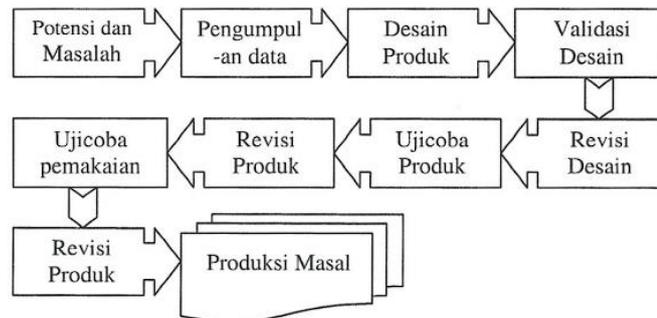


BAB III METODOLOGI PENELITIAN & PENGEMBANGAN

A. Metode Penelitian & Pengembangan

Metode penelitian *Research and Development* (R&D) adalah metode penelitian yang digunakan untuk mengembangkan suatu produk dan menguji keefektifan produk tersebut (Sugiyono, 2013, hlm. 297). Langkah – langkah dalam proses ini secara umum dikenal sebagai siklus R & D (*Research & Development*), di mana langkah – langkahnya terdiri dari: peninjauan dan pengkajian terhadap hasil – hasil penelitian sebelumnya yang memiliki keterkaitan dengan validitas komponen – komponen pada produk yang akan dikembangkan, pengembangan sebuah produk, pengujian terhadap produk, dan peninjauan ulang dan mengoreksi produk tersebut berdasarkan hasil uji coba (Sugiyono, 2013, hlm. 298). Penelitian & Pengembangan bertujuan untuk mengembangkan produk dan menguji keefektifan produk dalam mencapai tujuan. Sugiyono (Sugiyono, 2013, hlm. 298 – 311) menyatakan bahwa dalam pendekatan penelitian R&D mencakup tiga metode: deskriptif, evaluatif, dan eksperimen dengan serangkaian langkah – langkah yang harus ditempuh sebagaimana terdapat pada Gambar 3.1:



Gambar 3.1 Langkah - langkah *Research & Development*

(sumber: (Sugiyono, 2013, hlm. 298))

(a) metode deskriptif, meliputi:

- (1) potensi dan masalah, penelitian ini diawali mengidentifikasi masalah yang ada dalam penilaian esai melalui penyebaran kuesioner, dan observasi;
- (2) pengumpulan data, mengumpulkan informasi hasil kuesioner dan data dari studi literatur yang diperlukan pada penelitian ini untuk digunakan sebagai bahan perencanaan;

(b) metode evaluatif, meliputi:

- (3) desain produk, mengembangkan *prototype* aplikasi penilaian esai melalui analisis, perancangan, desain, dan penerapan algoritma ke dalam bentuk kode program;

- (4) validasi desain, melakukan uji coba awal desain produk yang dikembangkan kepada kelompok subjek dalam skala terbatas yakni meminta penilaian dari ahli di bidang teknik informatika dan *artificial intelligence*;
 - (5) revisi desain, *prototype* aplikasi penilaian esai akan dievaluasi kemudian melakukan perbaikan kekurangan dan kesalahan yang terjadi saat melakukan uji coba;
 - (6) uji coba produk, *prototype* aplikasi penilaian esai yang telah diperbaiki, selanjutnya akan diuji coba kembali untuk membandingkan efektivitas dan kemampuan sistem kerja yang baru dengan yang sebelumnya;
 - (7) revisi produk, melakukan perbaikan pada *prototype* aplikasi penilaian esai dengan sistem kerja yang baru setelah uji coba sebelumnya;
 - (8) ujicoba pemakaian, menguji *prototype* aplikasi penilaian esai kepada subjek di mana subjek penelitian ini adalah dosen yang ada di fakultas informatika dan komputer;
- (c) metode eksperimen, meliputi:
- (9) revisi produk, melakukan perbaikan akhir pada *prototype* aplikasi penilaian esai yang dikembangkan untuk menghasilkan produk akhir;
 - (10) produksi massal, *prototype* aplikasi penilaian esai yang telah dikembangkan akan diproduksi dan disebarakan setelah dinyatakan layak, tetapi pada jumlah produksi yang terbatas.

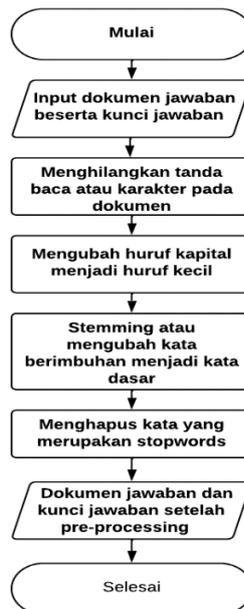
B. Model / Metode yang diusulkan

Penelitian ini mengusulkan tiga model / metode yang terdiri dari: model teoritis sebagai model yang akan digunakan untuk memecahkan masalah, model konseptual sebagai model analitis yang menjabarkan komponen – komponen produk akan dikembangkan serta keterkaitan antar komponen, dan model prosedural sebagai model yang menjabarkan atau menguraikan langkah – langkah yang harus diikuti dalam pengembangan produk.

1. Model Teoritis, Algoritma *Winnowing* dan *Cosine Similarity*

Pada model teoritis ini, terdapat beberapa langkah – langkah di antaranya:

- (1) melakukan *text pre-processing*, di mana teks akan diolah dengan cara mengubah huruf kapital menjadi huruf kecil, menghilangkan tanda baca, dan menghapus awalan dan akhiran teks, yang akan dijabarkan dalam bentuk *flowchart* sebagaimana dijelaskan pada gambar 3.2:



Gambar 3.2 Flowchart Text Pre-processing

penjelasan mengenai *flowchart text pre-processing* adalah sebagai berikut:

- (1) meng-inputkan dokumen jawaban ke dalam *function text_preprocessing*;
- (2) tanda baca, karakter, dan sebagainya pada dokumen dihilangkan terlebih dahulu sehingga hanya menyisakan kata – kata saja;
- (3) huruf kapital yang berada di awal kalimat diubah ke dalam bentuk huruf kecil;
- (4) kata yang memiliki awalan (prefix) beserta akhiran (suffix) akan dihapus awalan dan akhirannya;
- (5) kata – kata yang tidak diperlukan dihilangkan, contoh: adalah, yang, bahwa;
- (6) kata yang sudah melalui *pre-processing* siap digunakan untuk tahap selanjutnya;

