Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.202072236
Vol. 7, No. 5, Agustus 2020, hlm. 727-736 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 30/E/KPT/2018 e-ISSN: 2528-6579

MODEL PREDIKSI INTERAKSI SENYAWA DAN PROTEIN UNTUK *DRUG REPOSITIONING* MENGGUNAKAN *DEEP SEMI-SUPERVISED LEARNING*

Larasati*[1], Wisnu Ananta Kusuma[2] , Annisa[3]

[1,2,3]Departemen Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor
[2]Pusat Studi Biofarmaka Tropika, Institut Pertanian Bogor
Email: [1]larasati_ayudia@apps.ipb.ac.id, [2]ananta@apps.ipb.ac.id, [3]annisa@apps.ipb.ac.id
*Penulis Korespondensi

(Naskah masuk: 18 Juli 2019, diterima untuk diterbitkan: 29 Agustus 2019)

Abstrak

Drug repositioning adalah penggunaan senyawa obat yang sudah lolos uji sebelumnya untuk mengatasi penyakit baru selain penyakit awal obat tersebut ditujukan. *Drug repositioning* dapat dilakukan dengan memprediksi interaksi senyawa obat dengan protein penyakit yang bereaksi positif. Salah satu tantangan dalam prediksi interaksi senyawa dan protein adalah masalah ketidakseimbangan data. *Deep semi-supervised learning* dapat menjadi alternatif untuk menangani model prediksi dengan data yang tidak seimbang. Proses *pre-training* berbasis *unsupervised learning* pada *deep semi-supervised learning* dapat merepresentasikan input dari *unlabeled data* (data mayoritas) dengan baik dan mengoptimasi inisialisasi bobot pada *classifier*. Penelitian ini mengimplementasikan *Deep Belief Network* (DBN) sebagai *pre-training* dan *Deep Neural Network* (DNN) sebagai *classifier*. Data yang digunakan pada penelitian ini adalah *dataset* ion channel, GPCR, dan nuclear receptor yang bersumber dari pangkalan data KEGG BRITE, BRENDA, SuperTarget, dan DrugBank. Hasil penelitian ini menunjukkan pada *dataset* tersebut, *pre-training* berupa ekstraksi fitur memberikan efek optimasi dilihat dari peningkatan performa model DNN pada akurasi (3-4.5%), AUC (4.5%), *precision* (5.9-6%), dan F-measure (3.8%).

Kata kunci: *deep belief network, deep neural network, deep semi-supervised learning, drug repositioning, imbalanced data, restricted boltzman machine*

COMPOUND-PROTEIN INTERACTION PREDICTION FOR DRUG REPOSITIONING USING DEEP SEMI-SUPERVISED LEARNING

Abstract

Drug repositioning is the reuse of an existing drug to treat a new disease other than its original medical indication. Drug repositioning can be done by predicting the interaction of drug compounds with disease proteins that react positively. One of the challenges in predicting the interaction of compounds and proteins is imbalanced data. Deep semi-supervised learning can be an alternative to handle prediction models with imbalanced data. The unsupervised learning based pre-training process in deep semi-supervised learning can represent input from unlabeled data (majority data) properly and optimize initialization of weights on the classifier. This study implements the Deep Belief Network (DBN) as a pre-training with Deep Neural Network (DNN) as a classifier. The data used in this study are ion channel, GPCR, and nuclear receptor dataset sourced from KEGG BRITE, BRENDA, SuperTarget, and DrugBank databases. The results of this study indicate that pre-training as feature extraction had an optimization effect. This can be seen from DNN performance improvement in accuracy (3-4.5%), AUC (4.5%), precision (5.9-6%), and F-measure (3.8%).

Keywords: *deep belief network, deep neural network, deep semi-supervised learning, drug repositioning, imbalanced data, restricted boltzman machine*

1. PENDAHULUAN

Pengembangan industri farmasi memiliki tantangan besar di mana permintaan terhadap obat-obatan baru yang inovatif melonjak seiring munculnya berbagai penyakit mematikan seperti kanker, diabetes, dan lainnya. Adapun untuk menemukan obat baru dibutuhkan biaya dan waktu yang tidak sedikit. Csermely dkk. (2013) menyatakan setidaknya dibutuhkan waktu 12-15 tahun dan satu juta USD untuk menemukan obat baru dan membawanya hingga lolos uji *Food and Drug Administration* (FDA) di Amerika Serikat. Untuk

menangani hal tersebut alternatif yang dilakukan adalah dengan melakukan *drug repurposing* atau *repositioning,* yaitu menggunakan senyawa obat yang sudah lolos uji sebelumnya untuk mengatasi penyakit baru (Ashburn dan Thor, 2004). Bagi masyarakat, penggunaan obat yang sudah lolos uji sebelumnya tentu menjadi jaminan bahwa obat tersebut layak dan aman dikonsumsi (Novac, 2013; Ezzat, 2016; Bahi dan Batouche, 2018a).

Penelitian mengenai *drug repositioning* dapat dilakukan dengan mengidentifikasi dan memprediksi interaksi senyawa obat dengan protein penyakit yang bereaksi positif melalui tiga pendekatan, yaitu pendekatan berbasis ligan, pendekatan berbasis *docking*, dan pendekatan *chemogenomics*. Pendekatan berbasis ligan memprediksi interaksi berdasarkan kemiripan ligan antar protein penyakit. Kelemahan pendekatan ini adalah sedikitnya informasi ligan yang dimiliki (Jacob dan Vert, 2008). Pendekatan berbasis simulasi *docking* memprediksi interaksi berdasarkan kemiripan struktur antar protein penyakit. Meskipun pendekatan ini memberikan hasil prediksi yang baik, pendekatan ini membutuhkan informasi struktur 3 dimensi dari protein yang mana informasi tersebut belum tersedia di banyak protein (Xie dkk., 2011). Pendekatan yang terakhir adalah *chemogenomics* yang memprediksi interaksi berdasarkan fitur senyawa obat dan protein penyakit. Ada beberapa metode pada *chemogenomics* seperti metode berbasis kernel yang memanfaatkan *similarity matrix* senyawa dan protein untuk prediksi, metode berbasis jaringan seperti yang dilakukan Yamanishi (2013) dan Kurnia (2017), serta metode berbasis fitur di mana senyawa dan protein direpresentasikan dalam suatu set deskriptor seperti yang dilakukan Ezzat (2016).

Kurnia (2017) melakukan prediksi interaksi senyawa dan protein berbasis jaringan menggunakan Bipartite Local Model Network Interaction-Profile Inferring (BLMNII) pada *golden standard dataset* Yamanishi dkk. (2008). Pada penelitian tersebut data memiliki sifat yang tidak seimbang terkait hubungan antar senyawa dan protein dengan rasio data positif berjumlah 0.0001%. Sifat data yang tidak seimbang menyebabkan algoritme BLMNII memprediksi hubungan senyawa dan protein berdasarkan kemiripan antar senyawa tanpa adanya data interaksi senyawa dan protein secara langsung. Hal ini menyebabkan waktu komputasi menjadi lambat sehingga BLMNII dinilai kurang efektif untuk digunakan. Permasalahan ini dapat diatasi melalui dua pendekatan. Pendekatan pertama yaitu menggunakan teknik pengambilan sampel pada level data dengan melakukan *resampling* sehingga proporsi kelas data menjadi seimbang, seperti yang dilakukan oleh Rahmi (2018) yang mengimplentasikan *hybrid sampling technique* menggunakan *Complementary Fuzzy Support Vector Machine* dan SMOTE pada *dataset* Pruengkarn (2017). Pendekatan kedua adalah dengan melakukan modifikasi pada algoritme klasifikasi (Ali dkk., 2015).

