

BAB 4 PERANCANGAN

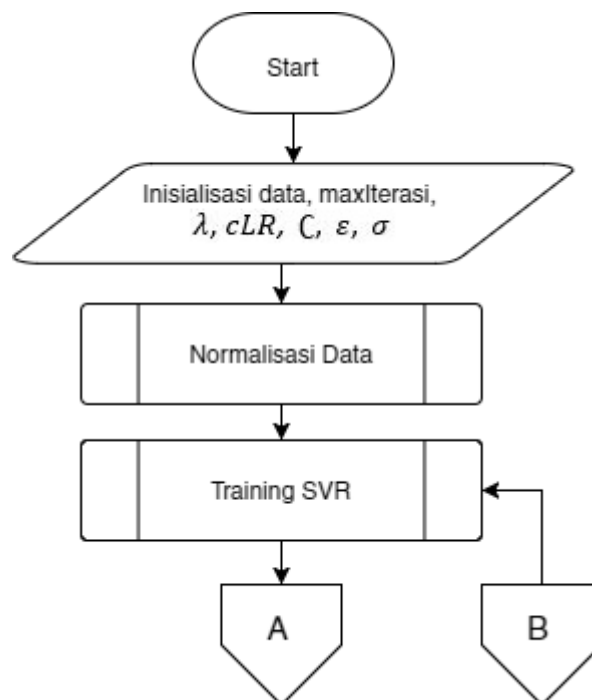
Pada bab ini akan dilakukan pembahasan mengenai perancangan dalam sistem prediksi indeks harga konsumen kelompok air, listrik, gas dan bahan bakar dengan metode SVR yang meliputi 4 tahapan yaitu formulasi permasalahan secara sederhana, penyelesaian permasalahan dengan metode *Support Vector Regression*, perancangan antarmuka pengguna dan perancangan pengujian sistem.

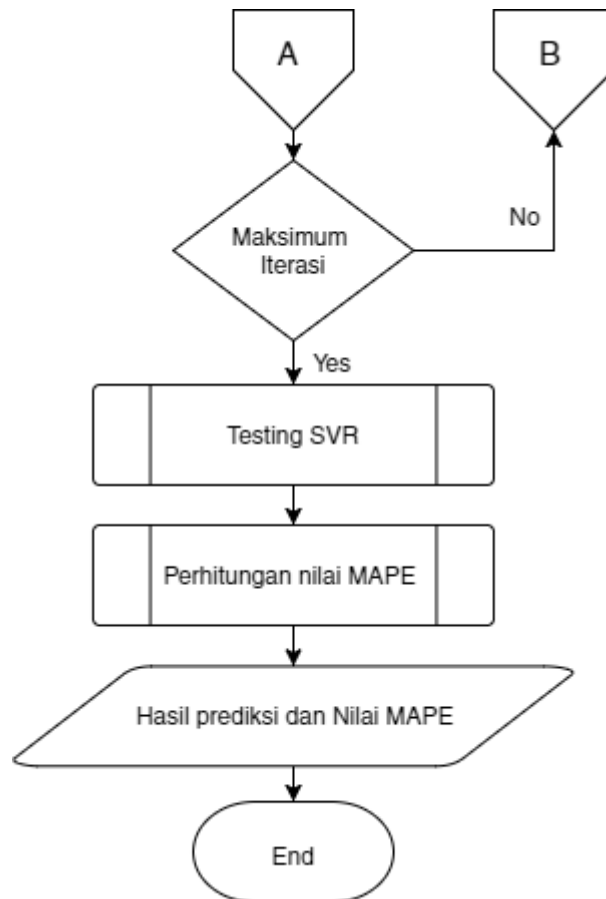
4.1 Formulasi Permasalahan

Berdasarkan uraian latar belakang permasalahan pada Bab 1 serta kajian pustaka yang telah diuraikan pada Bab 2, maka dapat diketahui bahwa metode SVR dapat digunakan untuk memprediksi indeks harga konsumen kelompok perumahan, air, listrik, gas dan bahan bakar. Berdasarkan analisis kebutuhan yang telah dijelaskan pada Bab 3, dalam menyelesaikan permasalahan tersebut diperlukan masukan berupa data indeks harga konsumen kelompok perumahan, air, listrik, gas dan bahan bakar pada tahun 2011-2016. Data tersebut kemudian akan diolah menggunakan metode SVR sehingga dapat menghasilkan data prediksi indeks harga konsumen kelompok perumahan, air, listrik, gas dan bahan bakar beserta nilai evaluasi yang didapatkan dengan membandingkan selisih antara data aktual dengan data hasil prediksi menggunakan metode *Mean Absolute Percentage Error (MAPE)*.

4.2 Penyelesaian Permasalahan Dengan *Support Vector Regression*

Penyelesaian permasalahan dengan metode SVR secara umum ditunjukkan pada Gambar 4.1.





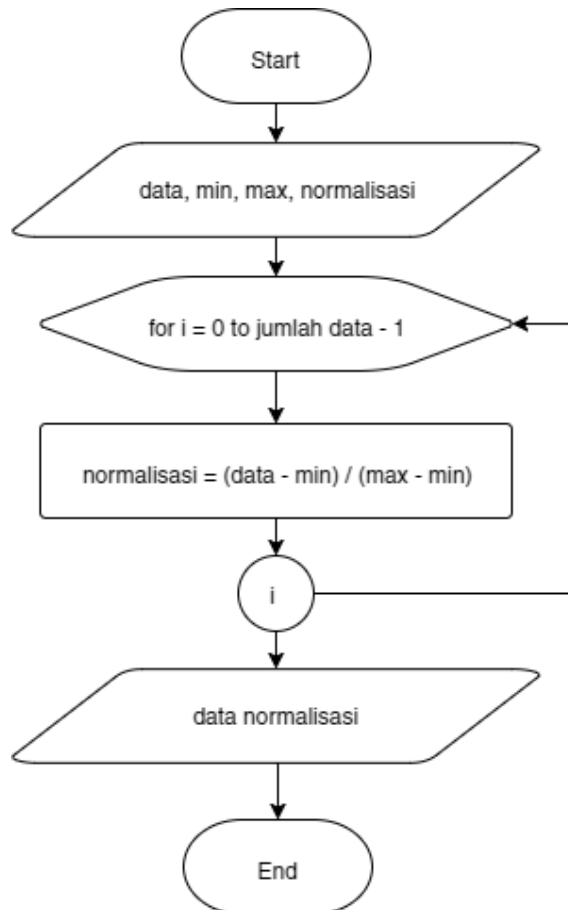
Gambar 4.1 Diagram Alir Metode Support Vector Regression

Tahap-tahap penyelesaian permasalahan dengan metode SVR adalah sebagai berikut.

1. *Input* berupa data berekstensi .csv dengan jumlah data *training* dan data *testing*, nilai parameter λ , cLR , C , ε , σ dan jumlah iterasi maksimum dimasukkan langsung oleh pengguna.
2. Proses normalisasi data untuk mentransformasi data menjadi data berskala 0 sampai 1 dengan metode *Min-Max*.
3. Proses *training* SVR yang meliputi perhitungan jarak antar data *training*, perhitungan matriks *Hessian* dengan fungsi kernel RBF dan perhitungan *sequential learning* yang akan diulangi hingga mencapai batas iterasi maksimum. Setelah itu dilakukan perhitungan nilai *error* dengan metode MAPE.
4. *Testing* SVR dilakukan oleh sistem dengan menghitung jarak antar data *testing*, perhitungan kernel RBF, dan perhitungan matriks *Hessian* dengan nilai *Lagrange Multiplier* dan parameter SVR yang telah diperoleh dari proses pelatihan kemudian menghitung nilai prediksi fungsi regresi $f(x)$ pada data *testing* untuk kemudian menghitung nilai *error* dengan MAPE.
5. *Output* yang dihasilkan adalah hasil prediksi serta nilai MAPE dari data *training* dan data *testing*.

4.2.1 Proses Normalisasi Data

Proses normalisasi data dilakukan untuk mentransformasi data menjadi data berskala 0 sampai dengan 1. Pada penelitian ini, metode normalisasi data yang digunakan ialah *Min-Max Normalization*. Diagram alir proses normalisasi data ditunjukkan pada Gambar 4.2.



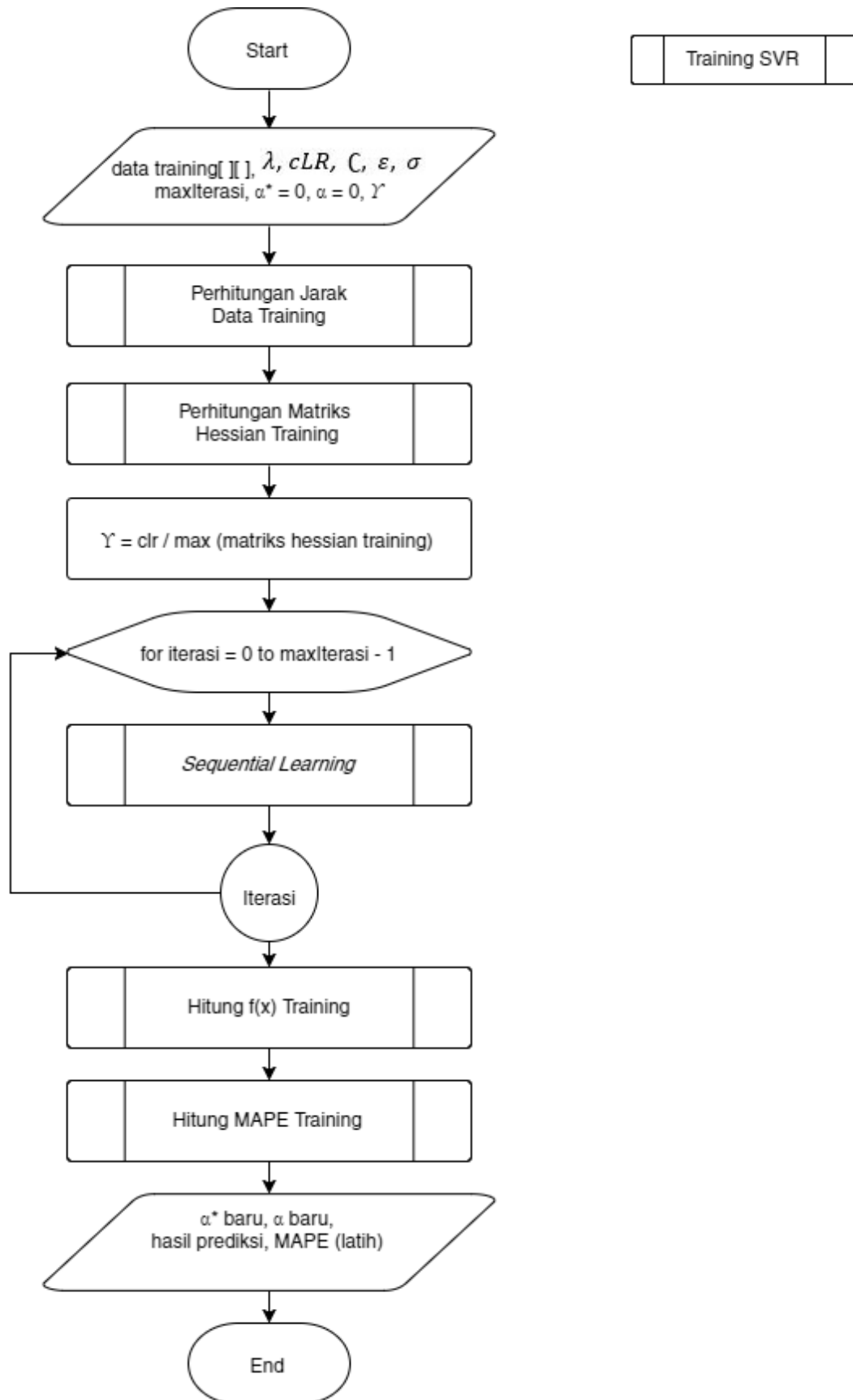
Gambar 4.2 Diagram Alir Proses Normalisasi Data

Langkah-langkah dalam proses normalisasi data adalah sebagai berikut.

