

Implementasi Deep Belief Network Sebagai Algoritma untuk Mendeteksi Penyakit Kanker Payudara

Wahyu Novitasari (2043201001), Hafidz Dinillah Al-Faqih (2043201072), Adistyari Husna (2043201102), dan Mukti Ratna Dewi, S.Si, M.Sc.
Departemen Statistika Bisnis, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember (ITS)
Kampus ITS Sukulilo-Surabaya 60111, Indonesia
e-mail:204320@its.ac.id

Abstrak—Tingkat prevalensi kanker payudara terus meningkat. Keberhasilan deteksi kanker sejak dini sangat penting dalam meningkatkan prognosis dan keselamatan pasien. Dalam upaya meningkatkan akurasi deteksi dini, teknologi *Deep Belief Network* (DBN) adalah suatu pengembangan dari *deep learning* yang merupakan tumpukan atau *stack* dari beberapa algoritma atau metode yang bertujuan *feature extraction* yang memanfaatkan seluruh *resource* seoptimal mungkin yang telah menjadi sorotan algoritma potensial. Data yang digunakan merupakan data sekunder yaitu dataset “*breast cancer wisconsin*”. Analisis yang dilakukan adalah *deep belief network*, *fine tuning*, dan *confusion matrix*. DBN menunjukkan performa yang baik. Model klasifikasi berhasil memprediksi dengan benar sebanyak 98,5% dari total prediksi. Selain itu, model tersebut juga memiliki *sensitivity* dan *specificity* yang tinggi, sehingga dapat mendeteksi kanker payudara ganas dan kanker payudara jinak dengan baik.

Kata Kunci—*Deep Belief Network, Deep Learning, Fine Tuning, Kanker Payudara*

I. PENDAHULUAN

Penyakit kanker payudara merupakan salah satu masalah kesehatan global yang mendesak, karena penyakit ini merupakan penyakit mematikan yang dapat mempengaruhi jutaan Perempuan di seluruh dunia. Menurut data organisasi kesehatan dunia (WHO), kanker payudara merupakan jenis kanker yang paling umum terjadi di kalangan perempuan, dengan tingkat prevalensi yang terus meningkat [1]. Keberhasilan deteksi dini sangat penting dalam meningkatkan prognosis dan keselamatan pasien, sehingga pasien dapat mengantisipasi sejak dini. Deteksi dini memberikan peluang yang lebih baik untuk antisipasi dan pengobatan yang lebih efektif.

Dalam upaya meningkatkan akurasi deteksi dini, teknologi di bidang kecerdasan buatan, seperti *Deep Belief Network* (DBN), telah menjadi sorotan algoritma potensial. Deep Belief Network sendiri secara prinsip dapat dikatakan berada di antara penelitian pada bidang pemrosesan sinyal, jaringan syaraf tiruan, pemodelan grafik, optimasi dan pengenalan pola [2]. Metode ini setidaknya memiliki dua fitur penting yaitu greedy learning strategis, yang merupakan efisiensi setiap lapisan untuk menginisialisasi jaringan, dan fine tuning semua bobot yang diinginkan [2]. Penggunaan *Deep Belief Network* sebagai algoritma untuk mendeteksi penyakit kanker payudara

menjanjikan kemampuan untuk meningkatkan Tingkat akurasi, sensitivitas, dan spesifisitas diagnosis.

Data yang digunakan adalah Database “*breast cancer wisconsin*” yang bersumber dari laman *uc Irvine machine learning repository*. Variabel yang digunakan dalam analisis adalah *clump thickness*, *uniformity of cell shape*, *uniformity of cell shape*, *marginal adhesion*, *single epithelial cell size*, *bare nuclei*, *bland chromatin*, *normal nucleoli*, dan *class target binary*.

