BAB 2

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian Thamrin (2021) menggunakan data 1648 jawaban esai dengan teknik klasifikasi dan algoritma similaritas dalam *essay grading* dengan klasifikasi *Support Machine Vector* (SVM) dan *K-Nearest Neighbors* (kNN) pada penelitian Thamrin, dengan klasifikasi TF-IDF memperoleh nilai RMSE sebesar 2,73. (Thamrin et al., 2021)

Pada tahun 2021, Verdikha (2021) menggunakan data yang sama pada penelitian Thamrin (2021). Penelitian dilakukan dengan judul "Regression and Oversampling Method for Indonesian Language Automated Essay Scoring" menggunakan metode *Support Vector Regression* (SVR)*, Logistic Regression* (LR)*,* dan *MLP Regression* (MLP-R) dengan parameter SVR, memperoleh nilai RMSE yang lebih baik dengan skor 2,16. (Verdikha et al., 2021)

Nambiar (2019), pada penelitiannya menggunakan dataset dari *Refrence Energy Disaggregation Data Set* (REDD) yang disediakan oleh MIT berupa data dengan frekuensi rendah untuk dua rumah dengan tipe 4 dan 5. Metode yang digunakan adalah SVR(sigmoid), kNN, DTR, Fully Connected NN, LSTM, dan SVR(rbf) menghasilkan nilai evaluasi RMSE dengan metode SVR(sigmoid) sebesar 49,23 (Nambiar et al., 2019)

Penelitian terhadap citra gambar yang dilakukan Jebadurai (2017), pembelajaran berbasis *super-resolution* (SR) dalam menghasilkan gambar *high-resolution* (HR) dari gambar *low-resolution* (LR) menggunakan metode SVR menggunakan model kesalahan *sigmoid-kernel* SVR dengan pendekatan algoritma NN, Bicubic, NE + LLE, SCSR, SC+SVR, SLSVR dengan evaluasi nilai menggunakan *peak signal-to-noise* (PSNR) membuktikan nilai evaluasi dari setiap algoritma dengan nilai rata-rata PSNR sebesar 7,29 , 0,98 , 3,56 , 0,67 , 0,32 , 0, 24 (Jebadurai & Peter, 2017)

Pada penelitian selanjutnya Jebadurai (2018) melakukan penelitian kembali yaitu pendekatan *super resolution* (SR) untuk gambar retina pada Kesehatan IoT menggunakan fundoskopi *smartphone*. Algoritma SR yang diusulkan menggunakan dukungan multikernel (SVR) antara lain neighbor embendding (NE), *sparse coding* (SC), SC+SVR, *self-learning* (SR), dan *sigmoid kernel* SVR (SK-SVR) dengan mengevaluasi nilai menggunakan *peak-signal-to-noise ratio* (PSNR) and *mean squared error* (MSE). Menghasilkan hasil experimental terbaik menggunakan SK-SVR dengan nilai evaluasi PSNR dan MSE masing-masing 0,19 dan 0,68 (Jebadurai & Peter, 2018).

2.2 Teori Dasar Penelitian

Teori dasar merupakan kumpulan sistem, cara, metode, tahapan, dan pemerosesan segala bentuk materi yang menyangkut setiap pembahasan di dalam penelitian.

2.2.1 *Natural Language Processing*

NLP adalah salah satu cabang teknologi kecerdasan buatan yang mengelola bahasa alami manusia. Cara kerja NLP adalah dengan melakukan representasi dari interaksi antara mesin dengan manusia dengan menggunakan data yang direkam lalu dikelola sehingga dapat merespon layaknya percakapan manusia. Contoh yang dapat kita lihat penerapan NLP pada kehidupan adalah penggunaan "*asisten google*" pada teknologi android yang ada di *smartphone*.

NLP dapat dikatakan sebagai area *Question Answering System* karena kemampuannya yang dapat menyelesaikan masalah pertanyaan dengan Bahasa yang dilontarkan oleh manusia. Kajian NLP mencangkup proses segmentasi ucapan (*speech segmentation*), segmentasi teks (*text segmentation*), penandaan kelas kata (*part-of-speech tagging*), dan masalah terbuka pemrosesan Bahasa alami (*word sense disambiguation*) (Herwin & Andesa, 2019)

2.2.2 *Auto Essay Scoring*

AES sangat berperan penting di dalam bidang pendidikan. Kemampuan AES dalam menilai jawaban esai Berbahasa Indonesia secara otomatis sangat

membantu pengelolaan data yang besar. Penilaian AES diperoleh dengan menghitung kunci jawaban dengan kemiripan teks jawaban.

