

## DAFTAR PUSTAKA

- Adugna, T., Xu, W., & Fan, J. (2022). Comparison of Random Forest and Support Vector Machine Classifiers for Regional Land Cover Mapping Using Coarse Resolution FY-3C Images. *Remote Sensing*, 14(3), Article 3. <https://doi.org/10.3390/rs14030574>
- Al Amrani, Y., Lazaar, M., & El Kadiri, K. E. (2018). Random Forest and Support Vector Machine based Hybrid Approach to Sentiment Analysis. *Procedia Computer Science*, 127, 511–520. <https://doi.org/10.1016/j.procs.2018.01.150>
- Alsayat, A. (2022). Improving Sentiment Analysis for Social Media Applications Using an Ensemble Deep Learning Language Model. *Arabian Journal for Science and Engineering*, 47(2), 2499–2511. <https://doi.org/10.1007/s13369-021-06227-w>
- Arista, W., & Rusmini, R. (2022). PENEGAKAN HUKUM TERHADAP LAYANAN APLIKASI PINJAMAN ONLINE DI MASA PANDEMI DI KOTA PALEMBANG SEBAGAI UPAYA PERLINDUNGAN KONSUMEN. *Lex Librum: Jurnal Ilmu Hukum*, 9(1), Article 1. <https://doi.org/10.46839/ljih.v9i1.660>
- Baiti, F. N., & Iswandi, I. (2022). ANALISIS TRANSAKSI KREDIT SECARA ONLINE PADA APLIKASI AKULAKU DALAM PERSPEKTIF HUKUM POSITIF DAN HUKUM ISLAM: Studi Pada Mahasiswa IAI AL-AZIS. *Jurnal Penelitian Multidisiplin Ilmu*, 1(3), Article 3. <https://doi.org/10.59004/metta.v1i3.169>
- Cholissodin, I., & Riyandani, E. (2016). *Swarm Intelligence*.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297. <https://doi.org/10.1007/BF00994018>
- Guo, R., Zhao, Z., Wang, T., Liu, G., Zhao, J., & Gao, D. (2020). Degradation state recognition of piston pump based on ICEEMDAN and XGBoost. *Applied Sciences*, 10, 6593. <https://doi.org/10.3390/app10186593>
- Hall, K., Chang, V., & Jayne, C. (2022). A review on Natural Language Processing Models for COVID-19 research. *Healthcare Analytics*, 2, 100078. <https://doi.org/10.1016/j.health.2022.100078>
- Hamsyah, F., & Sulistyowati, E. (2022). ANALISIS YURIDIS PENCANTUMAN KLAUSULA BAKU PADA SYARAT DAN KETENTUAN PERJANJIAN PINJAMAN ONLINE PT ARTHA DANA TEKNOLOGI. *NOVUM: JURNAL HUKUM*, 121–130. <https://doi.org/10.2674/novum.v0i0.44452>

- Hidayat, A., Azizah, N., & Ridwan, M. (2022). Pinjaman Online dan Keabsahannya Menurut Hukum Perjanjian Islam. *Jurnal Indragiri Penelitian Multidisiplin*, 2(1), 1–9. <https://doi.org/10.58707/jipm.v2i1.115>
- Kamil, I., Anggraini, D., & Prihanto, H. (2022). Sosialisasi Financial Technology (Fintech): Cerdas Dalam Memilih Dan Memutuskan Penggunaan Aplikasi Pinjaman Dana Online. *Jurnal Pustaka Dianmas*, 2(1), Article 1.
- Li, Z., Zhang, Q., Wang, Y., & Wang, S. (2020). Social Media Rumor Refuter Feature Analysis and Crowd Identification Based on XGBoost and NLP. *Applied Sciences*, 10, 4711. <https://doi.org/10.3390/app10144711>
- Manconi, A., Armano, G., Gnocchi, M., & Milanese, L. (2022). A Soft-Voting Ensemble Classifier for Detecting Patients Affected by COVID-19. *Applied Sciences*, 12(15), Article 15. <https://doi.org/10.3390/app12157554>
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space* (arXiv:1301.3781). arXiv. <http://arxiv.org/abs/1301.3781>
- Mochtar, H. N., & Rusdiana, E. (2022). ANALISIS YURIDIS PENYEBARAN DAN PENGGUNAAN DATA PRIBADI UNTUK KEPENTINGAN LAIN OLEH KREDITUR PINJAMAN ONLINE DAN/ATAU FINTECH P2P LENDING ADAKAMI DITINJAU DARI PERSPEKTIF HUKUM PIDANA. *NOVUM: JURNAL HUKUM*, 218–224. <https://doi.org/10.2674/novum.v0i0.47683>
- Nugroho, A. S. (2008). *SUPPORT VECTOR MACHINE: PARADIGMA BARU DALAM SOFTCOMPUTING*.
- Oliveira, G. P. de, Fonseca, A., & Rodrigues, P. C. (2022). Diabetes diagnosis based on hard and soft voting classifiers combining statistical learning models. *Brazilian Journal of Biometrics*, 40(4), Article 4. <https://doi.org/10.28951/bjb.v40i4.605>
- Özçift, A. (2020). *MEDICAL SENTIMENT ANALYSIS BASED ON SOFT VOTING ENSEMBLE ALGORITHM*.
- Peppes, N., Daskalakis, E., Alexakis, T., Adamopoulou, E., & Demestichas, K. (2021). Performance of Machine Learning-Based Multi-Model Voting Ensemble Methods for Network Threat Detection in Agriculture 4.0. *Sensors (Basel, Switzerland)*, 21(22), 7475. <https://doi.org/10.3390/s21227475>
- Rahutomo, F., Ikawati, D. S. E., & Rohman, O. A. (2019). *EVALUASI FITUR WORD2VEC PADA SISTEM UJIAN ESAI ONLINE. 04*.

- Rodrigues, A. P., Fernandes, R., A. A., B. A., Shetty, A., K. A., Lakshmana, K., & Shafi, R. M. (2022). Real-Time Twitter Spam Detection and Sentiment Analysis using Machine Learning and Deep Learning Techniques. *Computational Intelligence and Neuroscience*, 2022, e5211949. <https://doi.org/10.1155/2022/5211949>
- Rong, X. (2016). *Word2vec Parameter Learning Explained* (arXiv:1411.2738). arXiv. <http://arxiv.org/abs/1411.2738>
- Rosiwan, G. M. A., & Lasmanah. (2022). Pengaruh Kepercayaan dan Kemudahan Pengguna terhadap Penggunaan Sistem Kredit Online Pada Aplikasi Kredivo di Masa Pandemi Covid-19 di Kota Bandung. *Bandung Conference Series: Business and Management*, 2(1), Article 1. <https://doi.org/10.29313/bcsbm.v2i1.306>
- Samih, A., Ghadi, A., & Fennan, A. (2023). Enhanced sentiment analysis based on improved word embeddings and XGboost. *International Journal of Electrical and Computer Engineering*, Vol.13, 1827–1836. <https://doi.org/10.11591/ijece.v13i2.pp1827-1836>
- Sandag, G. A. (2020). Prediksi Rating Aplikasi App Store Menggunakan Algoritma Random Forest. *CogITO Smart Journal*, 6(2), Article 2. <https://doi.org/10.31154/cogito.v6i2.270.167-178>
- Sasongko, T. B. (2016). Komparasi dan Analisis Kinerja Model Algoritma SVM dan PSO-SVM (Studi Kasus Klasifikasi Jalur Minat SMA). *Jurnal Teknik Informatika Dan Sistem Informasi*, 2(2), Article 2. <https://doi.org/10.28932/jutisi.v2i2.627>
- Shafila, 16611022 Gregy Addis. (2020). *Implementasi Metode Extreme Gradient Boosting (XGBOOST) untuk Klasifikasi pada Data Bioinformatika (Studi Kasus: Penyakit Ebola, GSE122692)*. <https://dspace.uui.ac.id/handle/123456789/29276>
- Shaik, T., Tao, X., Li, Y., Dann, C., McDonald, J., Redmond, P., & Galligan, L. (2022). A Review of the Trends and Challenges in Adopting Natural Language Processing Methods for Education Feedback Analysis. *IEEE Access*, 10, 56720–56739. <https://doi.org/10.1109/ACCESS.2022.3177752>
- Shami, T. M., El-Saleh, A. A., Alswaitti, M., Al-Tashi, Q., Summakieh, M. A., & Mirjalili, S. (2022). Particle Swarm Optimization: A Comprehensive Survey. *IEEE Access*, 10, 10031–10061. <https://doi.org/10.1109/ACCESS.2022.3142859>
- Srivastava, D. K., & Bhambhu, L. (2005). *DATA CLASSIFICATION USING SUPPORT VECTOR MACHINE*.