Salah satu pendekatan dengan modifikasi algoritme klasifikasi yang dapat menangani data yang tidak seimbang adalah *deep semi-supervised learning*. Pemodelan berbasis *deep semi-supervised learning* memanfaatkan *unsupervised learning* seperti *Stacked Auto Encoder* (SAE) dan *Deep Belief Network* (DBN) sebagai *pre-training* pada data tidak seimbang untuk meningkatkan akurasi model prediksi berbasis *supervised learning* (Erhan dkk., 2010). Erhan dkk. (2010) telah membandingkan kedua metode tersebut pada proses *pre-training* untuk melakukan klasifikasi pada *dataset* MNIST menggunakan *Multi Layer Neural Network*. Hasil penelitian tersebut membuktikan bahwa penggunaan *unsupervised learning* sebagai *pre-training* dapat meningkatkan performa dari pemodelan berbasis *supervised learning* secara signifikan, namun perbedaan performa DBN dan SAE tidak berbeda secara signifikan. Hasil penelitian Erhan dkk. (2010) menjadi inspirasi bagi penelitian *deep semi-supervised learning* lainnya, termasuk pada prediksi interaksi senyawa obat dan protein penyakit di mana datanya tidak seimbang. Seperti Bahi dan Batouche (2018b) mengusulkan metode DSSL-DTI dimana SAE digunakan untuk mereduksi dimensi data dan menginisialisasi bobot untuk model prediksi *Deep Neural Network* (DNN) pada data interaksi senyawa obat dan protein penyakit yang tidak seimbang. Penelitian tersebut memberikan hasil yang sangat baik dengan nilai akurasi (AR) 98.68%.

Melihat keberhasilan *deep semi-supervised learning* pada penelitian prediksi interaksi senyawa obat dan protein penyakit sebelumnya, maka dari itu penelitian ini akan membuat model prediksi interaksi senyawa aktif dan protein dengan data yang tidak seimbang menggunakan DBN sebagai *pre-training* dan DNN sebagai model prediksi. Evaluasi model dilakukan menggunakan *area under curve* (AUC), AR, *precision* (PR), *recall* (RE), dan F-measure.

2. METODE PENELITIAN

2.1. *Deep Semi-Supervised Learning*

Arsitektur *deep semi-supervised learning* yang digunakan pada penelitian ini terdiri atas algoritme RBM, DBN, dan DNN.

2.1.1. *Restricted Boltzman Machine*

Restricted Boltzman Machine (RBM) merupakan salah satu model generatif yang banyak digunakan dalam jaringan *deep learning* karena kemampuan historisnya dan kesederhanaannya. RBM dapat memodelkan berbagai jenis data termasuk data citra berlabel atau pun tidak berlabel, data suara, data teks pada dokumen, maupun data temporal berdimensi tinggi seperti video (Hinton, 2012). Sebagai salah satu tipe khusus dari Markov random field, RBM merupakan suatu jaringan saraf

tiruan berlapis dua yang membentuk graf *bipartite* di mana lapisan input disebut dengan unit *visible* (v) dan *hidden layer* disebut sebagai unit *hidden* (h). Terdapat suatu batasan hubungan antar unit *visible* dan unit *hidden*, yaitu node dari lapisan yang sama tidak boleh saling terhubung.

Jika terdapat model dengan parameter θ=[W, b, a] maka fungsi energi didefinisikan seperti pada persamaan (1).

$$E(\boldsymbol{v},\boldsymbol{h};\theta) = -\sum_{i=1}^{I}\sum_{j=1}^{J} w_{ij} v_i h_j - \sum_{i=1}^{I} b_i v_i - \sum_{j=1}^{J} a_j h_j \quad (1)$$

w_{ij} merupakan bobot yang menghubungkan unit *visible* v_i dengan jumlah total I dan unit *hidden* h_j dengan jumlah total J. b_i merupakan nilai bias dari unit *visible* dan a_j merupakan nilai bias dari unit *hidden*. Distribusi gabungan keseluruhan unit kemudian dihitung berdasarkan fungsi energi $E(\boldsymbol{v},\boldsymbol{h};\theta)$ dengan persamaan (2).

$$p(\boldsymbol{v},\boldsymbol{h};\theta) = \frac{exp(-E(\boldsymbol{v},\boldsymbol{h};\theta))}{Z} \quad (2)$$

Di mana $Z = \sum_{h;v} exp(-E(\boldsymbol{v},\boldsymbol{h};\theta))$ adalah fungsi partisi atau faktor normalisasi. Kemudian probabilitas bersyarat dari unit *visible* dan unit *hidden* dihitung dengan persamaan (3) dan (4).

$$p(h_j = 1|v;\theta) = \delta\left(\sum_{i=1}^{I} w_{ij} v_i + a_j\right) \quad (3)$$

$$p(v_i = 1|h;\theta) = \delta\left(\sum_{i=1}^{I} w_{ij} h_j + b_i\right) \quad (4)$$

δ didefinisikan sebagai fungsi logistik. RBM dilatih untuk memaksimalkan peluang bersama. Pembelajaran nilai W dilakukan melalui metode yang disebut dengan *contrastive divergence* (CD) (Zhao dkk., 2019).

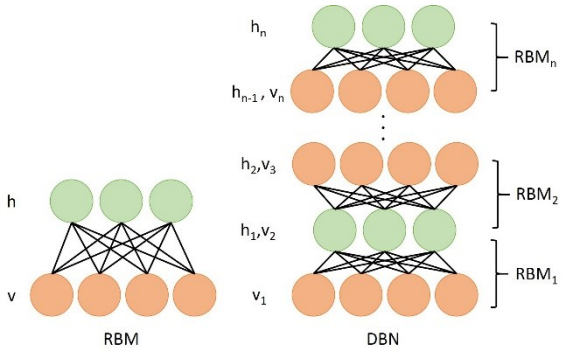
Setelah probabilitas dihitung, langkah selanjutnya adalah memperbarui unit *hidden* berdasarkan probabilitas bersyarat yang telah dihitung dan unit *visible* untuk mendapat rekonstruksi input (*output*). Hal tersebut dilakukan dengan memperbarui nilai bobot W dengan persamaan (5).

$$\Delta w_{ij} = \varepsilon\left(\langle v_i h_j\rangle_{data} - \langle v_i h_j\rangle_{recon}\right) \quad (5)$$

Dengan ε didefinisikan sebagai *learning rate* dari RBM, $\langle v_i h_j\rangle_{data}$ merupakan unit *visible* dan *hidden* dari input data dan $\langle v_i h_j\rangle_{recon}$ merupakan unit *visible* dan *hidden* hasil rekonstruksi input (Hinton dan Salakhutdinov 2006).