Pseudocode text pre-processing dapat dilihat sebagaimana dijelaskan pada tabel 3.1;

Tabel 3.1 Pseudocode Text Pre-processing

Pseudocode
<pre> Function text_preprocessing(input dokumen: string){ string hapusTandaBaca ← dokumen.replace(/[^a-zA-Z0-9\s]/g, "") string lowerCase ← hapusTandaBaca.toLowerCase() string stemmingKata ← stemmer.stem(lowerCase) string stopwords ← stemmingKata.setStopWords('Indoneisa') string hasilAkhir ← stopwords return hasilAkhir } </pre>

Bentuk penerapan dari *text pre-processing* dapat dilihat sebagaimana dijelaskan pada tabel 3.2;

Tabel 3.2 Kalimat Sebelum *Text Pre-processing*

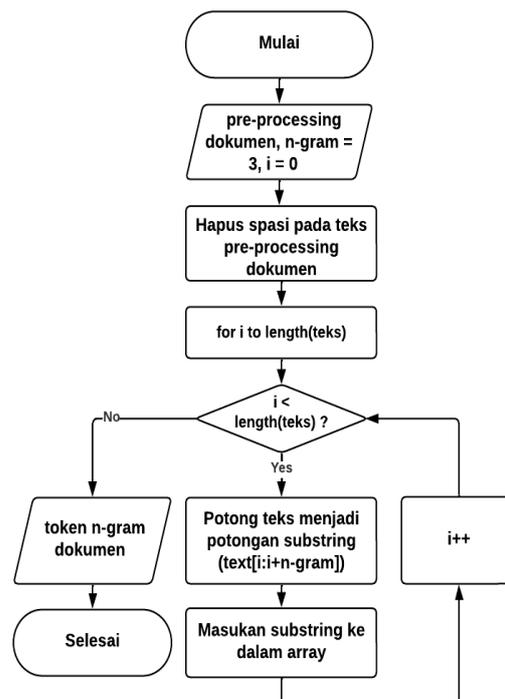
Jawaban	Kunci Jawaban
Database adalah sekumpulan data yang dikelola dan saling berhubungan dan terstruktur pada suatu sistem.	Database adalah sekumpulan data yang saling terkait dan terhubung yang disimpan secara terstruktur di dalam suatu sistem

Kalimat setelah melalui tahap *text pre-processing* sebagaimana dijelaskan pada tabel 3.3;

Tabel 3.3 Kalimat Setelah *Text Pre-processing*

Jawaban	Kunci Jawaban
database kumpul data kelola hubung struktur sistem	database kumpul data kait hubung disimpan struktur sistem

- (2) tokenisasi, dilakukan dengan cara memotong setiap kata pada teks menjadi beberapa potongan huruf; proses tokenisasi ini menggunakan metode k-gram; secara alur kerja, metode k-gram dijabarkan melalui *flowchart* sebagaimana dijelaskan pada gambar 3.3;



Gambar 3.3 *Flowchart* Tokenisasi

penjelasan mengenai *flowchart* dan *pseudocode* tokenisasi adalah sebagai berikut:

- (1) dokumen jawaban yang telah di *pre-processing* dimasukkan ke dalam *function text_tokenizing*;
- (2) data yang diinputkan terdiri dari: dokumen jawaban, jumlah k-gram, dan iterasi ke-l;
- (3) dokumen jawaban dipisah – pisah isi kalimatnya sehingga menjadi susunan potongan kata dan disusun ke dalam *array*;
- (4) melakukan iterasi sebanyak panjangnya kalimat teks jawaban, jika variabel iterasi ke-i lebih kecil dari panjangnya kalimat teks, iterasi akan terus dilakukan;
- (5) selama proses iterasi, setiap susunan kata pada *array* akan dipotong hingga digabungkan menjadi dua buah kata dan dikonversi ke dalam string tanpa ada spasi;
- (6) dua buah kata tersebut dimasukkan ke dalam data struktur *array* yang baru, pada *array* baru inilah teks jawaban telah ditokenisasi;
- (7) mengembalikan dokumen jawaban yang telah ditokenisasi dalam sebuah *array* untuk dapat digunakan pada tahap berikutnya;

Tabel 3.4 Tokenisasi

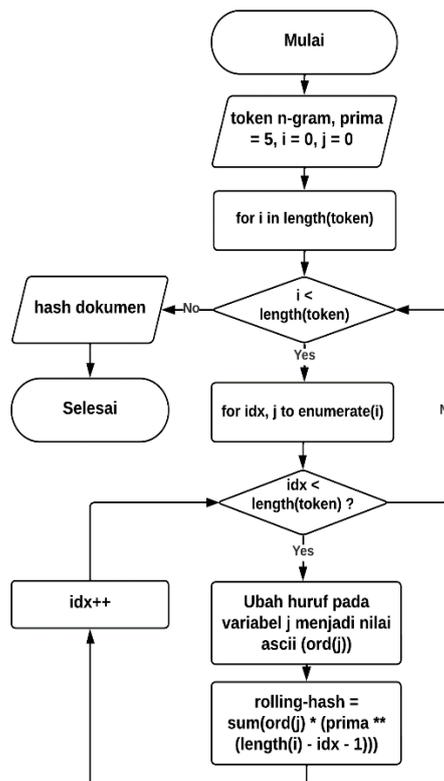
Pseudocode
<pre> Function text_tokenizing(input preprocessedDoc: string, n_grams: int, i: int){ string splittingDoc ← preprocessedDoc.split() array n_gramsResult for i in each element in length(preprocessedDoc) – n_grams{ string n_gram ← preprocessedDoc.slice(i, i + n_grams).join(""); n_gramsResult.push(n_gram) } return n_gramsResult } </pre>

Hasil dari penerapan dari *text pre-processing* dapat dilihat sebagaimana dijelaskan pada tabel 3.5;

Tabel 3.5 Tokenisasi Kalimat

Jawaban	Kunci Jawaban
"da", " at", " ta", " ab", " ba", " as", " se", " ek", " ku", " um", " mp", " pu", " ul", " ld", " da", " ta", " ak", " ke", " el", " lo", " la", " ah", " hu", " bu", " un", " ng", " gs", " st", " tr", " ru", " uk", " kt", " tu", " ur", " rs", " si", " is", " te", " em"	"da", " at", " ta", " ab", " ba", " as", " se", " ek", " ku", " um", " mp", " pu", " ul", " ld", " da", " at", " ta", " ak", " ka", " ai", " it", " th", " hu", " ub", " bu", " un", " ng", " gd", " di", " is", " si", " im", " mp", " pa", " an", " ns", " st", " tr", " ru", " uk", " kt", " tu", " ur", " rs", " si", " is", " st", " te", " em"