- Sukhavasi, V. (2021). *A Comprehensive Survey of Sentiment Analysis: Techniques, Applications and Open Problems*.
- Suryati, E., Styawati, S., & Aldino, A. A. (2023). Analisis Sentimen Transportasi Online Menggunakan Ekstraksi Fitur Model Word2vec Text Embedding Dan Algoritma Support Vector Machine (SVM). *Jurnal Teknologi Dan Sistem Informasi*, 4(1), Article 1. <https://doi.org/10.33365/jtsi.v4i1.2445>
- Vohra, M. S., & Teraiya, J. (2013). Applications and Challenges for Sentiment Analysis: A Survey. *International Journal of Engineering Research*, 2(2).
- Widjaja, G. (2022). PEMAHAMAN KONSUMEN TENTANG PINJAMAN ONLINE (PINJOL) DI JAKARTA. *Jurnal Pengabdian Kepada Masyarakat*, 2(2), Article 2.
- Willy, W., Rini, D. P., & Samsuryadi, S. (2021). Perbandingan Algoritma Random Forest Classifier, Support Vector Machine dan Logistic Regression Clasifier Pada Masalah High Dimension (Studi Kasus: Klasifikasi Fake News). *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 5(4), 1720. <https://doi.org/10.30865/mib.v5i4.3177>
- Yulianti, S. E. H., Soesanto, O., & Sukmawaty, Y. (2022). Penerapan Metode Extreme Gradient Boosting (XGBOOST) pada Klasifikasi Nasabah Kartu Kredit. *Journal of Mathematics: Theory and Applications*, 21–26. <https://doi.org/10.31605/jomta.v4i1.1792>

## LAMPIRAN

### Lampiran 1. Hasil Prediksi Model

content	sentimentSVM	sentimentXGB	sentimentRF	sentimentSV
okkeeeeee	0	2	0	0
limit nya terlalu kecil	2	0	1	1
akulaku memang mantap	2	0	2	2
mudah dan terpercaya	2	0	2	2
semoga pelayanannya cepat dan sukses selalu ke...	2	0	2	2
ksh bintang ojasoalnya ribet amat sih mau daft...	2	2	2	2
limit masih tidak bisa digunakan	1	0	1	1
sangat membantu	0	2	0	0
mantap dan sangat membantuterimakasih akulaku	2	0	2	2
terima kasi atas respon nyayg terbaik	2	2	1	2
keren sih udh sering banget ambil di akulaku b...	2	2	2	2
sangat membantu	0	2	0	0
terimakasih akulaku	2	2	2	2
lumayan membantu	2	2	1	2
baik sukses selalu	2	2	2	2
bagus banget	2	2	2	2

## Lampiran 2. Preprocessing Data

```

# Proses Case Folding
df['content'] = df ['content'].str.lower()

#Proses filter
def filtering_text(text):
    # mengubah tweet menjadi huruf kecil
    # text = text.lower()
    # menghilangkan url
    text = re.sub(r'https?:\V\S+', "", str(text))
    # menghilangkan mention, link, hastag
    text = ' '.join(re.sub("([@#][A-Za-z0-9+])|(\w+:\V\S+)", " ", text).split())
    #menghilangkan karakter byte (b')
    text = re.sub(r'(b'\{1,2})', "", text)
    # menghilangkan yang bukan huruf
    text = re.sub('[^a-zA-Z]', '', text)
    # menghilangkan digit angka
    text = re.sub(r'\d+', "", text)
    #menghilangkan tanda baca
    text = text.translate(str.maketrans("", "", string.punctuation))
    # menghilangkan whitespace berlebih
    text = re.sub(r"\s+", ' ', text).strip()
    return text

#Proses stopwords dan stemming
def stop_stem(text):
    #stopword
    with open('kamus.txt') as kamus:
        word = kamus.readlines()
        list_stopword = [line.replace("\n", "") for line in word]
    dictionary = ArrayDictionary(list_stopword)
    stopword = StopWordRemover(dictionary)
    text = stopword.remove(text)
    # stemming
    factory_stemmer = StemmerFactory()

```

```
stemmer = factory_stemmer.create_stemmer()
text = stemmer.stem(text)
return text

# tokenisasi
def word_tokenize_wrapper(text):
    return word_tokenize(text)
df['content_tokens'] = df['cleaned'].apply(word_tokenize_wrapper)

kamus_normalisasi = pd.read_csv("slang.csv")

kata_normalisasi_dict = {}

for index, row in kamus_normalisasi.iterrows():
    if row[0] not in kata_normalisasi_dict:
        kata_normalisasi_dict[row[0]] = row[1]

def normalisasi_kata(document):
    return [kata_normalisasi_dict[term] if term in kata_normalisasi_dict else term for
            term in document]

df['normalisasi'] = df['content_tokens'].apply(normalisasi_kata)

df['normalisasi'].head(10)

# stemming
stop=set(stopwords.words("indonesian"))
stop_bow = set(stopwords.words("indonesian"))
stop_bow.discard("tidak")
factory = StemmerFactory()
stemmer = factory.create_stemmer()
```

**Lampiran 3.** *Class* Random Forest

```

class RandomForestClassifier:
    def __init__(self, n_estimators=100, max_depth=None, min_samples_split=2, r
andom_state=None):
        self.n_estimators = n_estimators
        self.max_depth = max_depth
        self.min_samples_split = min_samples_split
        self.random_state = random_state
        self.trees = []

    def fit(self, X, y):
        np.random.seed(self.random_state)
        for i in range(self.n_estimators):
            # bootstrap sample
            idx = np.random.choice(X.shape[0], size=X.shape[0], replace=True)
            X_boot, y_boot = X[idx], y[idx]

            # train decision tree
            dt = DecisionTreeClassifier(max_depth=self.max_depth, min_samples_spl
it=self.min_samples_split)
            dt.fit(X_boot, y_boot)

            # save tree
            self.trees.append(dt)

    def predict(self, X):
        # make predictions for each tree
        predictions = np.array([tree.predict(X) for tree in self.trees])
        # calculate mode (most frequent class) for each sample
        return np.apply_along_axis(lambda x: Counter(x).most_common(1)[0][0], ax
is=0, arr=predictions)

```



**Lampiran 4. Class XGBoost**

```

class XGBoost:
    def __init__(self, n_trees=100, learning_rate=0.5, max_depth=13, reg_alpha=0,
reg_lambda=0):
        self.n_trees = n_trees
        self.learning_rate = learning_rate
        self.max_depth = max_depth
        self.reg_alpha = reg_alpha
        self.reg_lambda = reg_lambda
        self.trees = []
        self.f0 = None

    def fit(self, X, y):
        # Normalize the input features
        X = (X - X.mean(axis=0)) / X.std(axis=0)
        # Round the target labels to integer values
        y = np.round(y)
        # Convert y to integer type
        y = y.astype(int)
        # Convert y to one-hot encoding
        y_onehot = np.zeros((y.shape[0], np.max(y) + 1))
        y_onehot[np.arange(y.shape[0]), y] = 1
        # Initialize the model
        self.f0 = np.mean(y_onehot, axis=0)
        pred = np.full((X.shape[0], y_onehot.shape[1]), self.f0)
        for i in range(self.n_trees):
            # Compute the gradients of the loss function with respect to the predictions
            grad = y_onehot - pred
            trees = []
            for j in range(y_onehot.shape[1]):
                # Compute the gradients of the loss function with respect to the outputs
of the previous trees for each class
                grad_prev = grad[:, j] - self.reg_alpha * np.sign(pred[:, j]) -
self.reg_lambda * pred[:, j]

```

```

# Fit a new tree to the gradients of the current class
tree = DecisionTreeRegressor(max_depth=self.max_depth)
tree.fit(X, grad_prev)

# Add the new tree to the model for each class
trees.append(tree)

# Update the predictions for the current class
pred[:, j] += self.learning_rate * tree.predict(X)
self.trees.append(trees)
def predict_proba(self, X):
# Initialize the predictions
pred = np.full((X.shape[0], self.f0.shape[0]), self.f0)

# Compute the predictions by summing the outputs of each tree for each class
s
for i in range(self.n_trees):
    for j in range(len(self.trees[i])):
        pred[:, j] += self.learning_rate * self.trees[i][j].predict(X)

# Apply the softmax function to get the class probabilities
pred = np.exp(pred - np.max(pred, axis=1, keepdims=True))
return pred / np.sum(pred, axis=1, keepdims=True)

def predict(self, X):
# Predict the class with the highest probability
return np.argmax(self.predict_proba(X), axis=1)

