

BAB IV

HASIL DAN PEMBAHASAN

Berdasarkan analisa simulasi sistem yang diperoleh dari simulasi numerik menggunakan MATLAB didapatkan hasil sebagai berikut :

4.1 Analisis Potensi Efisiensi Sistem Berdasarkan Simulasi

Analisis efisiensi pembersihan filter udara dilakukan berdasarkan hasil simulasi MATLAB untuk mengetahui kinerja internal sistem secara teoretis dalam penggunaan energi dan waktu. Berdasarkan hasil simulasi yang ditampilkan oleh program, diperoleh parameter operasional satu siklus pembersihan sebagai berikut:

Tabel 4. 1 Hasil Simulasi Efisiensi Sistem

No	Parameter Operasional	Nilai	Keterangan Teknis
1	Waktu awal kotor terdeteksi	19,9s	Saat ADC/ ΔP melampaui ambang batas
2	Waktu bersih tercapai	27,7s	Sensor kembali di bawah ambang bawah
3	Durasi motor menyala	7.8s	Selisih waktu motor aktif
4	Daya motor	10W	Parameter input simulasi
5	Konsumsi energi sistem	78J	Durasi $\times$ daya motor

Durasi motor menyala merupakan selisih waktu antara saat sistem pertama kali aktif hingga sistem kembali nonaktif. Nilai ini menunjukkan bahwa sistem mampu merespons kondisi kotor secara cepat dan menghentikan operasi pembersihan setelah kondisi filter kembali berada pada batas yang ditentukan.

4.1.1 Perhitungan Efisiensi Waktu

Efisiensi waktu dihitung untuk membandingkan durasi pembersihan sistem otomatis terhadap metode pembersihan manual. Berdasarkan Bab III, efisiensi waktu dirumuskan sebagai:

$$\eta_t = \left(\frac{T_m - T_s}{T_m} \right) \times 100\%$$

$$\eta_t = \left(\frac{900 - 7,8}{900} \right) \times 100\%$$

$$\eta_t = \left(\frac{892,2}{900} \right) \times 100\%$$

$$\eta_t = 99,13\%$$

Keterangan:

η_t = efisiensi waktu (%)

T_m = waktu pembersihan secara manual (menit)

T_s = waktu pembersihan

Tabel 4. 2 Hasil Perhitungan Efisiensi Waktu

Parameter	Nilai
Waktu mulai pembersihan	19,9s
Waktu selesai pembersihan	27,7s
ADC awal pembersihan	7,8s
ADC akhir	99,13%

Nilai efisiensi yang diperoleh dari simulasi menunjukkan bahwa sistem memiliki potensi kinerja yang optimal berdasarkan parameter yang digunakan dalam model.

4.1.2 Perhitungan Efisiensi Energi

Efisiensi energi digunakan untuk mengevaluasi penghematan energi yang diperoleh dari penggunaan sistem pembersih otomatis. Berdasarkan Bab III, efisiensi energi dirumuskan sebagai:

$$E_{motor} = P \times t$$

$$E_{motor} = 10 \times 7,8$$

$$E_{motor} = 78 \text{ Joule}$$

Keterangan :

P = daya Motor

t = waktu motor aktif

Hasil Efisiensi Energi :

$$\eta_e = \left(\frac{E_m - E_s}{E_m} \right) \times 100\%$$

$$\eta_e = \left(\frac{5000 - 78}{5000} \right) \times 100\%$$

$$\eta_e = \left(\frac{4922}{5000} \right) \times 100\%$$

$$\eta_e = 98,44\%$$

Keterangan:

η_e = efisiensi energi (%)

E_m = energi listrik yang digunakan pada metode manual (Joule)

E_s = energi listrik yang digunakan sistem berbasis Arduino Uno

Tabel 4. 3 Hasil Perhitungan Efisiensi Energi

Parameter	Nilai
Daya Motor	10 W
Durasi motor aktif	7,8 s
Energi Konsumsi	78,0 J
Energi Manual	5000 J
Efisiensi Energi	98,44%

Nilai energi metode manual sebesar 5000 J digunakan sebagai parameter acuan teoretis dalam simulasi untuk merepresentasikan konsumsi energi sistem konvensional, sehingga perhitungan efisiensi energi sepenuhnya berbasis pada pemodelan numerik, bukan pengukuran eksperimental.

4.1.3 Perhitungan Efisiensi Tekanan Diferensial

Efisiensi tekanan diferensial digunakan untuk menilai kemampuan sistem dalam

menurunkan hambatan aliran udara akibat penumpukan kotoran pada filter. Berdasarkan Bab III, efisiensi tekanan dirumuskan sebagai:

$$\eta_p = \left(\frac{\Delta P_{bef} - \Delta P_{aft}}{\Delta P_{bef}} \right) \times 100\%$$

$$\eta_p = \left(\frac{116,94 - 37,12}{116,94} \right) \times 100\%$$

$$\eta_p = \left(\frac{79,82}{116,94} \right) \times 100\%$$

$$\eta_p = 68,29\%$$

Keterangan:

η_p = efisiensi pembersihan filter (%)

ΔP_{bef} = tekanan diferensial sebelum permbersihan (Pa)

ΔP_{aft} = tekanan diferensial sesudah pembersihan (Pa)

Tabel 4. 4 Hasil Efisiensi Penurunan Tekanan Diferensial

Parameter	Nilai
Tekanan Awal	116,94 Pa
Tekanan Akhir	37,12 Pa
Efisiensi tekanan	68,29%

Berdasarkan hasil simulasi, diperoleh nilai efisiensi penurunan tekanan diferensial sebesar 68,29%. Nilai ini menunjukkan bahwa proses pembersihan mampu menurunkan tekanan diferensial secara signifikan, sehingga aliran udara dapat kembali mendekati kondisi optimal