2.1.2. *Deep Belief Network*

Deep Belief Network (DBN) adalah salah satu kelas dari *deep generative* yang terdiri atas tumpukan beberapa *Restricted Boltzmann Machine* (RBM), di mana *output* dari lapisan ke-l (unit *hidden*) digunakan sebagai input dari lapisan ke-(l+1) (unit *visible*). Ilustrasi struktur jaringan DBN dapat dilihat pada Gambar 1. DBN dapat dilatih secara *greedy layer-wise* dan *unsupervised*. Hasil *pre-training* menggunakan DBN dapat menginisialisasi bobot pada *supervised learning* untuk menghasilkan model yang optimal dibandingkan dengan model dengan pengambilan bobot secara acak. Selain itu, DBN dapat digunakan secara efektif untuk melakukan *pre-training* untuk menginisialisasi pelatihan algoritme *backpropagation* (Ghasemi dkk., 2018; Zhao dkk., 2019).



Gambar 1. Ilustrasi jaringan RBM dan DBN

2.1.3. *Deep Neural Network*

Deep Neural Network (DNN) adalah suatu jaringan saraf tiruan umpan maju yang memiliki lebih dari satu *hidden layer* antara lapisan input dan *output*. Tiap *hidden layer* memiliki suatu fungsi aktivasi (persamaan (6)) seperti fungsi sigmoid, logistik, atau *hyperbolic tangent* (tanh) untuk memetakan input dari lapisan sebelumnya (x_j) ke *output* (y_j) yang akan dikirim pada lapisan setelahnya.

$$y_j = f(x_j) \quad x_j = b_j + \sum_i y_i w_{ij} \quad (6)$$

Di mana b_j merupakan bias dari unit j, i adalah indeks unit dari lapisan sebelumnya, dan w_{ij} adalah bobot yang menghubungkan unit i dan unit j.

DNN dapat dilatih secara diskriminatif dengan propagasi balik menggunakan turunan fungsi *cost* untuk mengukur selisih *output* target dan *output* aktual. Propagasi balik pada data latih yang besar dilakukan pada sebagian kecil dari data yang diambil secara acak sehingga lebih efisien dibandingkan dilakukan pada keselurahan data, sebelum memperbarui bobot secara proporsional dengan gradien seperti pada persamaan (7). Metode *stochastic gradient descent* ini dapat ditingkatkan menggunakan koefisien "momentum" (0< α <1) yang menghaluskan gradien pada *minibatch* t, sehingga mengurangi osilasi di jurang dan mempercepat kemajuan di jurang (Hinton dkk., 2012).

$$\Delta w_{ij}(t) = \alpha\, \Delta w_{ij}(t-1) - \epsilon\, f'(C) \quad (7)$$

DNN dengan jumlah *hidden layer* yang banyak sulit untuk dioptimasi. Pendekatan menggunakan *gradient descent* dari titik awal yang dibangkitkan secara acak mendekati nilai aktual tidak dapat menghasilkan suatu set bobot yang baik, kecuali dilakukan inisialisasi skala bobot secara hati-hati (Glorot dan Bengio 2010). Maka dari itu inisialisasi

bobot pada pemodelan DNN menjadi penting untuk meningkatkan performa pemodelan DNN.

2.2. Data Penelitian

Data pada penelitian ini bersumber dari *golden standard dataset* pada penelitian Yamanishi dkk. (2008) yang menjadi acuan bagi penelitian prediksi interaksi senyawa dan protein lainnya. *Dataset* Yamanishi dkk. (2008) terdiri atas pasangan ID KEGG senyawa (V2) dan protein (V1) yang memiliki interaksi (Gambar 2).

V1	V2
hsa:190	D00094
hsa:2099	D00066
hsa:2099	D00067
hsa:2099	D00105
hsa:2099	D00312

Gambar 2. *Dataset* Yamanishi dkk. (2008)

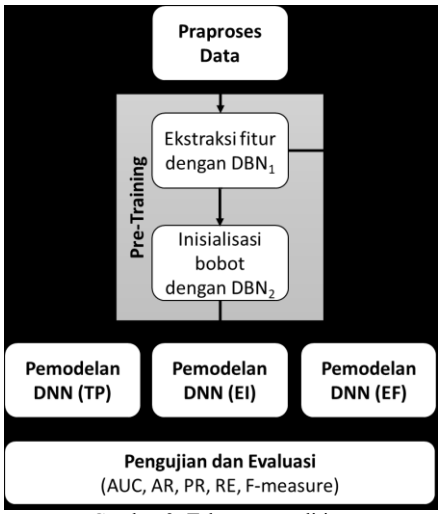
Dataset Yamanishi dkk. (2008) yang akan digunakan pada penelitian ini terdiri atas G–Protein-Coupled Receptor (GPCR), ion channel, dan nuclear receptor. Informasi statistik *dataset* Yamanishi dapat dilihat pada Tabel 1.

Tabel 1. Statistik *dataset* Yamanishi dkk. (2008)

Dataset	Jumlah Protein	Jumlah Senyawa	Jumlah Data	Rasio Interaksi (%)
Ion Channel	204	210	42840	3.4
GPCR	223	95	21185	2.9
Nuclear Receptor	26	54	1404	6.4

2.3. Tahapan Penelitian

Tahapan penelitian ini terdiri atas praproses data, *pre-training* menggunakan DBN, pemodelan DNN, serta pengujian dan evaluasi model (Gambar 3). Seluruh tahapan penelitian ini dilakukan pada masing-masing *dataset*.

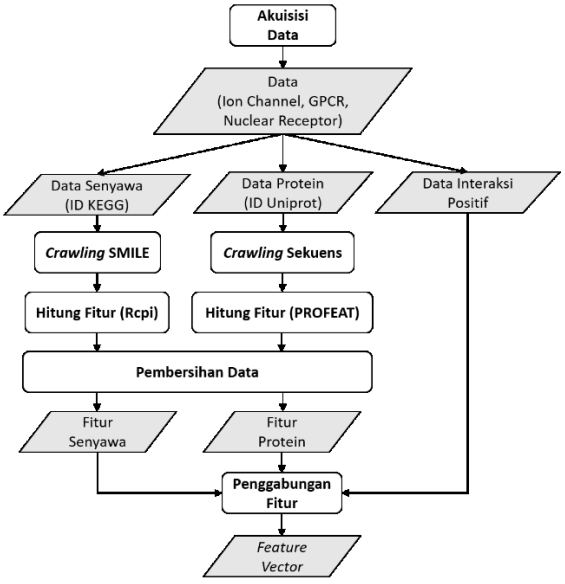


Gambar 3. Tahapan penelitian

Tahap *pre-training* terdiri atas dua kali pemodelan DBN untuk ekstraksi fitur (DBN_1) dan inisialisasi bobot DNN (DBN_2). Pada penelitian ini dilakukan perbandingan antara model tanpa *pre-training* (TP), model dengan *pre-training* ekstraksi fitur (EF), dan model dengan *pre-training* ekstraksi fitur + inisialisasi bobot (EI) untuk melihat apakah *pre-training* memberikan efek optimisasi pada model klasifikasi DNN. Evaluasi perbandingan model tersebut dilakukan dengan menghitung nilai AUC, AR, *precision*, *recall*, dan F-measure masing-masing model. Implementasi dilakukan menggunakan bahasa pemrograman Python 3.5 di mana implementasi DBN dilakukan dengan memodifikasi implementasi DBN oleh Albertbup (2017) sedangkan implementasi SAE dan DNN dilakukan menggunakan *package* h2o.