```

### Lampiran 5. Class Support Vector Machine

```

def projection_simplex(v, z=1):
    p = len(v)
    u = np.sort(v)[::-1]
    cssv = np.cumsum(u) - z
    ind = np.arange(p) + 1
    cond = u - cssv / ind > 0
    rho = np.where(cond)[0][-1] + 1
    theta = (cssv[cond][-1] / rho)
    w = (v - theta)
    w[w < 0] = 0
    return w

class MyMulticlassSVM(BaseEstimator, ClassifierMixin):
    def __init__(self, C=1, max_iter=50, tol=0.05, kernel='linear',
                 random_state=None, verbose=0, gamma=1):
        self.C = C
        self.max_iter = max_iter
        self.tol = tol,
        self.kernel = kernel
        self.random_state = random_state
        self.verbose = verbose
        self.gamma = gamma

    def _partial_gradient(self, X, y, i):
        # Partial gradient for the ith sample.
        if self.kernel == 'linear':
            gamma = np.dot(X[i], self.coef_.T) + 1
        elif self.kernel == 'rbf':
            # compute RBF kernel
            K = np.exp(-self.gamma * np.sum((X - X[i])**2, axis=1))
            gamma = np.dot(K, self.dual_coef_.T) + 1
        gamma[y[i]] -= 1
        return gamma

```

```

def _violation(self, gamma, y, i):
    # Optimality violation for the ith sample.
    smallest = np.inf
    for k in range(gamma.shape[0]):
        if k == y[i] and self.dual_coef_[k, i] >= self.C:
            continue
        elif k != y[i] and self.dual_coef_[k, i] >= 0:
            continue
        smallest = min(smallest, gamma[k])
    return gamma.max() - smallest

def _solve_subproblem(self, gamma, y, norms, i):
    # Prepare inputs to the projection.
    Ci = np.zeros(gamma.shape[0])
    Ci[y[i]] = self.C
    beta_hat = norms[i] * (Ci - self.dual_coef_[:, i]) + gamma / norms[i]
    z = self.C * norms[i]
    # Compute projection onto the simplex.
    beta = projection_simplex(beta_hat, z)
    return Ci - self.dual_coef_[:, i] - beta / norms[i]

def fit(self, X, y):
    n_samples, n_features = X.shape
    # Normalize labels.
    self._label_encoder = LabelEncoder()
    y = self._label_encoder.fit_transform(y)
    # Initialize primal and dual coefficients.
    n_classes = len(self._label_encoder.classes_)
    self.dual_coef_ = np.zeros((n_classes, n_samples), dtype=np.float64)
    self.coef_ = np.zeros((n_classes, n_features))
    # Pre-compute norms.
    norms = np.sqrt(np.sum(X ** 2, axis=1))
    # Shuffle sample indices.

```

```

rs = check_random_state(self.random_state)
ind = np.arange(n_samples)
rs.shuffle(ind)
violation_init = None
for it in range(self.max_iter):
    violation_sum = 0
    for ii in range(n_samples):
        i = ind[ii]
        # All-zero samples can be safely ignored.
        if norms[i] == 0:
            continue
        g = self._partial_gradient(X, y, i)
        v = self._violation(g, y, i)
        violation_sum += v
        if v < 1e-12:
            continue
        delta = self._solve_subproblem(g, y, norms, i)
        self.coef_ += (delta * X[i][:, np.newaxis]).T
        self.dual_coef_[i] += delta
    if it == 0:
        violation_init = violation_sum
    vratio = violation_sum / violation_init
    if vratio < self.tol:
        if self.verbose >= 1:
            print("Converged")
        break
return self

def predict(self, X):
    decision = np.dot(X, self.coef_.T)
    pred = decision.argmax(axis=1)
    return self._label_encoder.inverse_transform(pred)

```

## Lampiran 6. Optimasi Bobot

```

# Fungsi objektif untuk optimisasi PSO
def objective_function(weights, X_train, y_train, X_test, y_test):
    # Normalisasi bobot
    weights /= np.sum(weights)

    # Melakukan prediksi probabilitas menggunakan masing-
    # masing model dan bobot
    ensemble = (weights[0] * rf.predict_proba(X_train) + weights[1] * xgb_clf .pre
    dict_proba(X_train) + weights[2] * svm.predict_proba(X_train))

    # Menghitung akurasi menggunakan bobot
    y_pred = np.argmax(ensemble, axis=1)
    accuracy = accuracy_score(y_train, y_pred)

    # Mengembalikan nilai negatif akurasi untuk optimisasi maksimum
    return -accuracy

# Membuat objek optimizer PSO
options = {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
optimizer = ps.single.GlobalBestPSO(n_particles=50, dimensions=3, options=opti
ons)

# Melakukan optimisasi PSO untuk mendapatkan bobot terbaik
best_cost, best_weights = optimizer.optimize(objective_function, iters=100, X_tra
in=X_train, y_train=y_train, X_test=X_test, y_test=y_test)

# Normalisasi bobot terbaik
best_weights /= np.sum(best_weights)

# Implementasi bobot ke model soft voting
ensemble = (best_weights[0] * rf.predict_proba(X_test) + best_weights[1] * xgb_
clf .predict_proba(X_test) + best_weights[2] * svm.predict_proba(X_test))
y_pred = np.argmax(ensemble, axis=1)

```

### Lampiran 7. Contoh Sederhana Cara Kerja SVM Menghasilkan Prediksi

→ Mencari nilai *hyperplane* terbaik

Dengan menggunakan persamaan (6) kita dapat menghitung *hyperplane* dengan sederhana. Misalkan terdapat data dibawah ini:

dim1	dim2	label
0.87	-0.86	1
1.76	0.12	1
0.33	0.44	0
-0.23	0.12	0
0.21	-0.82	2
0.45	-0.12	2

Langkah awal yang harus dilakukan untuk menghitung *hyperplane* adalah dengan mencari nilai  $w$  yang dapat diperoleh dengan mengalikan *Koefisien Lagrange* yang diperoleh selama pelatihan SVM, label kelas dan vector fitur dari support vector. Misalkan nilai  $w$  yang dihasilkan pada setiap target fitur adalah:

- $w_0 = [0.013, -0.068]$
- $w_1 = [0.963, -0.394]$
- $w_2 = [0.399, -0.412]$

Kemudian selanjutnya untuk mencari nilai bias, dapat diperoleh dengan cara mengalikan vektor bobot  $w$  dengan vektor fitur dan kemudian mengurangnya dari label kelas. Hasil dari operasi ini adalah nilai bias  $b$  yang akan digunakan dalam rumus *hyperplane* untuk memisahkan kelas-kelas data. Misalkan nilai  $b$  yang dihasilkan adalah:

- $b_{\text{kelas}_0} = 0.02563$
- $b_{\text{kelas}_1} = -0.17725$
- $b_{\text{kelas}_2} = 1.77101$

Setelah mendapatkan nilai  $w$  dan  $b$  maka selanjutnya dilakukan perhitungan *hyperplane* dapat dihitung dengan menggunakan persamaan (6).

<b>Kelas 0:</b>	<b>Kelas 1:</b>	<b>Kelas 2:</b>
$w = [0.013, -0.068]$	$w = [0.963, -0.394]$	$w = [0.399, -0.412]$
$b = 0.02563$	$b = -0.17725$	$b = 1.77101$
Hyperplane Kelas 0:	Hyperplane Kelas 1:	Hyperplane Kelas 2:
$\text{dim1} * w[0] + \text{dim2} * w[1] + b = 0$	$\text{dim1} * w[0] + \text{dim2} * w[1] + b = 0$	$\text{dim1} * w[0] + \text{dim2} * w[1] + b = 0$
<b>Hyperplane Kelas 0:</b>	<b>Hyperplane Kelas 1:</b>	<b>Hyperplane Kelas 2:</b>
<b><math>0.013 * \text{dim1} - 0.068 * \text{dim2} + 0.02563 = 0</math></b>	<b><math>0.963 * \text{dim1} - 0.394 * \text{dim2} - 0.17725 = 0</math></b>	<b><math>0.399 * \text{dim1} - 0.412 * \text{dim2} + 1.77101 = 0</math></b>

*Hyperplane* yang dihasilkan dari ketiga kelas 3 tersebut sudah dapat digunakan untuk mengklasifikasikan sebuah data baru untuk menentukan apakah data tersebut termasuk ke kelas 0, 1 atau 2. Misalkan terdapat data dengan fitur  $\text{dim1} = -0.212$  dan  $\text{dim2} = 0.321$  akan diklasifikasikan kelasnya. Berikut perhitungan *hyperplane* untuk menentukan kelas dari data tersebut.