4.2 Analisis Potensi Efektivitas Sistem Pembersih Filter Udara Berdasarkan Simulasi

4.2.1 Analisis Efektivitas Aliran Udara (*Airflow Effectiveness*)

Efektivitas aliran udara dianalisis secara tidak langsung melalui perubahan tekanan diferensial filter. Tekanan akhir pasca pembersihan yang berada di bawah nilai target menunjukkan bahwa aliran udara telah kembali mendekati kondisi

optimal. Berdasarkan Bab II, rasio pembersihan dirumuskan sebagai:

a. Tekanan sebelum pembersihan (kondisi kotor):

$$Q_{bef} = 116,94 \text{ Pa}$$

b. Tekanan setelah pembersihan:

$$Q_{aft} = 37,12 \text{ Pa}$$

Debit relatif dihitung sebagai:

$$Q_{bef} = \frac{1}{\sqrt{116,94}} = 0,00855$$

$$Q_{aft} = \frac{1}{\sqrt{37,12}} = 0,02693$$

Sehingga :

$$\eta_{air} = \frac{Q_{aft} - Q_{bef}}{Q_{bef}} \times 100\%$$

$$\eta_{air} = \frac{0.02693 - 0.00855}{0.00855} \times 100\%$$

$$\eta_{air} = \frac{0.01838}{0.00855} \times 100\%$$

$$\eta_{air} = 215\%$$

Keterangan:

η_{air} = afektivitas peningkatan aliran udara (%)

Q_{bef} = debit aliran udara sebelum pembersihan (m^3/min)

Q_{aft} = debit aliran udara setelah pembersihan (m^3/min)

Tabel 4. 5 Hasil Efektivitas Sistem

Indikator	Nilai
Tekanan sebelum pembersihan	116,94 Pa
Tekanan setelah pembersihan	37,12 Pa
Debit relative sebelum pembersihan Q_{bef}	0,00855
Debit relative setelah pembersihan Q_{aft}	0,02693
Efektivitas aliran udara	215%
Status Airflow	Efektif

Berdasarkan hasil simulasi, sistem menunjukkan potensi pencapaian target pembersihan filter dan aliran udara secara teoretis.

4.2.2 Responsivitas Sensor MQ dan Keandalan Sistem

Responsivitas sensor MQ dan keandalan sistem dievaluasi berdasarkan perubahan nilai konsentrasi partikel udara yang terdeteksi sebelum dan sesudah proses pembersihan serta kestabilan siklus kerja motor. Parameter ini digunakan untuk menilai kemampuan sistem dalam merespons kondisi filter kotor secara tepat dan konsisten. Berdasarkan Bab III, rasio pembersihan dirumuskan sebagai:

$$T_{respon} = t_{deteksi} - t_{start}$$

$$T_{respon} = 19,9 - 0$$

$$T_{respon} = 19,9 \text{ s}$$

Keterangan:

T_{respon} = waktu respon sensor (s)

$t_{deteksi}$ = waktu sensor MQ menunjukkan hasil tabil

t_{start} = waktu sensor MQ memberikan pembacaan pertama

Perhitungan keandalan sistem dapat dihitung dengan rumus berikut:

$$R_{sys} = \frac{N_{berhasil}}{N_{total}} \times 100\% = 100\%$$

$$R_{sys} = \frac{1}{1} \times 100\% = 100\%$$

Tabel 4. 6 Hasil Responsivitas Sensor MQ

Parameter	Nilai
Waktu mulai pembacaan sensor t_{start}	0 s
Waktu deteksi kondisi kotor $t_{deteksi}$	20.2 s
Switching berulang	Tidak terjadi
Status keandalan	Stabil

Berdasarkan hasil simulasi, sistem mulai membaca sinyal sensor sejak awal waktu simulasi ($t_{\text{start}} = 0$ s), yang menunjukkan tidak adanya keterlambatan inisialisasi dalam model. Waktu deteksi kondisi filter kotor sebesar 19,9 s merepresentasikan respons dinamis sistem dalam mengenali perubahan parameter berdasarkan ambang yang ditetapkan. Tidak terjadinya switching berulang menunjukkan bahwa logika kendali bekerja stabil tanpa osilasi keputusan. Dengan status keandalan yang stabil, hasil ini mengindikasikan bahwa sistem memiliki potensi kinerja yang konsisten secara teoretis berdasarkan hasil simulasi.

4.2.3 Perhitungan Rasio Pembersihan Filter

Rasio pembersihan digunakan untuk mengevaluasi efektivitas sistem dalam menurunkan konsentrasi kotoran atau partikel pada filter. Berdasarkan Bab III, rasio pembersihan dirumuskan sebagai:

$$R_c = \left(\frac{C_1 - C_2}{C_1} \right) \times 100\%$$

$$R_c = \left(\frac{769,42 - 244,48}{769,42} \right) \times 100\%$$

$$R_c = 68,24\%$$

Keterangan:

R_c = rasio pembersihan partikel udara (%)

C_1 = konsentrasi partikel sebelum pembersihan (ppm atau $\mu\text{g}/\text{m}^3$)

C_2 = konsentrasi partikel setelah pembersihan (ppm atau $\mu\text{g}/\text{m}^3$)

Tabel 4. 7 Hasil Cleaning Ratio

Parameter	Nilai
Konsentrasi Awal	769,42 Ppm
Konsentrasi Akhir	244,48 Ppm
Cleaning Ratio	68,24%

Hasil simulasi menunjukkan bahwa nilai rasio pembersihan yang diperoleh adalah sebesar 68,24%. Nilai ini menandakan bahwa sistem pembersih filter mampu

mengurangi tingkat kontaminan udara secara efektif dalam satu siklus operasi.

4.3 Rekapitulasi Kinerja Sistem

Hasil simulasi menunjukkan bahwa sistem berpotensi memiliki efisiensi dan efektivitas secara teoretis berdasarkan parameter yang digunakan. Kinerja sistem pembersih filter dapat direkapitulasi melalui parameter efisiensi dan efektivitas sistem. Parameter kinerja yang direkap meliputi efisiensi waktu, efisiensi energi, efisiensi tekanan diferensial, serta rasio pembersihan filter, yang seluruhnya diperoleh dari hasil simulasi MATLAB–Simulink.

Untuk memperjelas hasil rekapitulasi kinerja sistem, disajikan tabel hasil analisis efisiensi dan efektivitas. Tabel berikut digunakan sebagai ringkasan kuantitatif memudahkan evaluasi performa sistem secara keseluruhan.