1. Sistem memproses data yang dimasukkan dengan menghitung normalisasi data menggunakan metode *Min-Max* dengan cara mengurangi setiap data dengan nilai data minimum dari keseluruhan data dibagi dengan selisih antara data maksimum dengan data minimum.
2. *Output* yang dihasilkan adalah data yang telah dinormalisasi, yaitu data berskala 0 sampai dengan 1.

4.2.2 Proses *Training* SVR

Training SVR bertujuan untuk mendapatkan nilai regresi pada data *training* yang dimulai dengan menghitung jarak antar data *training*, matriks *Hessian*, perhitungan *sequential learning*, perhitungan nilai *error* dan perubahan nilai *Lagrange Multiplier*. Gambar 4.3 adalah digram alir proses pelatihan SVR.



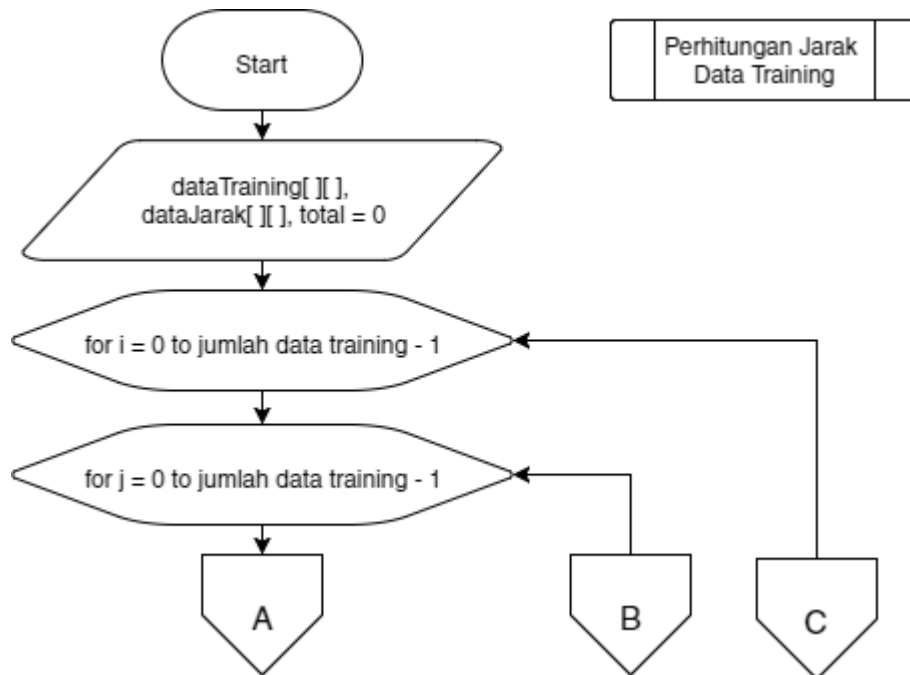
Gambar 4.3 Diagram Alir Proses Training SVR

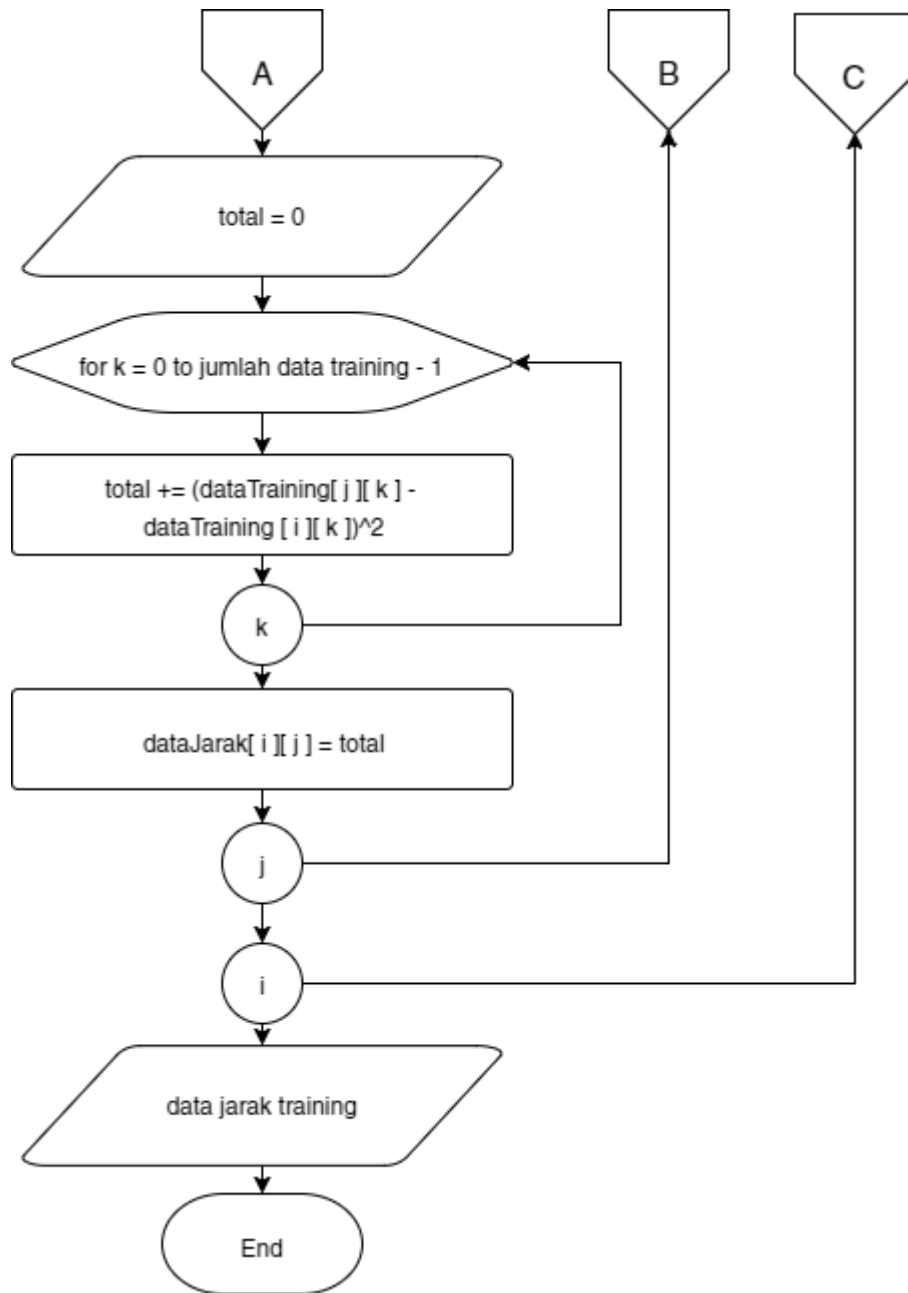
Langkah-langkah dalam proses *training* SVR adalah sebagai berikut.

1. *Input* data *training*, parameter-parameter SVR yang meliputi λ , cLR , C , ε , σ serta batas iterasi maksimum dan inialisasi nilai awal *sequential learning* α_i dan α_i^* dengan nilai 0.
2. Proses perhitungan jarak antar data *training* untuk digunakan pada proses perhitungan matriks *Hessian*.
3. Proses perhitungan matriks *Hessian training* dengan fungsi kernel RBF.
4. Proses perhitungan nilai γ dengan membagi cLR dengan nilai maksimum dari matriks *Hessian training*.
5. Proses perhitungan *sequential learning* untuk menghasilkan nilai *Lagrange Multiplier* yang kemudian akan digunakan pada proses pengujian SVR.
6. Proses perhitungan fungsi regresi $f(x)$ sebagai evaluasi dari nilai *Lagrange Multiplier* yang dihasilkan.
7. Proses perhitungan MAPE untuk mengetahui besarnya akurasi hasil prediksi pada tahap *training* yang diperoleh.
8. *Output* berupa nilai *sequential learning* α^* dan α yang akan digunakan pada tahap *testing* SVR serta nilai MAPE yang dihasilkan. Apabila MAPE yang dihasilkan cukup baik, maka nilai α^* dan α telah sesuai dengan data dan bisa digunakan untuk tahap *testing* SVR.

4.2.2.1 Proses Perhitungan Jarak Data Training

Proses ini dilakukan untuk menghitung jarak data *training* dilakukan untuk mengetahui posisi atau jarak antar data *training*. Diagram alir proses perhitungan data *training* ditunjukkan pada Gambar 4.4.