Sistem AES menggunakan metode *searching text similarity matching text* dapat dipakai sebagai tempat evaluasi hasil jawaban teks terhadap kunci jawaban karena mempunyai persentasi akurasi yang tinggi dalam pencocokan kesamaan teks. *Error rate* penggunaan *term frequency* (TF) lebih rendah dibandingkan dengan *inverse document* (IDF) dan *term frequency-inverse document* (TF-IDF) (Ahmad et al., 2018).

2.2.3 TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) adalah metode pengabungan konsep TF dan IDF untuk menghitung bobot pada suatu kata (term) terhadap dokumen. Semakin kecil jumlah dokumen yang mengandung *term*, maka nilai IDF yang diperoleh semakin besar. Menggabungkan dua konsep yaitu frekuensi kemunculan sebuah kata di dalam sebuah dokumen dan *inverse* frekuensi dokumen yang mengandung kata tersebut.

TF atau *term frequency* adalah pembobotan kata pada suatu dokumen yang didasarkan pada frekuensi jumlah kemunculan kata pada dokumen tersebut, semakin besar frekuensi kemunculan suatu kata maka semakin besar pula bobotnya (Liani et al., 2020). Persamaan TF pada persamaan 2.1 sebagai berikut:

$$d_{ij} = log\ log\ (d_{ij} + 1) \tag{2.1}$$

Dimana:

d_{ij} = jumlah kemunculan kata i dalam dokumen j

i = frekuensi kemunculan kata

j = *inverse* frekuensi yang mengandung kata i

Invers document frequency (IDF) berfungsi untuk menghitung kata-kata yang sering muncul atau hadir di semua dokumen atau secara keseluruhan data. Secara umum dapat digunakan persamaan 2.2 sebagai berikut:

$$idf(t) = log\ log\ \frac{1+n}{1+df(t)} + 1 \tag{2.2}$$

Dimana:

idf = *invers document frekuency*

t = *term* (kata)

n = total *dokument*

$df(t)$ = jumlah dokumen dalam kumpulan dokumen yang mengandung t

Untuk menentukan nilai TF-IDF yaitu dengan menggabungkan dua konsep perhitungan TF dan IDF dengan persamaan 2.3 sebagai berikut:

$$tf - idf(t,d) = tf(t,d) \times idf(t) \quad (2.3)$$

Dimana:

tf = *term Frekuency*

idf = *invers document frekuency*

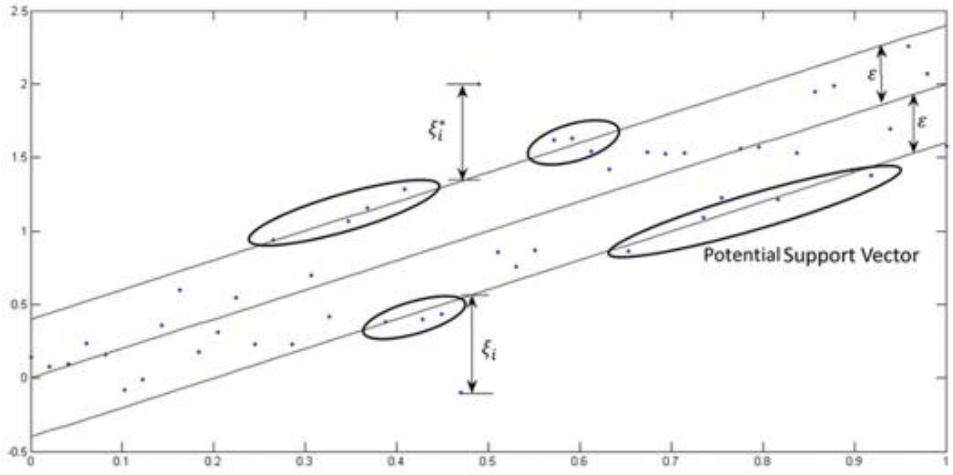
t = *term*

d = *document*

Maka hasil dari pembobotan TF-IDF adalah *term* yang sudah dinormalisasi. Normalisasi pada tahapan ini bertujuan mengurangi redudansi data ketika menjalankan program secara berulang, tujuannya agar program dapat bekerja optimal (Prayitno et al., 2018).

