

# PERAMALAN PRODUKSI PADI DI INDONESIA MENGUNAKAN JARINGAN SYARAF TIRUAN BILANGAN KOMPLEKS

Anifuddin Azis

Departemen Ilmu Komputer dan Elektronika, Fakultas MIPA, Universitas Gadjah Mada  
Sekip Utara, Bulaksumur, Yogyakarta  
E-mail : anifudin@ugm.ac.id

## ABSTRAK

Ketahanan pangan nasional tetap membutuhkan beras dalam memenuhi kecukupan pangan, namun demikian tekanan terhadap peningkatan produksi padi terkendala oleh permasalahan konversi lahan pertanian, kerusakan jaringan irigasi, perubahan iklim, dan serangan hama penyakit. Kemampuan memprediksi masa yang akan datang akan membuat petani bisa mengambil keputusan yang tepat dalam mengantisipasi keadaan di masa depan. Jaringan Syaraf Tiruan (JST) telah banyak digunakan dalam prediksi, termasuk dalam bidang pertanian. JST bilangan kompleks (JSBK) memiliki struktur pemrosesan bilangan kompleks sehingga memiliki kemampuan pelatihan yang lebih baik, kemampuan generalisasi yang lebih baik, konvergensi yang lebih cepat, kompleksitas yang lebih baik, dan data pelatihan yang lebih sedikit. Pada JST dengan arsitektur *multi-layer perceptron* (MLP) biasa yang bernilai real, nilai dari parameter yang ada, yaitu elemen dari matriks bobot dan vektor bias, dinyatakan dalam bilangan real, sedangkan pada JST bernilai kompleks, nilai dari parameter yang ada dinyatakan dalam bilangan kompleks. Jaringan Syaraf Tiruan (JST) bilangan kompleks dengan algoritma Backpropagation, umumnya digunakan dalam data citra atau suara, namun pada penelitian ini digunakan untuk memprediksi produksi padi tahun 2015 di 34 provinsi Indonesia. Data produksi padi diperoleh dari Badan Pusat Statistik (BPS), sedangkan data curah hujan, suhu, kelembaban, tekanan udara diperoleh dari Badan Meteorologi, Klimatologi, dan Geofisika (BMKG). Atribut masukan pada pelatihan JST pada penelitian ini adalah data iklim dari 34 provinsi di Indonesia dari tahun 2011-2015, yaitu nilai dari : curah hujan, kelembaban, rata-rata suhu, dan rata-rata tekanan udara. Atribut keluaran JST adalah nilai produktifitas padi. Berdasarkan hasil pengujian, JST Backpropagation bilangan kompleks berhasil memprediksi produksi padi tahun 2015 dengan sangat baik pada 32 provinsi (jika tidak menyertakan data provinsi Kepulauan Riau dan DKI Jakarta) dengan rata-rata akurasi 98,9 %.

**Kata Kunci:** prediksi, produksi padi, multi-layer perceptron, backpropagation, bilangan kompleks.

## 1. PENDAHULUAN

Ketahanan pangan nasional tetap membutuhkan beras dalam memenuhi kecukupan pangan, namun demikian tekanan terhadap peningkatan produksi padi terkendala oleh permasalahan konversi lahan pertanian, kerusakan jaringan irigasi, perubahan iklim, dan serangan hama penyakit. Kebijakan diversifikasi pangan tetap menjadi prioritas dalam pengendalian konsumsi beras. Dalam kurun waktu yang cukup panjang selama 1961-2013, fluktuasi produksi padi nasional mengalami pertumbuhan yang bervariasi. Target pertumbuhan produksi padi harus memenuhi kebutuhan nasional, apabila kurang mencukupi maka pemerintah melakukan impor beras. Secara umum pertumbuhan produksi padi mengalami pertumbuhan di atas pertumbuhan penduduk kecuali pada periode 1961-1966, 1998-2004, dan 2005-2007 (Kementerian Pertanian, 2015).

Kemampuan memprediksi masa yang akan datang akan membuat petani bisa mengambil keputusan yang tepat dalam mengantisipasi keadaan di masa depan. Jaringan Syaraf Tiruan (JST) menawarkan kemampuan yang baik untuk mesin pembelajaran, prediksi, termasuk

juga prediksi dalam bidang pertanian (Jabjone, Wannasang, 2014).