Pemanfaatan metode *deep belief learning* ke dalam proses deteksi kanker payudara, diharapkan dapat memberikan kontribusi positif dalam upaya global untuk menanggulangi beban kesehatan yang ditimbulkan oleh kanker payudara. dapat disesuaikan dengan keperluan Anda.

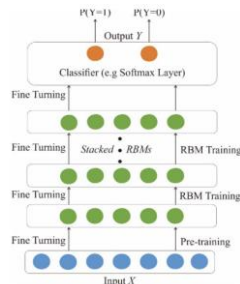
II. TINJAUAN PUSTAKA

A. Machine Learning

Machine learning dapat didefinisikan sebagai aplikasi komputer dan algoritma matematika yang diadopsi dengan cara pembelajaran yang berasal dari data dan pembelajaran yang berasal dari data dan menghasilkan prediksi di masa yang akan datang. Adapun proses pembelajaran yang dimaksud adalah suatu usaha dalam memperoleh kecerdasan yang melalui dua tahap antara lain latihan (*training*) dan pengujian (*testing*) [4].

B. Deep Belief Network

Deep Belief Network (DBN) adalah suatu pengembangan dari *Deep Learning* yang merupakan tumpukan atau *stack* dari beberapa algoritma atau metode yang bertujuan *feature extraction* yang memanfaatkan seluruh *resource* seoptimal mungkin. *Deep Learning* mencakup algoritma *unsupervised* dan *supervised learning* sehingga dapat memanfaatkan data yang berlabel maupun tidak berlabel. Pendekatan yang sering digunakan untuk mengimplementasikan *Deep Learning* adalah *graphical methods* atau *multi layer representation* atau *multi layer graphical model* seperti *Deep Belief Network* [2]. Berikut ini merupakan contoh proses *backpropagation* untuk menentukan bobot dari setiap proses.



Gambar 1 DBN Architectur

Parameter untuk *Restricted Boltzmann Machine* (RBM) dilatih oleh algoritma *Contrastive Divergence* (CD). CD adalah pembelajaran tanpa supervisi dan data berlabel tidak diperlukan pada tahap ini. Pada tahap kedua, jaringan yang telah dilatih sebelumnya akan disesuaikan dengan model pembelajaran terawasi seperti softmax/logistic regression atau backpropagation. Namun, parameter *deep belief network* hampir tetap setelah CD, dan tahap kedua hanya mengatur parameter model [6].

C. Fine Tuning

Dalam penelitian ini, *backpropagation* dan *softmax* digunakan dalam penyempurnaan DBN.

1. Backpropagation

Backpropagation digunakan untuk menyesuaikan bobot dengan prinsip rantai dalam model kesalahan. Kesalahan akan disebarkan dari lapisan terakhir ke lapisan pertama. Dua parameter utama di sini adalah ukuran batch dan jumlah zaman. Dalam penyempurnaan, sampel pelatihan dibagi menjadi beberapa kelompok dengan ukuran yang sama [5]. Berikut ini merupakan turunan untuk lapisan akhir

$$\frac{\partial L}{\partial W_{ij}} = \delta_j \frac{\partial \mu_j}{\partial W_{ij}} + \delta_j \frac{\partial p_j^2}{\partial W_{ij}} \quad (1)$$

$$\delta_j = \frac{\partial L}{\partial \mu_j} = \frac{\partial L}{\partial O_j} \frac{\partial o_j}{\partial n_j} \frac{\partial n_j}{\partial \mu_j} \quad (2)$$

$$\tau_j = \frac{\partial L}{\partial p_j^2} = \frac{\partial L}{\partial O_j} \frac{\partial o_j}{\partial n_j} \frac{\partial n_j}{\partial p_j^2} \quad (3)$$

Keterangan :