2.2.4 *Support Vector Regression*

SVR adalah penerapan dari *Support Vector Machine* (SVM) yang diperkenalkan oleh Cortes dan Vapnik untuk kasus regresi (Cortes & Vapnik, 1995). Tujuan dari SVR yaitu menentukan sebuah fungsi pemisah berupa fungsi regresi yang mana sesuai dengan semua input data dengan sebuah *error*. Hasil keluaran SVR berupa bilangan *riil* dan *kontinu*.



Gambar 2.1 Ilustrasi SVR (Yudhawan, 2020)

Melihat dari Gambar 2.1 di atas, *hyperplane* (garis diagonal tengah) di apit oleh dua garis pembatas nilai "− & +" seperti gambar di atas. Dapat di lihat ε sebagai jarak antara garis *hyperplane* dengan garis pembatas. Titik-titik yang dilingkari adalah *potential support vectors*. Titik titik yang tersebar di antara *hyperplane* dengan garis pembatas ataupun yang diluar garis pembatas merupakan *data points* yang bisa menjadi calon pembatas, sehingga semua data poin bisa masuk ke dalam satu kluster, dengan tetap sebisa mungkin meminimalisir nilai ε nya. Jika divisualisasikan, garis *hyperplane* sebisa mungkin melewati semua titik-titik *data points* tersebut.

SVR diperuntukkan dalam menentukan nilai fungsi menggunakan data training. Rumus untuk menghitung data training yang umum digunakan pada penelitian-penelitian sebelumnya dapat di lihat persamaan 2.4 berikut:

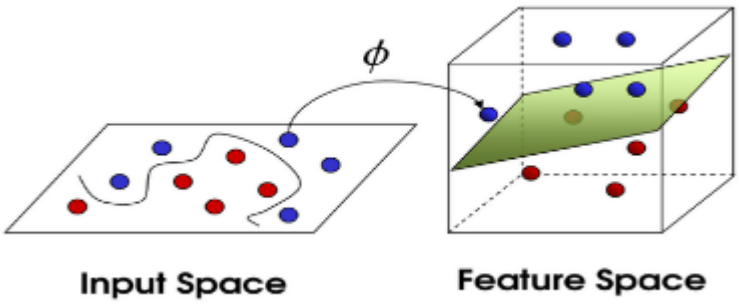
$$\{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\} \subset X \times R \tag{2.4}$$

Dimana X input vector dari R . Dalam konsep regresi, Vapnik (1995) menjelaskan bahwa tujuan dari SVR adalah mencari fungsi $f(x)$ yang mempunyai deviasi maksimal sebesar ε untuk menghasilkan nilai target y_i dari semua *training data*. Nilai kesalahan diterima jika kurang dari nilai ε. Sebaliknya, jika nilai kesalahan tidak diterima bila nilainya melebihi ε (Prakoso, 2017).

2.2.5 Fungsi Kernel

Fungsi *kernel* merupakan bagian terpenting dari metode SVR. *Kernel* adalah

sebuah algoritma yang digunakan untuk analisis dan pengenalan pola. Tugas umum dari analisis dan pengenalan pola adalah menemukan dan mempelajari hubungan umum (misalnya klaste*r*, klasifikasi, korelasi, komponen utama) pada data seperti *sequence*, teks dokumen, vektor, citra, grafik, dan lain-lain (Souza, 2010).



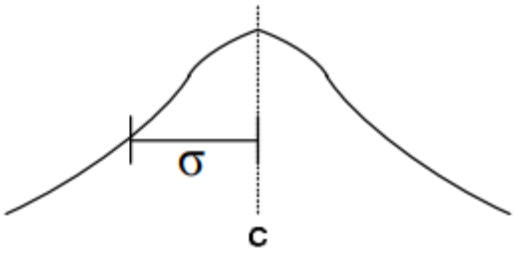
Gambar 2.2 Fungsi *Kernel* (Kesumawati, 2018)

Algoritma dapat dengan mudah dibawa ke ruang dimensi yang lebih tinggi tanpa harus secara eksplisit memetakan titik *input* data. Hal ini tentu memudahkan, karena terkadang ruang fitur dimensi bersifat *infinite-dimensional* dan tidak bisa dihitung. Terdapat 29 Metode *kernel* memetakan data ke ruang dimensi yang lebih tinggi, sehingga data lebih mudah dipisahkan atau lebih terstruktur. Dengan penggunaan fungsi *kernel*, salah satunya yang digunakan dalam penelitian ini adalah *kernel sigmoid* dalam membandingkan dengan *kernel radial-basis fuction* (RBF).