JST bilangan kompleks(JSBK) memiliki struktur pemrosesan bilangan kompleks sehingga memiliki kemampuan pelatihan yang lebih baik, kemampuan generalisasi yang lebih baik, konvergensi yang lebih cepat, kompleksitas yang lebih baik, dan data pelatihan yang lebih sedikit (El-Telbany, Refat, 2016).

Pada umumnya JSBK banyak diterapkan pada data citra dan suara. Seperti yang dilakukan oleh Amilia, dkk. (2015) yang menggunakan JSBK untuk melakukan pengenalan jenis kelamin pada citra wajah. Dataset yang digunakan adalah dataset wajah FERET, sedangkan ciri yang digunakan adalah Histogram of Oriented Gradients. Penelitian ini juga membandingkan hasil pada JSBK dengan JST bilangan real (JSBR). Rata-rata akurasi yang diperoleh dari JSBK adalah 80,2%, sedangkan rata-rata akurasi yang diperoleh dari JSBR adalah 78,2%. Sementara Virtue, dkk. (2017) dalam penelitiannya mengaplikasikan JSBK pada sistem *Magnetic Resonance Imaging* (MRI) untuk sidik jari. Sistem MRI bekerja dengan cara memindai bagian yang akan dipindai (dalam hal ini sidik jari) dengan menggunakan sejumlah

parameter tertentu, hingga kemudian didapat hasil yang selanjutnya akan diubah menjadi sinyal bernilai kompleks. Dengan JSBK, sinyal ini kemudian menjalani proses pelatihan, untuk kemudian dipetakan kembali ke parameter aslinya. Kesimpulan akhir dari penelitian ini adalah JSBK lebih akurat dibandingkan JSBR dalam sistem MRI sidik jari dengan sinyal yang bernilai kompleks. Penelitian lain yang menggunakan data citra, dilakukan oleh Popa (2017), yang melakukan penelitian untuk pengenalan citra menggunakan dataset MNIST dan CIFAR-10 menggunakan metode jaringan syaraf konvolusional bernilai kompleks. Untuk memperoleh fitur yang ada pada citra, penelitian ini menggunakan metode konvolusi bernilai kompleks. Hasil dari penelitian ini adalah, dengan arsitektur dan jumlah parameter yang sama, jaringan syaraf konvolusional bernilai kompleks bekerja lebih baik dibandingkan dengan jaringan syaraf konvolusional bernilai real. Kemudian JSBK dengan arsitektur yang berbeda digunakan oleh Arjovsky, dkk. (2015) yang menggunakan Recurrent Neural Network dengan matriks bobot yang bernilai kompleks, sedangkan Matsuda dan Murase (2016) melakukan klasifikasi menggunakan JSBK dengan satu layer dan digabungkan dengan algoritma *Synthetic Minority Over sampling Technique* (SMOTE). Algoritma SMOTE digunakan pada dataset yang terdapat ketidakseimbangan pada distribusi kelasnya (imbalanced dataset), sehingga terdapat kelas minoritas dan kelas mayoritas. Klasifikasi dilakukan pada lima imbalan *ced dataset* dari UCI LIVSVM. Hasil akhir dari penelitian ini adalah, JSBK yang dilengkapi dengan algoritma SMOTE secara umum lebih unggul dibandingkan JSBR pada beberapa parameter evaluasi yang ditentukan.

Prediksi produksi padi telah dilakukan pada penelitian sebelumnya oleh Azis (2018) menggunakan JSBR menggunakan data-data curah hujan, suhu, kelembaban, tekanan udara, luas lahan, dan produksi padi di Indonesia, dengan tingkat akurasi 86%. Pada penelitian ini digunakan JST Backpropagation bilangan kompleks (JSBK) menggunakan data-data yang sama Data-data tersebut berupa data numerik. Yang diperoleh dari Badan Meteorologi, Klimatologi, dan Geofisika (BMKG), dan Badan Pusat Statistik (BPS).

## 2. RUANG LINGKUP

Dalam penelitian ini permasalahan mencakup:

1. Penentuan model JST Backpropagation bilangan kompleks yang tepat untuk peramalan produksi padi di Indonesia.
2. Data yang digunakan adalah data-data faktor-faktor curah hujan, suhu, kelembaban, tekanan udara, luas lahan, dan data produksi di Indonesia tahun 2011-2015.
3. Hasil yang direncanakan dalam penelitian ini adalah model JST Backpropagation bilangan kompleks yang memiliki akurasi tinggi dalam peramalan produksi padi di Indonesia.