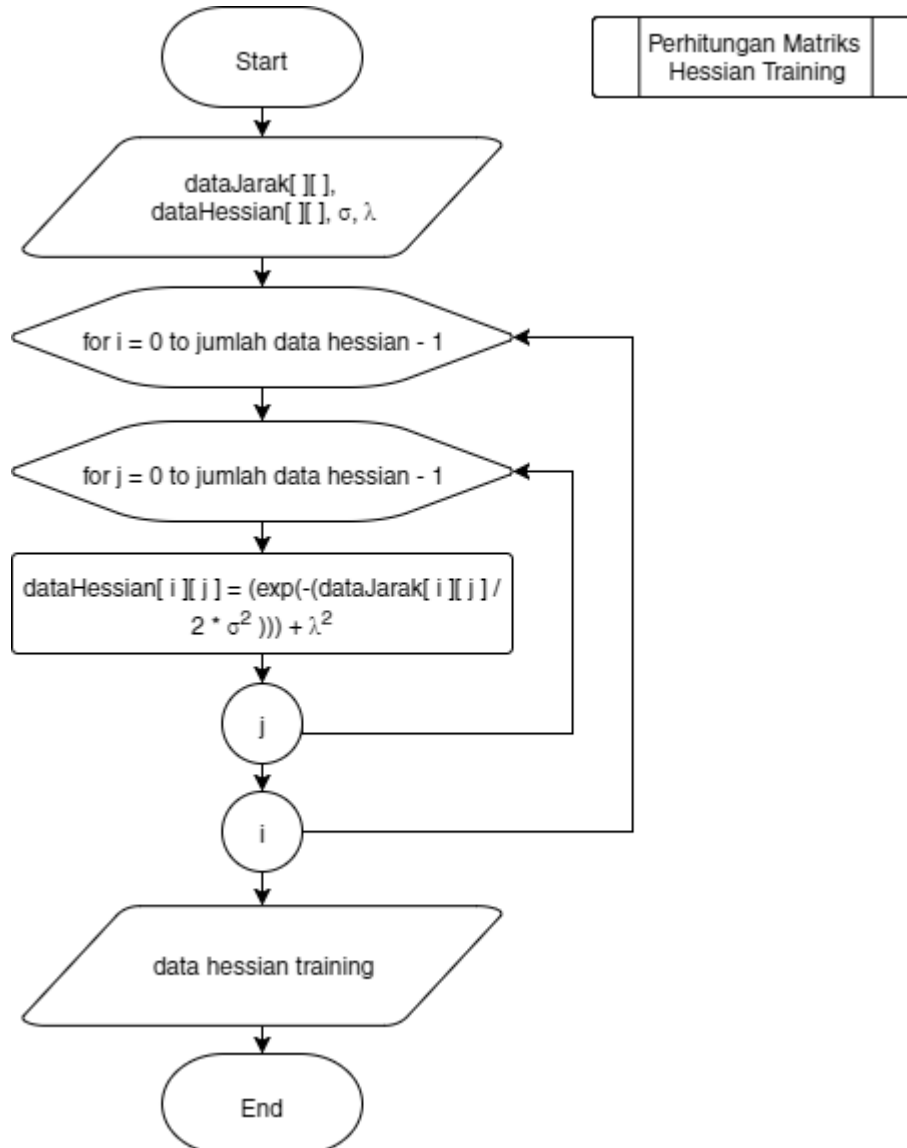
Gambar 4.4 Diagram Alir Proses Perhitungan Jarak Data *Training*

Langkah-langkah dalam proses perhitungan jarak data *training* adalah:

1. Sistem memproses masukan data *training* dengan menghitung selisih antar data *training* kemudian dikuadratkan.
2. *Output* yang dihasilkan adalah data jarak data *training* yang selanjutnya digunakan untuk menghitung matriks *Hessian training*.

4.2.2.2 Proses Perhitungan Matriks *Hessian Training*

Proses ini dilakukan untuk memetakan data menggunakan kernel sehingga kemampuan pembelajaran *linear* dapat ditingkatkan. Didalam perhitungan nilai kernel diperlukan perhitungan jarak antar data yang berguna untuk mengetahui hubungan kedekatan antar data. Gambar 4.5 menunjukkan diagram alir proses hitung matriks *Hessian training*.



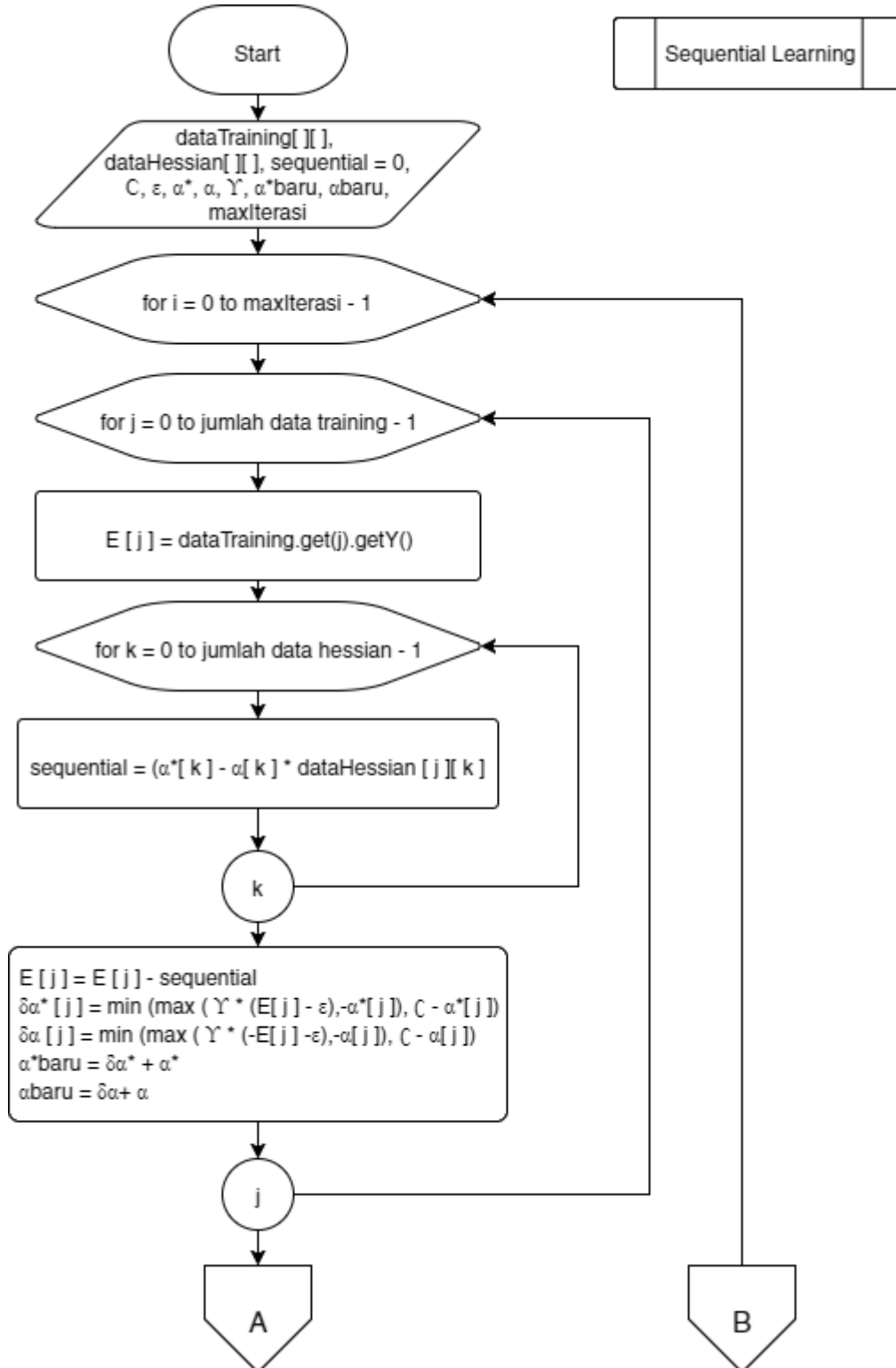
Gambar 4.5 Diagram Alir Perhitungan Matriks *Hessian Training*

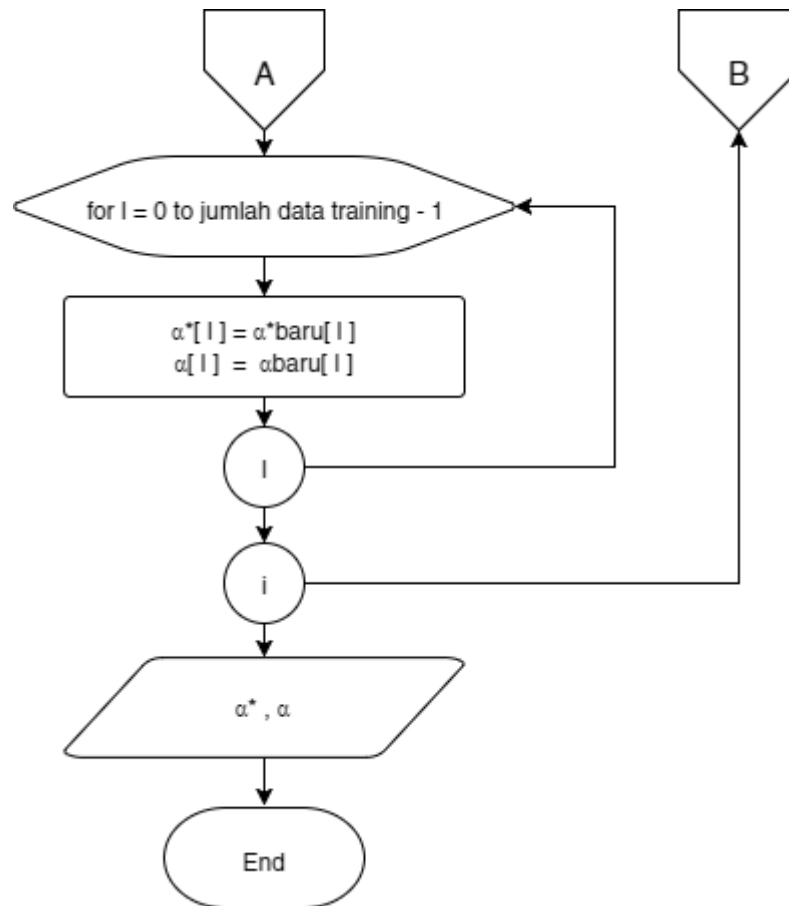
Langkah-langkah dalam proses perhitungan matriks *Hessian training* adalah sebagai berikut.

1. *Input* data berupa nilai parameter σ dan λ serta data jarak data *training*.
2. Menghitung matriks *Hessian training* dengan kernel RBF dan hasil perhitungan jarak antar data *training*.
3. *Output* yang dihasilkan berupa nilai matriks *Hessian*.

4.2.2.3 Proses *Sequential Learning*

Proses *sequential learning* terdiri dari 3 tahapan, yaitu menghitung nilai *error* pada setiap *training point*, menghitung nilai *Lagrange Multiplier* ($\delta\alpha_i^*$, $\delta\alpha_i$) dan menghitung nilai *sequential learning* (α_i^* , α_i). Proses *sequential learning* ditunjukkan oleh diagram alir pada Gambar 4.6.