Jarak ke *hyperplane* Kelas 0:

- $\text{jarak\_kelas\_0} = |(0.013 * -0.212) - (-0.068 * 0.321) + 0.02563| / \sqrt{(0.013^2) + (-0.068^2)}$
- $\text{jarak\_kelas\_0} = |(-0.002756) - (-0.021808) + 0.02563| / \sqrt{0.000169 + 0.004624}$
- $\text{jarak\_kelas\_0} = 0.029194 / 0.068085$
- $\text{jarak\_kelas\_0} = 0.4288$

Jarak ke *hyperplane* Kelas 1:

- $\text{jarak\_kelas\_1} = |(0.963 * -0.212) - (-0.394 * 0.321) - 0.17725| / \sqrt{(0.963^2) + (-0.394^2)}$
- $\text{jarak\_kelas\_1} = |(-0.203256) - (-0.125574) - 0.17725| / \sqrt{0.927369 + 0.155236}$
- $\text{jarak\_kelas\_1} = 0.146422 / 0.993191$
- $\text{jarak\_kelas\_1} = 0.1475$

Jarak ke *hyperplane* Kelas 2:

- $\text{jarak\_kelas\_2} = |(0.399 * -0.212) - (-0.412 * 0.321) + 1.77101| / \sqrt{(0.399^2) + (-0.412^2)}$
- $\text{jarak\_kelas\_2} = |(-0.084588) - (-0.132352) + 1.77101| / \sqrt{0.159201 + 0.169744}$



0.170144)

- jarak\_kelas\_2 = 1.553874 / 0.607328
- jarak\_kelas\_2 = 2.5568

Selanjutnya, kita dapat menghitung probabilitasnya menggunakan rumus:

Probabilitas Kelas 0:

$$\text{probabilitas\_kelas\_0} = \exp(-\text{jarak\_kelas\_0}) / (\exp(-\text{jarak\_kelas\_0}) + \exp(-\text{jarak\_kelas\_1}) + \exp(-\text{jarak\_kelas\_2}))$$

Probabilitas Kelas 1:

$$\text{probabilitas\_kelas\_1} = \exp(-\text{jarak\_kelas\_1}) / (\exp(-\text{jarak\_kelas\_0}) + \exp(-\text{jarak\_kelas\_1}) + \exp(-\text{jarak\_kelas\_2}))$$

Probabilitas Kelas 2:

$$\text{probabilitas\_kelas\_2} = \exp(-\text{jarak\_kelas\_2}) / (\exp(-\text{jarak\_kelas\_0}) + \exp(-\text{jarak\_kelas\_1}) + \exp(-\text{jarak\_kelas\_2}))$$

Mari kita hitung nilai probabilitasnya:

Probabilitas Kelas 0:

$$\text{probabilitas\_kelas\_0} = \exp(-0.4288) / (\exp(-0.4288) + \exp(-0.1475) + \exp(-2.5568))$$

$$\text{probabilitas\_kelas\_0} = 0.6528 / (0.6528 + 0.8636 + 0.0772)$$

Probabilitas Kelas 1:

$$\text{probabilitas\_kelas\_1} = \exp(-0.1475) / (\exp(-0.4288) + \exp(-0.1475) + \exp(-2.5568))$$

$$\text{probabilitas\_kelas\_1} = 0.8636 / (0.6528 + 0.8636 + 0.0772)$$

Probabilitas Kelas 2:

$$\text{probabilitas\_kelas\_2} = \exp(-2.5568) / (\exp(-0.4288) + \exp(-0.1475) + \exp(-2.5568))$$

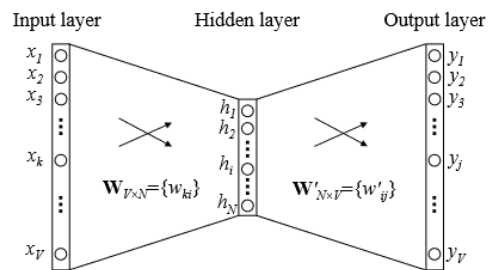
$$\text{probabilitas\_kelas\_2} = 0.0772 / (0.6528 + 0.8636 + 0.0772)$$

Jadi, nilai probabilitas untuk setiap kelas adalah:

- probabilitas\_kelas\_0 = 0.41
- probabilitas\_kelas\_1 = 0.54
- probabilitas\_kelas\_2 = 0.05

jadi dengan melihat nilai *hyperplane* dan probabilitas yang dihasilkan dapat disimpulkan bahwa data tersebut diprediksi sebagai kelas 1.

## Lampiran 8. Contoh Sederhana Cara Kerja Word2Vec



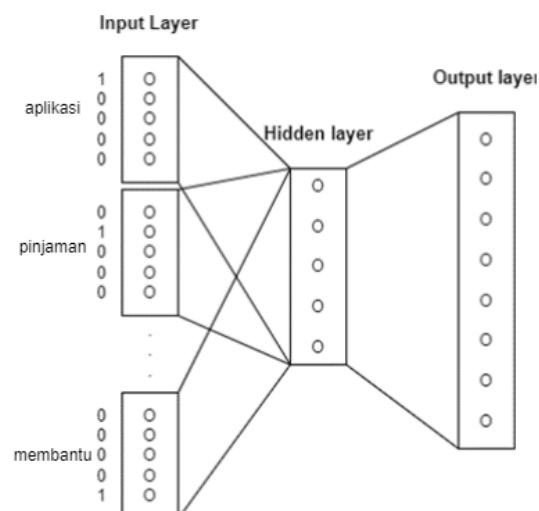
### 1) Pembuatan Vocabulary

Sebagai contoh, digunakan data korpus yang berisi 5 kata/vocab, yaitu “aplikasi pinjaman online bermanfaat membantu”. Kemudian dari korpus tersebut dilakukan one-hot vector encoding, yaitu mengubah kata target menjadi 1 dan lainnya menjadi 0.

	aplikasi	pinjaman	online	bermanfaat	membantu
aplikasi	1	0	0	0	0
pinjaman	0	1	0	0	0
online	0	0	1	0	0
bermanfaat	0	0	0	1	0
membantu	0	0	0	0	1

### 2) Input Layer

Ilustrasi proses pada input layer:



### 3) Hidden Layer dan Output Layer

Proses penentuan bobot dalam word2vec terjadi di lapisan tersembunyi (hidden layer), dan terdapat dua jenis bobot yaitu W1 dan W2. Dalam implementasi word2vec default, kedua bobot tersebut diinisialisasi dengan angka acak sesuai dengan ukuran  $m \times d$ . Namun, dalam perhitungan ini, W1 dan W2 diinisialisasi dengan nilai acak antara -1 dan 1.

Tahap pertama adalah mencari representasi vektor tersembunyi (hidden layer) dari sebuah kata wt.

$$h = W^T x = W_{(k,.)}^T := V_{wI}^T$$

$$[1 \ 0 \ 0 \ 0 \ 0] \text{np.dot} \begin{bmatrix} 0.78523 & -0.23679 & 0.14682 & -0.81247 & 0.94632 \\ 0.71293 & 0.53621 & -0.23456 & 0.67492 & -0.97261 \\ -0.78324 & 0.12345 & -0.90123 & -0.61582 & -0.01234 \\ -0.27364 & 0.56789 & -0.95173 & -0.78901 & 0.67291 \\ -0.82746 & -0.61529 & -0.53621 & -0.34192 & -0.26538 \end{bmatrix}$$

$$\rightarrow [0.78523 \ -0.23679 \ 0.14682 \ -0.81247 \ 0.94632]$$

Jadi hidden layer(h) yang diperoleh adalah  $[0.78523 \ -0.23679 \ 0.14682 \ -0.81247 \ 0.94632]$

Setelah itu, dilakukan perhitungan pada lapisan output, di mana hasilnya adalah perkalian antara vektor tersembunyi (hidden layer) h dan bobot W2.