2.3.1. Praproses Data

Tahap ini terdiri atas beberapa sub tahap, yaitu akuisisi *golden standard dataset*, *crawling* data deskriptor senyawa dan protein dari pangkalan data KEGG, penghitungan fitur, pembersihan data, dan penggabungan fitur (Gambar 4). Kerangka tahapan praproses data pada penelitian ini mengacu pada penelitan Ezzat dkk. (2016) serta Bahi dan Batouche (2018a).

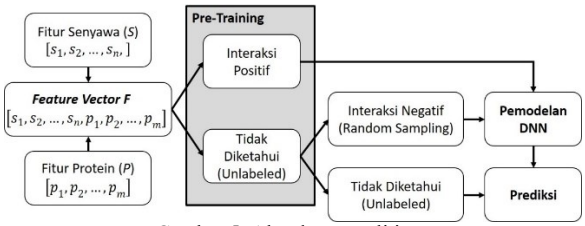


Gambar 4. Tahap praproses data (Ezzat dkk. 2016; Bahi dan Batouche 2018a)

Akuisisi data dilakukan dengan mengunduh *dataset* ion channel, GPCR, dan nuclear receptor pada http://web.kuicr.kyotou.ac.jp/supp/yoshi/drugtarget. Masing-masing *dataset* terdiri atas pasangan ID KEGG senyawa dan ID KEGG Protein yang memiliki interaksi positif. Data ID KEGG senyawa dan protein selanjutnya digunakan pada tahap *crawling* data untuk mendapatkan deskriptor berupa sidik jari molekular (SMILE) senyawa dan sekuens protein dalam fail teks bertipe FASTA. *Crawling* data dilakukan menggunakan *package* Rcpi pada R.

Data deskriptor senyawa kemudian dihitung fiturnya berdasarkan struktur konstitusional, topologi, dan geometri senyawa menggunakan fungsi extractDrugAIO pada *package* Rcpi. Adapun fitur protein dihitung menggunakan PROFEAT Webserver yang mencakup komposisi asam amino, komposisi asam pseudo-amino, dan deskriptor CTD (*composition*, *transition*, *distribution*). Perhitungan fitur senyawa menghasilkan suatu *feature vector S* berdimensi *n* ($S=[s_1, s_2,\ldots, s_n]$). Adapun perhitungan fitur protein menghasilkan suatu *feature vector P* berdimensi *m* ($P=[p_1, p_2,\ldots, p_m]$). Pembersihan data terhadap *S* dan *P* dilakukan dengan normalisasi data pada rentang [0 1], identifikasi dan penghapusan atribut yang tidak memberi makna, dan pengisian *missing value* dengan nilai *mean*.

Penggabungan fitur dilakukan dengan menggabungkan *S* dan *P* ke dalam satu *feature vector F* ($F=[s_1, s_2,\ldots,s_n, p_1,p_2,\ldots,p_m]$). *Feature vector F* dari seluruh data kemudian digunakan pada proses *pre-training*. Setelah itu data dipisahkan menjadi data latih DNN dan data untuk prediksi. Pada data latih DNN informasi interaksi ditambahkan sebagai atribut pada *feature vector F*. Data latih DNN terdiri atas data senyawa dan protein yang diketahui berinteraksi (interaksi positif) ditambah data yang belum diketahui interaksinya yang diambil secara acak sejumlah data interaksi positif. Data pasangan senyawa dan protein yang diketahui berinteraksi diberi label +1 pada atribut interaksi sedangkan selainnya diberi label -1 (interaksi negatif). Pembagian data untuk tiap tahapan penelitian dapat dilihat pada alur data (Gambar 5).



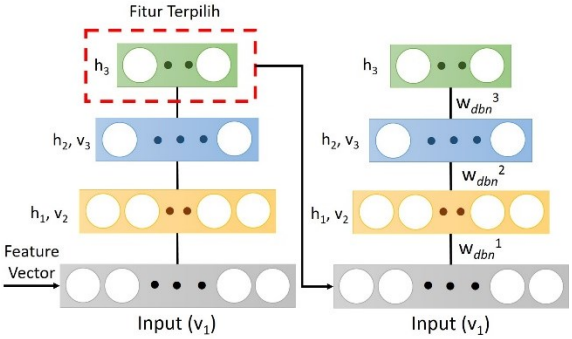
Gambar 5. Alur data penelitian

2.3.2. *Pre-Training*

Tahap ini terdiri dari dua bagian, yaitu ekstraksi fitur (DBN_1) dan inisialisasi bobot (DBN_2). Proses *unsupervised pre-training* untuk ekstraksi fitur menggunakan DBN terdiri atas tahapan berikut:

1 Inisialisasi parameter W, b, dan c secara acak.
2 Latih lapisan pertama dan kedua sebagai RBM. *Feature vector* dihasilkan pada tahap sebelumnya dijadikan lapisan *visible* (v_1). Fungsi aktivasi yang digunakan adalah sigmoid.
3 Latih lapisan kedua dan ketiga sebagai RBM, dengan lapisan kedua sebagai lapisan *visible* yang merepresentasikan lapisan ketiga (v_2).
4 Ulangi tahap nomor 3 untuk lapisan 3 dan 4.
5 Setelah seluruh *epoch* selesai, ambil lapisan ke-4 sebagai fitur terpilih.

Setelah data input ditransformasi ke dalam fitur penting, model DBN_2 dilatih (Gambar 6). Proses pelatihan DBN_2 sama dengan DBN_1 hanya saja yang diambil adalah bobot akhir yang dihasilkan tiap *layer*.



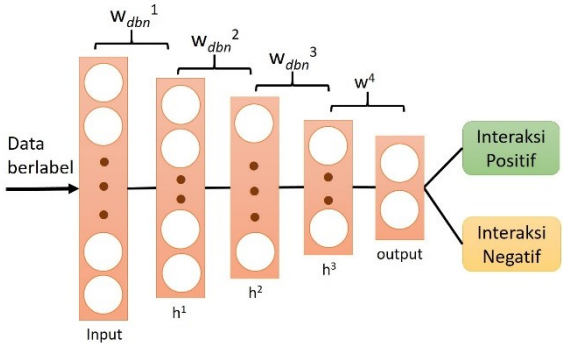
Gambar 6. Model DBN_1 (kiri) dan DBN_2 (kanan)

2.3.3. Klasifikasi DNN

Tahap selanjutnya adalah proses *supervised fine-tuning* dengan membuat model prediksi DNN berdasarkan hasil *pre-training* (Gambar 7). Model DNN yang dibangun memiliki jumlah *hidden layer* yang sama dengan DBN yaitu 3 lapisan.

Tahapan pada DNN terdiri atas:

1 Transformasi data latih DNN mejadi fitur penting menggunakan DBN_1
2 Gunakan parameter (bobot dan bias) hasil pemodelan DBN_2
3 Inisialisasi bobot lapisan *output* (W^4) dari DNN secara acak, fungsi aktivasi yang digunakan adalah fungsi tanh.
4 *Fine-tune* seluruh parameters DNN dengan *stochastic gradient descent* menggunakan *backpropagation*.