2.2.5.1 *Kernel RBF*

Kenel radial-basis function (RBF) RBF merupakan *kernel* yang digunakan dalam analisis data yang tidak terpisah secara linear. *Kernel* RBF memiliki dua parameter yaitu *Gamma* dan *Cost*. Parameter *cost* (*C*) bekerja sebagai optimaliasi SVM untuk menghindari misklasifikasi pada setiap training test sedangkan Parameter *Gamma* menentukan seberapa jauh efek dari satu sampel training dataset dengan nilai rendah berarti “jauh”, dan nilai tinggi berarti “dekat” (Kesumawati, 2018).

Pada *kernel* RBF fungsi aktivasi memiliki nilai *center* di tengah dari lembar fungsi basis, sehinggal yang dihitung adalah area antara nilai *center* pada fungsi dengan jarak antara titik dengan garis fungsi *gaussian*. Fungsi aktivasi pada *kernel* RBF dapat direpresentasikan dengan grafis fungsi RBF pada gambar 2.3 di bawah.



Gambar 2. 3 Representasi grafis fungsi *kernel* RBF (Fauzan et al., 2011)

Fungsi *kernel* RBF menghitung antara dua vektor, rumus persamaan yang digunakan menggunakan rumus persamaan 2.5 sebagai berikut.

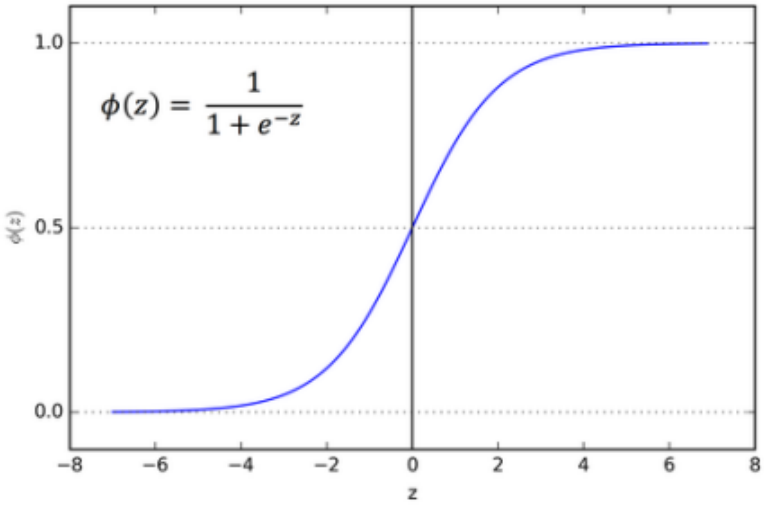
$$k(x, y) = exp(-\gamma \parallel x - y \parallel^2) \quad (2.5)$$

Dimana x dan y adalah inpu vektor. γ dikenal sebagai kemiringan. *Kernel* RBF dikenal juga *kernel* varian *Gaussian* (Pedregosa et al , 2011).

2.2.5.2 *Kernel Sigmoid*

Kernel hyperbolic tangent juga dikenal sebagai *kernel sigmoid* atau kernel *multilayer perceptron* (MLP). *Kernel sigmoid* berasal dari bidang *Neural Network* di mana fungsi *sigmoid* sering digunakan sebagai fungsi aktivasi untuk *neuron* buatan. Sangat menarik untuk dicatat bahwa model SVM menggunakan fungsi *kernel sigmoid* setara dengan dua *layer perceptron* pada jaringan syaraf tiruan.

Kernel sigmoid cukup popular untuk mendukung mesin vektor karena asalnya dari teori jaringan saraf. Selain itu, meskipun hanya kondisional positif yang pasti, ia telah terbukti berkinerja baik dalam praktiknya (Fadilah, 2018). Fungsi aktivasi *logistic* merupakan sebutan untuk fungsi aktivasi *sigmoid.* Grafik fungsi aktivasi pada kernel sigmoid dapat di lihat pada gambar (2.4) di bawah.