L : Fungsi Kerugian

O_j: Keluaran dari Simpul j-th

Masukkan algoritma kedalam fungsi 19

Untuk lapisan tersembunyi, turunan dapat dituliskan sebagai berikut

$$\frac{\partial L}{\partial W_{ij}} = \delta_j \frac{\partial \mu_j}{\partial W_{ij}} + \tau_j \frac{\partial p_j^2}{\partial W_{ij}} \quad (4)$$

$$\delta_j = a_j \frac{\partial n_j}{\partial \mu_j} \quad (5)$$

$$\tau_j = a_j \frac{\partial n_j}{\partial p_j^2} \quad (6)$$

$$a_j = \left[\sum_k \left\{ \delta_k \frac{\partial \mu_k}{\partial O_j} + \frac{\partial p_k^2}{\partial O_j} \right\} \right] \frac{\partial O_j}{\partial n_j} \quad (7)$$

2. Softmax Regression

Regresi Softmax [3] adalah perluasan dari regresi logistik

regresi, yang dapat diterapkan pada lebih dari dua kelas. Probabilitas dari $Y=I$ yang diberikan output RBM X' , matriks koefisien C dan intersepsi d ditunjukkan pada Gambar 4.2. matriks koefisien C dan intersepsi d ditunjukkan pada Persamaan 24. Prediksi akhir adalah kelas dengan probabilitas terbesar dalam Persamaan 25.

$$P(Y = I | X', C, d) = \frac{e^{C I^{Tx'} + d I}}{\sum_k e^{C k^{Tx'} + d k}} \quad (8)$$

$$\text{Prediksi} = \max_{I \in L\{P(Y=I|X', C, d)\}} \quad (9)$$

D. Confusion Matrix

Confusion Matrix adalah metode untuk mengetahui informasi yang berisi data aktual dan data prediksi dari hasil klasifikasi hasil klasifikasi yang dilakukan oleh sistem. Dalam klasifikasi, diharapkan dapat mengklasifikasikan data dengan tepat, dan memiliki hasil yang baik dengan kesalahan yang kecil. Oleh karena itu, metode ini ada untuk membantu mengetahui seberapa keberhasilan suatu klasifikasi [4]. Terdapat tiga hasil yang merupakan hasil utama dari *confusion matrix*, dan hasil tersebut adalah akurasi, spesifikasi, dan sensitivitas. Untuk mendapatkan hasil tersebut, hal-hal yang perlu diidentifikasi adalah True Positive (TP), False Positive (FP), False Negatif (FN), dan True Negatif (TN) [6] [7]. Tabel confusion matrix dapat dilihat pada Tabel 1.

Tabel confusion matrix.

Tabel 1.		
Klasifikasi Hasil Confussion Matrix		
Aktual	Klasifikasi	
	+	-
+	True Positive (TP)	False Negative (FN)
-	False Positive (FP)	True Negative (TN)

Setelah mendapatkan parameter TP, FN, FP, dan TN, maka dilakukan perhitungan untuk mencari akurasi, spesifikasi, dan sensitivitas yang dapat dilihat pada Persamaan 4,5,6.

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \quad (10)$$

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

$$\text{Spesifikasi} = \frac{TN}{TN + FP} \quad (12)$$

III. METODOLOGI PENELITIANBAB

A. Jenis Penelitian

Penelitian mengenai deteksi kanker payudara dengan metode *Deep Belief Network* (DBN) merupakan penelitian deksripsi kuantitatif. Proses penelitian ini melibatkan perhitungan matematis dan menghasilkan analisis deteksi kanker payudara. Tujuan dari penelitian ini nantinya dapat memberikan kontribusi untuk membantu pihak medis dalam melakukan identifikasi dini terhadap penyakit kanker payudara

B. Sumber Data dan Variabel Penelitian

Data yang digunakan merupakan data sekunder yaitu dataset “*breast cancer Wisconsin*” yang bersumber dari website *Breast Cancer Wisconsin (Original) - UCI Machine Learning Repository*. Data yang diperoleh telah dilakukan proses augmentasi menjadi 678 data. Data ini akan dibagi menjadi data training dan data testing dengan jumlah masing-masing 468 dan 200 data.