$$[0.78523 \ -0.23679 \ 0.14682 \ -0.81247 \ 0.94632] \text{np.dot} \begin{bmatrix} -0.68471 & 0.92739 & 0.15324 & -0.28691 & 0.75982 \\ -0.45678 & 0.98765 & -0.12345 & 0.54321 & -0.87654 \\ 0.27481 & -0.61948 & 0.81357 & -0.35894 & 0.72613 \\ -0.89236 & 0.46281 & -0.63729 & 0.19547 & -0.88163 \\ 0.65432 & 0.78239 & -0.34817 & -0.91624 & 0.41968 \end{bmatrix}$$

$$\rightarrow [0.955066 \ 0.767769 \ 0.457307 \ -1.432486 \ 2.024249]$$

Jadi output layer yang diperoleh adalah  $[0.955066 \ 0.767769 \ 0.457307 \ -1.432486 \ 2.024249]$

Hitung Probabilitas Vector

aplikasi	0.955066	0.184091
pinjaman	0.767769	0.152238
online	0.457307	0.111825
bermanfaat	-1.432486	0.016852
membantu	2.024249	0.535994

#### 4) Hitung Nilai Error dan SUM of Different

Selanjutnya mencari jumlah SUM of Different pada wt atau  $\sum I$ . Sebagai contoh pada kata “aplikasi”, mempunyai  $y_{pred}$  “pinjaman” dan “online”. Maka, pada kata “aplikasi” akan dicari jumlah error yang diperoleh dari  $y_{pred}$  “pinjaman” dan “online”. Sedangkan untuk mencari error sendiri diperoleh dari hasil pengurangan one-hot encoding dengan softmax.

0.184091	0	aplikasi	0.184091
0.152238	1	pinjaman	-0.84776
0.111825	0	online	0.111825
0.016852	0	bermanfaat	0.016852
0.535994	0	membantu	0.535994

0.184091	0	aplikasi	0.184091
0.152238	0	pinjaman	0.152238
0.111825	1	online	-0.888
0.016852	0	bermanfaat	0.016852
0.535994	0	membantu	0.535994

Sehingga jumlah SUM of Different pada wt adalah

0.368182
-0.69552
-0.77618
0.033704
1.071988

#### 5) Implementasi Backpropagation

Secara sederhana, proses ini bertujuan untuk menyesuaikan kembali setiap bobot (weight) dan bias berdasarkan kesalahan (error) yang diperoleh. Pada tahap ini, nilai delta untuk bobot  $W_1$  akan dicari.

Langkah 1 → mencari dot dari  $\sum I$  dan W2.

$$\begin{bmatrix} -0.68471 & 0.92739 & 0.15324 & -0.28691 & 0.75982 \\ -0.45678 & 0.98765 & -0.12345 & 0.54321 & -0.87654 \\ 0.27481 & -0.61948 & 0.81357 & -0.35894 & 0.72613 \\ -0.89236 & 0.46281 & -0.63729 & 0.19547 & -0.88163 \\ 0.65432 & 0.78239 & -0.34817 & -0.91624 & 0.41968 \end{bmatrix} \text{np.dot} \begin{bmatrix} 0.368182 \\ -0.695524 \\ -0.776175 \\ 0.033704 \\ 1.071988 \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} -0.211213 \\ -1.680594 \\ 0.666876 \\ -1.094306 \\ 0.38599 \end{bmatrix}$$

Langkah 2 → Untuk menghasilkan delta W1 dari hasil perkalian dot tersebut, dilakukan perkalian outer antara hasil dot tersebut (langkah 1) dengan one-hot encoding dari wt.

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{np.outer} \begin{bmatrix} -0.211213 \\ -1.680594 \\ 0.666876 \\ -1.094306 \\ 0.38599 \end{bmatrix} \rightarrow \begin{bmatrix} -0.211213 & -1.680594 & 0.666876 & -1.094306 & 0.38599 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Untuk menghitung nilai delta W2, dilakukan dengan cara mengalikan (outer) nilai h dan  $\sum I$

$$\begin{bmatrix} 0.78523 \\ -0.23679 \\ 0.14682 \\ -0.81247 \\ 0.94632 \end{bmatrix} \text{np.outer} \begin{bmatrix} 0.368182 \\ -0.695524 \\ -0.776175 \\ 0.033704 \\ 1.071988 \end{bmatrix} \rightarrow \begin{bmatrix} 0.28910755 & -0.54614631 & -0.6094759 & 0.02646539 & 0.84175714 \\ -0.08718182 & 0.16469313 & 0.18379048 & -0.00798077 & -0.25383604 \\ 0.05405648 & -0.10211683 & -0.11395801 & 0.00494842 & 0.15738928 \\ -0.29913683 & 0.56509238 & 0.6306189 & -0.02738349 & -0.87095809 \\ 0.34841799 & -0.65818827 & -0.73450993 & 0.03189477 & 1.01444368 \end{bmatrix}$$

## 6) Update Weight

Update Weight = Weight – learning rate (0.025) \* Delta Weight(W1), sehingga hasil update W1 diperoleh vector sebagai berikut.

$$0.79051 \quad -0.19478 \quad 0.130148 \quad 0.1575058 \quad 0.93667$$

berdasarkan vector diatas maka kalimat “aplikasi pinjaman online bermanfaat membantu” memiliki nilai vektor sebesar = 0.79051 -0.19478 0.130148 -0.785112 0.93667.

Referensi dari (Rahutomo et al., 2019)

### Lampiran 9. Contoh Sederhana Cara Kerja Random Forest

Dengan menggunakan data pada lampiran 7, maka dapat dilakukan langkah-langkah sebagai berikut:

1. Langkah 1: Inisialisasi Jumlah Pohon  
Menggunakan 10 pohon dalam hutan Random Forest.
2. Langkah 2: Ambil Sampel Acak  
Dalam contoh ini, akan digunakan seluruh data sebagai sampel untuk setiap pohon.
3. Langkah 3: Inisialisasi Fitur yang Dipilih  
Memilih dua fitur secara acak dari tiga fitur yang ada, yaitu "dim1" dan "dim2".
4. Langkah 4: Pisahkan Data Berdasarkan Nilai Fitur Terpilih  
Dalam contoh ini, akan dipilih "dim1" dan "dim2" sebagai fitur terpilih. Kemudian dipisahkan data menjadi kelompok berdasarkan nilai-nilai fitur tersebut:

Kelompok 1: Data dengan nilai "dim1"  $\geq 0$

dim1 dim2 label

0.87 -0.86 1

1.76 0.12 1

0.33 0.44 0

0.21 -0.82 2

0.45 -0.12 2

Kelompok 2: Data dengan nilai "dim1"  $< 0$

dim1 dim2 label

-0.23 0.12 0

5. Langkah 5: Hitung Impurity  
Menggunakan Gini impurity sebagai metrik impurity.

Gini impurity Kelompok 1:

- Jumlah total data: 5
- Jumlah data dengan label 1: 2
- Jumlah data dengan label 0: 1
- Jumlah data dengan label 2: 2

$$\text{Gini impurity} = 1 - (p(\text{label } 1)^2 + p(\text{label } 0)^2 + p(\text{label } 2)^2)$$

$$= 1 - ((2/5)^2 + (1/5)^2 + (2/5)^2)$$

$$= 1 - (4/25 + 1/25 + 4/25)$$

$$= 1 - 9/25$$

$$= 16/25$$

$$= 0.64$$

Gini impurity Kelompok 2:

- Jumlah total data: 1
- Jumlah data dengan label 0: 1

$$\text{Gini impurity} = 1 - (p(\text{label } 0))^2$$

$$= 1 - (1/1)^2$$

$$= 0$$

#### 6. Langkah 6: Pilih Pemisah (*Splitter*)

Dalam contoh ini, kita memilih pemisah berdasarkan nilai ambang (*threshold*) pada "dim1". Misalkan kita memilih nilai ambang 0 untuk memisahkan data menjadi dua kelompok.

#### 7. Langkah 7: Buat Simpul Pemisahan (*Split Node*)

Pada pohon pertama, kita membuat simpul pemisahan (*split node*) berdasarkan pemisah yang telah dipilih ("dim1  $\geq$  0").

#### 8. Langkah 8: Lanjutkan Pemisahan Secara *Rekursif*

Karena data di Kelompok 2 hanya memiliki satu data, maka pembangunan pohon berhenti karena sudah memenuhi kriteria penghentian. Setelah semua pohon selesai dibangun, langkah-langkah selanjutnya adalah menggunakan data baru untuk melakukan prediksi dengan melakukan voting dari setiap pohon dalam hutan.

Data baru yang akan kita gunakan adalah:

dim1	dim2
------	------

0.5	-0.5
-----	------

Kita akan memasukkan data baru ini ke setiap pohon dalam hutan dan melakukan voting untuk memprediksi label yang paling sering muncul.