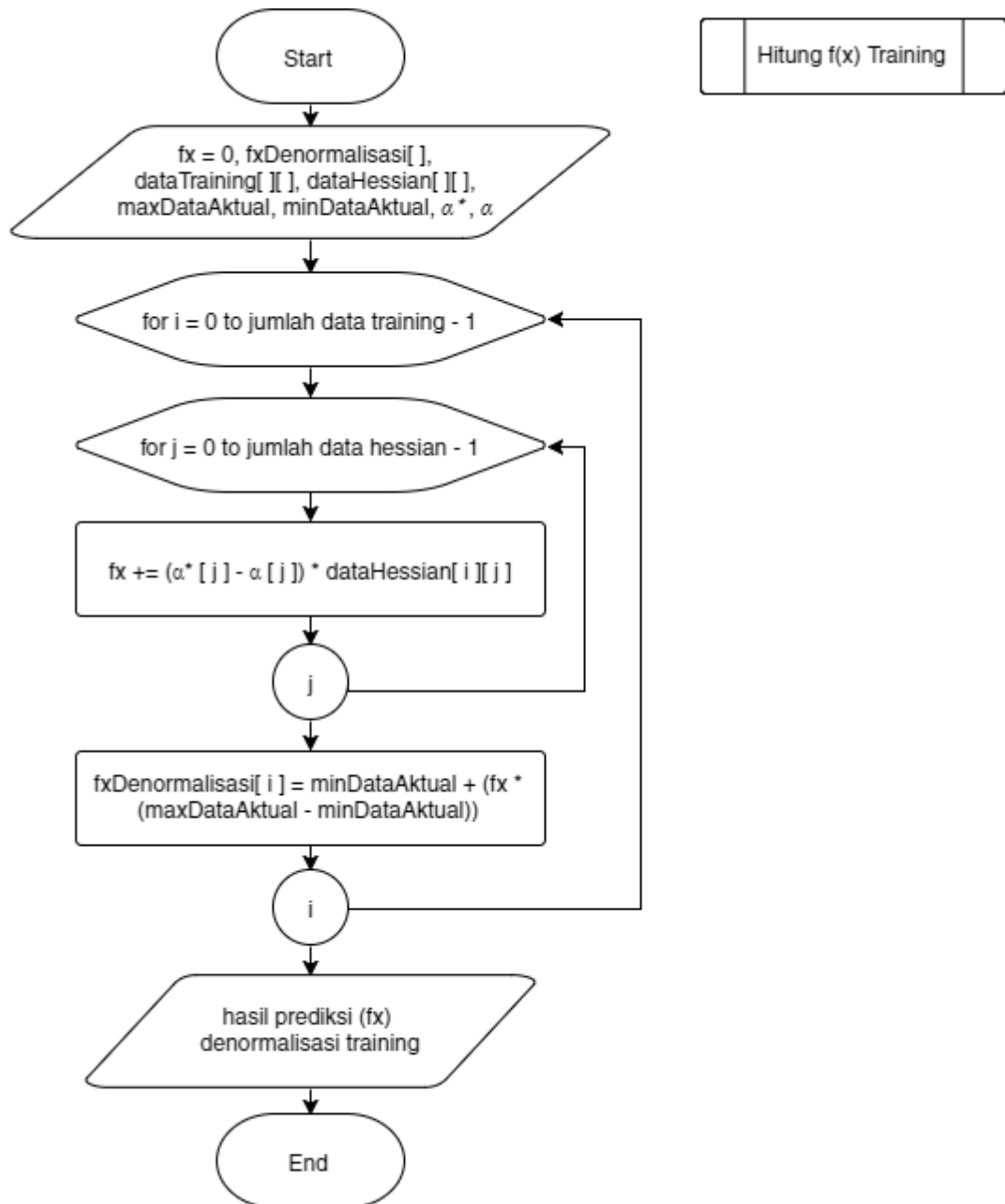
Gambar 4.6 Diagram Alir Proses *Sequential learning*

Langkah-langkah proses *sequential learning* adalah sebagai berikut.

1. *Input* berupa matriks *Hessian*, nilai α^* , nilai α , nilai γ , nilai ϵ , nilai C , jumlah data dan iterasi maksimal.
2. Menghitung nilai *error*.
3. Menghitung nilai $\delta\alpha^*$ dan $\delta\alpha$.
4. Melakukan pembaruan nilai *Lagrange Multiplier* yaitu nilai α^* dan nilai α .
5. Mengulangi proses perhitungan sampai mencapai batas iterasi maksimal yang telah ditentukan atau hingga mencapai nilai konvergen.
6. *Output* yang dihasilkan adalah nilai *sequential learning* yaitu nilai α^* dan α yang baru setelah mencapai iterasi maksimal atau syarat konvergensi.

4.2.2.4 Proses Perhitung $f(x)$ Training

Proses hitung $f(x)$ training merupakan proses untuk mendapatkan nilai regresi atau hasil prediksi pada tahap pelatihan. Pada *training SVR* proses hitung $f(x)$ digunakan sebagai eksplorasi nilai *Lagrange Multiplier* yang didapatkan. pada proses ini juga dilakukan proses denormalisasi terhadap data hasil prediksi. Diagram alir proses perhitungan $f(x)$ training ditunjukkan pada Gambar 4.7.



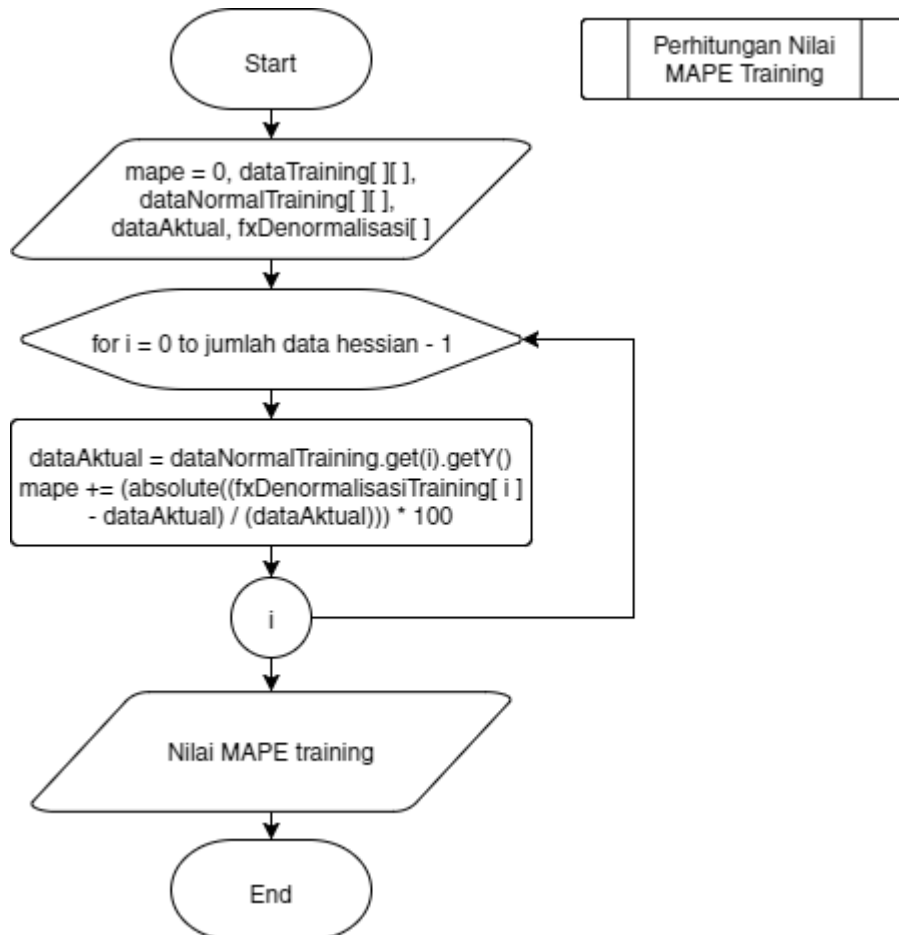
Gambar 4.7 Diagram Alir Proses Perhitungan $f(x)$ Training

Langkah-langkah dalam menghitung nilai fungsi regresi $f(x)$ training adalah:

1. *Input* berupa α^* dan α hasil *sequential learning*, *Hessian training*, nilai maksimal dan nilai minimal dari data aktual dan inialisasi $f(x) = 0$.
2. Melakukan perhitungan $f(x)$ atau hasil prediksi.
3. Proses denormalisasi terhadap nilai $f(x)$ yang dihasilkan dengan menggunakan Persamaan 2.21.
4. *Output* yang didapatkan adalah nilai $f(x)$ atau hasil prediksi dari tahap pelatihan.

4.2.2.5 Proses Perhitungan MAPE Training

Proses perhitungan nilai MAPE pada tahap *training* atau pelatihan digunakan sebagai evaluasi dari nilai *sequential learning* yang dihasilkan pada tahap pelatihan, apakah sudah cukup baik untuk digunakan pada tahap pengujian atau *testing*. Diagram alir hitung MAPE Training ditunjukkan pada Gambar 4.8.



Gambar 4.8 Diagram Alir Proses Perhitungan MAPE Training

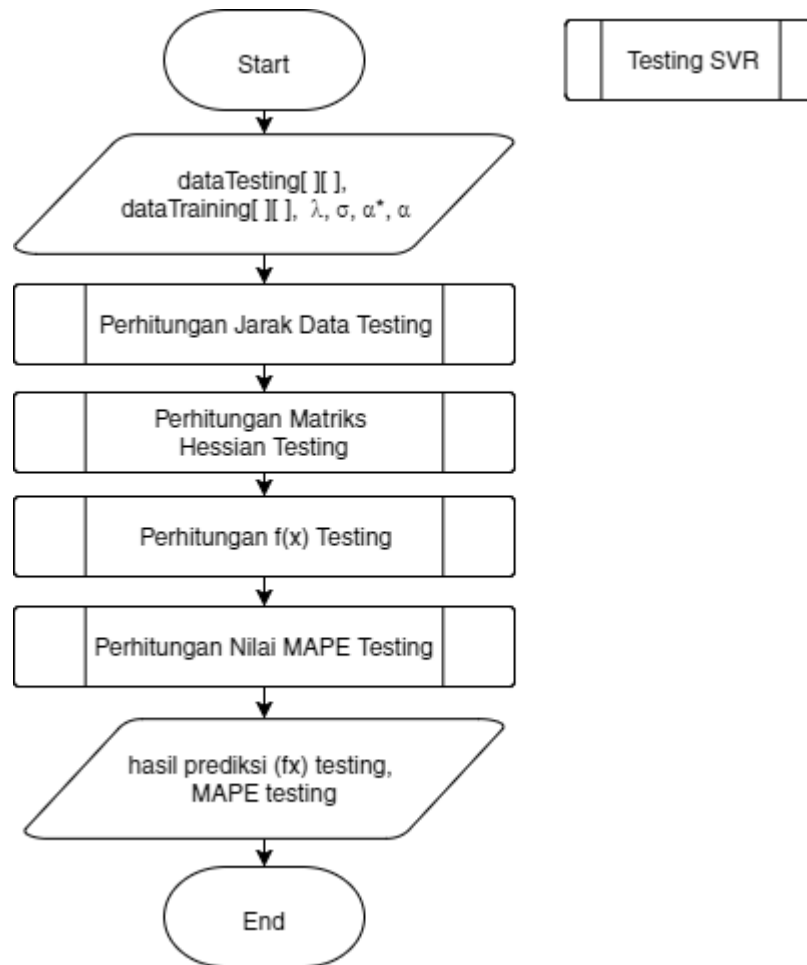
Langkah-langkah perhitungan nilai MAPE adalah sebagai berikut.

1. *Input* proses berupa data *training*, data aktual, hasil *fx training*, dan inialisasi nilai variabel $mape = 0$.
2. Melakukan perhitungan nilai MAPE sesuai dengan Persamaan 2.23 dengan menghitung nilai *absolute* dari selisih aktual dengan hasil prediksi dibagi aktual kemudian dikalikan dengan nilai 100 untuk menghasilkan persentase nilai *error rate*.
3. *Output* yang dihasilkan yaitu nilai MAPE.