Tabel 4. 8 Rekapitulasi Kinerja Sistem

Parameter	Nilai	Interpretasi Teknik
Efisiensi Waktu	99,13 %	Sistem bekerja jauh lebih cepat dibandingkan nilai metode manual yang digunakan sebagai acuan teoretis dalam simulasi
Efisiensi Energi	98,44 %	Konsumsi energi sangat rendah
Efisiensi Tekanan Diferensial	68,29%	Penurunan tekanan signifikan
Rasio Pembersihan	68,24%	Filter dibersihkan secara efektif
Durasi Motor	7,8 s	Motor bekerja singkat dan tidak kontinu
Respons Sensor	Stabil	Sensor merespons perubahan kondisi secara konsisten.
Keandalan Sistem	Baik	Sistem bekerja stabil tanpa hidup–mati berulang.

Respons sensor yang cepat juga berkontribusi terhadap nilai efisiensi waktu sebesar 99,13%, yang diperoleh dari perbandingan antara waktu kerja sistem simulasi

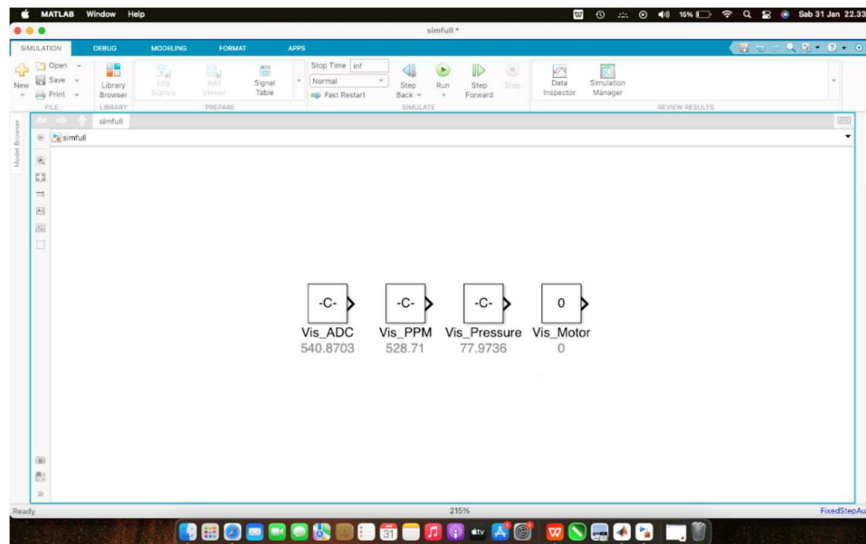
dengan waktu metode manual yang diasumsikan dalam model teoretis.

4.4 Simulasi Sistem

Subbab ini menjelaskan tahapan implementasi dan pelaksanaan simulasi sistem pembersih filter udara menggunakan MATLAB yang terintegrasi dengan Simulink. Prosedur simulasi dilakukan secara sistematis mulai dari inisialisasi lingkungan kerja, penetapan parameter, hingga eksekusi model dengan logika kendali ON–OFF berbasis hysteresis. Tahapan ini bertujuan untuk memastikan model berjalan sesuai rancangan serta menghasilkan data yang valid sebagai dasar analisis kinerja sistem.

4.4.1 Inisialisasi Lingkungan Kerja MATLAB

Tahap awal simulasi diawali dengan pembersihan lingkungan kerja MATLAB menggunakan perintah:



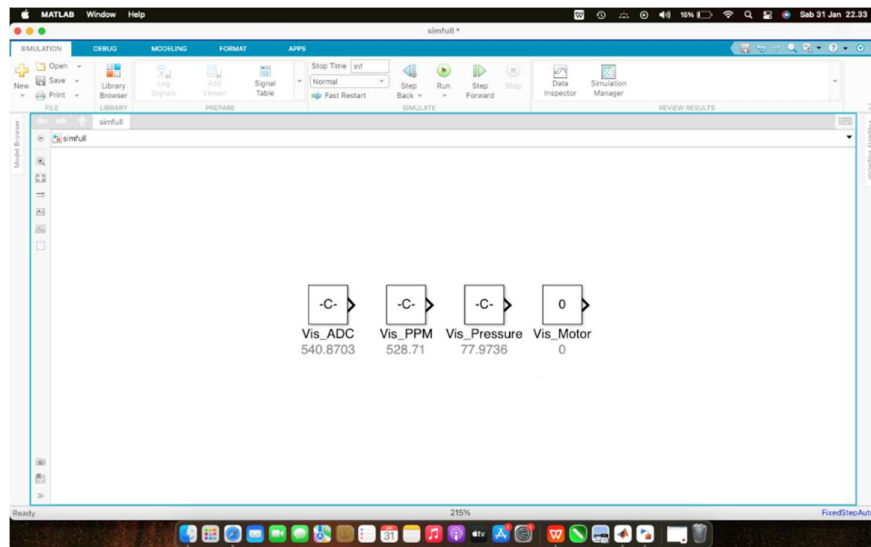
Gambar 4. 1 Inisialisasi Lingkungan Kerja MATLAB

Langkah ini bertujuan untuk menghapus seluruh variabel yang tersimpan di memori, membersihkan tampilan command window, serta menutup seluruh jendela grafik yang masih aktif. Inisialisasi ini penting untuk memastikan bahwa simulasi dijalankan pada kondisi awal yang bersih dan tidak terpengaruh oleh proses sebelumnya.

4.4.2 Persiapan dan Validasi Model Simulink

Setelah lingkungan kerja siap, langkah berikutnya adalah menentukan dan memverifikasi model Simulink yang digunakan sebagai media visualisasi sistem.

Pada bagian ini, script melakukan pengecekan keberadaan file Simulink untuk memastikan bahwa proses integrasi MATLAB–Simulink dapat berjalan dengan baik. Apabila file tidak ditemukan, simulasi dihentikan secara otomatis. Model kemudian dibuka dan diberikan jeda waktu agar seluruh blok visual siap menerima pembaruan data secara real-time.