## 3. BAHAN DAN METODE

JSBK menggunakan bilangan kompleks dalam operasinya. Pada bagian ini dijelaskan aljabar bilangan kompleks, kemudian dilanjutkan dengan fungsi bernilai kompleks, *phase-encoding*, *multi-layer perceptron* bernilai kompleks, *backpropagation* dengan *stochastic gradient descent* pada JSBK, fungsi aktivasi pada JSBK

### 3.1 Aljabar Bilangan Kompleks

Setiap bilangan kompleks  $z$  dapat ditulis dalam bentuk persamaan (1) :

$$z = a + bi \quad (1)$$

dengan  $a$  dan  $b$  adalah suatu bilangan real, dan  $i$  adalah bilangan imajiner, yaitu bilangan yang memenuhi  $i^2 = -1$ . Selanjutnya, nilai  $a$  disebut sebagai bagian real dari  $z$ , ditulis  $\text{Re}(z)$ , dan nilai  $b$  disebut sebagai bagian imajiner dari  $z$ , ditulis  $\text{Im}(z)$ . Setiap bilangan kompleks dapat dipandang sebagai sebuah titik pada suatu bidang koordinat dua dimensi dengan sumbu  $x$  untuk bagian real dan sumbu  $y$  untuk bagian imajiner, yang selanjutnya disebut dengan bidang kompleks.

Misalkan  $z = a + bi$ . Terdapat beberapa operasi pada  $z$  sebagai berikut, yaitu konjugat, modulus, dan argumen seperti pada persamaan (2), persamaan (3), dan persamaan (4) :

$$1. \text{Konjugat } z, \text{ yaitu } \bar{z} = a - bi \quad (2)$$

$$2. \text{Modulus atau magnitudo } z, \text{ yaitu } |z| = \sqrt{a^2 + b^2} \quad (3)$$

Modulus dari  $z$  dapat dipandang sebagai jarak dari titik  $(a, b)$  ke titik pusat pada bidang kompleks, yaitu  $0 + 0i$ .

3. Argumen, atau fase dari  $z$ , ditulis dengan  $\arg(z)$ , yaitu

$$\arg(z) = \arctan\left(\frac{b}{a}\right) \quad (4)$$

$\arg(z)$  dapat dipandang sebagai sudut yang dibentuk oleh sumbu  $x$ , dengan segmen garis yang dibentuk oleh titik  $(0, 0)$  dan titik  $(a, b)$  pada bidang kompleks.

Selain dapat direpresentasikan dengan bentuk  $z = a + bi$ , suatu bilangan kompleks juga dapat direpresentasikan dengan bentuk seperti pada persamaan (5) :

$$z = re^{i\theta} \quad (5)$$

dengan  $r$  adalah

modulus dari  $z$ , dan  $\theta$  adalah argumen utama dari  $z$ . Representasi lain yang dapat digunakan adalah  $z = r(\cos \theta + i \sin \theta)$ . Untuk selanjutnya,  $\cos x + i \sin x$  ditulis dengan  $\text{cis}(x)$ .

Untuk operasi aritmatika bilangan kompleks, misalkan  $z = a + bi$  dan  $w = c + di$ . Operasi penjumlahan seperti pada persamaan (6), pengurangan menggunakan persamaan (7), perkalian sesuai persamaan (8), dan pembagian menggunakan persamaan (9) :

$$1. z + w = (a + c) + (b + d)i \quad (6)$$

$$2. z - w = (a - c) + (b - d)i \quad (7)$$

$$3. zw = (ac - bd) + (ad + bc)i \quad (8)$$

$$4. \frac{z}{w} = \frac{\overline{zw}}{ww} = \frac{ac + bd}{c^2 + d^2} + \frac{bc - ad}{c^2 + d^2}i \quad (9)$$

Operasi-operasi ini akan digunakan pada backpropagation pada JSBK.