4.2.3 Proses Testing SVR

Proses *testing* atau pengujian SVR dilakukan untuk menghasilkan nilai regresi pada data *training* dan data *testing*. Proses pengujian SVR diperoleh dengan menghitung jarak antar data *testing*, menghitung matriks *Hessian testing*

kemudian menghitung nilai $f(x)$ untuk mendapatkan hasil prediksi dan MAPE untuk mendapatkan evaluasi hasil prediksi. Diagram alir proses pengujian SVR ditunjukkan pada Gambar 4.9.



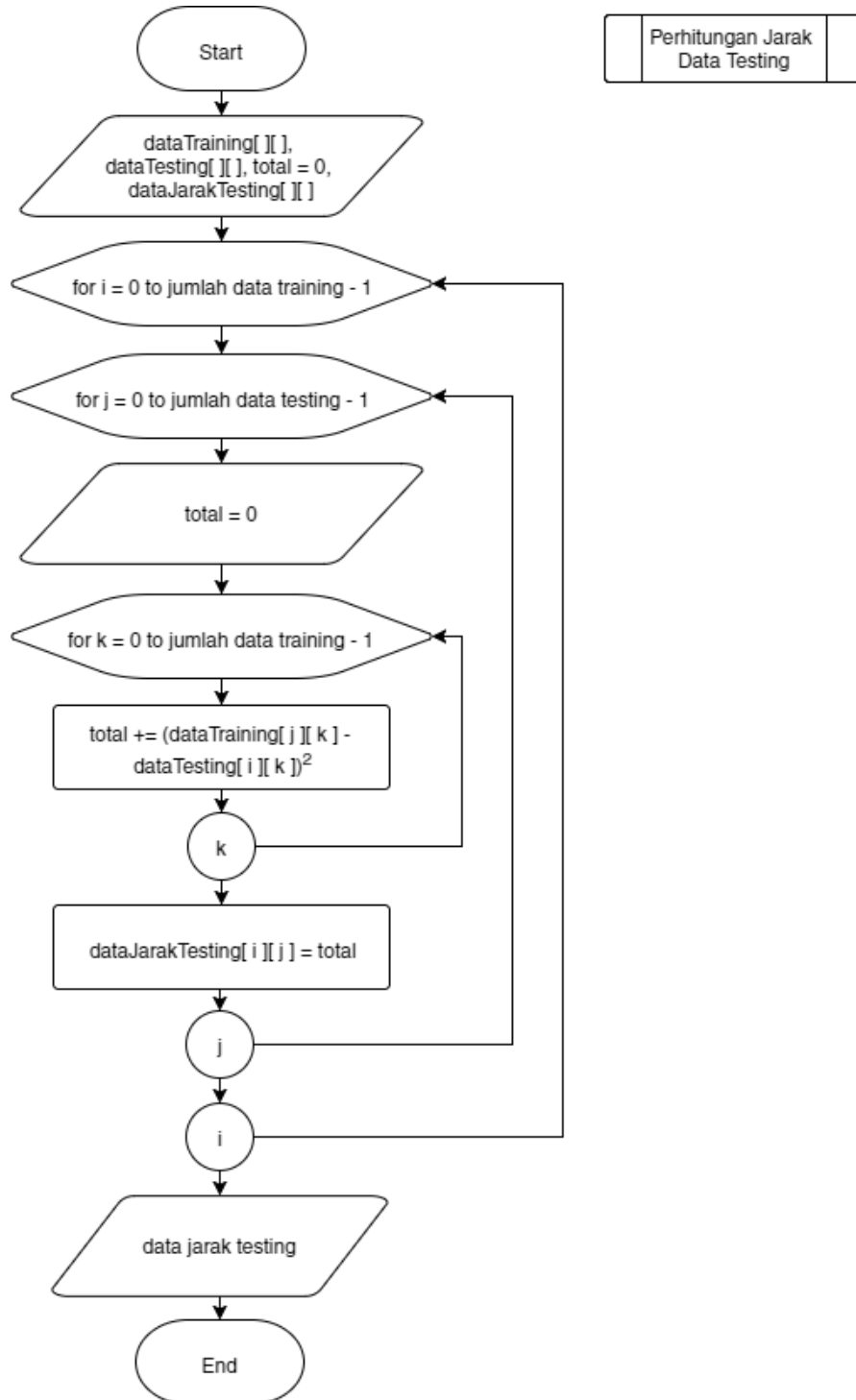
Gambar 4.9 Diagram Alir Proses Testing SVR

Langkah-langkah proses pengujian SVR adalah sebagai berikut.

1. *Input* berupa data *testing* dan parameter SVR yaitu σ dan λ , nilai *sequential learning* (α^* dan α) dari hasil *training* serta data *training* untuk hitung jarak.
2. Proses perhitungan jarak data *testing* dengan data *training* untuk mengetahui posisi atau jarak kedua data tersebut.
3. Menghitung matriks *Hessian testing* dengan melakukan perhitungan kernel RBF untuk mendapatkan nilai matriks *Hessian*.
4. Proses perhitungan fungsi regresi dengan cara menghitung nilai $f(x)$ pada data *testing*.
5. Proses perhitungan nilai *error rate* menggunakan metode MAPE.
6. *Output* yang dihasilkan berupa hasil prediksi dan MAPE *testing*.

4.2.3.1 Proses Perhitungan Jarak Data *Testing*

Perhitungan jarak data *testing* dilakukan dengan menghitung selisih antara data *testing* dengan data *training* untuk diketahui posisi antar kedua data tersebut yang kemudian digunakan untuk menghitung matriks *Hessian testing*. Diagram alir proses perhitungan jarak data *testing* ditunjukkan pada Gambar 4.10.



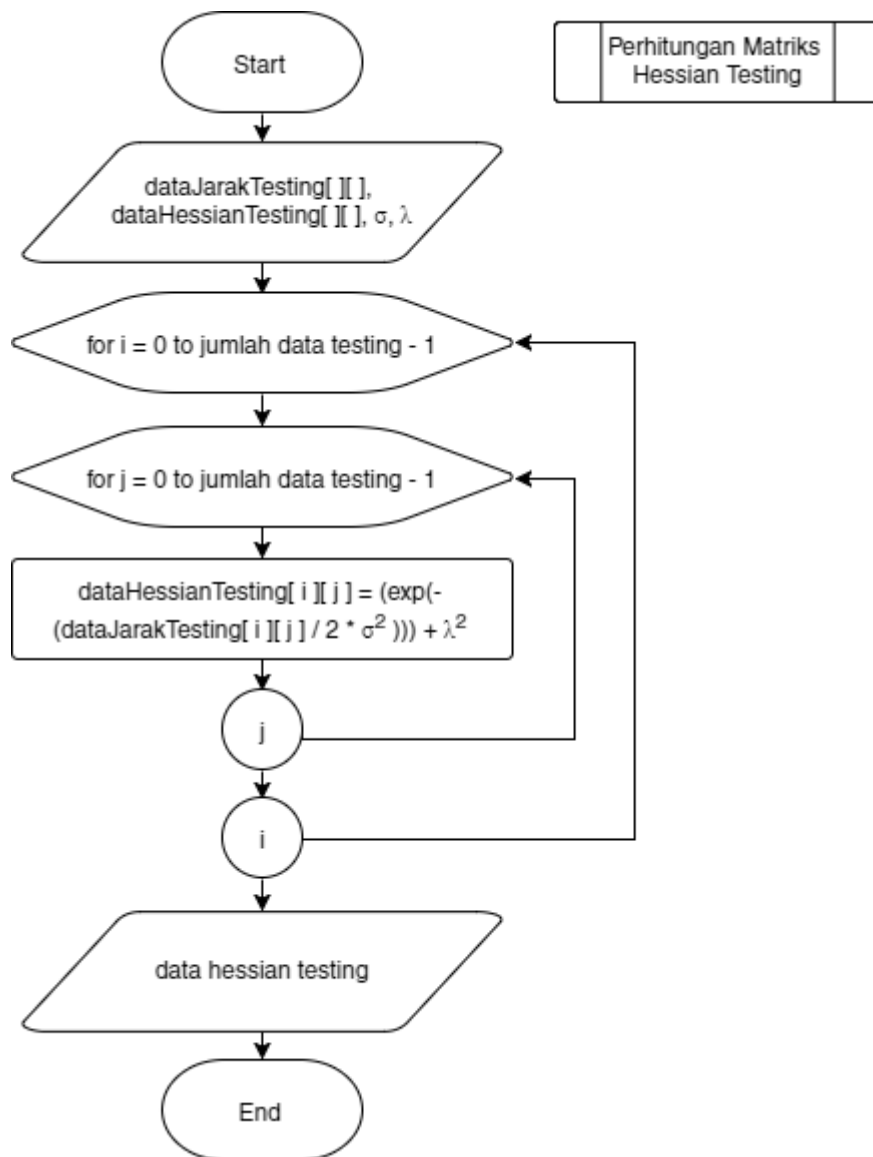
Gambar 4.10 Diagram Alir Proses Perhitungan Jarak Data *Testing*

Langkah-langkah dalam proses perhitungan jarak data *testing* adalah:

1. Sistem memproses masukan data *training* dan data *testing* dengan menghitung selisih antara data *training* dengan data *testing* kemudian dikuadratkan.
2. *Output* yang dihasilkan adalah data jarak data *testing* yang selanjutnya digunakan untuk menghitung matriks *Hessian testing*.

4.2.3.2 Proses Perhitungan Matriks *Hessian Testing*

Menghitung matriks *Hessian* juga diperlukan pada tahap *testing* atau pengujian SVR dengan cara yang sama seperti pada tahap pelatihan SVR, yaitu dengan menghitung jarak antar data dan kernel terlebih dahulu. Diagram alir matriks *Hessian testing* ditunjukkan pada Gambar 4.11.



