

M3

TP 1: Percobaan 2 Kondisi

[home](#)

[\[KEMBALI KE MENU SEBELUMNYA\]](https://septinam211002.blogspot.com/2024/03/tp-1-modul-1.html#)

DAFTAR ISI

[1. Prosedur](https://septinam211002.blogspot.com/2024/03/tp-1-modul-1.html#kondisi)

[2. Hardware dan diagram blok](https://septinam211002.blogspot.com/2024/03/tp-1-modul-1.html#gambar)

[3. Rangkaian Simulasi dan Prinsip kerja](https://septinam211002.blogspot.com/2024/03/tp-1-modul-1.html#video)

[4. Flowchart](https://septinam211002.blogspot.com/2024/03/tp-1-modul-1.html#prinsip) dan Listing Program

[5. Kondisi](https://septinam211002.blogspot.com/2024/03/tp-1-modul-1.html#link)

[6. Video Simulasi](https://septinam211002.blogspot.com/2024/03/tp-1-modul-1.html#link)

[7. Download File](https://septinam211002.blogspot.com/2024/03/tp-1-modul-1.html#link)

</div></div></center>
<div style="text-align: center;">MODUL 3</div><div style="text-align: center;">PERCOBAAN 1 KONDISI 6</div><div>
</div>1. Prosedur <a

[\[Kembali\]](https://septinam211002.blogspot.com/2024/03/tp-1-modul-1.html#home)

1. Rangkai semua komponen sesuai kondisi yang dipilih

2. Buat program di aplikasi arduino IDE

3. Setelah selesai masukkan program ke arduino di proteus

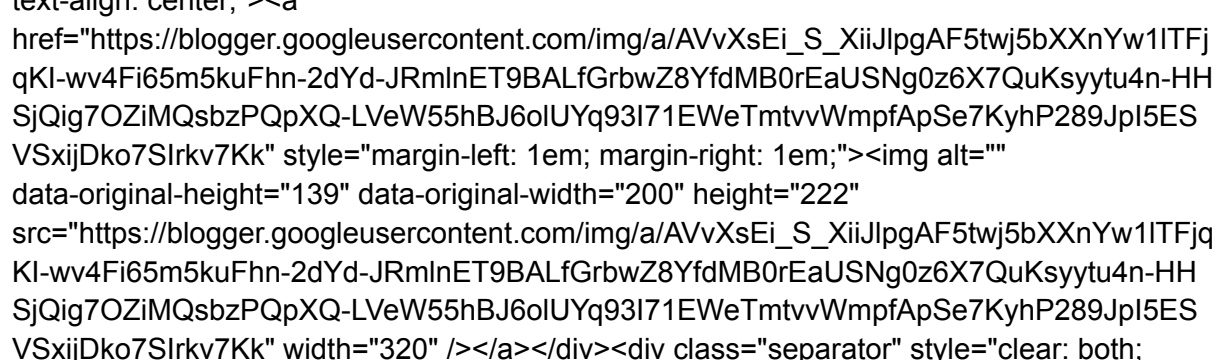
4. Jalankan program pada simulasi dan cobakan sesuai dengan modul dan kondisi

5. Selesai

2. Hardware dan diagram blok



1. Arduino Uno



color: #222222; font-family: Georgia, Utopia, "Palatino Linotype"; Palatino, serif; font-size: 15.4px; text-align: center;">
</div><div class="separator" style="clear: both; text-align: center;"><div class="separator" style="clear: both; color: #222222; font-family: Georgia, Utopia, "Palatino Linotype"; Palatino, serif; font-size: 15.4px; text-align: center;"></div><div style="text-align: left;">Spesifikasi: </div><div style="text-align: left;"><div class="separator" style="clear: both; text-align: center;"></div>

</div></div>2. Dipswitch</div><div style="color: #222222; font-family: Georgia, Utopia, "Palatino Linotype"; Palatino, serif; font-size: 15.4px;"><div class="separator" style="clear: both; text-align: center;"></div>Spesifikasi: </div><div style="color: #222222; font-family: Georgia, Utopia, "Palatino Linotype"; Palatino, serif; font-size: 15.4px;"><div class="separator" style="clear: both; text-align: center;"></div>

[illegible]

href="https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEhV2ixMvFot7h3acvHg1HR99Jd9MTmHQGGGfpuDHRCa2_MA5Bla8e3lj6WhOdPnl25OvDYMjofzvSvqH-ifgsoykhMBIXIEx9ZWDAstNr9Vk7MQGBNAITHQcilu_yhnJDKaK5PMucQmO302hew8Krr8-Wou7Bhoxn95rhyphenhyphenFlmmr-FGyvuWsiD5jBpUTwfQ/s1573/P1K6.png" style="margin-left: 1em; margin-right: 1em;"></div><div class="separator" style="clear: both; text-align: center;">
</div></div><div style="color: #222222; font-family: Georgia, Utopia, "Palatino Linotype", Palatino, serif; font-size: 15.4px;"><div class="separator" style="clear: both; text-align: left;"><b style="color: black; font-family: times, "times new roman", serif; font-size: 16.5px;">→<span style="color: #990000; font-family: times,