Pohon 1:

Prediksi menggunakan nilai ambang "dim1  $\geq$  0":

- Data baru memiliki nilai "dim1"  $\geq$  0, kita akan ke anak simpul sebelah kanan.

Pohon 2:

Prediksi menggunakan nilai ambang "dim1  $\geq$  0":

- Data baru memiliki nilai "dim1"  $\geq$  0, kita akan ke anak simpul sebelah kanan.

Pohon 3:

Prediksi menggunakan nilai ambang "dim1  $\geq$  0":

- Data baru memiliki nilai "dim1"  $\geq 0$ , kita akan ke anak simpul sebelah kanan.

...

Pohon 10:

Prediksi menggunakan nilai ambang "dim1  $\geq 0$ ":

- Data baru memiliki nilai "dim1"  $\geq 0$ , kita akan ke anak simpul sebelah kanan.

Setelah menghitung prediksi dari setiap pohon, kita akan menggunakan mayoritas suara untuk menentukan prediksi akhir.

Misalkan dari 10 pohon dalam hutan, 7 pohon memprediksi label 1 dan 3 pohon memprediksi label 0 untuk data baru tersebut. Maka, hasil prediksi menggunakan mayoritas suara adalah label 1. Jadi, hasil prediksi untuk data baru "dim1 = 0.5" dan "dim2 = -0.5" adalah label 1.

Dalam implementasi yang sebenarnya, proses ini akan dilakukan untuk setiap data baru yang ingin diprediksi menggunakan hutan Random Forest yang telah dibangun.

Pohon 1:

Prediksi: Label 1

Pohon 2:

Prediksi: Label 1

Pohon 3:

Prediksi: Label 0

...

Pohon 10:

Prediksi: Label 1

Selanjutnya, kita akan menghitung probabilitas kelas berdasarkan voting dari setiap pohon. Misalkan kita mendapatkan 7 pohon memprediksi label 1 dan 3 pohon memprediksi label 0.

Probabilitas kelas 1 = (Jumlah pohon yang memprediksi label 1) / (Total jumlah pohon)

$$= 7 / 10$$

$$= 0.7$$

Probabilitas kelas 0 = (Jumlah pohon yang memprediksi label 0) / (Total jumlah pohon)

$$= 3 / 10$$

$$= 0.3$$



Probabilitas kelas 2 = 0 (Karena tidak ada pohon yang memprediksi label 2)  
Jadi, probabilitas prediksi untuk data baru "dim1 = 0.5" dan "dim2 = -0.5" adalah sebagai berikut:

Probabilitas kelas 0: 0.3  
Probabilitas kelas 1: 0.7  
Probabilitas kelas 2: 0

Dengan demikian, Random Forest memberikan estimasi probabilitas untuk setiap kelas berdasarkan voting dari setiap pohon dalam hutan.

## Lampiran 10. Contoh Sederhana Cara Kerja XGBoost

Dengan menggunakan data pada lampiran 7, maka dapat dilakukan langkah-langkah sebagai berikut:

Langkah 1: Inisialisasi dan Pengaturan Hyperparameter

- Memiliki 6 data latih dan 2 fitur (dim1 dan dim2).
- Membangun 3 pohon keputusan untuk 3 kelas yang berbeda.
- Menentukan hyperparameter seperti jumlah pohon (3) dan nilai lambda (misalnya, 0.1).

Langkah 2: Hitung Prediksi Awal

Hitung dengan menggunakan rata-rata dari label sebagai prediksi awal untuk setiap kelas.

Prediksi awal untuk kelas 0:

$$\text{prediksi\_awal\_kelas\_0} = \text{rata-rata}([0, 0]) = 0$$

Prediksi awal untuk kelas 1:

$$\text{prediksi\_awal\_kelas\_1} = \text{rata-rata}([1, 1]) = 1$$

Prediksi awal untuk kelas 2:

$$\text{prediksi\_awal\_kelas\_2} = \text{rata-rata}([2, 2]) = 2$$

Langkah 3: Iterasi untuk Setiap Pohon

Pohon 1:

Hitung gradien residu kesalahan (gradient):

$$\begin{aligned} \text{gradient} &= \text{label} - \text{prediksi\_awal} \\ &= [1, 1, 0, 0, 2, 2] - [0, 0, 1, 1, 2, 2] \\ &= [1, 1, -1, -1, 0, 0] \end{aligned}$$

Hitung hessian residu kesalahan (hessian):

$$\begin{aligned} \text{hessian} &= \text{prediksi\_awal} * (1 - \text{prediksi\_awal}) \\ &= [0, 0, 1, 1, 2, 2] * (1 - [0, 0, 1, 1, 2, 2]) \\ &= [0, 0, 1, 1, 2, 2] * [1, 1, 0, 0, -1, -1] \\ &= [0, 0, 0, 0, -2, -2] \end{aligned}$$

Buat pohon keputusan baru:

Dalam pohon pertama, kita membagi data latih menjadi dua kelompok berdasarkan pemisahan "dim1 <= 0.45".

Di node akar, hitung penurunan gradien terbaik untuk setiap pemisahan dan pilih pemisahan dengan penurunan gradien terbesar.

Pilih pemisahan " $\text{dim1} \leq 0.45$ " dengan penurunan gradien sebesar  $1 + 1 + 0 + 0 = 2$ .

Buat node anak kiri dengan prediksi daun  $-(1 + 1) / (0 + \lambda)$ , dan node anak kanan dengan prediksi daun  $-(0 + 0) / (0 + \lambda)$ .

Hitung prediksi untuk setiap daun pohon:

Prediksi daun di node anak kiri:  $-2 / (2 + \lambda)$

Prediksi daun di node anak kanan:  $-(0 + 0) / (0 + \lambda)$

Pohon 2 dan Pohon 3:

Lakukan langkah yang sama seperti pada Pohon 1 untuk menghitung gradien, hessian, dan membuat pohon keputusan baru.

Langkah 4: Gabungkan Kontribusi Pohon

Hitung prediksi akhir dengan menggabungkan kontribusi pohon dari setiap pohon:

prediksi akhir = prediksi awal + prediksi daun di setiap daun pohon

Prediksi akhir untuk data latih adalah hasil gabungan prediksi dari ketiga pohon.

Prediksi untuk Data Baru:

Misalkan kita memiliki data baru dengan fitur  $(0.5, -0.5)$ .

Kita akan melakukan prediksi kelas untuk data baru ini menggunakan prediksi akhir dari pohon-pohon yang telah dibangun sebelumnya.

Langkah 1: Hitung Prediksi Awal

Prediksi awal untuk kelas 0: 0

Prediksi awal untuk kelas 1: 1

Prediksi awal untuk kelas 2: 2

Langkah 2: Hitung Kontribusi Pohon

Hitung kontribusi pohon untuk setiap pohon:

Kontribusi Pohon 1: Hitung prediksi pada daun yang sesuai dengan pemisahan " $\text{dim1} \leq 0.45$ ".

Kontribusi Pohon 2: Hitung prediksi pada daun yang sesuai dengan pemisahan pada pohon 2.

Kontribusi Pohon 3: Hitung prediksi pada daun yang sesuai dengan pemisahan pada pohon 3.

Langkah 3: Gabungkan Kontribusi Pohon

Hitung prediksi akhir dengan menggabungkan kontribusi pohon dari setiap pohon:

prediksi akhir = prediksi awal + kontribusi pohon

Prediksi akhir untuk data baru adalah hasil gabungan prediksi dari ketiga pohon.

**Lampiran 11.** Skenario Performa Baru dari Penambahan Jumlah Data

Jumlah data yang digunakan dapat dilihat pada tabel 4.2.1. Untuk skenario penambahan data baru, digunakan data sebanyak 2.637 ulasan, sehingga jumlah data menjadi:

Kelas	Jumlah Ulasan
Positif	8784
Negatif	8788
Netral	2704
Jumlah	20.277

Performa baru yang dihasilkan:

Model	Akurasi
Random Forest	79.4%
XGBoost	78.3%
SVM	79.4%
Soft Voting Classifier	85%

## LEMBAR PERBAIKAN SKRIPSI

### “PERBANDINGAN METODE KLASIFIKASI RANDOM FOREST, XGBOOST DAN SVM PADA ANALISIS SENTIMEN APLIKASI KREDIT DAN PINJAMAN ONLINE”


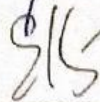


**OLEH:**

**PAHRUL  
D121191020**

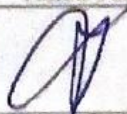
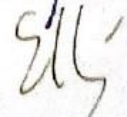
Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 7 Juni 2023.