Gambar 7. Arsitektur DNN dengan bobot hasil *pre-training*

2.3.4. Evaluasi Model

Hasil pengujian akan dideskripsikan dalam bentuk *confusion matrix* (Tabel 2) untuk mempermudah pengukuran evaluasi.

Tabel 2. *Confusion matrix*

	Predicted Positive	***Predicted Negative***
Actual Positive	*True Positive* (TP)	*False Negative* (FN)
Actual Negative	*False Positive* (FP)	*True Negative* (TN)

Nilai pengukuran evaluasi kemudian dihitung menggunakan perhitungan statiska berikut:

- *Area under curve* (AUC): mengukur kemampuan model dalam membedakan kelas.

$$AUC = \left(\frac{TP}{TP+FP} + \frac{TN}{TN+FN}\right) \times 100 \quad (8)$$

- Akurasi (AR): mengukur persentase data uji yang diklasifikasikan dengan benar

$$AR = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (9)$$

- *Precision*: mengukur rasio kelas positif yang diklasifikasikan dengan benar dari semua yang diprediksikan positif

$$Precision = \frac{TP}{TP+FP} \quad (10)$$

- *Recall*: mengukur rasio kelas positif yang diklasifikasikan dengan benar dari kelas positif

$$Recall = \frac{TP}{TP+FN} \quad (11)$$

- F-measure: mengukur performa kelas minoritas

$$F - measure = \frac{2*Precision*Recall}{Precision+Recall} \quad (12)$$

2.4. Peralatan Penelitian

Penelitian yang dilakukan menggunakan perangkat keras dan perangkat lunak sebagai berikut:

1 Perangkat keras berupa komputer dengan spesifikasi:
   - Processor Intel Core i5-4460 CPU 3.20GHz.
   - RAM 8.00 GB.
   - Hardisk internal 1 TB.

2 Perangkat lunak yang digunakan:
   - Sistem Operasi Ubuntu 18.04.02 LTS.
   - Bahasa pemrograman Python versi 3.6. dan R versi 3.6.
   - Aplikasi RStudio versi 1.2.1335 sebagai IDE (*Integrated Development Environment*) dari bahasa pemrograman R.
   - Aplikasi PyCharm Community Edition versi 2019.1.3 sebagai IDE dari bahasa pemrograman Python.
   - *Package* 'Rcpi', 'rcdk', dan 'Biostring' pada bahasa pemrograman R untuk tahap praproses.
   - *Package* 'h2o' versi 3.24.0.3 pada bahasa pemrogaman Python untuk pemodelan SAE dan DNN.

3. HASIL DAN PEMBAHASAN

Pada penelitian ini dilakukan tiga percobaan yaitu pemodelan DNN dengan *pre-training* berupa ekstraksi fitur dan inisialisasi bobot (EI), pemodelan DNN dengan *pre-training* berupa ekstraksi fitur saja (EF), dan pemodelan DNN tanpa *pre-training* (TP). Pemodelan DBN-DNN dibangun berdasarkan hasil *tuning* parameter dengan teknik *grid search* terhadap parameter *hidden layer*, *adaptive rate*, *learning rate*, *rate annealing*, *momentum start*, *momentum stop*, *fold assignment*, dan *balance class*. Evaluasi model dilakukan dengan menerapkan *k-fold cross validation* dengan k=10. Penerapan *k-fold cross validation* menghasilkan 10 model pada masing-masing model EI, EF, dan TP. Evaluasi performa model dilakukan dengan membandingkan nilai rataan tiap ukuran evaluasi dari 10 model tersebut. Hasil *tuning* parameter dan evaluasi model DNN dengan DBN sebagai *pre-training* secara rinci ada pada sub bab berikut.

3.1. *Tuning* Parameter

Tuning parameter menggunakan teknik *grid search* dilakukan untuk mencari parameter terbaik pada pemodelan DBN_1, DBN_2, dan DNN. Parameter hasil *grid search* dapat dilihat pada Tabel 3. Pada DBN_1 parameter *hidden layer* terbaik untuk seluruh *dataset* bernilai sama, yaitu 1000 neuron pada *hidden layer* pertama, 750 neuron pada *hidden layer* kedua, dan 500 neuron pada *hidden layer* ketiga. Parameter *hidden layer* DBN_2 hasil *grid search* kemudian akan digunakan untuk memodelkan DBN_2 dan DNN. Hal ini disebabkan model bobot DBN_2 digunakan untuk menginisialisasi bobot DNN sehingga jumlah neuron pada kedua model tersebut harus sama.

Tabel 3. Parameter hasil *grid search*

Model	Parameter	*Dataset*		
		Ion Channel	GPCR	Nuclear Receptor
DBN_1	*Hidden Layer*	[1000, 750,500]	[1000, 750,500]	[1000, 750,500]
DBN_2	*Hidden Layer*	[300, 180, 108]	[300, 180, 108]	[100, 50, 25]
DNN	*Adaptive rate*	*True*	*False*	*False*
	Learning rate	0.005	0.005	0.005
	Rate annealing	10^{-6}	10^{-8}	10^{-8}
	Momentum start	0	0.5	0
	Momentum stop	0	0.5	0
	Fold assignment	*Stratified*	*Stratified*	*Stratified*
	Balance class	*True*	*True*	*True*

3.2. *Dataset* Nuclear Receptor

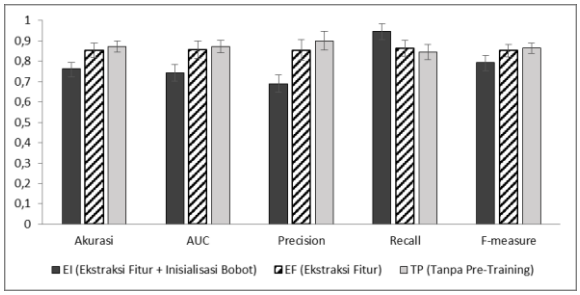
Dataset nuclear receptor merupakan *dataset* dengan ukuran terkecil pada golden standard *dataset* dengan data latih untuk *pre-training* sebanyak 1404 baris dan data latih untuk DNN sebanyak 180 baris. Hasil evaluasi model pada *dataset* ini berdasarkan *k-fold cross validation* dapat dilihat pada Tabel 4.