Gambar 4. 2 Persiapan dan Validasi Model Simulink

4.4.3 Penetapan Parameter Sistem dan Penentuan Kondisi Lingkungan

```

20 open_system(PoseUserName);
21 pause(1); % Jeda sebentar agar window terbuka sempurna
22
23 % ===== 1. KONFIGURASI PENGGUNA =====
24 % ===== A. SPESIFIKASI ALAT =====
25 Daya_Sistem_Watt = 18; %
26 Faktor_Efisiensi = 0.815;
27
28 % ===== B. ANGGARAN BATAS =====
29 Threshold_Atas_ADC = 880; % Motor ON
30 Threshold_Bawah_ADC = 250; % Motor OFF
31 Threshold_Atas_dp = 120;
32 Threshold_Bawah_dp = 40;
33
34 % ===== C. LINGKUNGAN =====
35 durasi = 60;
36 dt = 0.1;
37 ADC_Awal = 100;
38 dp_Awal = 20;
39 Laju_Motor_ADC = 35;
40 Laju_Motor_dp = 5;
41
42 % ===== D. DATA MANUAL =====
43 Waktu_Manual_Merit = 15;
44 Energi_Manual_Toule = 1000;
45
46 % ===== 2. LOOP SIMULASI (ANIMASI REALTIME) =====
47
48 Receptan_Bersih = Daya_Sistem_Watt * Faktor_Efisiensi;
49 t = 0:dt:durasi; n = length(t);
50

```

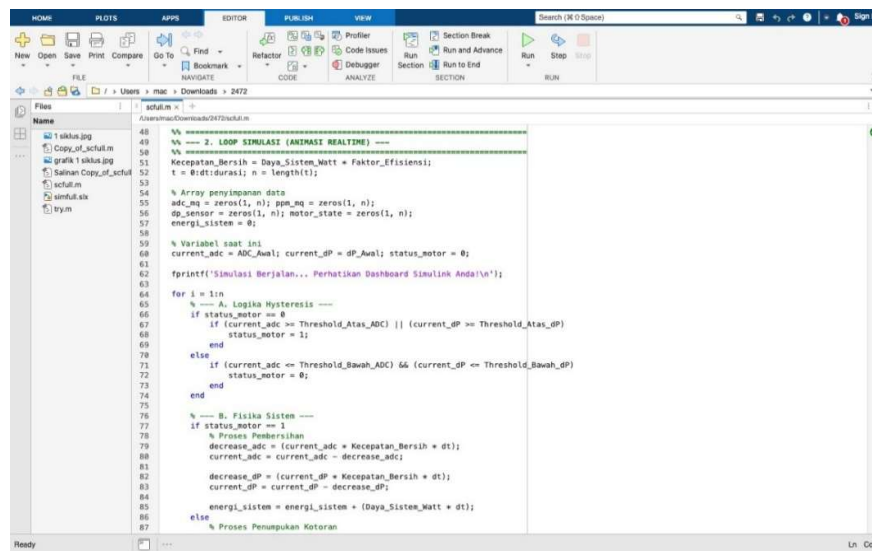
The screenshot shows the MATLAB Editor with a script file open. The script contains configuration parameters for a system simulation, including user information, device specifications, budget limits, environmental conditions, and manual data. The script is organized into sections with comments and uses MATLAB syntax for variable assignment and function calls. The bottom status bar shows the editor is at line 1, column 1.

Gambar 4. 3 Penetapan Parameter Sistem dan Penentuan Kondisi Lingkungan

Tahap selanjutnya adalah penetapan parameter sistem dan batas operasi yang digunakan dalam simulasi. Kondisi lingkungan simulasi ditetapkan untuk menggambarkan dinamika sistem selama periode pengujian. Parameter ini mencakup durasi simulasi, langkah waktu diskrit, kondisi awal sensor, serta laju penumpukan kotoran pada filter. Penetapan nilai-nilai tersebut bertujuan untuk memodelkan kondisi operasi yang mendekati keadaan nyata.

4.4.4 Inisialisasi Variabel Simulasi dan Penyimpanan Data

Sebelum simulasi dijalankan, seluruh variabel dan array penyimpanan data diinisialisasi sebagai berikut:



```

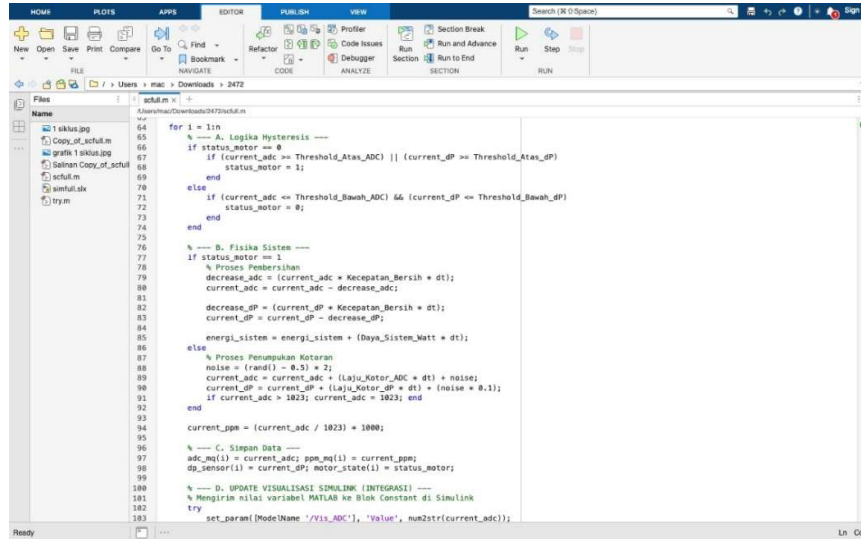
48 % ===== 2. LOOP SIMULASI (ANIMASI REALTIME) =====
49 % =====
50 % =====
51 Kecepatan_Bersih = Daya_Sistem_Matt * Faktor_Efisiensi;
52 t = 0:dt:durasi; n = length(t);
53
54 % Array penyimpanan data
55 adc_mq = zeros(1, n); ppm_mq = zeros(1, n);
56 dp_sensor = zeros(1, n); motor_state = zeros(1, n);
57 energi_sistem = 0;
58
59 % Variabel saat ini
60 current_adc = ADC_Awal; current_dp = dp_Awal; status_motor = 0;
61
62 fprintf('Simulasi Berjalan... Perhatikan Dashboard Simulasi Anda!\n');
63
64 for i = 1:n
65     % --- A. Logika Hysteresis ---
66     if status_motor == 0
67         if (current_adc >= Threshold_Atas_ADC) || (current_dp >= Threshold_Atas_dp)
68             status_motor = 1;
69         end
70     else
71         if (current_adc <= Threshold_Bawah_ADC) && (current_dp <= Threshold_Bawah_dp)
72             status_motor = 0;
73         end
74     end
75
76     % --- B. Fisika Sistem ---
77     if status_motor == 1
78         % Proses Pembersihan
79         decrease_adc = (current_adc * Kecepatan_Bersih * dt);
80         current_adc = current_adc - decrease_adc;
81
82         decrease_dp = (current_dp * Kecepatan_Bersih * dt);
83         current_dp = current_dp - decrease_dp;
84
85         energi_sistem = energi_sistem + (Daya_Sistem_Matt * dt);
86     else
87         % Proses Penumpukan Kotoran