Telah dilakukan perbaikan penulisan dan isi skripsi berdasarkan usulan dari penguji dan pembimbing skripsi.

Persetujuan perbaikan oleh tim penguji:

	Nama	Tanda Tangan
Ketua	Dr. Amil Ahmad Ilham, S.T., M.IT.	
Sekretaris	Elly Warni, S.T., M.T.	
Anggota	Dr. Ir. Zahir Zainuddin, M.Sc.	
	Dr. Eng. Zulkifli Tahir, S.T., M.Sc.	

Persetujuan perbaikan oleh pembimbing:

Pembimbing	Nama	Tanda Tangan
I	Dr. Amil Ahmad Ilham, S.T., M.IT.	
II	Elly Warni, S.T., M.T.	



KEMENTERIAN PENDIDIKAN , KEBUDAYAAN  
RISET DAN TEKNOLOGI  
UNIVERSITAS HASANUDDIN

FAKULTAS TEKNIK  
DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Matano, Gowa  
<http://eng.unhas.ac.id/informatika>, Email : [informatika@unhas.ac.id](mailto:informatika@unhas.ac.id)

**DAFTAR HADIR SEMINAR HASIL**


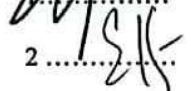
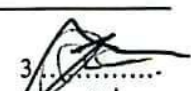
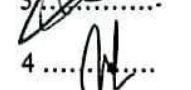
Nama/Stambuk : 1. Pahrul D121191020

Judul Skripsi/T.A : "Perbandingan Metode Klasifikasi Random Forest. XGBoost dari SVM pada Analisis Sentimen Aplikasi Kredit dan Pinjaman Online"


Hari/Tanggal : Jumat, 3 Mei 2023

Jam : 09- 00 Wita – Selesai

Tempat : Ruang Lab. UBICON Departemen Teknik Informatika Gowa

No.	Jabatan	Nama Dosen	Tanda Tangan
L.	Pembimbing I	1. Dr. Ir. Amil Ahmad Ilham,ST.,M.IT	
	Pembimbing II	2. Elly Warni,ST.,M.T	2 ..... 
II.	Anggota Penguji	3. Dr. Ir. Zahir Zainuddin,M.Sc	3 ..... 
		4. Dr. Eng. Zulkifli Tahir,ST.,M.Sc	4 ..... 

PANITIA UJIAN

Ketua,  
  
Dr. Amil Ahmad Ilham,ST.,M.IT

Sekretaris,  
  
Elly Warni,ST.,M.T



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN  
RISET DAN TEKNOLOGI  
UNIVERSITAS HASANUDDIN

FAKULTAS TEKNIK  
DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa  
<http://eng.unhas.ac.id/informatika>, Email : [informatika@unhas.ac.id](mailto:informatika@unhas.ac.id)

**BERITA ACARA SEMINAR HASIL**

Pada hari ini Jumat, tanggal 3 Mei 2023 Pukul 09.00 WITA - Selesai bertempat di Ruang Lab.UBICON Departemen Teknik Informatika Gowa, telah dilaksanakan Seminar Hasil bagi Saudara :

Nama : Pahrul  
No. Stambu : D121191020  
Fakultas/Departemen : Teknik/Teknik Informatika  
Judul Skripsi : "Perbandingan Metode Klasifikasi Random Forest. XGBoost dari SVM pada Analisis Sentimen Aplikasi Kredit dan Pinjaman Online"

Yang dihadiri oleh Tim Penguji Seminar Hasil sebagai berikut :

No.	N a m a	Jabatan	Tanda tangan
1.	Dr. Ir. Amil Ahmad Ilham,ST.,M.IT	Pemb I/Ketua	
2.	Elly Wami,ST.,M.T	Pemb II/Sekretaris	
3.	Dr. Ir. Zahir Zainuddin,M.Sc	Anggota	
4.	Dr. Eng. Zulkifli Tahir,ST.,M.Sc	Anggota	

Hasil keputusan Tim Penguji Seminar Hasil : **Lulus / Tidak lulus** dengan nilai angka ..... dan huruf .....

Gowa, 3 Mei 2023  
Ketua/Sekretaris Panitia Ujian,

Dr. Ir. Amil Ahmad Ilham,ST.,M.IT



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,  
RISET DAN TEKNOLOGI  
UNIVERSITAS HASANUDDIN  
FAKULTAS TEKNIK  
DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa  
<http://eng.unhas.ac.id/informatika>, Email : [informatika@unhas.ac.id](mailto:informatika@unhas.ac.id)

Nomor : 396/UN4.7.7/TD.06/2023  
Lamp : -  
Hal : Penerbitan Surat Penugasan Panitia  
Seminar Hasil Strata Satu (S1)

Kepada Yth :

Wakil Dekan Bidang Akademik dan Kemahasiswaan  
Fakultas Teknik Universitas Hasanuddin

Di-

Gowa

Dengan hormat,  
Berdasarkan Persetujuan Pembimbing Mahasiswa, Bersama ini diusulkan susunan Panitia Seminar Hasil Strata Satu (S1) bagi mahasiswa Departemen Teknik Informatika Fakultas Teknik tersebut di bawah ini :

Nama / Stambuk : Pahrul D121191020  
Judul TA : Perbandingan Metode Klasifikasi Random Forest, XGBoost dan SVM Pada Analisis Sentimen Aplikasi Kredit dan Pinjaman Online

Dengan ini kami sampaikan Susunan Panitia Seminar Hasil Program Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :

Pembimbing I/ Ketua : 1. Dr. Amil Ahmad Ilham, ST., M.IT  
Pembimbing II / Sekretaris : 2. Elly Warni, ST.,M.T  
Anggota : 3. Dr. Ir. Zahir Zainuddin, M.Sc.  
4. Dr.Eng. Zulkifli Tahir, ST., M.Sc.

Untuk dapat diterbitkan surat penugasannya

Demikian penyampaian kami, atas perhatian dan kerjasamanya diucapkan terima kasih.

Gowa, 26 April 2023  
Ketua Departemen Tek.Informatika,



Prof. Dr. Ir. Indrabayu.,ST, MT, M.Bus.Sys., IPM, ASEAN.Eng  
Nip.19750716 200212 1 004

Tembusan :  
1. Arsip



**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,  
RISET DAN TEKNOLOGI  
UNIVERSITAS HASANUDDIN  
FAKULTAS TEKNIK**

Pacos Malina Km.6Bontomaranna(92172) Gowa, Sulawesi Selatan 92172, Sulawesi Selatan  
Telp. (0411) 586015, 586262 Fax (0411) 586015  
<http://eng.unhas.ac.id>, Email : [teknika@unhas.ac.id](mailto:teknika@unhas.ac.id)

**SURAT PENUGASAN**  
No. 8081/UN4.7.1.TD.06/2023

- Dari : Dekan Fakultas Teknik Universitas Hasanuddin  
Kepada : Mereka yang tercantum namanya dibawah ini  
Isi : 1. **Bahwa berdasarkan peraturan Akademik Universitas Hasanuddin Tahun 2018 pasal 18 (SK.Rektor Unhas nomor : 2781/UN4.1/KEP/2018), dengan ini menugaskan Saudara sebagai PANITIA SEMINAR HASIL Program Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :**

**Pembimbing I / Ketua : 1. Dr. Amil Ahmad Ilham, ST., M.IT**  
**Pembimbing II / Sekretaris : 2. Elly Warni, ST.,M.T**  
**Anggota : 3. Dr. Ir. Zahir Zainuddin, M.Sc.**  
**4. Dr.Eng. Zulkifli Tahir, ST., M.Sc.**

Untuk menguji bagi mahasiswa tersebut dibawah ini :

- |                      |                                                                                                                                    |                   |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| Nama/NIM             | : <b>Pahrul</b>                                                                                                                    | <b>D121191020</b> |
| Program Studi        | : Teknik Informatika                                                                                                               |                   |
| Judul thesis/Skripsi | : <b>Perbandingan Metode Klasifikasi Random Forest, XGBoost dan SVM Pada Analisis Sentimen Aplikasi Kredit dan Pinjaman Online</b> |                   |
- 2 Waktu seminar ditetapkan oleh Panitia Seminar Hasil Program Strata Satu (S1)
  - 3 Agar Surat Penugasan ini dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab
  - 4 Surat penugasa ini berlaku sejak tanggal ditetapkan sampai dengan berakhirnya seminar tersebut dengan ketentuan bahwa segala sesuatunya akan ditinjau dan diperbaiki sebagaimana mestinya apabila dikemudian hari terdapat kekeliruan dalam keputusan ini.