Tabel 4. Evaluasi model DBN-DNN pada *dataset* nuclear receptor

Evaluasi	EI	EF	TP
Akurasi	0,762863	0,853945	**0,872163**
AUC	0,74423	0,8574	**0,872425**
Precision	0,689858	0,855278	**0,900635**
Recall	**0,945952**	0,863568	0,843925

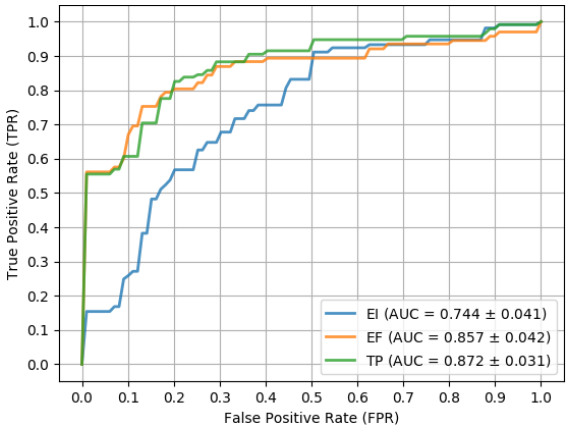
F-measure	0,793816	0,852341	**0,864091**

Pada percobaan dilakukan perbandingan performa antara model pada perlakuan EI, EF, dan TP (Gambar 8). Gambar 8 menunjukkan bahwa meskipun model TP unggul dalam akurasi, *precision*, dan F-measure dibandingkan model lainnya, perbedaan nilai tersebut tidak signifikan secara statistik. Hal itu terlihat dari *error bar* model TP yang bersinggungan dengan model EF. Nilai *recall* tertinggi ada pada model EI dengan nilai 94.5952% namun nilai *precision* model tersebut merupakan yang terrendah dengan nilai 68.1273%. Hal ini mengindikasikan meskipun model EI mampu memprediksi kelas aktual positif dengan baik, terdapat banyak kesalahan klasifikasi kelas negatif menjadi kelas positif.



Gambar 8. Grafik perbandingan model DBN-DNN pada *dataset* nuclear receptor

Kurva ROC (Gambar 9) menunjukkan model TP dan EF memiliki pergerakan nilai yang berbeda tipis di mana model TP sedikit lebih unggul dari EF. Perbedaan nilai tersebut tidak signifikan secara statistik di mana pada Gambar 8 *error bar* model TP bersinggungan dengan model EF. Adapun model EI memiliki pergerakan nilai yang lebih lambat dibanding model lainnya dan memiliki nilai *true positive rate* yang lebih kecil pada titik *false positive rate*=0.



Gambar 9. Perbandingan kurva ROC model DBN-DNN pada *dataset* nuclear receptor

Maka dari itu dapat disimpulkan pada percobaan model DBN-DNN dengan *dataset* nuclear receptor, *pre-training* menggunakan DBN tidak berhasil memberi efek optimasi. Hal ini terlihat dari model EI yang memiliki performa lebih buruk dari TP dan model EF yang performanya tidak berbeda signifikan dengan model TP. Sehingga untuk *dataset* nuclear receptor untuk model perlakuan TP merupakan model terbaik.

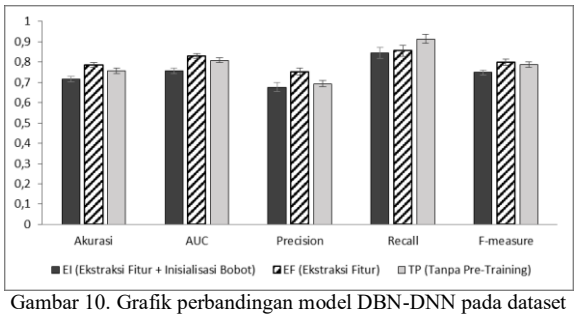
Pada *dataset* ini *pre-training* tidak memberikan efek optimasi pada model klasifikasi DNN. Hal ini sejalan dengan pernyataan Erhan dkk. (2010) bahwa *pre-training* berbasis *unsupervised learning* dapat mengoptimasi model klasifikasi hanya ketika data latih memiliki ukuran yang besar. Maka dari itu dapat disimpulkan bahwa untuk *dataset* berukuran kecil seperti nuclear receptor, model DNN tanpa *pre-training* mampu memberikan performa yang lebih baik dibandingkan model DNN dengan *pre-training* berbasis *unsupervised learning*.

3.3. *Dataset* GPCR

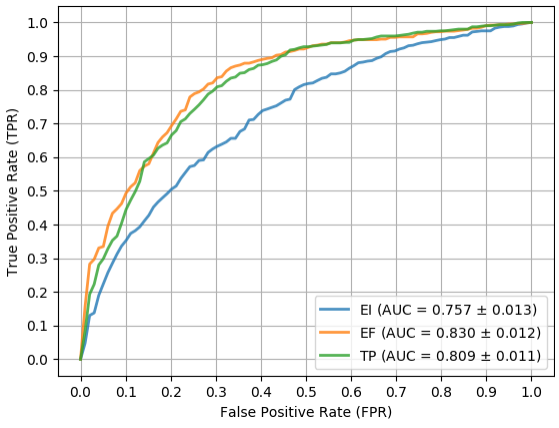
Hasil evaluasi model dari masing-masing perlakuan pada *dataset* GPCR berdasarkan *k-fold cross validation* dapat dilihat pada Tabel 5. Adapun visualisasi perbandingan performa model dapat dilihat pada Gambar 10. Analisis performa AUC divisualisasikan dengan kurva ROC pada Gambar 11.

Tabel 5. Evaluasi model DBN-DNN pada *dataset* GPCR

Evaluasi	EI	EF	TP
Akurasi	0,716868	**0,787087**	0,75582
AUC	0,75738	**0,830287**	0,80859
Precision	0,677022	**0,752543**	0,693517
Recall	0,84579	0,856872	**0,914697**
F-measure	0,748744	**0,799503**	0,788157



Gambar 10. Grafik perbandingan model DBN-DNN pada dataset GPCR



Gambar 11. Perbandingan kurva ROC model antar perlakuan DBN-DNN pada *dataset* GPCR

Gambar 10 menunjukkan bahwa model EF unggul dalam akurasi dan *precision* dibandingkan model lainnya dan nilai tersebut signifikan secara statistik. Meskipun nilai F-measure model EF sedikit lebih unggul dari model TP, *error bar* kedua model tersebut bersinggungan sehingga dapat dikatakan kedua model tersebut tidak berbeda secara signifikan. Nilai *recall* tertinggi ada pada model TP dengan nilai 91.4697% dan *error bar* nilai tersebut tidak bersinggungan dengan model lainnya. Akan tetapi nilai *precision* model tersebut tergolong kecil dengan nilai 69.3517%.

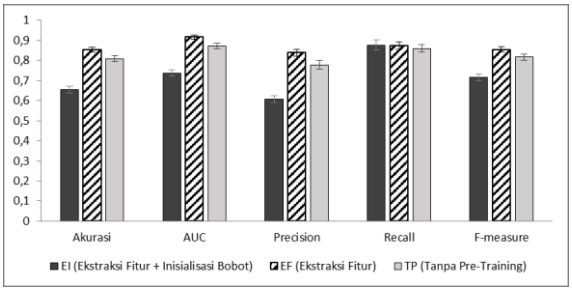
Analisis performa AUC yang divisualisasikan pada kurva ROC (Gambar 11) menunjukkan model EF memiliki pergerakan nilai yang lebih cepat mendekati titik (0,1) meskipun berbeda tipis dengan model TP. Adapun model EI memiliki pergerakan yang lebih lambat.. Jika dilihat *error bar* nilai AUC model EF bersinggungan sedikit dengan model TP sehingga dapat dikatakan kedua model tersebut tidak berbeda signifikan. Jika dibandingkan secara keseluruhan model yang memiliki performa terbaik pada *dataset* GPCR adalah model EF. Hal ini menunjukkan bahwa *pre-training* berupa ekstraksi fitur menggunakan DBN terbukti memberikan efek optimasi yang signifikan pada model klasifikasi DNN dengan *dataset* berukuran besar.