```

Gambar 4. 4 Inisialisasi Variabel Simulasi dan Penyimpanan Data

Array digunakan untuk menyimpan riwayat nilai sensor dan status motor selama simulasi, yang selanjutnya dimanfaatkan dalam analisis dan visualisasi hasil.

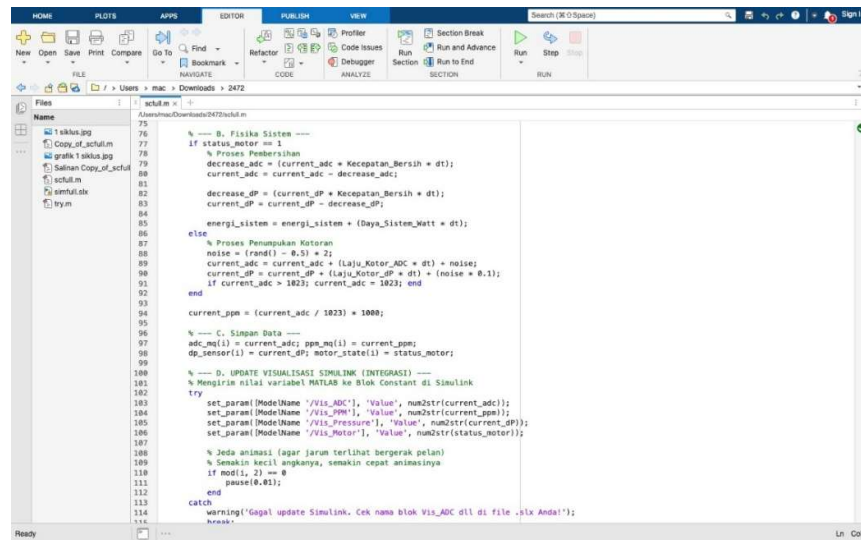
4.4.5 Proses Loop Simulasi dan Logika Kontrol Hysteresis



Gambar 4. 5 Proses Loop Simulasi dan Logika Kontrol Hysteresis

Proses utama simulasi dijalankan di dalam loop waktu diskrit yang merepresentasikan dinamika sistem terhadap waktu. Logika ini memastikan bahwa motor pembersih hanya aktif ketika kondisi kotor melampaui ambang batas yang ditentukan dan akan berhenti setelah kondisi filter kembali berada pada batas aman.

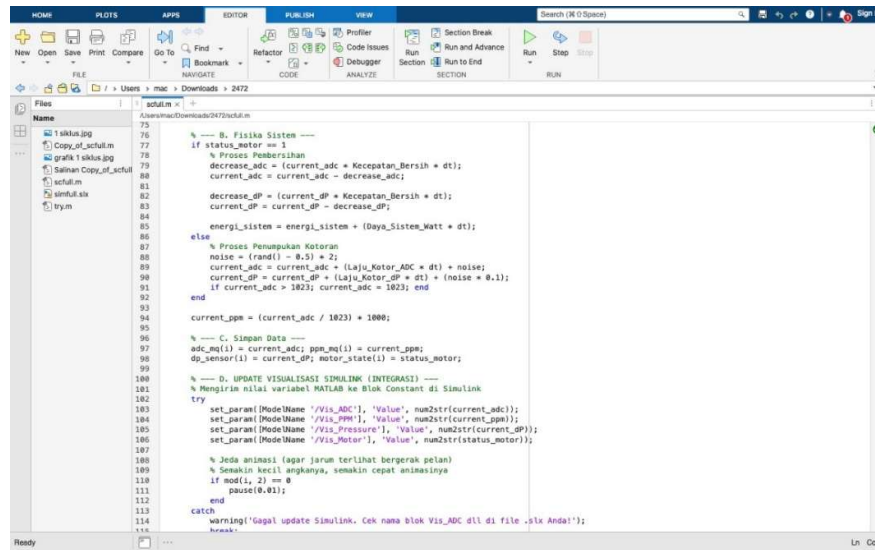
4.4.6 Pemodelan Dinamika Penumpukan dan Pembersihan Filter



Gambar 4. 6 Pemodelan Dinamika Penumpukan dan Pembersihan Filter

Setelah status motor ditentukan, dilakukan pemodelan dinamika fisika sistem. Ketika motor aktif, nilai sensor gas dan tekanan diferensial mengalami penurunan sebagai akibat proses pembersihan. Sebaliknya, saat motor tidak aktif, terjadi penumpukan kotoran yang dimodelkan dengan laju tertentu dan gangguan acak untuk menyerupai kondisi lingkungan nyata.

4.4.7 Integrasi Visualisasi Real-Time pada Simulink



Gambar 4. 7 Integrasi Visualisasi Real-Time pada Simulink

Nilai sensor dan status motor yang telah dihitung kemudian dikirimkan ke dashboard Simulink secara real-time. Tahap ini memungkinkan pengamatan langsung terhadap respon sistem, sehingga kesesuaian antara logika kontrol dan perilaku sistem dapat divalidasi secara visual.