- (3) *hashing* atau mengonversi token jawaban esai menjadi susunan kode angka; nilai *hash* selanjutnya dijadikan sebagai *document fingerprinting*; pada dokumen token jawaban esai, sejumlah karakter pertama pada token di-*hash* dengan cara karakter paling pertama dalam token dikonversi menjadi karakter ASCII dan begitupun seterusnya; setelah nilai *hash* pertama didapatkan, *hashing* berlanjut dengan menggeser satu karakter ke kanan atau dengan istilah menggeser *window*, *flowchart hashing* ditunjukkan sebagaimana dijelaskan pada gambar 3.4 dan *pseudocode hashing* ditunjukkan sebagaimana dijelaskan pada tabel 3.6;



Gambar 3.4 Flowchart Hashing

Penjelasan mengenai *flowchart hashing* adalah sebagai berikut:

- (1) dokumen berisi token k-gram diinputkan ke dalam *function rolling_hash* beserta dengan parameter lainnya seperti basis, dan panjang token yang dibutuhkan;
- (2) melakukan iterasi pada panjang token sampai paling akhir;
- (3) selama iterasi, setiap karakter huruf pada token diubah menjadi karakter ASCII;
- (4) setiap karakter dikalikan dengan basis yang dipangkatkan dengan jumlah panjang token, setiap masuk ke iterasi selanjutnya panjang jumlah token akan dikurangi satu sehingga nilai basisnya berbeda pada setiap karakter,

dan dijumlahkan secara keseluruhan untuk mendapatkan nilai *hash* pertama;

- (5) untuk menentukan nilai *hash* selanjutnya, panjang token akan digeser ke sebelah kanan satu karakter untuk mendapatkan susunan *string* baru, karakter awal pada token k-gram sebelumnya diambil;
- (6) karakter akhir pada token k-gram baru yang sedang diproses juga dipilih;
- (7) karakter awal dan karakter akhir diubah menjadi nilai ASCII;
- (8) nilai *hash* baru dihasilkan dengan cara nilai *hash* pertama dikurangi nilai ASCII karakter pertama yang telah dikalikan basis yang dipangkatkan dengan jumlah panjang karakter, lalu dikalikan basis dan kemudian ditambahkan dengan nilai ASCII karakter terakhir;
- (9) memasukkan nilai – nilai *hash* ke dalam data struktur berupa *array* pada variabel *hashResult*;
- (10) mengembalikan nilai *hashResult* agar dapat digunakan;

Tabel 3.6 *Rolling Hash*

Pseudocode
<pre> function rolling_hash (n_gram_token: string, panjang_token: int, basis: int){ int n ← length(n_gram_token) int[] power int[] hashResult for l = 1 to n power[l] = (power[l - 1] * base) % mod int currentHash = 0 for l = 0 to windowSize - 1 currentHash = (currentHash * base + ord(s[l])) % mod hashResult[0] = currentHash for l = 1 to n - windowSize currentHash = ((currentHash - power[windowSize - 1] * ord(s[l - 1])) % mod + mod) % mod currentHash = (currentHash * base + ord(s[l + windowSize - 1])) % mod hashResult [l] = currentHash return hashResult } </pre>

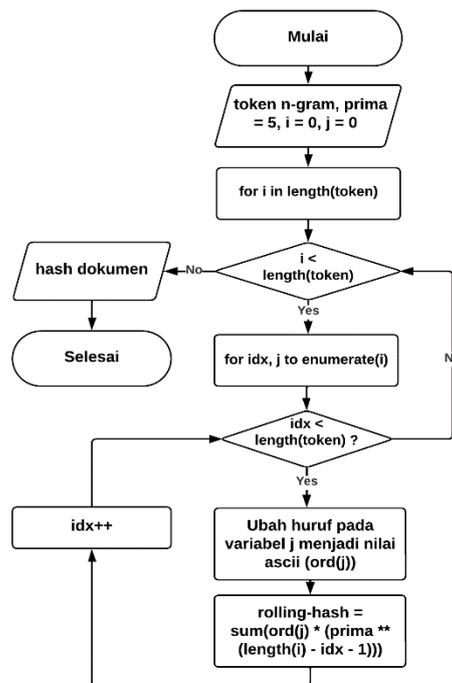
Hasil dari proses *hashing* pada token k-gram jawaban dan kunci jawaban adalah sebagaimana ditunjukkan pada tabel 3.7;

Tabel 3.7 *Hashing*

Jawaban	Kunci Jawaban
597, 601, 677, 583, 587, 600, 676, 612,	597, 601, 677, 583, 587, 600, 676, 612,
652, 694, 657, 677, 693, 640, 597, 601,	652, 694, 657, 677, 693, 640, 597, 601,
677, 592, 636, 613, 651, 663, 637, 589,	677, 592, 632, 590, 641, 684, 637, 683,

Jawaban	Kunci Jawaban
637, 683, 607, 695, 653, 630, 691, 694, 687, 692, 651, 697, 699, 685, 680, 640, 691, 681, 614	607, 695, 653, 615, 605, 640, 680, 634, 657, 657, 595, 665, 691, 694, 687, 692, 651, 697, 699, 685, 680, 640, 691, 681, 614

(4) menentukan *fingerprinting*, nilai *hashing* yang telah didapatkan dikelompokkan ke dalam *window*; ukuran *window* menentukan berapa banyak nilai *hash* yang ditampilkan; dari kumpulan nilai – nilai *hash* dipilih nilai *hash* minimum, *flowchart fingerprinting* dijelaskan pada gambar 3.5 dan *pseudocode*-nya pada tabel 3.8:



Gambar 3.5 Flowchart Fingerprint

penjelasan mengenai *flowchart* dan *pseudocode fingerprinting* adalah sebagai berikut:

- (1) nilai token *hash* hasil dari proses *rolling hash* dimasukkan ke dalam *function fingerprinting* beserta parameter lainnya seperti ukuran *w* atau *window*, dan panjang nilai token *hash*;
- (2) nilai token *hash* diiterasi sampai paling akhir;
- (3) nilai token *hash* dipotong dan dikelompokkan berdasarkan ukuran *window*,
- (4) setiap *window*, dipilih nilai *hash* minimum dari nilai *hash* yang lain dalam satu *window*;
- (5) memasukkan nilai *hash* minimum ke dalam data struktur *array windowResult* sebagai dokumen berisi *fingerprinting*;