3.4. *Dataset* Ion Channel

Hasil evaluasi model dari masing-masing perlakuan pada *dataset* ion channel berdasarkan *k-fold cross validation* dapat dilihat pada Tabel 6. Adapun perbandingan performa model EI, EF, dan TP pada dataset ion channel divisualisasikan dalam diagram balok pada Gambar 12.

Tabel 6. Evaluasi model DBN-DNN pada *dataset* ion channel

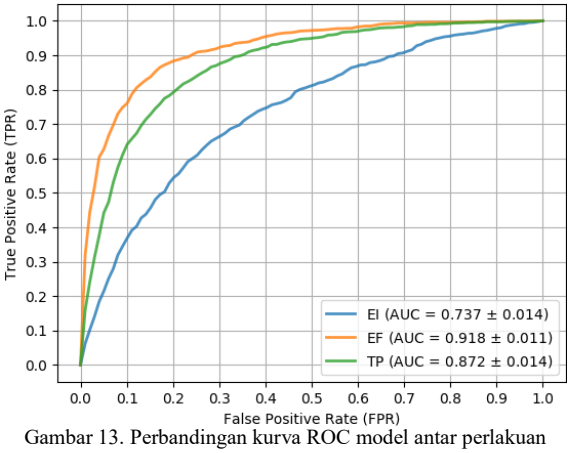
Evaluasi	EI	EF	TP
Akurasi	0,65487	**0,853723**	0,809047
AUC	0,737424	**0,917682**	0,872311
Precision	0,607085	**0,838696**	0,778952
Recall	**0,87734**	0,873846	0,860419
F-measure	0,716208	**0,855136**	0,816698



Gambar 12. Grafik perbandingan model antar perlakuan DBN-DNN pada *dataset* ion channel

Gambar 12 menunjukkan bahwa model EF mengungguli model lainnya dalam akurasi, *precision*, dan F-measure secara signifikan. Hal ini terbukti dari *error bar* model EF pada nilai tersebut yang tidak bersinggungan dengan model EI dan TP. Model EI mengungguli EF dalam *recall*, namun perbedaan nilai *recall* kedua model tersebut sangat tipis yaitu sebesar 0.3494%. *Error bar* nilai *recall* dari ketiga model tersebut juga bersinggungan sehingga dapat dikatakan bahwa ketiga model tersebut tidak berbeda secara signifikan.

Analisis terhadap AUC yang divisualisasikan pada kurva ROC (Gambar 13) menunjukkan bahwa model yang paling mendekati titik (0,1) merupakan model EF. Selain itu pergerakan nilai model EF menuju titik konstan lebih cepat dan lebih tinggi dibanding model lainnya. Selain itu, Gambar 12 menunjukkan bahwa *error bar* nilai AUC dari model EF tidak bersinggungan dengan model lainnya. Maka dari itu dapat dikatakan bahwa model EF pada *dataset* ion channel merupakan model dengan performa terbaik. Hal ini menunjukkan semakin besar ukuran *dataset* maka efek optimasi *pre-training* menggunakan DBN semakin signifikan.



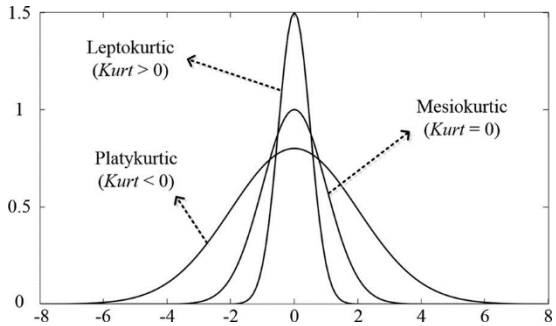
Gambar 13. Perbandingan kurva ROC model antar perlakuan DBN-DNN pada *dataset* ion channel

3.5. *Pre-Training* Inisialisasi Bobot

Meskipun pada penelitian Erhan dkk. (2010) penerapan *pre-training* berupa inisialisasi bobot (EI) pada data citra memberikan hasil yang memuaskan, model EI pada data fitur senyawa dan protein di penelitian ini cenderung tidak memberikan efek optimasi dan justru memperburuk performa klasifikasi DNN meskipun memiliki nilai *recall* yang relatif baik. Menurut Erhan dkk. (2009) *pre-training* inisialisasi bobot dapat memperparah performa model klasifikasi ketika nilai bobot dan bias yang dihasilkan memiliki distribusi *fat-tailed* atau leptokurtic (Gambar 14).

Leptokurtic merupakan distribusi data yang memiliki nilai kurtosis lebih dari 3 (*excess kurtosis* > 0) sedangkan distribusi normal memiliki nilai kurtosis yang mendekati 0 (Larasati dkk. 2018). Maka dari itu dilakukan pengecekan nilai kurtosis pada bobot dan bias yang dihasilkan model DBN_2 pada setiap *dataset*. Pada *dataset* nuclear receptor dihasilkan

excess kurtosis sebesar 17.35, pada *dataset* GPCR sebesar 17.2 dan pada *dataset* ion channel sebesar 31.8. Ketiga *dataset* tersebut memiliki inisialisasi bobot dan bias dengan nilai kurtosis lebih dari 0 sehingga masuk ke dalam kategori leptokurtic.



Gambar 14. Ilustrasi distribusi leptokurtic (Zhong dkk., 2016)

Menurut Larasati dkk. (2018) distribusi data yang leptokurtic dapat menyebabkan peningkatan kesalahan klasifikasi pada *artificial neural network* sehingga menurunkan performa model klasifikasi. Pernyataan tersebut dikuatkan oleh penemuan pada penelitian ini di mana dengan bobot berdistribusi leptokurtic nilai akurasi dan *precision* yang dihasilkan model EI DBN-DNN jauh lebih kecil dari model lain meskipun memiliki nilai *recall* tinggi. Dengan kata lain pada model EI DBN-DNN terjadi banyak kesalahan klasifikasi terutama kesalahan klasifikasi kelas negatif menjadi kelas positif. Maka dari itu pada *dataset* penelitian ini *pre-training* inisialisasi bobot pada DBN-DNN tidak cocok digunakan karena menghasilkan bobot dengan distribusi leptokurtic sehingga menurunkan performa klasifikasi DNN.

4. KESIMPULAN

Penelitian ini membuktikan bahwa pada *dataset* dengan ukuran kecil seperti nuclear receptor model klasifikasi DNN tanpa bantuan *pre-training* dapat memberikan performa yang lebih baik dibanding model dengan *pre-training* dengan akurasi 87,2%. Pada *dataset* dengan ukuran yang lebih besar seperti GPCR dan ion channel model *pre-training* dengan ekstraksi fitur oleh DBN memberikan hasil yang paling baik dibandingkan SAE. Peningkatan performa DNN dengan *pre-training* DBN pada *dataset* GPCR antara lain peningkatan akurasi sebesar 3.127% dan *precision* sebesar 5.9%. Pada *dataset* ion channel dihasilkan peningkatan pada akurasi sebesar 4.468%, AUC sebesar 4.537%, *precision* sebesar 5.974%, dan F-measure sebesar 3.844%. Peningkatan dihitung berdasarkan selisih nilai evaluasi model DNN dengan *pre-training* dengan model DNN tanpa *pre-training*.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Kementerian Riset, Teknologi, dan Pendidikan Tinggi yang telah memberikan bantuan melalui program hibah Penelitian Tesis Magister (PTM) dengan nomor 4408/IT3.L1/PN/2019 tanggal 4 April 2019 sehingga naskah jurnal ini dapat diterbitkan. Semoga hasil penelitian ini dapat bermanfaat bagi peneliti lain maupun masyarakat luas.