Gambar 2. 4 Grafik fungsi aktivasi *Sigmoid* (Tejakusuma, 2019)

Ada dua parameter yang dapat disesuaikan dalam *kernel sigmoid*. Nilai yang umum untuk *alpha* adalah *1/N*, dimana *N* adalah dimensi data. *Kernel sigmoid* di definisikan menggunakan persamaan 2.6 berikut:

$$k(x, y) = tan\, h(\gamma x^T y + c_0) \tag{2.6}$$

Dimana:

x, y = input vektor

γ = dikenal sebagai kemiringan

c_0 = dikenal sebagai intersep

Penjelasan atas kedua *kernel* tersebut dimasukan guna untuk melihat secara spesifik fungsional dari masing-masing *kernel* terhadap penerapan dalam metode SVR.

2.2.6 *Root Mean Square Error*

RMSE adalah aturan untuk menghitung bobot rata-rata dalam penilaian kuadrat dalam mengukur tingkat kesalahan didalamnya. Akar kuadrat dari rata-rata perbedaan kuadrat nantinya akan di prediksi dengan mengamati nilai sebenarnya. RMSE bergantung pada skala dari variabel dependen. RMSE digunakan sebagai pengukuran relatif untuk membandingkan prediksi pada series

yang sama dalam model berbeda. Semakin kecil nilai *error*, maka semakin baik kemampuan prediksi metode. Nilai RMSE dihitung dengan mengkuadratkan nilai *residual* dibagi dengan jumlah data, lalu diakarkan.

Nilai *residual* adalah selisih dari data aktual dengan data hasil prediksi (Fadilah, 2018). Berikut dapat dilihat rumus persamaan di bawah.

$$MSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \tag{2.7}$$

kumpulan dataset di lambangkan dalam n, untuk $\hat{y}_i$ melambangkan nilai model prediksi, y_i adalah nilai aktual.

2.2.7 *Text Preprocessing*

Preprocessing adalah proses normalisasi data agar data teks menjadi mendekati Bahasa baku. Fungsi *preprocessing* dalam menentukan evaluasi tentu memiliki keterkaitan penting di dalamnya dalam hal akurasi dalam hal pembobotan. Terdapat beberapa tahap yang terdapat di dalam *text preprocessing*, antara lain:

2.2.7.1 *Case Folding*

Pada dasarnya setiap teks dokumen tidak semua konsisten dalam menggunakan huruf kapital. Maka, peran *case folding* dibutuhkan dalam konversi data keseluruhan teks dalam dokumen menjadi bentuk standar yaitu *lowercase* (huruf kecil). Contohnya seseorang ingin mencari informasi komputer dengan mengetik kata kunci 'KOMPUTER', 'KomPUter', atau 'komputer'. Tetap hasil yang di temukan yakni 'komputer'. *Case folding* hanya mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf 'a' sampai 'z' yang diterima. Karakter selain dari huruf tersebut akan di hilangkan dan di anggap *delimiter* atau urutan pembatas) (Ahmad et al., 2018).

2.2.7.2 *Tokenizing*

Tokenizing atau bisa juga disebut *parsing*. *Parsing* memotong *input string* dalam kata-kata yang tersusun didalamnya termasuk *spasi* yang digunakan

sebagai pemisah dari setiap kata dalam susunan kata. Contohnya, “saya belajar *information retrieval*”, setelah melalui *tokenizing* akan menjadi seperti ini “saya”, “belajar”, “*information*”, “*retrieval*” (Ahmad et al., 2018).

2.2.7.3 *Filtering*

Langkah mengambil kata-kata yang penting dari hasil *token*. Algoritma yang digunakan biasanya *stoplist* (membuang kata kurang penting). *Stoplist/stopword* adalah kata-kata yang tidak deskriptif yang dapat di hapus menggunakan pendekatan *bag-of-words*. Contohnya “yang”, “dan”, “di”, “dari” dan seterusnya (Ahmad et al., 2018).

2.2.7.4 *Stemming*

Ini adalah sebuah proses transformasi kata yang terdapat dalam sebuah dokumen ke kata-kata inti asli (*root word*) dengan menggunakan aturan-aturan tertentu. Proses *stemming* pada teks berbahasa Indonesia berbeda jika dibandingkan dengan Bahasa lain. Contoh, pada Bahasa inggris, proses yang dilakukan hanya menghilangkan *sufiks* (akhiran kata).