4.4.8 Analisis Data Berdasarkan Satu Siklus Operasi

Setelah simulasi selesai, dilakukan analisis data dengan fokus pada satu siklus operasi pembersihan.

```

120 % ===== 3. ANALISIS DATA (FORMAT REVISI SINKRON) =====
121 %
122 % Kita fokus pada perhitungan 1 SIKLUS (Start pertama sampai Stop pertama)
123 % Agar data Durasi Motor sesuai dengan selisih waktu
124
125 idx_start_clean = find(motor_state == 1, 1, 'first');
126
127 if isempty(idx_start_clean)
128     % Jika motor tidak pernah nyala
129     idx_stop_clean = n;
130     Waktu_Sisten_Detik = 0;
131     val_bef_ppm = 0; val_aft_ppm = 0; val_bef_dp = 0; val_aft_dp = 0;
132 else
133     % Cari kapan motor mati SETELAH nyala pertama kali
134     temp_idx = find(motor_state(idx_start_clean:end) == 0, 1, 'first');
135     if isempty(temp_idx)
136         idx_stop_clean = n; % Masih nyala sampai akhir
137     else
138         idx_stop_clean = idx_start_clean + temp_idx - 1;
139     end
140
141 % Hitung Durasi berdasarkan selisih waktu real simulasi
142 Waktu_Sisten_Detik = (idx_stop_clean - idx_start_clean) * dt;
143
144 % Ambil data sensor pada titik tersebut
145 val_bef_ppm = ppm_mq(idx_start_clean);
146 val_aft_ppm = ppm_mq(idx_stop_clean);
147 val_bef_dp = dp_sensor(idx_start_clean);
148 val_aft_dp = dp_sensor(idx_stop_clean);
149
150 end
151
152 % Hitung Ulang Energi berdasarkan durasi fix 1 siklus
153 energi_sisten_fix = Waktu_Sisten_Detik * Daya_Sisten_Watt;
154
155 % Konversi & Rumus Efisiensi
156 Waktu_Sisten_Minut = Waktu_Sisten_Detik / 60;
157
158 if Waktu_Manual_Menit == 0, Eff_Waktu = (Waktu_Manual_Menit - Waktu_Sisten_Minut) / Waktu_Manual_Menit * 100; else, Eff_Waktu = 0; end
159
160 % ===== 4. VISUALISASI GRAFIK MATLAB =====
161 %
162 %
163
164 figure('Name', ['History Data: ', num2str(Daya_Sisten_Watt), ' Watt'], 'Color', 'w');
165 subplot(3,1,1);
166 yyaxis left; plot(t, adc_mq, 'b-', 'LineWidth', 1.5); ylabel('ADC (0-1023)'); ylim([0 1024]);
167 yyaxis right; plot(t, ppm_mq, 'r-', 'LineWidth', 1.0); ylabel('PPM');
168 yline(Threshold_Atas_ADC, 'g-', 'Batas ON'); yline(Threshold_Bawah_ADC, 'b-', 'Batas OFF');
169 title('Logika Sensor'); grid on;
170
171 subplot(3,1,2);
172 plot(t, dp_sensor, 'k-', 'LineWidth', 1.5); ylabel('Tekanan (Pa)');
173 yline(Threshold_Atas_dp, 'r-', 'Max'); yline(Threshold_Bawah_dp, 'g-', 'Target');
174 title('Tekanan Diferensial'); grid on;
175
176 subplot(3,1,3);
177 area(t, motor_state, 'FaceColor', [0.2 0.6 0.8], 'FaceAlpha', 0.5);
178 ylabel('Status'); ylim([-0.2 1.2]); title('Aktuator Motor'); grid on;
179
180 % ===== 5. OUTPUT COMMAND WINDOW (FORMAT LAPORAN) =====
181 %
182 %
183 fprintf('===== \n');
184 fprintf('HASIL SIMULASI SISTEM PEMBERSIH FILTER \n');
185 fprintf('===== \n');
186 fprintf('SETTING PENGUNDAH: \n');
187 fprintf(' - Threshold ADC : ON > %d | OFF < %d \n', Threshold_Atas_ADC, Threshold_Bawah_ADC);
188 fprintf(' - Threshold Pp : ON > %d | OFF < %d \n', Threshold_Atas_dp, Threshold_Bawah_dp);
189 fprintf(' - Daya Motor : %d Watt \n', Daya_Sisten_Watt);
190 fprintf('===== \n');

```

Gambar 4. 8 Analisis Data Berdasarkan Satu Siklus Operasi

Pendekatan ini digunakan untuk memperoleh durasi kerja motor dan parameter performa yang representatif terhadap satu proses pembersihan filter.