### 3.2 Fungsi Bernilai Kompleks

Misalkan  $z = x + iy$  adalah variabel kompleks dan  $f : C \rightarrow C$  adalah fungsi bernilai kompleks, yaitu fungsi yang memetakan suatu bilangan kompleks ke bilangan kompleks. Fungsi  $f(z) = f(x + iy)$  juga dapat dinyatakan dalam bentuk  $f(z) = u(x, y) + iv(x, y)$  dengan  $u$  dan  $v$  masing-masing adalah fungsi dengan pemetaan dari  $R^2$  ke  $R$ . Dengan kata lain, fungsi  $u$  dan  $v$  keduanya adalah fungsi dalam  $x$  dan  $y$ . Fungsi  $u$  dan  $v$  juga dapat ditulis berturut-turut dengan  $\text{Re}(f)$  dan  $\text{Im}(f)$ . Untuk derivatif dari fungsi bernilai kompleks, misalkan  $f$  dan  $g$  adalah fungsi

bernilai kompleks, menurut Trabelsi, dkk. (2018), nilai dari  $\frac{\partial f}{\partial g}$  seperti pada persamaan (10) :

$$\frac{\partial f}{\partial g} = \frac{\partial f}{\partial \text{Re}(g)} + i \frac{\partial f}{\partial \text{Im}(g)} \quad (10)$$

Sedangkan untuk menghitung  $\frac{\partial f}{\partial g}$  dengan aturan rantai,

misalkan  $h$  adalah fungsi kompleks, dapat dihitung menggunakan persamaan (11) :

$$\frac{\partial f}{\partial g} = \frac{\partial f}{\partial h} \frac{\partial h}{\partial g} = \frac{\partial f}{\partial \text{Re}(h)} \frac{\partial \text{Re}(h)}{\partial g} + \frac{\partial f}{\partial \text{Im}(h)} \frac{\partial \text{Im}(h)}{\partial g} \quad (11)$$

Persamaan (1) dan (2) akan digunakan untuk menghitung rumus perubahan parameter pada algoritma backpropagation.

### 3.3 Phase-encoding

Untuk input atau target yang nilainya berupa bilangan real, nilai tersebut harus diubah terlebih dahulu ke bilangan kompleks. Salah satu cara untuk mengubah nilai ini adalah dengan *phase-encoding*. Secara formal, *phase-encoding* adalah fungsi  $p : R \rightarrow C$ . Untuk setiap input bernilai real  $x$ , input ini dipetakan ke suatu

bilangan real lain  $\phi = p(x)$  terlebih dahulu, yang kemudian nilai  $\phi$  digunakan sebagai fase dari bilangan kompleks  $z$  yang akan dijadikan input, yaitu  $z = \text{cis}(\phi)$ . Salah satu cara *phase-encoding* yang ada adalah yang digunakan dalam Amin dan Murase (2009) seperti pada persamaan (12) :

$$\phi = p(x) = \left( \frac{x-a}{b-a} \right) \pi \quad (12)$$

dengan  $a$  dan  $b$  berturut-turut adalah nilai terkecil dan terbesar pada vektor input. Nilai  $\pi$  dapat diganti dengan  $2\pi$  atau  $\pi/2$ . Namun, pada penelitian tersebut, hasil terbaik didapat ketika menggunakan nilai  $\pi$ .

Phase-encoding tersebut memungkinkan adanya dua atau lebih input bilangan real yang diubah ke satu bilangan kompleks yang sama. Akibatnya, input bilangan real tersebut akan diperlakukan sebagai sebuah bilangan kompleks yang sama, padahal input tersebut memberikan informasi yang berbeda. Untuk itu, *phase-encoding* yang digunakan haruslah merupakan fungsi bijektif, sehingga satu input bilangan real dipetakan ke tepat satu bilangan kompleks. Penelitian ini mengusulkan penggunaan *phase-encoding* baru. Fungsi yang diusulkan sebagai *phase-encoding* ini adalah fungsi bijektif  $p : R \rightarrow C$  seperti pada persamaan (13) :

$$z = p(x) = \text{cis} \left( \frac{2\pi}{1 + \exp(-x)} \right) \quad (13)$$

dengan  $x$  adalah input bilangan real dan  $z$  adalah bilangan kompleks hasil *phase-encoding* dari  $x$  yang akan dijadikan input pada JSBK.