DAFTAR PUSTAKA

ALBERTBUP. 2017. *A python implementation of deep belief networks built upon numpy and tensorflow with scikit-learn compatibility*. [online] Tersedia pada: <https://github.com/albertbup/deep-belief-network> [diunduh: 2019 Mar 6].

ALI, A., SHAMSUDDIN, S.M. & RALESCU, A.L. 2015. Classification with Class Imbalance Problem: A Review. *International Journal Advance Soft Computing Application*, 7(3), pp.176, 204.

ASHBURN, T.T. dan THOR, K.B. 2004. Drug Repositioning: Identifying and Developing New Uses for Existing Drugs. *Nature Reviews Drug discovery*, 3(8), pp.673.

BAHI, M. dan BATOUCHE, M. 2018a. Drug-Target Interaction Prediction in Drug Repositioning Based on Deep Semi-Supervised Learning. Dalam: *6th International Conference on Computer Intelligence and Its Applications (CIIA)*. Oran (DZ): Springer International Publishing, pp.302-313.

BAHI, M. dan BATOUCHE, M. 2018b. Deep Semi-Supervised Learning for DTI Prediction Using Large Datasets and H2O-Spark Platform. Dalam: *2018 International Conference on Intelligent Systems and Computer Vision (ISCV)*. Fez (MA): IEEE, pp.1-7.

CAO, D., XIAO, N., XU, Q. & CHEN, A.F. 2015. Rcpi: R/Bioconductor package to generate various descriptors of proteins, compounds and their interactions. *Bioinformatics*, 31(2), pp.279, 281.

CSERMELY, P., KORCSMÁROS, T., KISS. H.J.M., LONDON, G. & NUSSINOV, R. 2013. Structure and Dynamics of Molecular Networks: A Novel Paradigm of Drug Discovery: A Comprehensive Review. *Pharmacology and Therapeutics*, 138(3), pp.333, 408.

ERHAN, D., BENGIO, Y., COURVILLE, A., MANZAGOL, P.A., VINCENT, P. & BENGIO, S. 2010. Why Does Unsupervised Pre-Training Help Deep Learning?. *Journal of Machine Learning Research*, 11(Feb), pp.625, 660.

ERHAN, D., MANZAGOL, P.A., BENGIO, Y., BENGIO, S. & VINCENT, P. 2009. The Difficulty of Training Deep Architectures

and The Effect of Unsupervised Pre-Training. Dalam: *12th International Conference on Artificial Intelligence and Statistics (AISTATS)*. Florida (AS): PMLR. pp.153-160.

EZZAT, A., WU, M., LI, X.L. & KWOH, C.K. 2016. Drug-Target Interaction Prediction Via Class Imbalance-Aware Ensemble Learning. *BMC Bioinformatics*, 17(19), pp.267, 276.

GHASEMI, F., MEHRIDEHNAVI, A., FASSIHI, A. & PÉREZ-SÁNCHEZ, H. 2018. Deep Neural Network in QSAR Studies Using Deep Belief Network. *Applied Soft Computing*, 62, pp.251, 258.

GLOROT, X. dan BENGIO, Y. 2010. Understanding The Difficulty of Training Deep Feedforward Neural Networks. Dalam: *Thirteenth International Conference on Artificial intelligence and Statistics*. Sardiana (IT): Journal of Machine Learning Research. pp.249-256.

HINTON, G.E. & SALAKHUTDINOV, R.R. 2006. Reducing The Dimensionality of Data with Neural Networks. *Science 313*, 5786, pp.504, 507.

HINTON, G.E. 2012. A practical guide to training restricted Boltzmann machines. Dalam: Neural networks: Tricks of the trade. Berlin (DE): Springer. pp.599-619.

HINTON, G.E., DENG, L., YU, D., DAHL, G.E., MOHAMED, A.R., JAITLY, N., SENIOR, A., VANHOUCKE, V., NGUYEN, P., SAINATH, T.N. & KINGSBURY, B. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6), pp.82-97.

JACOB, L. & VERT, J.P. 2008. Protein-Ligand Interaction Prediction: An Improved Chemogenomics Approach. *Bioinformatics*, 24(19), pp.2149, 2156.

KURNIA, A. 2017. Prediksi formula jamu berkhasiat menggunakan teknik link prediction dari jejaring bipartite senyawa aktif dan protein. [skripsi] Institut Pertanian Bogor, Indonesia.

LARASATI, A., DWIASTUTIK, A., RAMADHANTI, D. & MAHARDIKA, A. 2018. The effect of Kurtosis on the accuracy of artificial neural network predictive model. Dalam: *MATEC Web of Conferences*, 204, pp.02018.

NOVAC, N. 2013. Challenges and Opportunities of Drug Repositioning. *Trends in pharmacological sciences*, 34(5), pp.267, 272.

PRUENGKARN, R., WONG, K.W. & FUNG, C.C. 2017. Imbalanced Data Classification Using Complementary Fuzzy Support Vector Machine Technique and SMOTE. Dalam: *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Banff (CA): IEEE. pp.978-983.

RAHMI, A.S. 2018. Implementasi Hybrid Sampling Technique untuk Prediksi Interaksi Senyawa Aktif dan Protein pada Data yang Tidak Seimbang. Dalam: *ICDALC International Conference on Digital Agriculture from Land to Consumer*. Bogor (ID):IPB Press.

XIE, L., EVANGELIDIS, T., XIE, L., BOURNE, P.E. 2011. Drug Discovery Using Chemical Systems Biology: Weak Inhibition of Multiple Kinases May Contribute to The Anti-Cancer Effect of Nelfinavir. *PLoS Computational Biology*, 7(4), pp.e1002037.

YAMANISHI, Y. 2013. Chemogenomic Approaches to Infer Drug–Target Interaction Networks. *Data Min Syst Biol*, 939, pp.97, 113.

YAMANISHI, Y., ARAKI, M., GUTTERIDGE, A., HONDA, W. & KAMEHISA, M. 2008. Prediction of Drug–Target Interaction Networks from The Intefration of Chemical and Genomic Spaces. *Bioinformatics*, 24(8), pp.i232, i240.

ZHAO, R., YAN, R., CHEN, Z., MAO, K., WANG, P. & GAO, R.X. 2019. Deep Learning and Its Applications to Machine Health Monitoring. *Mechanical Systems and Signal Processing*, 115, pp.213, 237.

ZHONG, L., CHENG, L., XU, H., WU, Y., CHEN, Y., & LI, M. 2016. Segmentation of individual trees from TLS and MLS data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(2), pp.774, 787.