Variabel penelitian yang digunakan untuk penelitian ini adalah sebagai berikut

Tabel 2. Variabel Penelitian

Variabel	Keterangan	Kategori	Skala
Y	Cancer	2 = Begign 4 = Malignant	Nominal
X ₁	Clump Thickness	-	Rasio
X ₂	Uniformity of Cell Size	-	Rasio
X ₃	Uniformity of Cell Shape	-	Rasio
X ₄	Marginal Adhesion	-	Rasio
X ₅	Single Epithelial Cell Size	-	Rasio
X ₆	Bare Nuclei	-	Rasio
X ₇	Normal Nucleoli	-	Rasio

C. Langkah Analisis

Langkah-langkah pengerjaan penelitian ini terdiri dari dua tahapan besar, yaitu algoritma DBN dan tahap *fine-tuning*. Langkah penyelesaian tahapan-tahapan tersebut adalah sebagai berikut.

a. Algoritma Deep Belief Network (DBN)

1. Baca dataset: program membaca dataset dengan format .csv dengan menggunakan fungsi yang telah disediakan oleh *Python*.
2. Normalisasi: bagian ini adalah proses mengubah rentang nilai pada setiap kolom agar seragam. Normalisasi dilakukan dengan menggunakan metode normalisasi Min-Max yaitu mengurangi nilai minimum dari setiap kolom dan membaginya dengan selisih antara nilai maksimum dan minimum. Hasil normalisasi disimpan kembali dalam *DataFrame*. Dataset yang telah dinormalisasi ke dalam file CSV dengan *path* yang ditentukan oleh variabel ``path``.
3. *Positive phase*: merupakan proses mendapatkan nilai *hidden unit* (*h*) dengan menambah nilai bias (*b*) dengan jumlah hasil perkalian setiap input (*x*) dengan *weights*-nya (*w*) masing-masing pada setiap *layer* dan kemudian diaktivasi dengan fungsi aktivasi sigmoid (σ).
4. *Negative phase*: merupakan proses untuk membarui nilai input atau *visible units* (*v*) yang dipakai untuk mendapatkan nilai *hidden units* sekali lagi.
5. *Update weights*: merupakan proses memperbarui *weights* (*w*) dengan menambah *weights* lama dengan *learning rate* (*l*) dikali selisih antara hasil *positive phase* dan *negative phase*.
6. Proses tersebut dijalankan sebanyak jumlah *layer* DBN yang telah di-set sebelumnya, jika proses belum berjalan sebanyak jumlah *layer*, maka proses akan kembali diulang dari tahap *positive phase* sampai *update weights*. Jika sudah berjalan sebanyak jumlah *layer*, maka tahap

selanjutnya adalah melakukan *fine-tuning*.

b. *Fine-tuning* (Algoritma *Backpropagation*)

Tahap ini bertujuan untuk menghasilkan *output* yang merupakan prediksi dari data input. Berikut merupakan langkah-langkah dalam algoritma *backpropagation*.