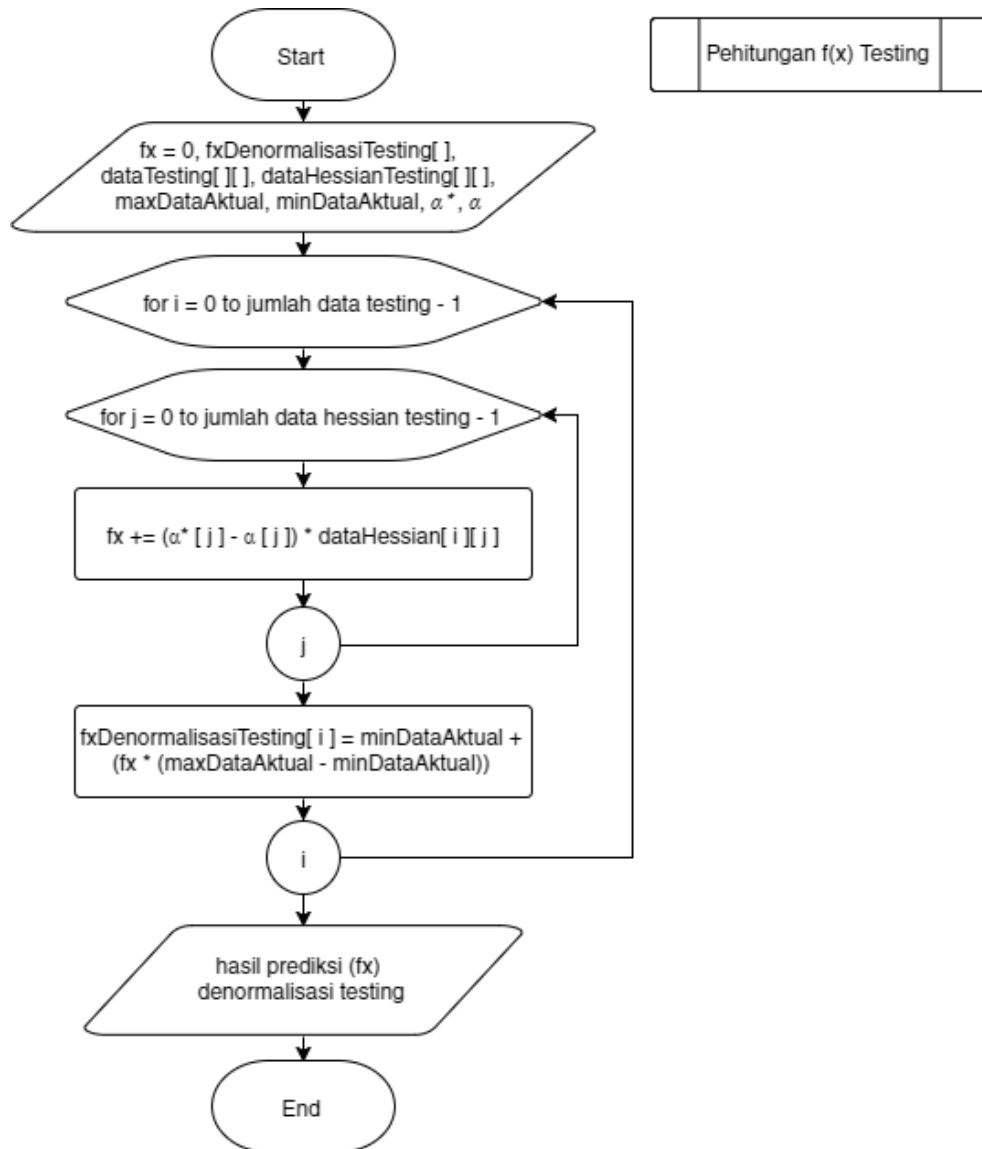
Gambar 4.11 Diagram Alir Proses Perhitungan Matriks *Hessian Testing*

Langkah-langkah dalam proses perhitungan matriks *Hessian* adalah sebagai berikut.

1. *Input* data berupa data *training* dan data *testing*, nilai parameter σ dan λ .
2. Menghitung matriks *Hessian testing* berdasarkan perhitungan kernel dengan menggunakan hasil perhitungan jarak data *testing*.
3. *Output* yang dihasilkan berupa nilai matriks *Hessian* uji.

4.2.3.3 Proses Perhitungan $f(x)$ Testing

Proses hitung $f(x)$ testing merupakan proses untuk mendapatkan nilai regresi atau hasil prediksi pada tahap pengujian. Pada proses ini juga dilakukan proses denormalisasi data untuk mengembalikan nilai normalisasi menjadi nilai aktual. Diagram alir proses perhitungan $f(x)$ testing ditunjukkan pada Gambar 4.12.



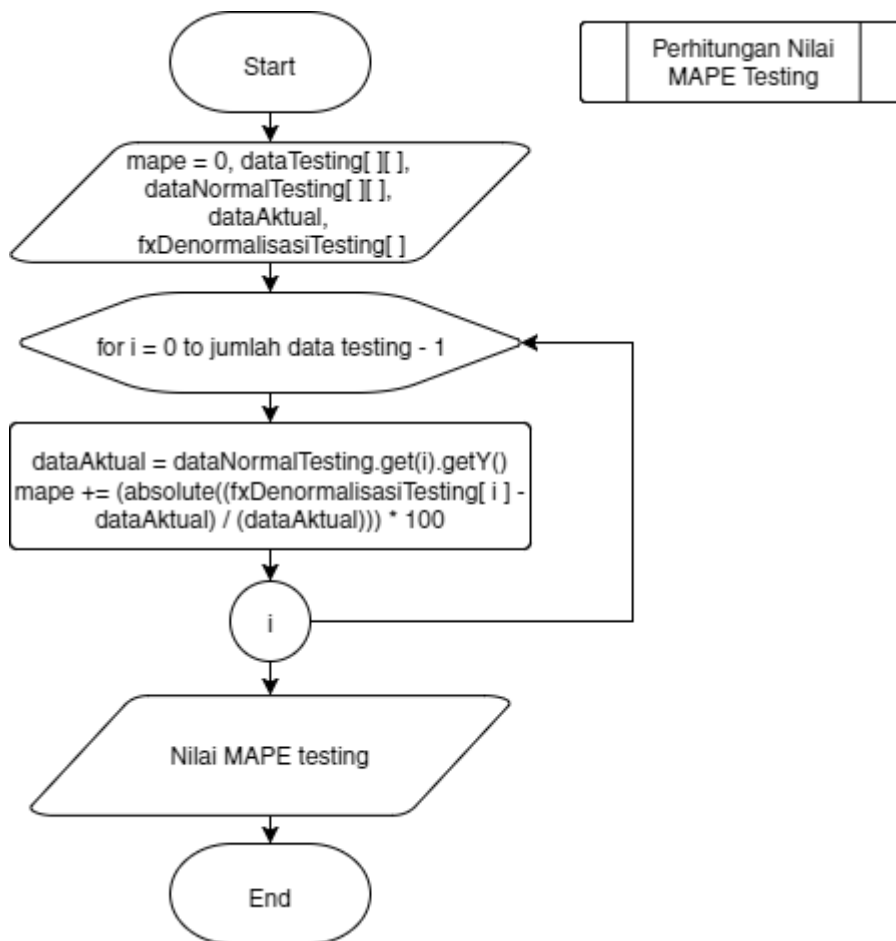
Gambar 4.12 Diagram Alir Proses Perhitungan $f(x)$ Testing

Langkah-langkah dalam menghitung nilai fungsi regresi $f(x)$ testing adalah:

1. *Input* berupa α^* dan α dari tahap *training*, *Hessian testing*, nilai maksimal dan nilai minimal dari data aktual dan inisialisasi variabel $fx = 0$.
2. Melakukan perhitungan hasil prediksi.
3. Melakukan proses denormalisasi terhadap data hasil prediksi
4. *Output* yang didapatkan adalah nilai $f(x)$ atau hasil prediksi.

4.2.3.4 Proses Perhitungan MAPE Testing

Proses perhitungan nilai MAPE digunakan untuk evaluasi tingkat *error* antara hasil prediksi dan data aktual. Diagram alir perhitungan MAPE untuk proses *testing* dijelaskan pada Gambar 4.13.



Gambar 4.13 Diagram Alir Proses Perhitungan MAPE Testing

4.2.4 Perhitungan Manualisasi SVR

Perhitungan manualisasi bertujuan untuk mengetahui tahapan implementasi metode SVR terhadap prediksi indeks harga konsumen kelompok perumahan, air, listrik, gas dan bahan bakar secara manual. Data yang digunakan pada tahap manualisasi adalah data indeks harga konsumen (IHK) kelompok perumahan, air, listrik, gas dan bahan bakar periode Januari 2011 sampai Desember 2011.

Tabel 4.1 Data IHK Kelompok Perumahan, Air, Listrik, Gas dan Bahan Bakar

Bulan/Tahun	Indeks Harga
Januari 2011	120,37
Februari 2011	120,85
Maret 2011	121,2
April 2011	121,46
Mei 2011	121,76
Juni 2011	122,13
Juli 2011	122,36
Agustus 2011	122,76
September 2011	123,08
Oktober 2011	123,33
November 2011	123,6
Desember 2011	123,95

Data pada Tabel 4.1 kemudian dinormalisasi dengan *Min-Max* lalu dibagi kedalam dua bagian, yaitu data *training* dan data *testing*. Data *training* untuk perhitungan manualisasi diambil data bulan Mei 2011 hingga bulan Agustus 2011 sedangkan untuk data *testing* menggunakan data bulan September 2011 hingga Desember 2011. Selanjutnya dilakukan penentuan jumlah fitur yang digunakan untuk tahap manualisasi, yaitu 4 data sekuensial dari 4 bulan sebelumnya untuk memprediksi bulan berikutnya. x_1 , x_2 , x_3 dan x_4 merupakan fitur data dari 4 bulan sebelumnya sedangkan Y merupakan nilai data aktual pada bulan yang akan diprediksi. Penggunaan 4 data sekuensial dilakukan dengan berdasarkan penelitian yang dilakukan oleh Li, et al., (2014). Hasil normalisasi data ditampilkan pada Tabel 4.2 sedangkan pembagian data *training* dan data *testing* dengan fitur berupa data dari 4 bulan sebelumnya dijelaskan pada Tabel 4.3 dan Tabel 4.4.

Tabel 4.2 Hasil Normalisasi Data

Bulan/Tahun	Indeks Harga
Januari 2011	0,425959781
Februari 2011	0,443510055
Maret 2011	0,45630713
April 2011	0,465813528
Mei 2011	0,47678245
Juni 2011	0,490310786
Juli 2011	0,498720293
Agustus 2011	0,513345521
September 2011	0,525045704
Oktober 2011	0,534186472
November 2011	0,544058501
Desember 2011	0,556855576

Tabel 4.3 Data *Training* dengan Fitur Data 4 Bulan Sebelumnya

<i>Data training</i>					
Bulan/Tahun	x1	x2	x3	x4	Y
Mei 2011	0,42595	0,44351	0,45630	0,46581	0,47678
Juni 2011	0,44351	0,45630	0,46581	0,47678	0,49031
Juli 2011	0,45630	0,46581	0,47678	0,49031	0,49872
Agustus 2011	0,46581	0,47678	0,49031	0,49872	0,51334

Tabel 4.4 Data *Testing* dengan Fitur Data 4 Bulan Sebelumnya

<i>Data testing</i>					
Bulan/Tahun	x1	x2	x3	x4	Y
September 2011	0,47678	0,49031	0,49872	0,51334	0,52504
Oktober 2011	0,49031	0,49872	0,51334	0,52504	0,53418
November 2011	0,49872	0,51334	0,52504	0,53418	0,54405
Desember 2011	0,51334	0,52504	0,53418	0,54405	0,55685

Setelah melakukan pemilihan fitur, selanjutnya dilakukan proses perhitungan metode SVR. Perhitungan manualisasi metode SVR adalah sebagai berikut.

1. Inisialisasi parameter metode SVR yang digunakan

Parameter metode SVR diantaranya adalah λ (lambda), ϵ (epsilon), C (*complexity*), cLR (*constant learning rate*) dan σ (sigma). Pemilihan parameter pada penelitian ini didasarkan pada percobaan rentang nilai parameter yang telah dicari sebelumnya dan juga mengacu pada penelitian-penelitian sebelumnya. Tabel 4.5 menjelaskan batasan parameter yang telah dilakukan pada penelitian-penelitian sebelumnya.