- (6) Mengembalikan nilai *windowResult* agar dapat digunakan pada tahap selanjutnya;

Tabel 3.8 *Pseudo Fingerprinting*

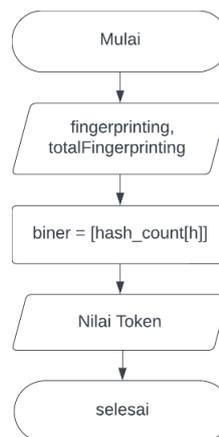
Pseudocode
<pre> Function fingerprinting (hashValues, w, l ← length(hashValues)){ int[] windowHashes for i to l - w { int[] window ← hashValues.slice(i, i+k) int minimumHash ← Math.min(window) windowHashes.push(minimumHash) } return windowHashes } </pre>

Hasil dari proses *fingerprinting* pada *hashing* jawaban dan kunci jawaban adalah sebagaimana ditunjukkan pada tabel 3.9;

Tabel 3.9 *Fingerprinting* Kalimat

Jawaban	Kunci Jawaban
583, 583, 583, 583, 587, 600, 612, 612,	583, 583, 583, 583, 587, 600, 612, 612, 597,
597, 597, 597, 592, 592, 592, 592, 592,	597, 597, 592, 592, 590, 590, 590, 590, 590,
592, 589, 589, 589, 589, 589, 589, 589,	590, 590, 607, 607, 605, 605, 605, 605, 605,
607, 607, 607, 630, 630, 630, 651, 651,	605, 595, 595, 595, 595, 595, 595, 595, 651,
651, 640, 640, 640, 614	651, 651, 651, 640, 640, 640, 614

- (5) Menghitung frekuensi kemunculan *fingerprinting* jawaban dan kunci jawaban. Nilai token pada *fingerprint* jawaban dan kunci jawaban digabungkan dan menghapus token yang sama. Kemudian masing – masing token jawaban dan kunci jawaban dihitung frekuensi kemunculannya berdasarkan gabungan dari *fingerprint* sebelumnya; *flowchart fingerprinting* dijelaskan pada gambar 3.6 dan *pseudocode*-nya pada tabel 3.10:



Gambar 3.6 *Flowchart* Menghitung Kemunculan Hashing

penjelasan mengenai *flowchart* dan *pseudocode* konversi biner adalah sebagai berikut:

- (1) Menginputkan nilai *fingerprinting* dari jawaban dan kunci jawaban yang telah digabungkan;
- (2) Menghitung frekuensi token *fingerprinting* pada variabel *hash* yang ada pada gabungan token *fingerprinting*;
- (3) Mengembalikan frekuensi kemunculan token;

Tabel 3.10 *Pseudocode* Menghitung Kemunculan Hashing

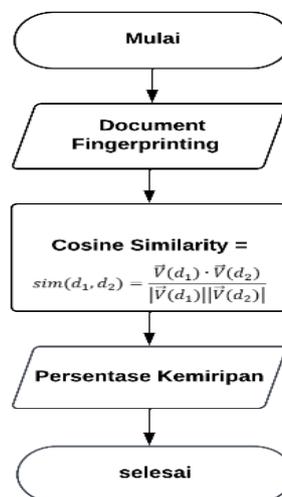
Pseudocode
<pre> biner_representation(hashes, all_hashes){ hash_count = {h: 0 for h in all_hashes} for h in hashes: hash_count[h] += 1 return [hash_count[h] for h in all_hashes] } </pre>

Hasil dari proses menghitung kemunculan hashing ditunjukkan pada tabel 3.11;

Tabel 3.11 Hasil Konversi Biner

Jawaban	Kunci Jawaban
[583] : 4; [587] : 1; [600] : 1; [612] : 2;	[583] : 4; [587] : 1; [600] : 1; [612] : 2;
[597] : 3; [592] : 6; [589] : 7; [607] : 3;	[597] : 3; [592] : 2; [589] : 0; [607] : 2;
[630] : 3; [651] : 3; [640] : 3; [614] : 1;	[630] : 0; [651] : 4; [640] : 3; [614] : 1;
[590] : 0; [605] : 0; [595] : 0	[590] : 7; [605] : 6; [595] : 7

- (6) mengukur tingkat similaritas antara dokumen jawaban esai dengan dokumen kunci jawaban dengan menggunakan algoritma *cosine similarity* sebagaimana dijelaskan pada gambar 3.7 mengenai *flowchart cosine similarity*;



Gambar 3.7 *Flowchart Cosine Similarity*

penjelasan mengenai *flowchart* dan *pseudocode* penilaian esai adalah sebagai berikut:

- (1) dokumen *fingerprinting* yang merupakan bentuk *fingerprinting* dari jawaban esai dan kunci jawaban esai dimasukkan ke dalam *function penilaianEsai*;
- (2) menghitung *dot product* dengan cara mengalikan nilai *hash* pada dokumen *fingerprinting* jawaban dengan dokumen *fingerprinting* kunci jawaban;
- (3) menghitung *magnitudeA* dan *magnitudeB* melalui setiap nilai *hash* dari dokumen *fingerprinting* jawaban esai dan kunci jawaban;
- (4) mendapatkan nilai tingkat kemiripan antara jawaban dengan kunci jawaban dengan rumus perhitungan nilai kemiripan:

$$= \frac{(4 * 4) + \dots + (0 * 7)}{\text{sqrt}(3^2 + \dots + 1^2) * \text{sqrt}(3^2 + \dots + 1^2)};$$

$$= \frac{71}{12,3693 * 14,1067} = 0,406898;$$

untuk mendapatkan nilai perolehan, nilai kemiripan dikalikan dengan bobot soal yang diberikan misal 5. Sehingga nilai perolehan didapatkan dengan cara:

$$= 0,406898 \times 5.$$

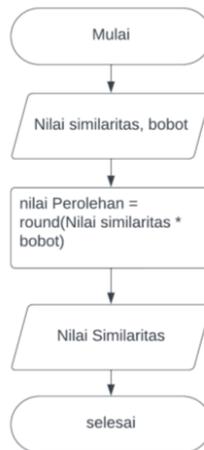
$$= 2$$

untuk *pseudocode* algoritma *Cosine Similarity* dapat dilihat sebagaimana ditunjukkan pada tabel 3.12.