1. Ambil *weights* terakhir untuk *training backpropagation*: setiap *weights* (*w*) yang didapat dari setiap layer pada DBN yang telah dijalankan sebelumnya untuk dijadikan bobot pada *backpropagation*.
2. *Forward propagation*: tahap ini adalah untuk mendapatkan nilai *hidden layer* dan *output layer* dengan menambah nilai bias (*b*) dengan jumlah hasil perkalian setiap input (*x*) dengan *weights*-nya (*w*) masing-masing pada setiap *layer* dan kemudian diaktivasi dengan fungsi aktivasi sigmoid (σ).
3. *Backward propagation*: proses ini bertujuan untuk menghitung *error* (δ) dari setiap unit, yang dimana nilainya akan digunakan untuk mendapatkan nilai perubahan *weights* (Δw) dan bias (Δb).
4. *Update weights & bias*: proses ini bertujuan untuk membarui nilai *weights* (*w*) dan bias (*b*) dengan menambahkan nilai *weights* lama dan bias lama dengan masing-masing perubahan yang telah didapatkan setelah menghitung nilai *error* (δ) setiap unit.
5. Proses tersebut diulangi sebanyak jumlah iterasi yang telah di-set sebelumnya, jika proses belum berjalan sebanyak jumlah iterasi, maka proses akan kembali diulang mulai dari tahap *forward propagation* sampai *update weights & bias*. Jika sudah berjalan sebanyak jumlah iterasi, maka tahap selanjutnya adalah melakukan prediksi data.
6. Prediksi data: tahapan ini merupakan tahap evaluasi di mana proses yang dijalankan sama seperti proses *forward propagation*, tetapi dengan nilai *weights* dan bias baru yang telah didapatkan dari tahapan *training*. Tahap ini mengevaluasi algoritma dengan menggunakan data *non-training*, yaitu data di luar yang dipakai pada tahapan *training*.

IV. ANALISIS DAN PEMBAHASAN

Hasil dari penelitian ini adalah akurasi klasifikasi yang bertujuan untuk mengukur keberhasilan identifikasi. Semakin tinggi akurasi yang dihasilkan, semakin besar keberhasilan identifikasi. Selanjutnya, pengujian dalam penelitian ini dibagi menjadi dua, yaitu: *training* dan evaluasi. Bagian *training* untuk melakukan *training* algoritma dan evaluasi pada tahap *fine-tuning*. Nilai *epoch* & *learning rate* RBM, *layer* DBN, dan *layer backpropagation* untuk tahap *training* dapat identifikasi melibatkan dua tahap, yaitu *pre-processing* dan klasifikasi.

A. Implementasi Program

a. Positive phase

Gambar 2. Menunjukkan implementasi *positive phase* untuk *meng-update hidden node* pada *code* dengan menggunakan fungsi `np.dot` untuk melakukan perkalian antara *array* input dan masing-masing *weights*-nya dan ditambah dengan bias, kemudian dilakukan aktivasi dengan menggunakan fungsi `sigmoid ()`.

Gambar 2. Positive Phase

```
pos_hidden_activations = np.dot(train_data, weights)
pos_hidden_probs = sigmoid(pos_hidden_activations)
```

b. Negative phase

Gambar 3. Menunjukkan implementasi *negative phase* untuk *meng-update visible node* pada *code* dengan menggunakan fungsi `np.dot` untuk melakukan perkalian antara hasil dari *positive phase* dan masing-masing *weights*-nya, kemudian dilakukan aktivasi dengan menggunakan fungsi `sigmoid ()`.

Gambar 3. Negative Phase

```
neg_visible_activations = np.dot(pos_hidden_states, weights.T)
neg_visible_probs = sigmoid(neg_visible_activations)
```

c. Update weights

Implementasi *update weights* dapat dilihat pada gambar 4.

Gambar 4 Update Weight

```
# Update weights.
weights += learning_rate * ((pos_associations - neg_associations) / num_examples)
```

B. Hasil Pengujian

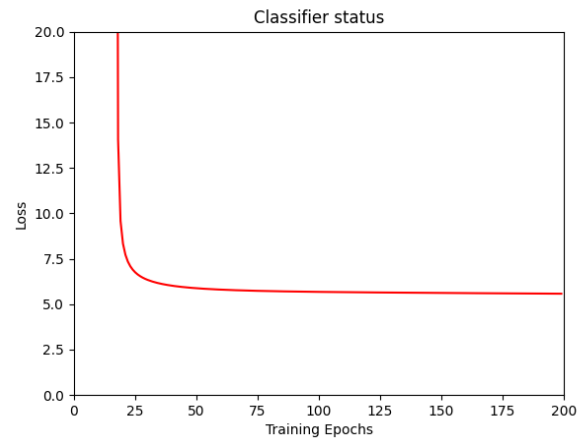
Pengujian dalam penelitian ini dibagi menjadi dua, yaitu *training* dan evaluasi. Bagian *training* untuk melakukan *training* algoritma dan evaluasi pada tahap *fine-tuning*. Nilai *epoch* & *learning rate* RBM, *layer* DBN, dan *layer backpropagation* untuk tahap *training* dapat dilihat pada Gambar 6.