Tabel 4.5 Inisialisasi Parameter pada Penelitian Sebelumnya

Parameter	Penelitian Sebelumnya	Batas Bawah	Batas Atas
σ	Li, et al., (2005)	8	100
	Chen, et al., (2013)	0,01	10
λ	Vijayakumar & Wu, (1999)	0	5
ϵ	Li, et al., (2005)	0,008	0,01
	Li, et al., (2014)	0,0	2
	Chen, et al., (2013)	0,005	0,05
cLR	Vijayakumar & Wu, (1999)	0,0001	0,6
C	Li, et al., (2005)	0,8	1
	Chen, et al., (2013)	0,1	100

Pada penelitian ini, inisialisasi nilai parameter yang digunakan pada perhitungan manualisasi ditunjukkan pada Tabel 4.6.

Tabel 4.6 Inisialisasi Parameter SVR

λ	C	ϵ	σ	cLR
0,027	100	0,0002	10	0,24

2. Training SVR

Proses *training* SVR dilakukan untuk menghitung data *training* sehingga dapat menghasilkan nilai regresi. Pelatihan SVR terdiri dari beberapa tahapan, diantaranya adalah perhitungan matriks *Hessian training* dan *sequential learning*.

a. Perhitungan Jarak Data *Training*

Perhitungan matriks *Hessian* dimulai dengan menghitung kernel. Penelitian ini menggunakan fungsi kernel RBF sehingga untuk mendapatkan nilai fungsi kernel terlebih dahulu dilakukan perhitungan jarak antar data *training*. Berikut contoh perhitungan jarak antar data *training* pada fitur ke-1 dan ke-2.

$$\begin{aligned} \|x_1 - x_2\|^2 &= (0,42596 - 0,44351)^2 + (0,44351 - 0,45631)^2 \\ &\quad + (0,45630 - 0,46581)^2 + (0,46581 - 0,47678)^2 \\ &= 0,000682466 \end{aligned}$$

Perhitungan seluruh jarak antar data akan menghasilkan matriks 4x4 yang hasilnya ditunjukkan pada Tabel 4.7.

Tabel 4.7 Perhitungan Jarak Antar Data *training*

Distance (D_{ij})				
D_{ij}	1	2	3	4
1	0	0,000682466	0,002437761	0,004934477
2	0,000682466	0	0,00055747	0,001998068
3	0,002437761	0,00055747	0	0,000464425
4	0,004934477	0,001998068	0,000464425	0

b. Perhitungan Matriks *Hessian Training*

Tahap selanjutnya adalah menghitung matriks *Hessian* menggunakan kernel RBF. Tabel 4.8 adalah hasil matriks *Hessian* yang didapatkan setelah melakukan perhitungan kernel RBF. Contoh perhitungan matriks *Hessian* menggunakan fungsi kernel RBF pada data *training* baris ke-1 dan kolom ke-2 adalah sebagai berikut.

$$R_{ij} = \exp\left(-\frac{0,000682466}{2(10)^2}\right) + 0,027^2 = 1,000725588$$

Tabel 4.8 Matriks *Hessian* Data *training*

Matriks <i>Hessian</i> (R_{ij})				
R_{ij}	1	2	3	4
1	1,000729	1,000725588	1,000716811	1,000704328
2	1,000725588	1,000729	1,000726213	1,00071901
3	1,000716811	1,000726213	1,000729	1,000726678
4	1,000704328	1,00071901	1,000726678	1,000729

Setelah memperoleh nilai matriks *Hessian* selanjutnya dilakukan perhitungan untuk mendapatkan nilai γ (gamma). Berikut adalah contoh perhitungan nilai *gamma*.

$$\gamma = \frac{0,24}{1,000729} = 0,239825167$$

c. Perhitungan *Sequential learning*

Proses perhitungan *sequential learning* terdiri dari 3 tahapan, yaitu menghitung nilai *error*, menghitung perubahan nilai *Lagrange Multiplier* yaitu, nilai δ^* dan δ , dan melakukan *update* nilai *Lagrange Multiplier* yang baru, yaitu α_i^* dan α_i . Jumlah iterasi maksimum yang digunakan pada tahap ini sebanyak 10

iterasi dan inialisasi nilai α_i^* dan α_i sebesar 0. Inialisasi nilai α_i^* dan α_i ditampilkan pada Tabel 4.9 sejumlah banyaknya data *training*.

Tabel 4.9 Nilai Awal α_i^* dan α_i

α_i^*	α_i
0	0
0	0
0	0
0	0

Proses selanjutnya adalah melakukan perhitungan nilai *error*. Berikut ini adalah contoh perhitungan nilai *error* untuk data *training* ke-1 pada iterasi ke-1 dimana nilai α_i^* dan α_i didapatkan dari inialisasi awal yaitu 0.

$$E_i = 0,476782 - ((0 - 0)1,000729 + (0 - 0) 1,000725588 + (0 - 0) 1,000716811 + (0 - 0)1,000704328) = 0,47678245$$

Nilai y_i merupakan nilai aktual y pada data *training* ke- x_i . Selanjutnya perhitungan nilai *Lagrange Multiplier* δa_i^* dan δa_i sesuai dengan Persamaan 2.15 dan Persamaan 2.16.

$$\delta a_i^* = \min\{\max[0,239825167(0,47678245 - 0,0002), - 0], 100 - 0\} = 0,114296466$$

$$\delta a_i = \min\{\max[0,239825167(-0,47678245 - 0,0002), - 0], 100 - 0\} = 0,000000000$$

Setelah mendapatkan nilai δa_i^* dan δa_i , proses selanjutnya adalah melakukan *update* nilai α_i^* dan α_i baru.

$$a_i^*(baru) = 0,114296466 + 0,000000000 = 0,114296466$$

$$a_i (baru) = 0,000000000 + 0,000000000 = 0,000000000$$

Proses *sequential learning* terus dilakukan hingga mencapai iterasi maksimum yang telah ditentukan, dalam kasus ini adalah sebanyak 10 iterasi. Hasil iterasi pertama dan iterasi ke-10 ditunjukkan pada Tabel 4.10 dan Tabel 4.11 berikut.

Tabel 4.10 Hasil Iterasi ke-1

E	δa_i^*	δa_i	a_i^*	a_i
0,47678245	0,114296466	0	0,114296466	0
0,490310786	0,117540901	0	0,117540901	0
0,498720293	0,119557713	0	0,119557713	0
0,513345521	0,123065211	0	0,123065211	0

Tabel 4.11 Hasil Iterasi ke-10

E	$\delta\alpha_i^*$	$\delta\alpha_i$	α_i^*	α_i
-0,01418382	-0,003449602	0,003353672	0,086640359	0,027315836
-0,000655483	-0,000205166	0,000109236	0,119084009	0,001360352
0,007754023	0,001811645	0	0,139252134	0
0,022379251	0,005319143	0	0,174327656	0

Nilai α_i^* dan α_i pada iterasi paling akhir adalah nilai yang akan digunakan untuk perhitungan $f(x)$ pada tahap pengujian.

d. Hitung Hasil Prediksi atau Nilai $f(x)$

Nilai $f(x)$ atau hasil prediksi dihitung dengan menggunakan nilai *Lagrange Multiplier* (α_i^* dan α_i) dari hasil iterasi paling akhir yang diperoleh dari proses *sequential learning*. Perhitungan nilai $f(x)$ pada pelatihan SVR bertujuan untuk mengecek apakah nilai *Lagrange Multiplier* yang dihasilkan telah sesuai sehingga memberikan hasil prediksi yang baik pada pelatihan SVR.

$$\begin{aligned}
 f(x) &= ((0,086640359 - 0,027315836)1,000729) \\
 &\quad + ((0,119084009 - 0,001360352)1,000725588) \\
 &\quad + ((0,139252134 - 0,000000000)1,000716811) \\
 &\quad + ((0,174327656 - 0,000000000)1,000704328) \\
 &= 0,490979238
 \end{aligned}$$

Nilai $f(x)$ tersebut kemudian dinormalisasi sehingga kembali didapatkan nilai aktual.

$$x = 108,72 + (0,490979238(136,07 - 108,72)) = 122,1482821$$

Hasil $f(x)$ yang diperoleh secara lengkap ditunjukkan pada Tabel 4.12. Dengan nilai Y adalah nilai data *training* aktual dan $f(x)$ adalah data hasil prediksi.

Tabel 4.12 Nilai $f(x)$ atau Hasil Prediksi Data *training*

Data <i>training</i> ke- i	$f(x)$ (Normalisasi)	Y (Normalisasi)	$f(x)$ (Denormalisasi)	Y (Aktual)
1	0,490979238	0,47678245	122,1482821	121,76
2	0,490983305	0,490310786	122,1483934	122,13
3	0,490984182	0,498720293	122,1484174	122,36
4	0,490982675	0,513345521	122,1483761	122,76

e. Hitung MAPE

Tahapan terakhir adalah menghitung nilai evaluasi menggunakan metode *Mean Absolute Percentage Error* (MAPE). Berikut adalah contoh perhitungan MAPE data *training*.

$$\begin{aligned}
 MAPE = \frac{1}{4} & \left(ABS \left(\frac{(122,1482821 - 121,76)}{121,76} \times 100\% \right) \right. \\
 & + ABS \left(\frac{(122,1483934 - 122,13)}{122,13} \times 100\% \right) \\
 & + ABS \left(\frac{(122,1484174 - 122,36)}{122,36} \times 100\% \right) \\
 & \left. + ABS \left(\frac{(122,1483761 - 122,76)}{122,76} \times 100\% \right) \right) = 0,251274337\%
 \end{aligned}$$

3. Testing SVR

Pengujian SVR merupakan perhitungan nilai regresi untuk data *testing* berdasarkan perhitungan *sequential* data *training*. Tahapan *testing* atau pengujian SVR terdiri dari perhitungan jarak antara data *training* dan data *testing*, perhitungan matriks *Hessian* dan perhitungan nilai $f(x)$ dan perhitungan MAPE. Perhitungan *sequential learning* tidak dilakukan pada tahap *testing* SVR.

a. Perhitungan Matriks *Hessian Testing*

Pada pengujian, perhitungan jarak antar data yang digunakan adalah jarak antar data *training* dengan data *testing*. Sehingga hasil perhitungannya akan berbentuk sebuah matriks berukuran 4x4 yang ditunjukkan pada Tabel 4.13. Cara perhitungan jarak antar data pada proses *testing* sama dengan proses perhitungan jarak antar data pada proses *training*. Berikut adalah contoh perhitungan jarak antar data *training* pada baris ke-1 dengan data *testing* baris ke-1.

$$\begin{aligned}
 \|x_1 - x_2\|^2 &= (0,42595 - 0,47678)^2 + (0,44351 - 0,49031)^2 \\
 &+ (0,45630 - 0,49872)^2 + (0,46581 - 0,51334)^2 \\
 &= 0,008831419
 \end{aligned}$$

Tabel 4.13 Perhitungan Jarak Antar Data *Training* dan Data *Testing*

Distance (Dij)				
Dij	1	2	3	4
1	0,008831419	0,013951051	0,019570935	0,026471797
2	0,004683014	0,008577817	0,013105221	0,01880291
3	0,002031222	0,004782476	0,008312584	0,012945867
4	0,00058795	0,002305011	0,004884078	0,008569261

b. Perhitungan Matriks *Hessian Testing*

Perhitungan matriks *Hessian testing* dilakukan dengan cara yang sama seperti pada tahap *training*. Sebagai contoh hasil perhitungan matriks *Hessian testing* ditunjukkan pada Tabel 4.14.