Ditetapkan di Gowa  
Pada tanggal 26 April 2023

a.n. Dekan,  
Wakil Dekan Bidang Akademik dan Kemahasiswaan  
Fakultas Teknik Unhas



Dr. Amil Ahmad Ilham, ST., M.IT  
NIP. 197310101998021001

**Tembusan**

1. Dekan Fak. Teknik Unhas
2. Ketua Departemen Teknik Informatika FT-UH
3. Mahasiswa yang bersangkutan





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN  
UNIVERSITAS HASANUDDIN  
FAKULTAS TEKNIK  
DEPARTEMEN TEKNIK INFORMATIKA  
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa  
<http://eng.unhas.ac.id/informatika>, Email : [informatika@unhas.ac.id](mailto:informatika@unhas.ac.id)

DAFTAR HADIR UJIAN SKRIPSI MAHASISWA  
FAKULTAS TEKNIK UNHAS

Nama/Stambuk : Pahrul D121191020

Judul Skripsi/T.A : "Perbandingan Metode Klasifikasi Random Forest. XGBoost dari SVM  
pada Analisis Sentimen Aplikasi Kredit dan Pinjaman Online"

Hari/Tanggal : Rabu, 7 Juni 2023

Jam : 13.00 Wita – Selesai

Tempat : Ruang Lab. UBICON Departemen Teknik Informatika

No.	Jabatan	Nama Dosen	Tanda Tangan
I.	Pembimbing I	1. Dr. Ir. Amil Ahmad Ilham, ST., M.IT	1.
	Pembimbing II	2. Elly Warni, ST., M.T	2.
II.	Anggota Penguji	3. Dr. Ir. Zahir Zainuddin, M.Sc	3.
		4. Dr. Eng. Zulkifli Tahir, ST., M.Sc	4.

PANITIA UJIAN

Ketua,

Dr. Ir. Amil Ahmad Ilham, ST., M.IT

Sekretaris,

. Elly Warni, ST., M.T



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN  
UNIVERSITAS HASANUDDIN  
FAKULTAS TEKNIK

DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa  
<http://eng.unhas.ac.id/informatika>, Email : [informatika@unhas.ac.id](mailto:informatika@unhas.ac.id)

**BERITA ACARA UJIAN SKRIPSI**

Pada hari ini Rabu, tanggal 7 Juni 2023 Pukul 13.00 WITA - Selesai bertempat di Ruang Lab. UBICON Departemen Teknik Informatika, telah dilaksanakan Ujian Skripsi bagi Saudara :

Nama : Pahrul  
No. Stambuk : D121191020  
Fakultas/Departemen : Teknik/Teknik Informatika  
Judul Skripsi : "Perbandingan Metode Klasifikasi Random Forest. XGBoost dari SVM pada Analisis Sentimen Aplikasi Kredit dan Pinjaman Online"

Yang dihadiri oleh Tim Penguji Ujian Skripsi sebagai berikut :

No.	Nama	Jabatan	Tanda tangan
1.	Dr. Ir. Amil Ahmad Ilham, ST., M.IT	Pemb I/Ketua	
2.	Elly Wami, ST., M.T	Pemb II/Sekretaris	
3.	Dr. Ir. Zahir Zainuddin, M.Sc	Anggota	
4.	Dr. Eng. Zulkifli Tahir, ST., M.Sc	Anggota	

Hasil keputusan Tim Penguji Ujian Skripsi/Tugas Akhir : Lulus / Tidak lulus dengan nilai angka ..... dan huruf .....

Gowa, 7 Juni 2023

Ketua/Sekretaris Panitia Ujian,

Dr. Ir. Amil Ahmad Ilham, ST., M.IT



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,  
RISET DAN TEKNOLOGI  
UNIVERSITAS HASANUDDIN  
FAKULTAS TEKNIK  
DEPARTEMEN TEKNIK INFORMATIKA  
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa  
<http://eng.unhas.ac.id/informatika>, Email : [informatika@unhas.ac.id](mailto:informatika@unhas.ac.id)

Gowa, 5 Juni 2023

Nomor : 464/UN4.7.7.1/TD.06/2023  
Lamp : -  
Hal : Usulan Susunan Panitia Ujian Sarjana

Yth. : Bapak Wakil Dekan Bidang Akademik dan Kemahasiswaan  
Fakultas Teknik Unhas  
Di  
Gowa

Dalam rangka penyelesaian studi pada Departemen Teknik Informatika Fakultas Teknik Unhas, bersama ini kami usulkan susunan Panitia Ujian Sarjana Program Strata Satu (S1) bagi mahasiswa Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin atas nama :

Pembimbing I / Ketua : 1. Dr. Amil Ahmad Ilham, ST.,M.IT  
Pembimbing II / Sekretaris : 2. Elly Warni, ST., M.T  
Anggota : 3. Dr. Ir. Zahir Zainuddin, M.Sc  
4. Dr.Eng. Zulkifli Tahir, ST., M.Sc

Untuk Bertugas sebagai Penguji/ Penanggap Ujian Sarjana bagi Mahasiswa :

Nama : Pahrul  
Stambuk : D121 19 1020

Dengan Judul Skripsi

“ Perbandingan Metode Klasifikasi Random Forest, XGBoost dan SVM Pada Analisis Sentimen Aplikasi Kredit dan Pinjaman Online “

Pada :  
Hari/Tanggal : Rabu, 7 Juni 2023  
Jam : 13.00 Wita - Selesai  
Tempat : Ruang Sidang Lab. Ubicon

Demikian penyampaian kami, atas perhatiannya diucapkan terimah kasih.

Ketua Departemen Tek.Informatika,



Prof. Dr. Ir. Indrabayu.,ST, MT, M.Bus.Sys., IPM, ASEAN.Eng  
Nip.197507016 200212 1 004

Tembusan :  
I. Arsip

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,  
RISET DAN TEKNOLOGI  
UNIVERSITAS HASANUDDIN  
FAKULTAS TEKNIK

Poros Malino Km.6Bontomarannu(92172) Gowa, Sulawesi Selatan 92172, Sulawesi Selatan  
Telp. (0411) 586015, 586262 Fax (0411) 586015  
<http://eng.unhas.ac.id>, Email : [teknik@unhas.ac.id](mailto:teknik@unhas.ac.id)

**SURAT PENUGASAN**

No. 11403/UN4.7.1/TD.06/2023

Dari : Dekan Fakultas Teknik Universitas Hasanuddin.

Kepada : Mereka yang tercantum namanya di bawah ini.

Isi : 1. Bahwa berdasarkan peraturan Akademik Universitas Hasanuddin Tahun 2018 pasal 19 (SK.Rektor Unhas nomor : 2781/UN4.1/KEP/2018), dengan ini menugaskan Saudara sebagai PANITIA UJIAN SARJANA Program Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :

Pembimbing I / Ketua : 1. Dr. Amil Ahmad Ilham, ST.,M.IT  
Pembimbing II / Sekretaris : 2. Elly Warni, ST., M.T  
Anggota : 3. Dr. Ir. Zahir Zainuddin, M.Sc  
4. Dr.Eng. Zulkifli Tahir, ST., M.Sc

untuk menguji bagi mahasiswa tersebut di bawah ini :

Nama/NIM : Pahrul D121191020  
Program Studi : Teknik Informatika  
Judul Thesis/Skripsi : Perbandingan Metode Klasifikasi Random Forest, XGBoost dan SVM Pada Analisis Sentimen Aplikasi Kredit dan Pinjaman Online

2. Waktu Ujian ditetapkan oleh Panitia Ujian Sarjana Program Strata Satu (S1).
3. Agar Surat penugasan ini dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab.
4. Surat penugasan ini berlaku sejak tanggal ditetapkan sampai dengan berakhirnya Ujian Sarjana tersebut, dengan ketentuan bahwa segala sesuatunya akan ditinjau dan diperbaiki sebagaimana mestinya apabila dikemudian hari ternyata terdapat kekeliruan dalam keputusan ini.

Ditetapkan di Gowa,  
Pada tanggal 5 Juni 2023

a.n. Dekan

Wakil Dekan Bidang Akademik dan Kemahasiswaan  
Fakultas Teknik Unhas



Dr. Amil Ahmad Ilham, ST., M.IT  
NIP.197310101998021001

**Tembusan :**

1. Dekan Fak. Teknik Unhas
2. Ketua Departemen Teknik Informatika FT-UH
3. Kasubag. Umum dan Perlengkapan FT-UH

