

LAMPIRAN

Lampiran 1. Script MATLAB Simulasi Sistem Pembersih Filter Udara Berbasis MATLAB–Simulink

Lampiran ini menyajikan script MATLAB yang digunakan untuk menjalankan simulasi sistem pembersih filter udara. Script ini berfungsi sebagai pengendali utama simulasi serta sebagai penghubung antara MATLAB dan model Simulink dalam menampilkan respons sistem secara real time.

```
%%
-----
%% PROGRAM SIMULASI FILTER UDARA (MATLAB DRIVING
SIMULINK)
% Integrasi: Script MATLAB menggerakkan Dashboard
'simfull.slx'
% Fitur: Auto-Link Fisika + Visualisasi Realtime +
Laporan Akhir
clear; clc; close all;
%% --- 0. PERSIAPAN KONEKSI SIMULINK ---
modelName = 'simfull'; %
% Cek keberadaan file
if ~exist([modelName '.slx'], 'file')
    error('File "%s.slx" tidak ditemukan. Pastikan nama
file sudah benar!', modelName);
end
% Buka Simulink
fprintf('Membuka          Simulink      Dashboard:
          %s...\n', modelName);
open_system(modelName);
pause(1); % Jeda sebentar agar window terbuka sempurna
%%
=====
=====
%% --- 1. KONFIGURASI PENGGUNA ---
%%
```

```

=====
=====
% --- A. SPESIFIKASI ALAT ---
Daya_Sistem_Watt          = 10;      % Coba ubah: 10, 50,
100 Watt
Faktor_Efisiensi          = 0.015;

% --- B. AMBANG BATAS ---
Threshold_Atas_ADC = 800; % Motor ON
Threshold_Bawah_ADC = 250; % Motor OFF
Threshold_Atas_dP      = 120;
Threshold_Bawah_dP     = 40;

% --- C. LINGKUNGAN ---
durasi = 60;
dt      = 0.1;
ADC_Awal = 100;
dP_Awal  = 20;
Laju_Kotor_ADC = 35;
Laju_Kotor_dP  = 5;

% --- D. DATA MANUAL ---
Waktu_Manual_Menit = 15;
Energi_Manual_Joule = 5000;%%
=====
=====

%% --- 2. LOOP SIMULASI (ANIMASI REALTIME) ---
%%
=====
=====

Kecepatan_Bersih = Daya_Sistem_Watt * Faktor_Efisiensi;
t = 0:dt:durasi; n = length(t);

```

```

% Array penyimpanan data
adc_mq = zeros(1, n); ppm_mq = zeros(1, n); dp_sensor =
zeros(1, n); motor_state = zeros(1, n); energi_sistem =
0;

% Variabel saat ini
current_adc      = ADC_Awal;    current_dP      =
    dP_Awal; status_motor = 0;

fprintf('Simulasi      Berjalan...  Perhatikan
      Dashboard Simulink Anda!\n');

for i = 1:n
% --- A. Logika Hysteresis --- if status_motor == 0
if      (current_adc              >=
      Threshold_Atas_ADC)
|| (current_dP >= Threshold_Atas_dP) status_motor = 1;

else

end

if      (current_adc      <=      Threshold_Bawah_ADC)
&&(current_dP <= Threshold_Bawah_dP)
status_motor = 0;
end
end

% --- B. Fisika Sistem --- if status_motor == 1
% Proses Pembersihan
decrease_adc = (current_adc * Kecepatan_Bersih
* dt);
current_adc = current_adc - decrease_adc;

```

```

decrease_dP = (current_dP * Kecepatan_Bersih * dt);
current_dP = current_dP - decrease_dP;
else
end
energi_sistem = energi_sistem +
(Daya_Sistem_Watt * dt);

% Proses Penumpukan Kotoran noise = (rand() - 0.5) * 2;
current_adc = current_adc + (Laju_Kotor_ADC * dt) +
noise;
current_dP = current_dP + (Laju_Kotor_dP * dt)
+ (noise * 0.1);
if current_adc > 1023; current_adc = 1023; end
current_ppm = (current_adc / 1023) * 1000;
% --- C. Simpan Data ---

adc_mq(i) = current_adc; ppm_mq(i) =
current_ppm; dp_sensor(i) = current_dP;
motor_state(i) = status_motor;

% --- D. UPDATE VISUALISASI SIMULINK (INTEGRASI) --
-
% Mengirim nilai variabel MATLAB ke Blok Constant di
Simulink
try

        set_param([ModelName '/Vis_ADC'], 'Value',
        num2str(current_adc));
        set_param([ModelName '/Vis_PPM'], 'Value',
        num2str(current_ppm));
set_param([ModelName '/Vis_Pressure'], 'Value',
num2str(current_dP));
set_param([ModelName '/Vis_Motor'],

```

```

        'Value', num2str(status_motor));

% Jeda animasi (agar jarum terlihat bergerak pelan)
%     Semakin kecil angkanya,          semakin
        cepat animasinya
if mod(i, 2) == 0 pause(0.01);
    end catch
        warning('Gagal update Simulink. Cek nama blok
Vis_ADC dll di file .slx Anda!');
break;
end
end
fprintf('Simulasi Selesai.\n');
%%
=====
=====
%% --- 3. ANALISIS DATA (FORMAT REVISI SINKRON) ---
%%
=====
=====
idx_start_clean = find(motor_state == 1, 1, 'first');

if isempty(idx_start_clean)
% Jika motor tidak pernah nyala idx_stop_clean = n;
Waktu_Sistem_Detik = 0;
val_bef_ppm = 0; val_aft_ppm = 0; val_bef_dP = 0;
val_aft_dP = 0; else
% Cari kapan motor mati SETELAH nyala pertama kali
    temp_idx = find(motor_state(idx_start_clean:end) ==
0, 1, 'first');

if isempty(temp_idx)
idx_stop_clean = n; % Masih nyala sampai akhir