Pada teks Bahasa Indonesia, baik *sufiks* maupun *prefiks* (awalan kata) juga di hilangkan. Contoh, ‘Bersama’, ‘kebersamaan’, ‘menyamai’, jika melalui proses *stemming* maka akan menjadi ‘sama’. Namun, kerja *stemming* juga bervariasi tergantung pada *domain* Bahasa yang digunakan (Ahmad et al., 2018).

2.2.8 *Data Split*

Data split adalah membagi dataset menjadi dua jenis data yaitu data training dan data testing. Data training digunakan untuk *fit* model *machine learning*, sedangkan data testing untuk evaluasi hasil *fit* model yang akan digunakan juga menghitung nilai evaluasi RMSE. Menurut Saifudin (2018), Data training digunakan untuk melatih algoritma klasifikasi, sedangkan data testing digunakan untuk menguji algoritma atau model yang telah dilatih (Saifudin, 2018)

2.3 Tinjauan Peneliti Terdahulu

Setelah mengetahui teori dasar, selanjutnya peneliti akan melakukan tinjauan terhadap penelitian yang memiliki keterkaitan dan permasalahan yang

sama dengan penelitian yang dilakukan. Penelitian terkait harus menggunakan metode yang sama dengan penelitian yang dilakukan sekarang, yaitu menggunakan metode regresi SVR parameter *kernel sigmoid*. Lalu penelitian-penelitian terkait akan disajikan dalam bentuk Tabel 2.1 sebagai berikut.

Tabel 2.1 Sepuluh Penelitian terkait

NO	Data	Metode	Evaluasi
1	Data Split train/ test = 70/30 dari 228 data pencarian kerja	SVR – ALO, kernel Sigmoid	RMSE 0,254027 (Akbar & Kurniawan, 2020)
2	Total Energy Data House type 4	SVR (Sigmoid), kNN, DTR, Fully Connected NN, LSTM, SVR (rbf)	SVR (Sigmoid) RMSE 49,23. MAE 10,94 (Nambiar et al., 2019)
3	Data Harga Saham Harian PT. Telkom	SVR Kernel Liner, polynomial, Sigmoid, Radial	SVR kernel Sigmoid, RMSE 82,67%. MAPE 2.13 (Fahmi, 2020)
4	Data SWH sejumlah 17568	Metode MLP kernel Sigmoid	MAPE 6,88. RMSE 0,12. MSE 0.01 (Nooriansyah et al., 2018)
5	Gambar retina yang di ambil menggunakan fundoskopi	NE, SC – spare coding-based SR SC+SVR, SL-SR, SK-SVR-Sigmoid kernel SVR	Rata-rata PSNR SK-SVR 0,19. Rata-rata MSE 0.68 (Jebadurai & Peter, 2018)

6	272 data opini dari media twitter	Algoritma SVM berbasis PSO k-fold = 5 kernel sigmoid	Akurasi terbaik 77,33% (Anggita, 2020)
7	Ulasan Pelanggan marketplace Shopee, Tokopedia, Bukalapak. Data training 3759 dan testing 940	SVM, kernel sigmoid parameter $C, y, dan\ r$	F1 Score 92% (Rianti et al., 2021)
8	Data social media twitter sebanyak 1977	SVM kernel Sigmoid	F-measure 0,82 (Aulia et al., 2021)
9	Training data tweet sebanyak 3649 menjadi data testing sejumlah 1095 tweet	Metode SVM kernel sigmoid parameter nu	Akurasi 96,25% dengan parameter nu 0,2 (Oktavia, 2018)
10	616 data abstrak skripsi Teknik Informatika UNSIKA tahun 2015-2019	SVM kernel Sigmoid free parameter	Akurasi 30% (Liani et al., 2020)

Tinjauan awal dari penelitian-penelitian di atas belum tentu bisa menjamin dalam menentukan nilai evaluasi RMSE adalah satu-satunya untung menghitung nilai kesalahan, tetapi bisa dijadikan acuan untuk membuktikan penggunaan *kernel sigmoid* dalam kasus tertentu masih mungkin dapat di gunakan di dalam penelitian-penelitian yang akan datang menggunakan metode dan data yang berbeda. Nantinya nilai yang di dapatkan di dalam penelitian ini bisa menjadi referensi di dalam penelitian yang akan datang.