### 3.4 JST Bilangan Kompleks (JSBK)

Pada JST dengan arsitektur *multi-layer perceptron* (MLP) biasa yang bernilai real, nilai dari parameter yang ada, yaitu elemen dari matriks bobot dan vektor bias, dinyatakan dalam bilangan real, sedangkan pada MLP bernilai kompleks, nilai dari parameter yang ada dinyatakan dalam bilangan kompleks.

#### 3.4.1 Algoritma Backpropagation pada JSBK

Terdapat dua proses dalam algoritma backpropagation, yaitu *forward-pass* dan *backward-pass*. *Forward-pass* dilakukan untuk menghitung vektor output  $x^{(L)}$ . *Forward-pass* dalam JSBK sama seperti pada JSBR. Perbedaannya hanya terletak pada bilangan yang dioperasikan. Parameter pada JSBK yang berupa bilangan kompleks dioperasikan mengikuti operasi yang telah dijelaskan sebelumnya.

Cara menghitung vektor output  $x^{(L)}$  dilakukan sebagai berikut. Untuk setiap  $k$  di  $\{1, 2, 3, \dots, L\}$ , didefinisikan vektor baru  $s^{(k)}$ , seperti pada persamaan (14) :

$$s^{(k)} = \left(W^{(k)}\right)^T \cdot x^{(k-1)} + b^{(k)} \quad (14)$$

Dengan vektor  $x$  adalah vektor input dari *layer* sebelumnya,  $W$  adalah matriks bobot dan  $b$  adalah vektor bias, dan semua nilainya dalam bilangan kompleks. Vektor output  $x$  dapat dihitung menggunakan persamaan (15) :

$$x^k = f_k(s^{(k)}) \quad (15)$$

Dengan  $f$  adalah fungsi aktivasi yang akan dijelaskan sebelumnya.

Tujuan algoritma *Backpropagation* adalah menentukan konfigurasi parameter pada matriks bobot  $W$  dan vektor bias  $b$ , sehingga nilai dari *loss function* atau fungsi *loss* menjadi sekecil mungkin, yaitu mendekati nol. Setelah melakukan proses *forward-pass*, proses *backward-pass* dilakukan untuk memperbarui konfigurasi nilai parameter tersebut, dengan menggunakan *loss function* dan nilai dari vektor output  $x^{(L)}$ , yang sudah dihitung pada proses *forward-pass* sebelumnya. Fungsi *loss* secara formal adalah fungsi  $E : \mathbb{R}^{n^L} \times \mathbb{R}^{n^L} \rightarrow \mathbb{R}$ . Salah satu fungsi *loss* yang dapat digunakan pada JSBK, mengacu pada Popa (2017), adalah seperti pada persamaan (16) :

$$E(x^{(L)}, t) = \frac{1}{2} \sum_{k=1}^{n^L} (x_k^{(L)} - t_k) \overline{(x_k^{(L)} - t_k)} \quad (16)$$

dengan  $t$  adalah vektor target atau label sebenarnya.

### 3.4.2 Stochastic Gradient Descent

*Stochastic gradient descent* (SGD) adalah salah satu metode optimasi yang dapat digunakan untuk permasalahan meminimumkan nilai suatu fungsi, dalam hal ini fungsi yang dimaksud adalah fungsi *loss*. Metode ini memperbarui nilai seluruh parameter setiap kali data baru diinputkan, dengan memanfaatkan *gradien* atau turunan dari fungsi *loss* terhadap parameter tersebut. Satu kali memperbarui nilai parameter disebut dengan satu iterasi. Untuk setiap  $k$  di  $\{1, 2, 3, \dots, L\}$ , matriks bobot

$\mathbf{b}^{(k)}$  dan vektor bias  $\mathbf{W}^{(k)}$  diperbarui sesuai persamaan (17) dan (18) :

$$\mathbf{b}^{(k)} = \mathbf{b}^{(k)} - \alpha \Delta \mathbf{b}^{(k)} \quad (17)$$

$$\mathbf{W}^{(k)} = \mathbf{W}^{(k)} - \alpha \Delta \mathbf{W}^{(k)} \quad (18)$$

dengan  $\alpha$  adalah nilai *learning rate* dan notasi  $\Delta A$  menyatakan  $\partial E / \partial A$ . Rumus untuk  $\Delta \mathbf{b}^{(k)}$  dan  $\Delta \mathbf{W}^{(k)}$  adalah seperti pada persamaan (19) dan (20) (Popa, 2017) :