4.4.9 Perhitungan Efisiensi dan Indikator Kinerja Sistem

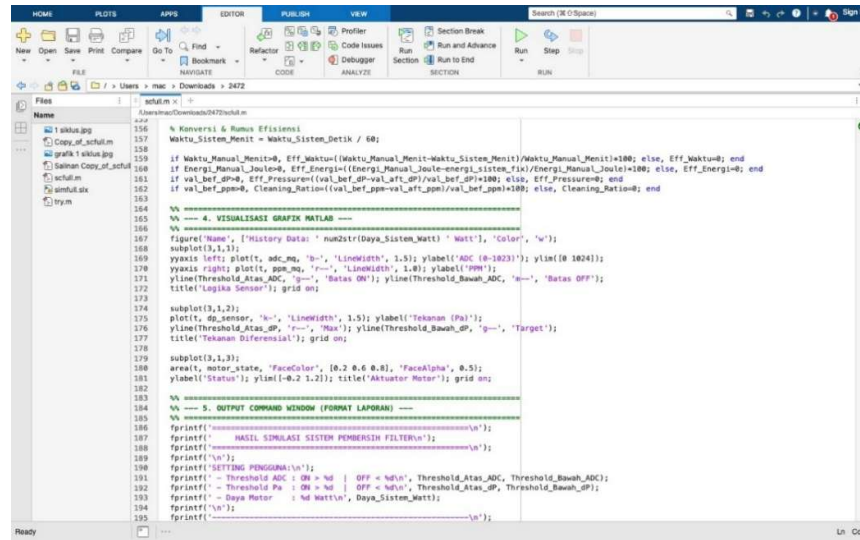
```

156 % Konversi & Rumus Efisiensi
157 Waktu_Sisten_Minut = Waktu_Sisten_Detik / 60;
158
159 if Waktu_Manual_Menit == 0, Eff_Waktu = (Waktu_Manual_Menit - Waktu_Sisten_Minut) / Waktu_Manual_Menit * 100; else, Eff_Waktu = 0; end
160
161 if Energi_Manual_Joule == 0, Eff_Energi = (Energi_Manual_Joule - energi_sisten_fix) / Energi_Manual_Joule * 100; else, Eff_Energi = 0; end
162
163 if val_bef_dp == 0, Eff_Pressure = (val_bef_dp - val_aft_dp) / val_bef_dp * 100; else, Eff_Pressure = 0; end
164
165 if val_bef_ppm == 0, Cleaning_Ratio = (val_bef_ppm - val_aft_ppm) / val_bef_ppm * 100; else, Cleaning_Ratio = 0; end
166
167 % ===== 4. VISUALISASI GRAFIK MATLAB =====
168 %
169 %
170
171 figure('Name', ['History Data: ', num2str(Daya_Sisten_Watt), ' Watt'], 'Color', 'w');
172 subplot(3,1,1);
173 yyaxis left; plot(t, adc_mq, 'b-', 'LineWidth', 1.5); ylabel('ADC (0-1023)'); ylim([0 1024]);
174 yyaxis right; plot(t, ppm_mq, 'r-', 'LineWidth', 1.0); ylabel('PPM');
175 yline(Threshold_Atas_ADC, 'g-', 'Batas ON'); yline(Threshold_Bawah_ADC, 'b-', 'Batas OFF');
176 title('Logika Sensor'); grid on;
177
178 subplot(3,1,2);
179 plot(t, dp_sensor, 'k-', 'LineWidth', 1.5); ylabel('Tekanan (Pa)');
180 yline(Threshold_Atas_dp, 'r-', 'Max'); yline(Threshold_Bawah_dp, 'g-', 'Target');
181 title('Tekanan Diferensial'); grid on;
182
183 subplot(3,1,3);
184 area(t, motor_state, 'FaceColor', [0.2 0.6 0.8], 'FaceAlpha', 0.5);
185 ylabel('Status'); ylim([-0.2 1.2]); title('Aktuator Motor'); grid on;
186
187 % ===== 5. OUTPUT COMMAND WINDOW (FORMAT LAPORAN) =====
188 %
189 %
190 fprintf('===== \n');
191 fprintf('HASIL SIMULASI SISTEM PEMBERSIH FILTER \n');
192 fprintf('===== \n');
193 fprintf('SETTING PENGUNDAH: \n');
194 fprintf(' - Threshold ADC : ON > %d | OFF < %d \n', Threshold_Atas_ADC, Threshold_Bawah_ADC);
195 fprintf(' - Threshold Pp : ON > %d | OFF < %d \n', Threshold_Atas_dp, Threshold_Bawah_dp);
196 fprintf(' - Daya Motor : %d Watt \n', Daya_Sisten_Watt);
197 fprintf('===== \n');

```

Gambar 4. 9 Perhitungan Efisiensi dan Indikator Kinerja Sistem

4.4.10 Visualisasi dan Penyajian Hasil Simulasi



Gambar 4. 10 Visualisasi dan Penyajian Hasil Simulasi

Hasil simulasi divisualisasikan dalam bentuk grafik untuk mempermudah analisis performa sistem. Grafik ini menunjukkan respon sensor terhadap waktu, perubahan tekanan diferensial, serta pola kerja motor pembersih. Seluruh hasil numerik juga ditampilkan dalam command window dalam format laporan teknis sebagai dasar pembahasan lanjutan pada subbab analisis kinerja sistem.

4.4.11 Penyajian dan Interpretasi Hasil Simulasi Sistem

Subbab ini menyajikan hasil keluaran (output) simulasi MATLAB yang diperoleh setelah seluruh proses simulasi selesai dijalankan. Hasil ditampilkan melalui command window MATLAB dalam format laporan numerik yang merepresentasikan kondisi operasional sistem pembersih filter udara.