Gambar 5. Epoch dan Layer Tahap Training

```
train_network(network=model, train=dataset, l_rate=0.1, n_epoch=200)
```

Hasil *error* dalam *training* DBN untuk setiap *epoch* yang merepresentasikan *output* adalah:

- Epoch = 0 ; *error* = 83,437
- Epoch = 25 ; *error* = 6,753
- Epoch = 50 ; *error* = 5,879
- Epoch = 75 ; *error* = 5,736
- Epoch = 100 ; *error* = 5,678
- Epoch = 125 ; *error* = 5,645
- Epoch = 150 ; *error* = 5,620
- Epoch = 175 ; *error* = 5,598
- Epoch = 199 ; *error* = 5,579



Gambar 6. Grafik Tingkat Error DBN

C. Klasifikasi

Confusion matrix adalah tabel yang digunakan untuk mengevaluasi kinerja model klasifikasi. Tabel ini menunjukkan hubungan antara prediksi model dengan hasil sebenarnya. Dalam kasus ini, confusion matrix menunjukkan hasil prediksi model klasifikasi terhadap dua kelas, yaitu:

- Malignant: kanker payudara ganas
- Benign: kanker payudara jinak

Tabel 3.

Confusion Matrix		
	Predicted Malignant	Predicted Benign
Actual Maglinant	64	1
Actual Benign	2	133

Tabel 1. merupakan tabel *confusion matrix* yang menunjukkan kinerja model klasifikasi secara keseluruhan. Pada Tabel 1. dapat diketahui bahwa 64 kasus diprediksi dengan benar sebagai *Maglinant* dan 133 sebagai *Benign*. Terdapat satu kasus dimana *Benign* diprediksi tetapi sebenarnya *Maglinant* dan dua kasus dimana *Maglinant* diprediksi tetapi sebenarnya *Benign*.

Tabel 4.

Hasil Akurasi, Sensitivity, Specificity, dan F1-Measure

	Nilai
Accuracy	98,5%
Sensitivity	98,46%
Specificity	98,52%
F1-measure	98,49%

Accuracy dihitung dengan membagi jumlah prediksi yang benar dengan total jumlah prediksi. Dalam kasus ini, classifier *accuracy* adalah 98,5%. Artinya, dari 137 prediksi, model berhasil memprediksi dengan benar sebanyak 135 prediksi.

Sensitivity adalah ukuran kinerja model dalam mendeteksi kelas positif. *Sensitivity* dihitung dengan membagi jumlah prediksi positif yang benar dengan total jumlah kelas positif. Dalam kasus ini, *sensitivity* adalah 98,46%. Artinya, dari 65 kanker payudara ganas, model berhasil memprediksi dengan benar sebanyak 64 kanker payudara ganas.

Specificity adalah ukuran kinerja model dalam mendeteksi kelas negatif. *Specificity* dihitung dengan membagi jumlah prediksi negatif yang benar dengan total jumlah kelas negatif. Dalam kasus ini, *specificity* adalah 98,52%. Artinya, dari 72 kanker payudara jinak, model berhasil memprediksi dengan benar sebanyak 71 kanker payudara jinak.

F1-measure adalah ukuran kinerja model yang menggabungkan *accuracy* dan *sensitivity*. *F1-measure* dihitung dengan membagi 2 kali hasil perkalian antara *accuracy* dan *sensitivity* dengan hasil penjumlahan antara *accuracy* dan *sensitivity*. Dalam kasus ini, *F1-measure* adalah 98,49%.