$$\Delta \mathbf{b}^{(k)} = \begin{cases} (\mathbf{W}^{(k+1)})^H \circ \overline{f'(s^{(k)})}, & k < L \\ (x^{(k)} - t) \circ f'(s^{(k)}), & k = L \end{cases} \quad (19)$$

$$\Delta \mathbf{W}^{(k)} = \Delta \mathbf{b}^{(k)} (x^{(k-1)})^H \quad (20)$$

dimana  $\mathbf{A}^H$  menyatakan konjugat transpos dari matriks  $\mathbf{A}$ , dan operasi  $\mathbf{A} \circ \mathbf{B}$  menyatakan operasi perkalian tiap entri dengan indeks yang sama dari matriks  $\mathbf{A}$  dan matriks  $\mathbf{B}$ .

### 3.4.3 Fungsi Aktivasi pada JSBK

Fungsi aktivasi pada JSBK tentu berbeda dengan JSBR, karena terdapat bagian imajiner pada bilangan kompleks. Jika fungsi pada JSBR adalah pemetaan dari  $\mathbb{R}$  ke  $\mathbb{R}$ , fungsi aktivasi pada JSBK adalah pemetaan dari  $\mathbb{C}$  ke  $\mathbb{C}$ . Pada penelitian ini fungsi aktivasi yang digunakan adalah yang diusulkan oleh Guberman (2016), yaitu fungsi aktivasi ReLU yang sudah dimodifikasi seperti pada persamaan (21) :

$$f(z) = \begin{cases} z, & \text{Arg}(z) \in \left[0, \frac{\pi}{2}\right] \\ 0, & \text{selainnya} \end{cases} \quad (21)$$

Perhatikan pula bahwa jika  $z$  real, maka fungsi ini adalah fungsi ReLU, yaitu  $f(z) = \max(0, z)$ .

## 3.5 Data Penelitian

Langkah pertama dalam penelitian ini adalah pengumpulan data. Data produksi padi menggunakan data dari BPS tahun 1993-2015. Sedangkan data iklim di Indonesia menggunakan data dari BMKG antara tahun 2000-2015. Untuk keperluan penelitian ini, data yang digunakan adalah data antara 2011-2015.

Data produksi padi terdiri dari data produksi padi 34 propinsi di Indonesia. Sedangkan data iklim yang diperoleh dari BMKG berdasarkan data dari setiap stasiun BMKG, yang berbeda jumlahnya di setiap propinsi, sehingga yang digunakan adalah nilai rata-rata di setiap propinsi. Pada tahap pemrosesan awal data, data-data iklim yang kosong diisi dengan nilai rata-rata data per propinsi.

Atribut masukan pada pelatihan JST pada penelitian ini adalah data iklim dari 34 propinsi di Indonesia dari tahun 2011-2015, yaitu nilai dari : curah hujan, kelembaban, rata-rata suhu, dan rata-rata tekanan udara. Sedangkan atribut keluaran JST adalah nilai produktifitas padi.

## 4. PEMBAHASAN

Pada tahap pemrosesan data, ditentukan data *input* untuk JST yang berjumlah 49, yaitu data :

1. Suhu minimum tahun 2011 - 2015.
2. Suhu rata-rata 2011 - 2015.
3. Suhu maksimum tahun 2011 - 2015
4. Tekanan udara tahun 2011 - 2015,
5. Penyinaran matahari tahun 2011 - 2015,
6. Kecepatan angin tahun 2011-2015,
7. Kelembaban tahun 2011 - 2015,
8. Curah hujan tahun 2011 - 2015,

9. Jumlah hari hujan tahun 2011 - 2015, yaitu banyak hari yang tercatat hujan pada provinsi dan tahun tersebut,
10. Produksi padi 2011 - produksi 2015, yaitu produksi padi yang tercatat pada provinsi dan tahun tersebut (ton),  
dengan target *output* adalah produksi padi 2015.

Kemudian dilakukan proses normalisasi pada data agar berada pada rentang  $[0,1]$ , menggunakan persamaan (21) :

$$x' = \frac{x - a}{b - a}$$

(21)

Dengan :

- x adalah nilai data yang akan dinormalisasi
- a adalah nilai minimum pada suatu atribut
- b adalah nilai maksimum pada suatu atribut
- x' adalah nilai hasil normalisasi

Untuk setiap sampel data, setelah dinormalisasi, setiap sampel tersebut dikenai operasi *phase-encoding*, yaitu proses mengubah setiap entri pada sampel tersebut dari bilangan real ke bilangan kompleks.