Tabel 3.12 *Cosine Smilarity*

Pseudocode
<pre>function cosineSimilarity (int[] docJ, int[] docKJ){ Int dotProduct ← 0 for i in length(docJ) { dotProduct ← docJ [i] * docKJ [i] } int magnitudeA = Math.squareroot(docJ.reduce((sum, val) => sum + val ** 2, 0)) int magnitudeB = Math.squareroot (docKJ.reduce((sum, val) => sum + val ** 2, 0)) const nilaiSimilarity = dotProduct / (magnitudeA * magnitudeB) return nilaiSimilarity }</pre>

nilai perolehan kemudian dibulatkan ke bilangan bulat terdekat sehingga nilai perolehan yang didapatkan adalah 2. Proses penilaian esai ini dijelaskan melalui *flowchart* penilaian esai sebagaimana ditunjukkan pada gambar 3.8.



Gambar 3.8 Flowchart Penilaian Esai

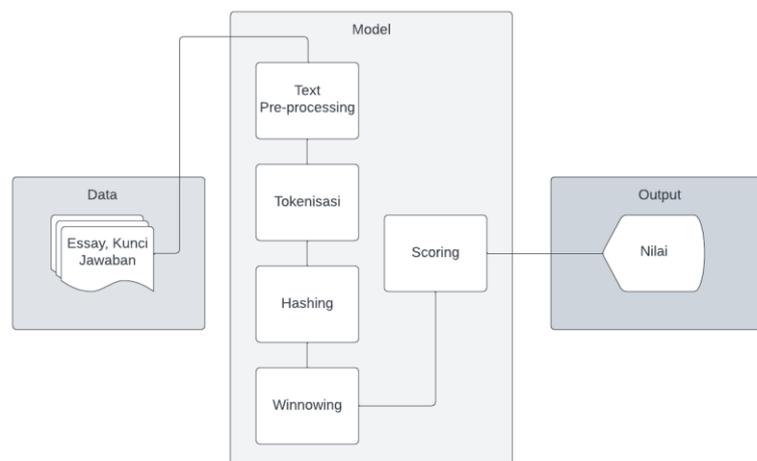
Sedangkan untuk *pseudocode* penilaian esai dapat dilihat pada tabel 3.13.

Tabel 3.13 Penilaian Esai

Pseudocode
<pre> function penilaianEsai(nilaiSimilaritas, bobot) { return round(nilaiSimilaritas * bobot) } </pre>

2. Model Konseptual, *Automated Essay Scoring*

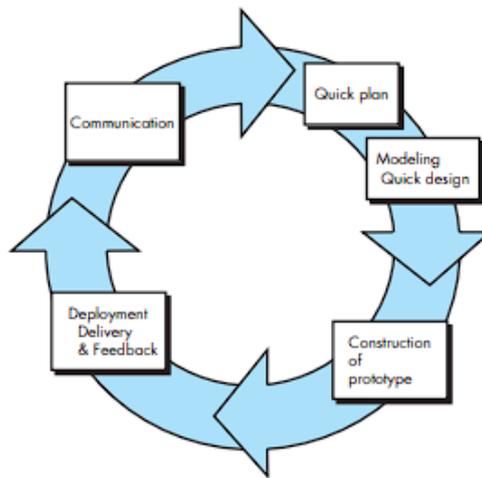
Model konseptual yang diusulkan pada penelitian ini adalah *Automated Essay Scoring*. *Automated Essay Scoring* terdiri dari beberapa komponen di antaranya: komponen data berupa jawaban esai beserta kunci jawabannya, komponen model berupa *agent* yang mengolah teks pada esai melalui proses: *text pre-processing*, *tokenization*, *hashing*, dan komponen *output* berupa hasil skor yang diperoleh setelah melalui proses pencocokan teks oleh model.



Gambar 3.9 Komponen *Automated Essay Scoring*

3. Model Prosedural, *Prototyping*

Model prosedural yang diajukan pada penelitian ini adalah model pengembangan *prototyping*. Dalam pengembangan perangkat lunak, pengguna mengidentifikasi kebutuhan dan persyaratan secara umum tetapi tidak mengidentifikasi persyaratan terperinci untuk fungsi pada perangkat lunak sehingga pengembang melakukan interaksi dengan pengguna. Dalam paradigma model *prototyping*, terdapat beberapa tahapan yang perlu dilakukan yang dibagi menjadi empat bagian, di antaranya:



Gambar 3.10 Tahapan *Prototyping*

(Sumber: Pressman, 2015, p.46)

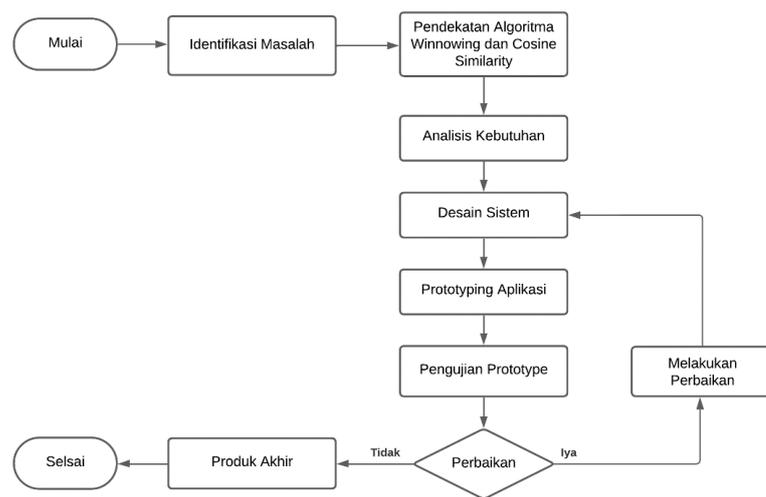
- (1) *communication*, pada tahap ini dilakukan proses identifikasi permasalahan yang terjadi dengan melakukan beberapa instrumen penelitian yakni melalui observasi, wawancara, lalu kemudian data yang dikumpulkan dilakukan analisis kebutuhan pengguna di mana dalam penelitian ini pengguna adalah dosen mata kuliah,
- (2) *quick plan*, tahapan ini dilakukan perencanaan pengembangan aplikasi penilaian esai, panduan berupa buku dan sebagainya mengenai *flowchart* dan alur kerja metode yang diusulkan, pihak yang akan dijadikan sebagai subjek uji coba serta pemilihan metode untuk menguji tingkat akurasi algoritma yang diusung,
- (3) *modelling quick design*, mendeskripsikan model sistem yang dikembangkan dengan menggunakan diagram UML (*Unified Modelling Language*) dan perancangan *interface* sistem,
- (4) *construction of prototype*, setelah *blueprint* sistem dibuat menggunakan UML dan telah melalui tahap evaluasi selanjutnya adalah menerjemahkan *blueprint*

tersebut ke dalam struktur kode program sebagai bentuk realisasi pengembangan *prototype* aplikasi penilaian esai,

- (5) *deployment delivery & feedback*, *prototype* aplikasi segera disebarakan kepada pengguna, yakni dosen mata kuliah, untuk diuji coba dan kemudian mendapatkan *feedback* berupa kritik dan saran yang akan dijadikan sebagai bahan evaluasi.