```

```

else
idx_stop_clean = idx_start_clean + temp_idx - 1;
end

    % Hitung Durasi berdasarkan selisih waktu real
simulasi
    Waktu_Sistem_Detik          =      (idx_stop_clean
    - idx_start_clean) * dt;

% Ambil data sensor pada titik tersebut
end
val_bef_ppm = ppm_mq(idx_start_clean); val_aft_ppm =
ppm_mq(idx_stop_clean); val_bef_dP =
dp_sensor(idx_start_clean); val_aft_dP =
dp_sensor(idx_stop_clean);

% Hitung Ulang Energi berdasarkan durasi fix 1 siklus
energi_sistem_fix              =      Waktu_Sistem_Detik
    * Daya_Sistem_Watt;
% Konversi & Rumus Efisiensi Waktu_Sistem_Menit =
Waktu_Sistem_Detik / 60;
if      Waktu_Manual_Menit>0,
Eff_Waktu=((Waktu_Manual_Menit-
Waktu_Sistem_Menit)/Waktu_Manual_Menit)*100;
    else,
Eff_Waktu=0; end
if      Energi_Manual_Joule>0,
Eff_Energi=((Energi_Manual_Joule-
energi_sistem_fix)/Energi_Manual_Joule)*100;
    else,
Eff_Energi=0; end
if      val_bef_dP>0,
    Eff_Pressure=((val_bef_dP-

```

```

val_aft_dP)/val_bef_dP)*100; else, Eff_Pressure=0; end
if      val_bef_ppm>0,      Cleaning_Ratio=((val_bef_ppm-
val_aft_ppm)/val_bef_ppm)*100; else, Cleaning_Ratio=0;
end

%%
=====
=====

%% --- 4. VISUALISASI GRAFIK MATLAB ---
%%
=====
=====

figure('Name',          ['History          Data:
      ' num2str(Daya_Sistem_Watt) ' Watt'], 'Color',
'w');
subplot(3,1,1);
yyaxis left; plot(t, adc_mq, 'b-', 'LineWidth', 1.5);
ylabel('ADC (0-1023)'); ylim([0 1024]);
yyaxis right; plot(t, ppm_mq, 'r--', 'LineWidth', 1.0);
ylabel('PPM');
yline(Threshold_Atas_ADC,          'g--',      'Batas
      ON'); yline(Threshold_Bawah_ADC, 'm--', 'Batas
OFF');
title('Logika Sensor'); grid on;
subplot(3,1,2);
plot(t,      dp_sensor,      'k-',      'LineWidth',
      1.5); ylabel('Tekanan (Pa)');
yline(Threshold_Atas_dP,          'r--',
      'Max'); yline(Threshold_Bawah_dP, 'g--',
'Target');
title('Tekanan Diferensial'); grid on;

subplot(3,1,3);

```

```

area(t, motor_state, 'FaceColor', [0.2 0.6 0.8],
'FaceAlpha', 0.5);
ylabel('Status'); ylim([-0.2 1.2]); title('Aktuator
Motor'); grid on;

%%
=====
=====
%% --- 5. OUTPUT COMMAND WINDOW (FORMAT LAPORAN) ---
%%
=====
=====
fprintf('=====
=====\\n');
fprintf('          HASIL SIMULASI SISTEM PEMBERSIH
FILTER\\n');

fprintf('=====
=====\\n');
fprintf('\\n'); fprintf('SETTING PENGGUNA:\\n');
fprintf(' - Threshold ADC : ON > %d | OFF < %d\\n',
Threshold_Atas_ADC, Threshold_Bawah_ADC);
fprintf(' - Threshold Pa : ON > %d | OFF < %d\\n',
Threshold_Atas_dP, Threshold_Bawah_dP);
fprintf('          - Daya Motor : %d
          Watt\\n', Daya_Sistem_Watt);
fprintf('\\n'); fprintf('
          \\n');
fprintf('-----
HASIL OPERASIONAL (Analisis 1 Siklus):\\n'); if
~isempty(idx_start_clean)
    fprintf(' - Detik Kotor Terdeteksi : %.1f
s\\n', t(idx_start_clean));
if idx_stop_clean < n

```

```

        fprintf(' - Detik Bersih Tercapai : %.1f s\n',
t(idx_stop_clean));
else
        fprintf(' - Detik Bersih Tercapai :
(Belum Selesai)\n');
end
else
        fprintf(' - Status : Sistem Stabil (Tidak Mencapai
Threshold)\n'); end
fprintf(' - Durasi Motor Menyala : %.2f
detik\n', Waktu_Sistem_Detik);
fprintf(' - Konsumsi Energi : %.2f
Joule\n', energi_sistem_fix);
fprintf('\n');

fprintf('-----
- \n');
fprintf('INDIKATOR EFISIENSI (Berdasarkan
Rumus PDF):\n');
fprintf(' 1. Efisiensi Waktu (eta_t) : %.2f
%%\n', Eff_Waktu);
fprintf(' 2. Efisiensi Energi (eta_e) : %.2f
%%\n', Eff_Energi);
fprintf(' 3. Efisiensi Tekanan (eta_p) : %.2f %%\n',
Eff_Pressure);
fprintf(' 4. Rasio Pembersihan (Rc) : %.2f
%%\n', Cleaning_Ratio);
fprintf('=====
=====\\n');

```