Setelah dilakukan pra-pemrosesan pada data selanjutnya dilakukan proses pelatihan dengan parameter sebagai berikut : Banyak epoch adalah 500.000 dan *Learning rate* bernilai 1.7.

Setelah didapat bobot pada model dari proses pelatihan, model ini digunakan untuk memprediksi produksi padi di setiap provinsi pada tahun 2015.

Hasil prediksi ini dibandingkan dengan produksi padi yang sebenarnya. Untuk mengukur seberapa dekat hasil prediksi dengan produksi yang sebenarnya, digunakan ukuran *Mean Absolute Error* sesuai persamaan (22) :

$$MAE = \frac{1}{34} \sum_{k=1}^{34} p_k - y_k$$

(22)

dengan:

- $p_k$  adalah hasil prediksi produksi padi pada tahun 2015 dari sampel (provinsi) ke-k
- $y_k$  adalah produksi padi yang sebenarnya pada tahun 2015 dari sampel (provinsi) ke-k

Tabel 1 memperlihatkan hasil prediksi JSTBK produksi padi 2015 pada setiap propinsi.

**Tabel 1. Hasil prediksi JSTBK**

Provinsi	Produksi Padi 2015	Prediksi Produksi	Selisih Absolut	Selisih Relatif (%)
Sumatera Utara	4044829	4039923	4906	0.12
Sumatera Barat	2550609	2535681	14928	0.59
Riau	393917	391077	2840	0.72
Jambi	541486	537782	3704	0.68
Sumatera Selatan	4247922	4227847	20075	0.47
Bengkulu	578654	574758	3896	0.67
Lampung	3641895	3623522	18373	0.5
Kepulauan Bangka Belitung	27068	32787	5719	21.13
Kepulauan Riau	959	18063	17104	1783.52
DKI Jakarta	6361	16299	9938	156.23
Jawa Barat	11373144	11368866	4278	0.04
Jawa Tengah	11301422	11299698	1724	0.02
DI Yogyakarta	945136	941677	3459	0.37
Jawa Timur	13154967	13142221	12746	0.1
Banten	2188996	2184965	4031	0.18
Bali	853710	852117	1593	0.19
Nusa Tenggara Barat	2417392	2410016	7376	0.31
Nusa Tenggara Timur	948088	942191	5897	0.62
Kalimantan Barat	1275707	1267539	8168	0.64
Kalimantan Tengah	893202	892116	1086	0.12
Kalimantan Selatan	2140276	2127742	12534	0.59
Kalimantan Timur	408782	408129	653	0.16
Kalimantan Utara	112102	112159	57	0.05
Sulawesi Utara	674169	669914	4255	0.63
Sulawesi Tengah	1015368	1012222	3146	0.31
Sulawesi Selatan	5471806	5453064	18742	0.34
Sulawesi Tenggara	660720	659106	1614	0.24
Gorontalo	331220	330979	241	0.07
Sulawesi Barat	461844	460832	1012	0.22
Maluku	117791	117056	735	0.62
Maluku Utara	75265	73047	2218	2.95
Papua Barat	30219	31115	896	2.97
Papua	181769	181071	698	0.38

Dari nilai selisih relatif ini, prediksi produksi padi yang paling mendekati produksi sebenarnya adalah prediksi produksi padi pada provinsi Jawa Tengah, yang memiliki selisih relatif sebesar 0,02%. Prediksi produksi padi yang paling jauh dari produksi sebenarnya adalah prediksi produksi padi pada provinsi Kepulauan Riau, yang nilai selisih relatifnya mencapai 1783,52% dan rata-rata selisih relatif seluruh Indonesia adalah 59,9%. Jika data prediksi Kepulauan Riau dan DKI Jakarta tidak disertakan, maka rata-rata selisih relatif adalah 1,1 %.