C. Prosedur Pengembangan

Langkah – langkah pengembangan pada penelitian yang dilakukan adalah sebagaimana dijelaskan pada gambar 3.4:



Gambar 3.11 Alur Pengembangan

Dari gambar di atas, berikut adalah penjelasan mengenai alur pengembangan pada penelitian ini:

- (1) Mengidentifikasi masalah mengenai penilaian esai di mana identifikasi masalahnya adalah penilaian esai yang cenderung kurang objektif dan efektif dalam memberikan nilai.
- (2) Pada tahap ini adalah pendekatan solusi berupa algoritma *WInnowing* dan *Cosine Similarity* di mana *WInnowing* digunakan untuk melakukan *fingerprinting* dokumen jawaban dan *Cosine Similarity* untuk mengukur kemiripan *fingerprinting* dokumen.
- (3) Pada tahap ini analisis kebutuhan meliputi pengembangan sistem dengan metode *prototyping* lalu menentukan bobot pada soal, jumlah *window*, k-gram, dan angka untuk proses *hashing*.
- (4) Pada tahap ini dilakukan perancangan model dan alur pengembangan *prototype* aplikasi penilaian jawaban esai.

- (5) Pada tahap ini, model rancangan dikembangkan menjadi sebuah *prototype* aplikasi penilaian jawaban esai untuk memecahkan masalah yang teridentifikasi di tahap awal.
- (6) Pada tahap ini, *prototype* akan diuji coba oleh ahli di bidang *artificial intelligence* dan dosen sebagai pengguna untuk menguji apakah *prototype* aplikasi yang telah dikembangkan sudah sesuai dengan kebutuhan atau belum.
- (7) Pada tahap ini, dilakukan perbaikan pada kekurangan yang terdapat pada *prototype* aplikasi setelah tahap pengujian. Jika masih terdapat kekurangan, *prototype* aplikasi akan direvisi dan diuji kembali.
- (8) Hasil akhir merupakan tahap di mana *prototype* aplikasi dinyatakan telah memenuhi kebutuhan setelah melalui pengujian dan revisi. *Prototype* aplikasi dapat dikembangkan lebih lanjut menjadi aplikasi seutuhnya.

D. Uji Coba Produk

Untuk dapat mengukur tingkat ketepatan, keefektivan, dan/atau daya tarik dari produk yang dihasilkan, perlu dilakukan pengumpulan data yang akan digunakan sebagai dasar dari penelitian ini. Uji coba produk memiliki beberapa proses, berikut adalah proses pengujian produk:

1. Desain Uji Coba

Pada penelitian ini akan ada dua tahapan pengujian produk, yakni pengujian yang akan dilakukan kepada ahli dan pengujian yang akan dilakukan kepada pengguna.

a. Uji Coba dari Pengguna

Pengujian dari pihak pengguna dilakukan dengan melalui kuesioner di mana hasilnya dijadikan sebagai acuan dalam melihat tingkat kelayakan dari produk. Uji coba dari pengguna ini dilakukan oleh dosen.

b. Uji Coba dari Ahli

Pengujian kepada pihak ahli yang menguasai di bidang teknik informatika dan *artificial intelligence* untuk mengetahui ketepatan dan keefektifan dalam penerapan algoritma *Winnowing* dan *Cosine Similarity* melalui kuesioner yang disebarakan.

2. Subjek Uji Coba

Pihak yang menjadi subjek uji coba adalah dosen tetap yang berjumlah 24 orang sebagai responden di fakultas Informatika dan Komputer di mana dua orang di antaranya menjadi pihak ahli di bidang teknik informatika dan *artificial intelligence*.

3. Jenis Data

Data yang dibutuhkan untuk penelitian ini berupa berupa data jawaban esai dari mahasiswa, kunci jawaban dari mata kuliah yang diujikan, dan bobot soal.

4. Instrumen Pengumpulan Data

Pada penelitian ini pengumpulan data dilakukan dengan maksud memperkuat bukti pada permasalahan yang terjadi selama penilaian esai melalui instrumen berupa kuesioner. dan untuk menguji sistem.

a. Instrumen Pengidentifikasi Masalah

Instrumen yang digunakan untuk mengidentifikasi masalah untuk penelitian ini adalah kuesioner jenis pertanyaan tertutup. Butir – butir pertanyaan tertutup dapat dilihat sebagaimana ditunjukkan pada tabel 3.14:

Tabel 3.14 Instrumen Pengidentifikasi Masalah

No	Pertanyaan
1	Dalam melakukan pemeriksaan dan penilaian jawaban esai per mahasiswa, apakah penggunaan waktu yang digunakan relatif singkat?
2	Apakah selama melakukan penilaian esai terdapat kesulitan yang bapak / ibu alami?
3	Dalam melakukan penilaian esai, perlu adanya objektivitas selama melakukan penilaian seperti dengan cara tidak membeda – bedakan antar mahasiswa, memberikan nilai sesuai dengan kompetensi yang dimiliki. Tetapi pelaksanaannya terdapat beberapa kasus penilaian dilakukan secara subjektif seperti memberikan nilai berdasarkan gender di mana mahasiswi akan lebih diutamakan dalam memberikan bobot nilai sebagaimana dijelaskan pada situs: https://persfe.com/menilik-objektivitas-dosen-dalam-memberikan-nilai/ . Dari kasus tersebut, apakah bapak atau ibu pernah melakukan penilaian esai secara subjektif?

b. Instrumen Pengujian Sistem

Pengujian sistem yang dilakukan untuk penelitian ini terdiri dari pengujian dari ahli dan dari pengguna. Instrumen yang digunakan untuk ahli adalah *black box* dan untuk pengguna adalah PSSUQ.

