23, No. 4*, 589-609.
- Asyhadie, H. Z., & Sutrisno, B. (2012). *Hukum Perusahaan dan Kepailitan*. Jakarta: Erlangga.
- Brigham, E. F., & Ehrhardt, M. C. (2005). *Financial Management: Theory and Practice*. Boston: South-Western Cengage.
- Brigham, E. F., & Gapenski, L. C. (1993). *Intermediate Financial Management*. Florida: Dryden Press.
- Fausett, V. L. (1994). *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. New Jersey: Prentice-Hall.
- Glueck, W. F., & Jauch, L. R. (1984). *Business Policy and Strategic Management*. New York: McGraw-Hill.
- Hamid, N. A., Nawati, N. M., Ghazali, R., & Salleh, M. N. (2011). Accelerating Learning Performance of Back Propagation Algorithm by Using Adaptive Gain Together with Adaptive Momentum and Adaptive Learning Rate on Classification Problems. *The 2nd International Conference Ubiquitous Computing and Multimedia Applications* (hal. 559-570). London: Springer-Verlag.
- Hanafi, M. (2004). *Manajemen Keuangan*. Yogyakarta: BPFE - Universitas Gadjah Mada.
- Hanke, J. E., Reitsch, A. G., & Wichern, D. W. (1999). *Business Forecasting 7th Edition*. New Jersey: Prentice-Hall.
- Hsu, L. C., & Wang, C. H. (2008). Applied Multivariate Forecasting Model to Tourism Industry. *Original Scientific Paper Vol. 5, No. 2*, 156-172.
- Kim, S., & Kim, H. (2016). A New Metric of Absolute Percentage Error for Intermittent Demand Forecast. *International Journal of Forecasting*, 669-679.
- Lestari, D. Y. (2017). Jaringan syaraf Tiruan untuk Prediksi Penjualan Jamur Menggunakan Algoritma Backpropagation. *Jurnal ISD Vol. 2 No. 1*, 40-46.
- Malaka, A. S., & Hartojo. (2014). Model Prediksi Kepailitan Bank Umum Di Indonesia Menggunakan Algoritma Backpropagation. *Jurnal Ilmu Manajemen Vol. 2, No. 4*, 1714-1724.
- Manan, B. (1995). *Kekuasaan Kehakiman Republik Indonesia*. Bandung: Pusat Penerbitan Universitas, LPPM, Universitas Islam Bandung.
- Ridhani, M. N., Putri, R. R., & Sutopo. (2017). Peramalan Dosis Pupuk Berdasarkan Karakteristik dan Lingkungan Tanaman Jeruk Siam Menggunakan Metode Backpropagation. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol. 1, No. 11*, 1214-1223.

- Tambunan, R. W., Dwiatmanto, & Endang, W. N. (2015). Analisis Prediksi Kebangkrutan Perusahaan Dengan Menggunakan Metode Altman Z-Score (Studi Pada Subsektor Rokok Yang Listing dan Perusahaan Delisting di Bursa Efek Indonesia Tahun 2009-2013). *Jurnal Administrasi Bisnis, Vol. 2 No. 1*, 1-11.
- Wahyuni, I., Adam, N. R., Mahmudy, W. F., & Iriany, A. (2017). Modeling Backpropagation Neural Network for Rainfall Prediction in Tengger East Java. *International Conference on Sustainable Information Engineering and Technology (SIET)*, 170-175.
- Wasis. (1997). *Pengantar Ekonomi Perusahaan*. Bandung: Alumni.
- Werdiastu, D., Ratnawati, D. E., & Rahayudi, B. (2018). Estimasi Hasil Produksi Benih Berdasarkan Karakteristik Tanaman Kenaf Menggunakan Metode Backpropagation (Studi Kasus: Balai Tanaman Pemanis dan Serat Kota Malang). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol. 2, No. 11*, 5201-5209.
- Weston, J. F., & Copeland, T. E. (1991). *Managerial Finance 8th Edition*. London: Dryden Press.