## 5. KESIMPULAN

1. Pada proses pembentukan model JSBK pada penelitian ini diperoleh jumlah *epoch* optimal adalah 500 dengan *learning rate* 1,7. Berdasarkan model JSBK tersebut, dapat terlihat JSTBK dapat memprediksi produksi padi tahun 2015 dengan sangat baik, kecuali pada provinsi Kepulauan Riau dan DKI Jakarta, sehingga bisa disimpulkan JSBK berhasil digunakan pada data numerik, yaitu data-data curah hujan, kelembaban, rata-rata suhu, dan rata-rata tekanan udara tahun 201-2015 untuk memprediksi produksi padi tahun 2015. Bahkan tingkat akurasi yang diperoleh lebih baik daripada menggunakan JSBR. Tingkat akurasi pada penelitian ini dengan JSBK mencapai 98,9%, sedangkan pada penelitian sebelumnya dengan JSBR hanya mencapai 86%.

Prediksi produksi padi yang paling mendekati produksi sebenarnya adalah prediksi produksi padi pada provinsi Jawa Tengah, yang memiliki selisih relatif sebesar 0,02%. Prediksi produksi padi yang paling jauh dari produksi sebenarnya adalah prediksi produksi padi pada provinsi Kepulauan Riau, yang nilai selisih relatifnya mencapai 1783,52% dan rata-rata selisih relatif seluruh Indonesia adalah 59,9%. Jika data prediksi Kepulauan Riau dan DKI Jakarta tidak disertakan, maka rata-rata selisih relatif adalah 1,1 % .

## 6. SARAN

Untuk penelitian berikutnya bisa dicobakan algoritma JSBK menggunakan beberapa variasi fungsi aktifasi atau JSBK pada model JST yang lain seperti LVQ (Learning Vector Quantization LVQ memiliki proses perhitungan yang lebih sederhana sehingga waktu komputasinya lebih kecil, meskipun secara umum tingkat akurasi lebih rendah daripada Backpropagation.

## 7. DAFTAR PUSTAKA

Amilia, S., Sulistiyo, M. D., dan Dayawati, R. N. 2015.Face image-based gender recognition using complex-valued neural network. 3rd International

Conference on Information and Communication Technology (ICoICT). Kuta.

Arjovsky, M., Shah, A., dan Bengio, Y. 2015. Unitary Evolution Recurrent Neural Networks, <http://arxiv.org/abs/1511.06464>, diakses 3 Mei 2019.

Azis, A. 2018. Peramalan Produksi Padi di Indonesia Menggunakan Jaringan Syaraf Tiruan. Laporan Penelitian FMIPA UGM.

El-Telbany, M., Refat, S., 2016, Complex-Valued Neural Networks Training : A Particel Swarm Optimazion Strategy, IJACSA (International Journal of Advance Computer Science and Applications) Vol 7 No 1

Guberman, N., 2016, On Complex Valued Convolutional Neural Networks, <ht-tp://arxiv.org/abs/1602.09046>, diakses 2 Mei 2019.

Jabjone, S., Wannasang, S., 2014, Decision Support System Using Neural Network to Predict Rice Production in Phimi District, Thailand, *International Journal of Computer and Electrical Engineering*, Vol 6, NO. 2, April 2014

Kementerian Pertanian. 2015 Rencana Strategis Kementerian Pertanian Tahun 2015-2019

Matsuda, K. dan Murase, K. 2016. Single-Layered Complex-Valued Neural Network with SMOTE for Imbalanced Data Classification, 2016 Joint 8th International Conference on Soft Computing and Intelligent Systems (SCIS) and 17th International Symposium on Advanced Intelligent Systems (ISIS). Hokkaido

Popa, C. 2017. Complex-Valued Convolutional Neural Networks for Real-Valued Image Classification. International Joint Conference on Neural Networks (IJCNN). Anchorage.

Trabelsi, C., Bilaniuk, O., Serdyuk, D., Subramanian, S., Santos, J. F., Mehri, S., Rostamzadeh, N., Bengio, Y., dan Pal, C. J., 2018, Deep Complex Networks, <ht-tp://arxiv.org/abs/1705.09792>, diakses 2 Mei 2019.

Virtue, P., Yu, S. X., dan Lustig, M. 2017. Better Than Real: Complex-Valued Neural Nets for MRI Fingerprinting. IEEE International Conference on Image Processing (ICIP). Beijing.

## UCAPAN TERIMA KASIH

Terima kasih saya ucapkan kepada Departemen Ilmu Komputer dan Elektronika Fakultas MIPA UGM atas bantuan dana pada penelitian ini.