Hasil simulasi ditampilkan menggunakan perintah fprintf sebagai berikut:

```

156 % Konversi & Rumus Efisiensi
157 Waktu_Sistem_Menit = Waktu_Sistem_Detik / 60;
158
159 if Waktu_Manual_Menit > 0, Eff_Waktu = (Waktu_Manual_Menit - Waktu_Sistem_Menit) / Waktu_Manual_Menit; else, Eff_Waktu = 0; end
160 if Energi_Manual_Joule > 0, Eff_Energi = (Energi_Manual_Joule - Energi_Sistem_Fix) / Energi_Manual_Joule; else, Eff_Energi = 0; end
161 if va_bef_ppm > 0, Eff_Pressure = (va_bef_ppm - va_af_ppm) / va_bef_ppm; else, Eff_Pressure = 0; end
162 if va_bef_ppm > 0, Cleaning_Ratio = (va_bef_ppm - va_af_ppm) / va_bef_ppm; else, Cleaning_Ratio = 0; end
163
164 % ===== 4. VISUALISASI GRAFIK MATLAB =====
165
166 figure('Name', ['num2str(Daya_Sistem_Matt') ' Watt'], 'Color', 'w');
167 subplot(3,1,1);
168 yyaxis left; plot(t, adc_mq, 'b-', 'LineWidth', 1.5); ylabel('ADC (0-1023)'); ylim([0 1024]);
169 yyaxis right; plot(t, ppm_mq, 'r-', 'LineWidth', 1.0); ylabel('PPM');
170 ylim([Threshold_Atas_adc, 'g--', 'Batas ON']; ylim([Threshold_Bawah_adc, 'm--', 'Batas OFF']);
171 title('Logika Sensor'); grid on;
172
173 subplot(3,1,2);
174 plot(t, dp_sensor, 'k-', 'LineWidth', 1.5); ylabel('Tekanan (Pa)');
175 ylim([Threshold_Atas_dp, 'r--', 'Max']; ylim([Threshold_Bawah_dp, 'g--', 'Target']);
176 title('Tekanan Diferensial'); grid on;
177
178 subplot(3,1,3);
179 area(t, motor_state, 'FaceColor', [0.2 0.6 0.8], 'FaceAlpha', 0.5);
180 ylabel('Status'); ylim([-0.2 1.2]); title('Aktuator Motor'); grid on;
181
182 % ===== 5. OUTPUT COMMAND WINDOW (FORMAT LAPORAN) =====
183
184 fprintf('===== HASIL SIMULASI SISTEM PENDING FILTER =====\n');
185 fprintf('===== PENGUNCIAN PENGUNCIAN =====\n');
186 fprintf('===== PENGUNCIAN PENGUNCIAN =====\n');
187 fprintf('===== PENGUNCIAN PENGUNCIAN =====\n');
188 fprintf('===== PENGUNCIAN PENGUNCIAN =====\n');
189 fprintf('===== PENGUNCIAN PENGUNCIAN =====\n');
190 fprintf('===== PENGUNCIAN PENGUNCIAN =====\n');
191 fprintf('===== PENGUNCIAN PENGUNCIAN =====\n');
192 fprintf('===== PENGUNCIAN PENGUNCIAN =====\n');
193 fprintf('===== PENGUNCIAN PENGUNCIAN =====\n');
194 fprintf('===== PENGUNCIAN PENGUNCIAN =====\n');
195 fprintf('===== PENGUNCIAN PENGUNCIAN =====\n');

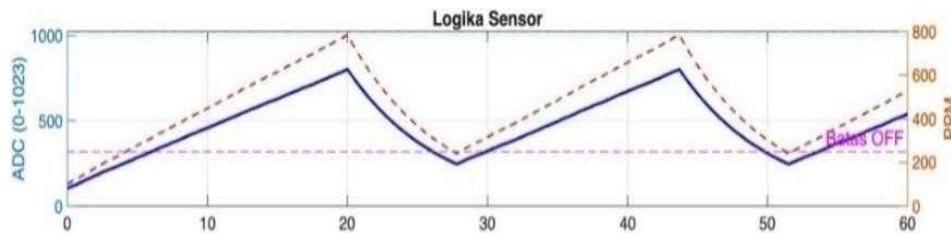
```

Gambar 4. 11 Kode Program Penyajian Hasil Simulasi

Kode di atas berfungsi untuk menyajikan hasil simulasi secara terstruktur dan sistematis, sehingga memudahkan proses evaluasi kinerja sistem berdasarkan parameter yang telah ditentukan. Hasil simulasi sistem pembersih filter ditampilkan pada Command Window MATLAB menggunakan fungsi fprintf. Output ini menyajikan parameter operasional satu siklus pembersihan, meliputi waktu aktivasi sistem, durasi kerja motor, konsumsi energi, serta indikator efisiensi sistem.

Sebagai pendukung rekapitulasi kinerja sistem, grafik hasil simulasi MATLAB–Simulink disajikan untuk memberikan representasi visual terhadap perilaku sistem selama simulasi berlangsung.

1) Logika Sensor

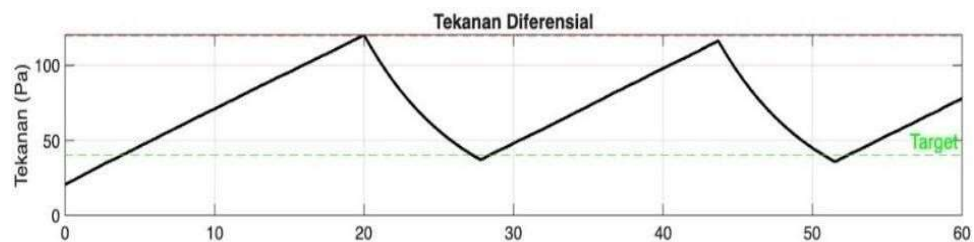


Gambar 4. 1 Grafik Logika Sensor

Grafik logika sensor menunjukkan peningkatan nilai ADC hingga melampaui

ambang batas atas, yang memicu aktivasi sistem pembersih pada waktu 19,9s. Setelah motor aktif, nilai konsentrasi partikel udara (PM) menurun hingga kondisi bersih tercapai pada 27,7s. Perubahan nilai sebelum dan sesudah pembersihan menghasilkan rasio pembersihan sebesar 68,24%, yang menunjukkan bahwa sistem mampu menurunkan konsentrasi partikel secara signifikan dalam satu siklus kerja. Respons sensor yang cepat juga berkontribusi terhadap nilai efisiensi waktu sebesar 99,13%.

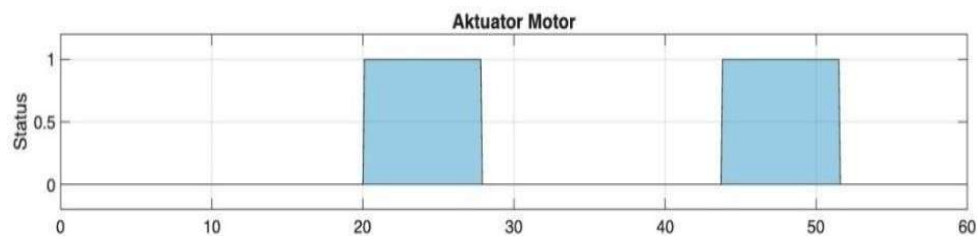
2) Tekanan Diferensial



Gambar 4. 2 Grafik Tekanan Diferensial

Grafik tekanan diferensial memperlihatkan peningkatan ΔP akibat penumpukan kotoran hingga mencapai ambang batas atas, kemudian menurun secara signifikan selama motor aktif. Penurunan tekanan diferensial dari kondisi awal hingga akhir pembersihan menghasilkan efisiensi tekanan diferensial sebesar 68,29%. Hasil ini menunjukkan bahwa aliran udara yang dihasilkan oleh sistem efektif dalam mengurangi hambatan aliran pada media filter selama satu siklus pembersihan.

3) Aktuator Motor



Gambar 4. 3 Grafik Aktuator Motor

Grafik status aktuator motor menunjukkan motor aktif selama 7,8 s dalam satu siklus kerja. Dengan daya motor sebesar 10W, konsumsi energi sistem tercatat sebesar 78 Joule.