Tabel 4.14 Matriks *Hessian Data testing*

Matriks <i>Hessian</i> (Rij)				
Rij	1	2	3	4
1	1,000684844	1,000659247	1,00063115	1,00059665
2	1,000705585	1,000686112	1,000663476	1,00063499
3	1,000718844	1,000705088	1,000687438	1,000664273
4	1,00072606	1,000717475	1,00070458	1,000686155

b. Hitung Hasil Prediksi atau Nilai $f(x)$

Contoh perhitungan nilai $f(x)$ atau hasil prediksi dengan data *testing* adalah sebagai berikut.

$$\begin{aligned}
 f(x) &= ((0,086640359 - 0,027315836)1,000684844) \\
 &\quad + ((0,119084009 - 0,001360352)1,000659247) \\
 &\quad + ((0,139252134 - 0,000000000)1,00063115) \\
 &\quad + ((0,174327656 - 0,000000000)1,00059665) \\
 &= 0,490938108
 \end{aligned}$$

Nilai $f(x)$ tersebut kemudian didenormalisasi sehingga kembali didapatkan nilai aktual.

$$x = 108,72 + (0,490938108(136,07 - 108,72)) = 122,1471573$$

Hasil perhitungan $f(x)$ terhadap data *testing* ditunjukkan pada Tabel 4.15.

Tabel 4.15 Nilai $f(x)$ atau Hasil Prediksi Data *Testing*

Data <i>training</i> ke-i	$f(x)$ (Normalisasi)	Y (Normalisasi)	$f(x)$ Denormalisasi	Y (Aktual)
1	0,490938108	0,525045704	122,1471573	123,08
2	0,490953687	0,534186472	122,1475833	123,33
3	0,490965149	0,544058501	122,1478968	123,6
4	0,490973237	0,556855576	122,148118	123,95

c. Hitung MAPE

Berikut adalah contoh perhitungan MAPE data *testing*.

$$\begin{aligned} MAPE &= \frac{1}{4} \left(ABS \left(\frac{(122,1471573 - 123,08)}{123,08} \times 100\% \right) \right. \\ &\quad + ABS \left(\frac{(122,1475833 - 123,33)}{123,33} \times 100\% \right) \\ &\quad + ABS \left(\frac{(122,1478968 - 123,6)}{123,6} \times 100\% \right) \\ &\quad \left. + ABS \left(\frac{(122,148118 - 123,95)}{123,95} \times 100\% \right) \right) \\ &= 1,086303867\% \end{aligned}$$

Hasil perhitungan nilai MAPE untuk data *training* dan data *testing* secara rinci ditunjukkan pada Tabel 4.16.

Tabel 4.16 Hasil MAPE Data *Training* dan Data *Testing*

Data <i>Training</i>	Data <i>Testing</i>
0,251274337%	1,086303867%

4.3 Perancangan *User Interface*

Perancangan *user interface* adalah proses untuk merancang antarmuka pengguna dengan sistem. Perancangan *user interface* pada penelitian ini terdiri dari 2 panel. Panel pertama berisi *form* untuk *Input* jumlah data *training* dan data *testing* serta nilai parameter-parameter SVR dan jumlah iterasi maksimum. Terdapat juga tombol yang digunakan untuk mengambil parameter yang selanjutnya akan diolah dengan metode SVR. Panel kedua terdiri dari 4 tab menu untuk 4 halaman yang masing-masing halaman berisi tabel yang menampilkan data-data, mulai dari data yang digunakan untuk prediksi, data hasil normalisasi, data hasil prediksi untuk tahap *training* dan data hasil prediksi untuk tahap *testing*. Pada halaman *training* dan halaman *testing* juga ditampilkan hasil perhitungan MAPE dari masing-masing tahapan. Perancangan *user interface* dapat dilihat pada Gambar 4.14.

JUDUL

	1 Data	2 Data Normalisasi	3 Training	4 Testing
<p style="text-align: center;">DATA</p> <p>Jumlah Data Training <input style="width: 50px;" type="text" value="6"/></p> <p>Jumlah Data Testing <input style="width: 50px;" type="text" value="7"/></p> <p style="text-align: center;">PARAMETER SVR</p> <p>Lambda (λ) <input style="width: 50px;" type="text" value="8"/></p> <p>Epsilon (ϵ) <input style="width: 50px;" type="text" value="9"/></p> <p>cLR <input style="width: 50px;" type="text" value="10"/></p> <p>Sigma (σ) <input style="width: 50px;" type="text" value="11"/></p> <p>Kompleksitas (C) <input style="width: 50px;" type="text" value="12"/></p> <p>Jumlah Iterasi <input style="width: 50px;" type="text" value="13"/></p> <p style="text-align: center;"><input style="width: 50px;" type="button" value="14"/></p> <p style="text-align: center;"><input style="width: 50px;" type="button" value="15"/></p>				
MAPE	<input style="width: 100%; height: 20px;" type="text" value="5"/>			

Gambar 4.14 Perancangan User Interface

Keterangan:

1. *Tab* Data untuk menampilkan model data yang digunakan
2. *Tab* Data Normalisasi untuk menampilkan data yang telah dinormalisasi
3. *Tab* yang sedang aktif, yaitu tab *Training*, untuk menampilkan hasil *training*
4. *Tab* *Testing* untuk menampilkan hasil *testing*
5. *TextField* untuk menampilkan hasil perhitungan MAPE
6. *TextField* untuk memasukkan jumlah bulan data *training*
7. *TextField* untuk memasukkan jumlah bulan data *testing*
8. *TextField* untuk memasukkan nilai parameter λ (lambda)
9. *TextField* untuk memasukkan nilai parameter ϵ (epsilon)
10. *TextField* untuk memasukkan nilai parameter *cLR*
11. *TextField* untuk memasukkan nilai parameter σ (sigma)
12. *TextField* untuk memasukkan nilai parameter C (*complexity*)
13. *TextField* untuk memasukkan jumlah iterasi maksimum
14. *Button* untuk melakukan proses prediksi
15. *Button* untuk menghapus masukan yang ada di layar

4.4 Perancangan Pengujian

Perancangan pengujian digunakan untuk menguji hasil prediksi sistem. Pengujian yang dilakukan antara lain:

1. Pengujian rentang nilai parameter SVR
2. Pengujian jumlah iterasi untuk proses *sequential learning*
3. Pengujian variasi jumlah data *training*

4.4.1 Perancangan Pengujian Rentang Nilai Parameter SVR

Perancangan Pengujian rentang nilai parameter SVR dilakukan untuk mengetahui nilai masing-masing parameter SVR agar menghasilkan kombinasi parameter SVR yang optimal sehingga didapatkan hasil prediksi yang baik. Nilai parameter yang diberikan akan mempengaruhi hasil prediksi. Pengujian akan dilakukan sebanyak 3 kali percobaan dengan 6 – 8 variasi batasan parameter yang berbeda. Tabel 4.17 adalah tabel untuk pengujian parameter σ (sigma), Tabel 4.18 untuk parameter λ (lambda), Tabel 4.19 untuk parameter *cLR*, Tabel 4.20 untuk parameter ϵ (epsilon) dan Tabel 4.21 untuk parameter *C* (*complexity*).

Tabel 4.17 Perancangan Uji Coba Rentang Nilai Parameter σ

Batas bawah σ	Batas atas σ	Nilai MAPE percobaan ke- <i>i</i>			Rata-rata nilai MAPE
		1	2	3	
0,01	0,1				
0,1	1				
1	5				
5	10				
10	50				
50	100				

Tabel 4.18 Perancangan Uji Coba Rentang Nilai Parameter λ

Batas bawah λ	Batas atas λ	Nilai MAPE percobaan ke- <i>i</i>			Rata-rata nilai MAPE
		1	2	3	
0,005	0,01				
0,01	0,05				
0,05	0,1				
0,1	0,5				
0,5	1				
1	5				
5	10				
10	50				

Tabel 4.19 Perancangan Uji Coba Rentang Nilai Parameter cLR

Batas bawah cLR	Batas atas cLR	Nilai MAPE percobaan ke- i			Rata-rata nilai MAPE
		1	2	3	
0,0001	0,0005				
0,0005	0,001				
0,001	0,005				
0,005	0,01				
0,01	0,05				

Tabel 4.20 Perancangan Uji Coba Rentang Nilai Parameter ϵ

Batas bawah ϵ	Batas atas ϵ	Nilai MAPE percobaan ke- i			Rata-rata nilai MAPE
		1	2	3	
0,00001	0,00005				
0,00005	0,0001				
0,0001	0,0005				
0,0005	0,001				
0,001	0,005				
0,005	0,01				

Tabel 4.21 Perancangan Uji Coba Rentang Nilai Parameter C

Batas bawah C	Batas atas C	Nilai MAPE percobaan ke- i			Rata-rata nilai MAPE
		1	2	3	
1	5				
5	10				
10	50				
50	100				
100	500				
500	1000				

4.4.2 Perancangan Pengujian Jumlah Iterasi

Uji coba ini dilakukan untuk menentukan jumlah iterasi yang digunakan untuk proses *sequential learning*. Pengujian ini dilakukan sebanyak 3 kali dengan 6 variasi jumlah iterasi yang dilakukan dengan parameter yang sama. Perancangan Pengujian jumlah iterasi ditunjukkan pada Tabel 4.22.

Tabel 4.22 Perancangan Uji Coba Jumlah Iterasi

Jumlah Iterasi	Nilai MAPE percobaan ke- <i>i</i>			Rata-rata nilai MAPE
	1	2	3	
10				
50				
100				
500				
1000				
5000				

4.4.3 Perancangan Pengujian Variasi Jumlah Data *Training*

Uji coba variasi data *training* dilakukan untuk mendapatkan nilai *error rate* terendah yang mampu dihasilkan dari banyaknya variasi data yang digunakan. Pengujian ini dilakukan dengan hasil parameter SVR yang telah ditentukan dan diperoleh berdasarkan hasil pengujian terbaik pada pengujian parameter algoritma SVR yang telah dilakukan sebelumnya. Terdapat 4 variasi pengujian jumlah data *training*. Perancangan Uji Coba variasi jumlah data *training* ditunjukkan pada Tabel 4.23.

Tabel 4.23 Perancangan Uji Coba Variasi Jumlah Data *Training*

Data <i>Training</i> (Bulan)	Data <i>Testing</i> (Bulan)	Nilai MAPE
12	12	
24	12	
36	12	
48	12	