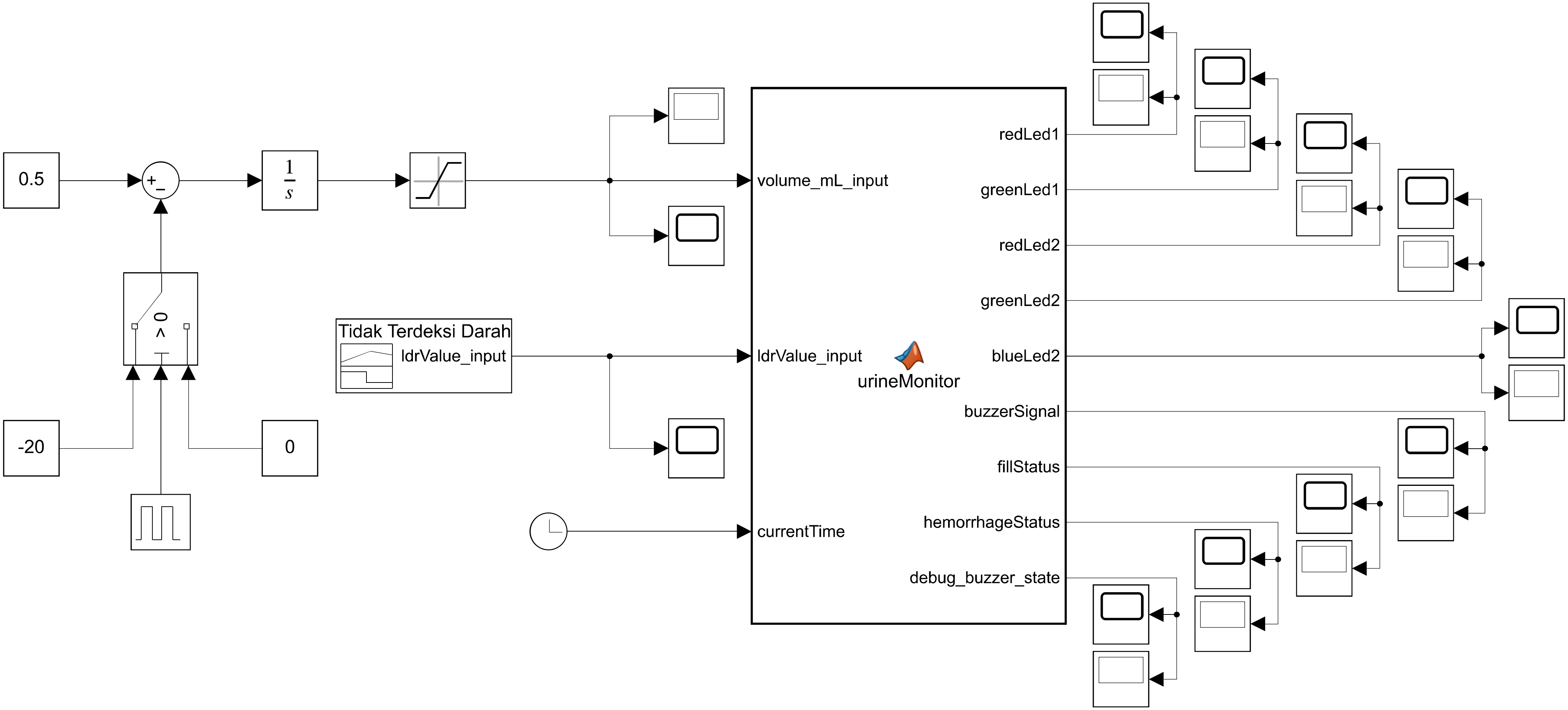
0.5
+
_
1
s
-20
0
> 0
Tidak Terdeksi Darah
ldrValue_input
volume_mL_input
ldrValue_input
currentTime
urineMonitor
redLed1
greenLed1
redLed2
greenLed2
blueLed2
buzzerSignal
fillStatus
hemorrhageStatus
debug_buzzer_state

```
function [redLed1, greenLed1, redLed2, greenLed2, blueLed2, buzzerSignal, fillStatus, hemorrhageStatus, debug_buzzer_state] = urineM

    persistent buzzer_state;
    persistent buzzer_hemorrhage_timer;
    persistent buzzer_full_count;
    persistent full_beep_timer;
    persistent full_beep_state;

    if isempty(buzzer_state)
        buzzer_state = 0;
    end
    if isempty(buzzer_hemorrhage_timer)
        buzzer_hemorrhage_timer = 0;
    end
    if isempty(buzzer_full_count)
        buzzer_full_count = 0;
    end
    if isempty(full_beep_timer)
        full_beep_timer = 0;
    end
    if isempty(full_beep_state)
        full_beep_state = 0;
    end

    LDR_THRESHOLD = 1000;
    LEVEL_KOSONG = 0;
    LEVEL_HAMPIR_PENUH_START = 70.0;
    LEVEL_PENUH_START = 90.0;
    MAX_VOLUME = 100.0;

    redLed1 = 0;
    greenLed1 = 0;
    redLed2 = 0;
    greenLed2 = 0;
    blueLed2 = 0;
    buzzerSignal = 0;
    fillStatus = 0;
    hemorrhageStatus = 0;

    isUrineNormal = (ldrValue_input > LDR_THRESHOLD);
    if ~isUrineNormal
        redLed1 = 1;
        greenLed1 = 0;
        hemorrhageStatus = 1;
    else
        redLed1 = 0;
        greenLed1 = 1;
        hemorrhageStatus = 0;
    end

    if volume_mL_input >= LEVEL_PENUH_START && volume_mL_input <= MAX_VOLUME
        fillStatus = 2;
        redLed2 = 1;
        greenLed2 = 0;
        blueLed2 = 0;
    elseif volume_mL_input >= LEVEL_HAMPIR_PENUH_START && volume_mL_input < LEVEL_PENUH_START
        fillStatus = 1;
        redLed2 = 0;
        greenLed2 = 0;
        blueLed2 = 1;
    else
    fillStatus = 0;
    redLed2 = 0;
    greenLed2 = 1;
    blueLed2 = 0;
end

current_buzzer_mode = 0;

if hemorrhageStatus == 1
    current_buzzer_mode = 2;
elseif fillStatus == 2
    current_buzzer_mode = 1;
elseif fillStatus == 1
    current_buzzer_mode = 3;
end

if current_buzzer_mode == 2
    if buzzer_state ~= 2
        buzzer_state = 2;
        buzzer_hemorrhage_timer = currentTime;