1) Instrumen Untuk Ahli

Pihak ahli yang terlibat dalam penelitian ini adalah dua orang dosen ahli di bidang teknik informatika dan *artificial intelligence*. Instrumen penelitian yang digunakan adalah pengujian menggunakan *black box*. Ini menguji kualitas *software* dari segi fungsional tanpa memperhatikan kode program dan desain *software*. Proses pengujian ini dilakukan dengan membuat *test case*. Hasil dari pengujian ini adalah kesesuaian fungsi yang berjalan apakah sesuai atau tidak. Bentuk pengujian ini sebagaimana dijelaskan pada tabel 3.15:

Tabel 3.15 Kuesioner Untuk Ahli

No	Skenario	Proses yang diuji	Hasil yang Diharapkan	Hasil Pengujian	
				Iya	Tidak
1	<p>a. NIP/NPM dan Password tidak diinputkan</p> <p>b. NIP/NPM dan Password yang diinputkan salah</p> <p>c. NIP/NPM dan Password yang diinputkan benar</p>	Halaman Login	<p>a. Menampilkan pesan bahwa NIP/NPM dan <i>password</i> wajib diisi</p> <p>b. Menampilkan pesan bahwa NIP/NPM dan <i>password</i> salah</p> <p>c. Masuk ke halaman <i>home</i></p>		
2	Halaman <i>home</i> dosen dan mahasiswa menampilkan <i>drop-down</i> tahun akademik dan <i>drop-down</i> semester	Halaman <i>Home</i>	Menampilkan data tahun akademik pada <i>drop down</i> tahun akademik, dan <i>drop data</i> semester pada <i>drop down</i> semester		
3	Pada <i>drop-down</i> tahun akademik bisa memilih tahun akademik dan <i>drop-down</i> semester bisa memilih semester	Halaman <i>Home</i>	Menampilkan data mata kuliah sesuai tahun akademik dan semester atau menampilkan pesan bahwa mata kuliah tidak tersedia		
4	Mengklik tombol Input Soal	Halaman <i>Home</i>	Masuk ke halaman input soal esai yang menampilkan tabel berisi data soal.		
5	<p>a. Menyimpan data dengan nilai data yang kosong</p> <p>b. Menyimpan data dengan nilai data yang terisi</p>	Halaman Input Soal Esai	<p>a. Menampilkan pesan konfirmasi, jika mengklik tombol Simpan akan menampilkan pesan bahwa data tidak boleh kosong</p> <p>b. Menampilkan pesan konfirmasi,</p>		

No	Skenario	Proses yang diuji	Hasil yang Diharapkan	Hasil Pengujian	
				Iya	Tidak
			jika mengklik tombol Simpan akan menampilkan pesan bahwa data berhasil disimpan		
6	Mengubah status soal apakah ditampilkan atau tidak	Halaman Input Soal Esai	Menampilkan pesan bahwa status soal berhasil diubah dan status pada data soal berubah		
7	Mengklik tombol Perbarui Data pada data soal	Halaman Input Soal Esai	Data soal, kunci jawaban, dan bobot soal ditampilkan pada <i>form</i> input soal		
8	a. Meng- <i>update</i> data dengan nilai data yang kosong b. Meng- <i>update</i> data dengan nilai data yang diubah.	Halaman Input Soal Esai	a. Menampilkan pesan konfirmasi, jika mengklik tombol Simpan akan menampilkan pesan bahwa nilai data harus diisi b. Menampilkan pesan konfirmasi, jika mengklik tombol Simpan akan menampilkan pesan bahwa data berhasil disimpan		
9	Mengklik tombol Jawaban Soal	Halaman <i>Home</i>	Masuk ke halaman jawaban esai mahasiswa yang menampilkan data jawaban esai dan nilai		
10	Mengklik tombol Soal UTS atau Soal UAS	Halaman <i>Home</i>	Menampilkan halaman yang berisi soal dan <i>form</i> untuk menjawab soal		

No	Skenario	Proses yang diuji	Hasil yang Diharapkan	Hasil Pengujian	
				Iya	Tidak
11	a. Menyimpan data dengan nilai data yang kosong b. Menyimpan data dengan nilai data yang terisi	Halaman Input Jawaban Esai	a. Menampilkan pesan konfirmasi, jika mengklik tombol Simpan akan menampilkan pesan bahwa nilai data harus diisi b. Menampilkan pesan konfirmasi, jika mengklik tombol Simpan akan menampilkan pesan bahwa data berhasil disimpan		
12	Jawaban esai mahasiswa dan nilai perolehannya ditampilkan di halaman Jawaban Esai	Halaman Jawaban Esai	Menampilkan jawaban esai mahasiswa dan nilai perolehannya		
13	Mengklik tombol Logout yang terdapat di halaman <i>Home</i>	Halaman <i>Home</i>	Menampilkan pesan konfirmasi apakah ingin keluar dari sistem atau tidak. Jika iya, akan kembali ke halaman <i>login</i>		

Dari tabel 3.16, pada kolom "No" merupakan nomor urutan skenario pengujian, pada kolom "Skenario" merupakan serangkaian skenario pengujian fungsional *software*, kolom "Proses yang diuji"

Tabel 3.16 Pernyataan Terbuka untuk Ahli

Kritik	Saran

2) Instrumen Untuk Pengguna

Penelitian ini menggunakan instrumen berupa kuesioner tertutup dan terbuka untuk pengguna. Hasil dari kuesioner ini berupa data kuantitatif yang dikonversi menjadi kualitatif menggunakan skala *Likert*

yakni membagi dalam bentuk interval. Kuesioner yang diberikan kepada pengguna merupakan berupa kuesioner *Post-Study System Usefulness Questionnaire* (PSSUQ) versi 3. (Sauro & Lewis, 2012, hlm. 193) menyatakan bahwa PSSUQ versi 3 terdiri dari 16 pernyataan. Berikut merupakan 16 pertanyaan kuesioner PSSUQ untuk mengukur kelayakan produk dalam menggunakan sistem yang terdapat pada tabel 3.17.