V. KESIMPULAN DAN SARAN

Berdasarkan hasil perancangan, implementasi, dan pengujian maka dapat diambil beberapa kesimpulan mengenai algoritma *deep belief network* (DBN) dalam mendeteksi penyakit kanker payudara, yaitu:

- a. Proses pendeteksian penyakit kanker payudara menggunakan DBN dilakukan dengan beberapa langkah, yaitu persiapan data, *training* DBN, dan evaluasi dengan *backpropagation*.
- b. DBN menunjukkan performa yang baik. Model klasifikasi berhasil memprediksi dengan benar sebanyak 98,5% dari total prediksi. Selain itu, model tersebut juga memiliki *sensitivity* dan *specificity* yang tinggi, sehingga dapat mendeteksi kanker payudara ganas dan kanker payudara jinak dengan baik.

Saran yang dapat diberikan untuk penelitian lebih lanjut adalah melakukan perbandingan model klasifikasi menggunakan metode machine learning lainnya agar dapat mendapatkan hasil yang terbaik. Dalam bidang medis, potensi penerapan model ini untuk mendukung diagnosis sejak dini perlu dieksplorasi lebih lanjut, sehingga bisa membuka peluang baru untuk perawatan yang lebih efektif, oleh karena itu pembaca diharapkan dapat merespon dan mendiskusikan potensi aplikasi di dalam bidang Kesehatan.

UCAPAN TERIMA KASIH

Penulis Wahyu Novitasari, Hafidz Dinilah Alfaqih dan Adistyari Husna mengucapkan terima kasih kepada dosen pengampu mata kuliah machine learning yang telah memberikan kesempatan dan dukungan untuk dapat mempelajari mata kuliah dengan baik.

DAFTAR PUSTAKA

- [1] H. Lasari, M. Amalia dan Sarmila, "Higeia Journal of Public Health Research and Development," 2021.
- [2] D. L dan Y. D, *Deep Learning: Methods and Application, foundation and trends in Signal Processing*, USA: Redmond, 2013.
- [3] G. Yanming, Y. Liu, A. Oerlemans, S. Laoi, S. Wu dan M. S. Lew, *Deep Learning for Visual Understanding*, USA: Neurocomputing, 2016.
- [4] Q.-Y. Zhu, C. Kheong Siew dan G. Bin Huang, *Extreme Learning Machine Theory and Applications*, vol. 70, USA: Neurocomputing, 2016.
- [5] B. Gopika, K. Srilaxmi, D. Alekhya, B. B. Rao, and B. Rama Mohan, "Face Recognition by Using Gabor Feature Extraction and Neural Networks," *IOSR J. Electron. Commun. Eng. Ver. II*, vol. 10, no. 2, pp. 2278–2834, 2015.
- [6] W. Feng and K. Kunkle, "A Deep Belief Network Based Machine Learning System for Risky Host Detection."
- [7] S. Visa, B. Ramsay, A. Ralescu, and E. Van Der Knaap, "Confusion matrix-based feature selection," *CEUR Workshop Proc.*, vol. 710, pp. 120–127, 2011.
- [8] D. A. Adyanti et al., "Analisis Citra Dental Panoramic Radiograph (Dpr) Pada Tulang Mandibula Untuk Deteksi Osteoporosis Menggunakan Metode GLCM – SVM Multiclass (Gray Level CoOccurrence Matrix – Support Vector Machine Multiclass)," 2018.
- [9] A. Z. Foeady, D. C. R. Novitasari, A. H. Asyhar, and M. Firmansjah, "Automated Diagnosis System of Diabetic Retinopathy Using GLCM Method and SVM Classifier," 2018 5th Int. Conf. Electr. Eng. Comput. Sci. Informatics, pp. 154–160, 2019.