        buzzer_full_count = 0;
        full_beep_state = 0;
        full_beep_timer = 0;
    end

    if (currentTime - buzzer_hemorrhage_timer) < 60
        if mod(floor(currentTime * 5), 2) == 0
            buzzerSignal = 1;
        else
            buzzerSignal = 0;
        end
    else
        buzzer_state = 0;
        buzzerSignal = 0;
        buzzer_hemorrhage_timer = 0;
    end

elseif current_buzzer_mode == 1
    if buzzer_state ~= 1
        buzzer_state = 1;
        buzzer_hemorrhage_timer = 0;
        buzzer_full_count = 0;
        full_beep_state = 0;
        full_beep_timer = 0;
    end

    switch full_beep_state
        case 0
            if buzzer_full_count < 3
                full_beep_state = 1;
                full_beep_timer = currentTime;
                buzzerSignal = 1;
                buzzer_full_count = buzzer_full_count + 1;
            else
                buzzerSignal = 0;
                buzzer_state = 0;
                buzzer_full_count = 0;
                full_beep_state = 0;
                full_beep_timer = 0;
            end
        case 1
```

```
                if (currentTime - full_beep_timer) < 0.2
                    buzzerSignal = 1;
                else
                    full_beep_state = 2;
                    full_beep_timer = currentTime;
                    buzzerSignal = 0;
                end
            case 2
                if (currentTime - full_beep_timer) < 0.3
                    buzzerSignal = 0;
                else
                    full_beep_state = 0;
                end
        end

    elseif current_buzzer_mode == 3
        if buzzer_state ~= 3
            buzzer_state = 3;
            buzzer_full_count = 0;
            full_beep_state = 0;
            buzzer_hemorrhage_timer = 0;

            buzzerSignal = 1;
            full_beep_timer = currentTime;
        end

        if buzzerSignal == 1 && (currentTime - full_beep_timer) >= 0.2
            buzzerSignal = 0;
        end

    else
        buzzer_state = 0;
        buzzerSignal = 0;
        buzzer_hemorrhage_timer = 0;
        buzzer_full_count = 0;
        full_beep_state = 0;
        full_beep_timer = 0;
    end

    debug_buzzer_state = buzzer_state;

end
```