Tabel 3.17 Kuesioner PSSUQ

No	Pertanyaan PSSUQ	5	4	3	2	1
1	Secara keseluruhan, saya puas dengan betapa mudahnya cara menggunakan sistem ini					
2	Sistem ini sederhana untuk digunakan					
3	Saya bisa menyelesaikan tugas dengan cepat menggunakan sistem ini.					
4	Saya merasa nyaman menggunakan sistem ini					
5	Sistem ini mudah untuk dipelajari					
6	Saya percaya saya jadi cepat produktif dengan menggunakan sistem ini					
7	Sistem memberikan pesan kesalahan yang secara jelas memberitahukan saya bagaimana cara memperbaiki masalahnya					
8	Setiap saya membuat kesalahan ketika menggunakan sistem ini, saya dapat memulihkannya dengan mudah dan cepat					
9	Informasi (seperti <i>online help</i> , <i>on-screen message</i> , dan dokumentasi lain) yang disediakan oleh sistem ini jelas					
10	Sistem ini mudah untuk mencari informasi yang saya butuhkan					
11	Informasi ini efektif dalam membantu melengkapi tugas dan skenario					
12	Penataan informasi pada sistem ini jelas					
13	Antarmuka sistem ini memuaskan					
14	Saya suka menggunakan antarmuka sistem ini					
15	Sistem ini memiliki semua fungsi dan kemampuan yang saya harapkan					
16	Secara keseluruhan, saya puas dengan sistem ini					

Dari 16 item pernyataan dalam kuesioner, dapat dikelompokkan menjadi empat aspek PSSUQ, yaitu: skor kepuasan secara keseluruhan (*Overall*), kegunaan sistem atau teknologi (*System Usefulness*), kualitas informasi (*Information Quality*), dan kualitas antarmuka (*Interface Quality*) (Sauro & Lewis, 2012, hlm. 192). Berikut ini adalah tabel standar penilaian PSSUQ sebagaimana dijelaskan pada tabel 3.18.

Tabel 3.18 Aturan Perhitungan Skor PSSUQ

Nama Skor	Rata – rata <i>Item Respons</i>
<i>Overall</i>	No. item 1 – 16
<i>System Usefulness (Sysuse)</i>	No. item 1 – 6
<i>Information Quality (Infoqual)</i>	No. item 7 – 12
<i>Interface Quality (Interqual)</i>	No. item 13 – 15

Selain kuesioner, terdapat juga pertanyaan terbuka yang digunakan untuk mengumpulkan masukan dari pengguna mengenai sistem, dan informasi tersebut digunakan sebagai bahan evaluasi produk. sebagaimana dijelaskan pada tabel 3.19.

Tabel 3.19 Pernyataan Terbuka untuk Pengguna

Kritik	Saran

c. Skala Penilaian

Skala penilaian yang digunakan pada penelitian ini di antaranya:

1) Skala *Likert*

Penelitian ini menggunakan skala *Likert* untuk mengukur pendapat responden dari kuesioner yang diberikan yakni PSSUQ. Jawaban pada setiap *item* instrumen mempunyai rentang dari sangat setuju hingga sangat tidak setuju (Sugiyono, 2013, hlm. 94). Setiap jawaban pada pernyataan diberikan skor sebagaimana dijelaskan oleh tabel 3.20.

Tabel 3.20 Skala *Likert*

No	Kategori	Skor
1	Sangat Setuju	5
2	Setuju	4
3	Netral	3
4	Tidak Setuju	2
5	Sangat Tidak Setuju	1

2) Skala *Guttman*

Selain menggunakan skala *Likert*, penelitian ini menggunakan skala *Guttman* untuk mengukur pendapat responden dari kuesioner yang

diberikan yakni *Black Box*. Jawaban pada *item* instrumen mempunyai dua nilai, yakni iya dan tidak (Sugiyono, 2013, hlm. 96–97). Setiap jawaban pada pernyataan diberikan skor sebagaimana dijelaskan oleh tabel 3.21.

Tabel 3.21 Skala *Guttman*

No	Kategori	Skor
1	Iya	1
2	Tidak	0

5. Teknik Analisis Data

a. Uji Produk

Dari data yang dikumpulkan melalui kuesioner tertutup dan terbuka yakni data kualitatif, selanjutnya data ini dijumlahkan dan dihitung tingkat persentase kelayakan dengan menggunakan rumus:

$$\text{Persentase Kelayakan (\%)} = \frac{\text{Skor yang didapatkan}}{\text{Skor maksimal}} \times 100\%$$

Persentase kelayakan digunakan untuk menilai kelayakan dari aspek-aspek yang diteliti. Menurut (Lubis dkk., 2023), persentase kelayakan ini diinterpretasikan ke dalam 5 kategori. Nilai maksimum yang diharapkan adalah 100%, sementara nilai minimum adalah 0%. Pembagian kelima kategori kelayakan dapat dilihat pada tabel 3.22.

Tabel 3.22 Kategori Kelayakan

No	Persentase Kelayakan	Kategori Kelayakan
1	$80 < X \leq 100\%$	Sangat Layak
2	$60 < X \leq 80\%$	Layak
3	$40 < X \leq 60\%$	Cukup
4	$20 < X \leq 40\%$	Tidak Layak
5	$0 < X \leq 20\%$	Sangat Tidak Layak

Sumber: (Lubis dkk., 2023, hlm. 85)

Tabel di atas dapat digunakan sebagai acuan dalam menentukan termasuk kategori apakah persentase kelayakan yang didapatkan.

b. Uji Hasil

Untuk mengukur tingkat ketepatan algoritma *Winnowing* dan *Cosine Similarity* dalam menilai jawaban esai, pengujian hasil menggunakan metode *Mean Absolute Percentage Error (MAPE)* dengan mengukur kesalahan relatif dengan menghitung rata – rata persentase selisih absolut antara nilai prediksi dengan nilai sebenarnya.

Rumus MAPE dapat dijabarkan sebagai berikut:

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^n \left| \frac{(x_i - y_i)}{x_i} \right| \times 100\%;$$

Keterangan:

i : urutan data;

x_i : nilai data sebenarnya (hasil nilai melalui perhitungan oleh dosen);

\hat{x}_i : nilai data prediksi yang dihasilkan oleh model;

N : jumlah data pada *database*;

semakin kecil nilai persentase yang dihasilkan, semakin baik pula model yang digunakan. Berdasarkan penelitian yang dilakukan oleh (Sumari dkk., 2020) interpretasi nilai MAPE dijelaskan pada tabel 3.23.

Tabel 3.23 Interpretasi Nilai MAPE

No	Persentase Nilai MAPE	Interpretasi
1	<10%	Sangat Baik
2	10% – 20%	Baik
3	20% – 50%	Layak
4	> 50%	Tidak Baik

Sumber: (Sumari dkk., 2020, hlm. 644)