"times new roman", serif; font-size: 16.5px;">Gambar Rangkaian Setelah Disimulasikan</div><div class="separator" style="clear: both; text-align: left;">
</div><div class="separator" style="clear: both; text-align: left;"><div class="separator" style="clear: both; text-align: center;"></div></div><div class="separator" style="clear: both; text-align: center;"><div class="separator" style="clear: both; text-align: center;">
</div></div></div><div style="color: #222222; font-family: Georgia, Utopia, "Palatino Linotype", Palatino, serif; font-size: 15.4px;">PRINSIP KERJA</div><div style="color: #222222; font-family: Georgia, Utopia, "Palatino Linotype", Palatino, serif; text-align: justify;"><div class="separator" style="box-sizing: border-box; clear: both; color: #555555; font-family: Roboto, Arial, Helvetica, sans-serif; text-align: left;"><b style="box-sizing: border-box;">Master:</div><div class="separator" style="box-sizing: border-box; clear: both; color: #555555; font-family: Roboto, Arial, Helvetica, sans-serif;"> Pada rangkaian di atas merupakan perocbaan Komunikasi UART. Arduino yang diatas bertindak sebagai master dalam sebuah sistem yang menggunakan beberapa tombol sebagai input. Setiap tombol dihubungkan ke pin digital pada Arduino dan masing-masing tombol memiliki aksi yang berbeda saat ditekan. yang mana pada pin DS1 sampai dengan DS4 merukanan tombol atau switch PULL DOWN dan DS5 sampai DS8 merupakan switch PULL UP Saat program berjalan, Arduino akan terus memeriksa status setiap tombol dengan

menggunakan fungsi `digitalRead()`. Jika sebuah tombol ditekan, Arduino akan mengirimkan karakter yang sesuai melalui koneksi serial (UART) dengan menggunakan fungsi `Serial.write()`. Karakter yang dikirimkan akan sesuai dengan tombol yang ditekan, dan akan dibaca oleh perangkat arduino lain yang terhubung dengan Arduino melalui koneksi serial. Dalam simulasi pada Proteus ini, jika menekan 1 switch pull-down dan 4 switch pull up akan mengirimkan data ke slave arduino dan nantinya mengaktifkan running led, dan jika 4 switch pull down dan 1 switch pull UP akan mengirimkan kondisi ini ke slave arduino dan nantinya akan mengaktifkan blinking led.

Pada arduino yang kedua bertindak sebagai Slave dimana nantinya akan menerima inputan dari Arduino master dan mengeluarkan output berupa LED. Prinsip kerja rangkaian di atas adalah untuk mengontrol LED-LED yang terhubung ke pin Arduino berdasarkan status tombol-tombol yang terhubung ke pin digital lainnya. Pada `setup()`, pin-pin yang terhubung ke tombol-tombol diatur sebagai input dengan resistansi pull-up atau pull-down sesuai dengan kondisi. Pada `loop()`, program akan terus memeriksa status tombol-tombol tersebut. Jika kondisi tertentu terpenuhi, seperti empat tombol dalam kondisi pull-down atau tidak ada tombol dalam kondisi pull-up yang ditekan, maka program akan menjalankan aksi tertentu. Misalnya, untuk kondisi pull-down empat tombol, program akan menyalakan LED secara berurutan (running LED), sementara untuk kondisi ketika tidak ada tombol pull-down atau pull-up yang ditekan, program akan membuat LED-LED tertentu berkedip (blinking LED). Jika tidak ada kondisi yang terpenuhi, maka semua LED akan dimatikan. Dengan demikian, prinsip kerja rangkaian tersebut adalah memantau status tombol-tombol dan mengatur tindakan yang sesuai terhadap LED-LED yang terhubung.

"Palatino Linotype", Palatino, serif, font-size: 15.4px;">

 4.

FlowChart [Kembali]</div><div style="color: #222222; font-family: Georgia, Utopia, "Palatino Linotype", Palatino, serif, font-size: 15.4px;">
</div><div style="color: #222222; font-family: Georgia, Utopia, "Palatino Linotype", Palatino, serif, font-size: 15.4px;">a. Listing Program </div><div style="color: #222222; font-family: Georgia, Utopia, "Palatino Linotype", Palatino, serif, font-size: 15.4px;">
</div><div style="color: #222222; font-family: Georgia, Utopia, "Palatino Linotype", Palatino, serif, font-size: 15.4px;">Master</div><div style="color: #222222;

```

font-family: Georgia, Utopia, &quot;Palatino Linotype&quot;, Palatino, serif; font-size:
15.4px;"><span style="font-family: inherit;"><div style="background-color: #1f272a; color:
#dae3e3; font-family: Consolas, &quot;Courier New&quot;, monospace; font-size: 14px;
line-height: 19px; white-space: pre;"><div><div style="line-height: 19px;"><div><div
style="line-height: 19px;"><br /><div><span style="color: #7f8c8d;">//Master
Arduino</span></div><div><span style="color: #c586c0;">#include</span><span
style="color: #7fcbcd;">&lt;SPI.h&gt;</span> <span style="color: #7f8c8d;">//Library for
SPI</span></div><div><span style="color: #0ca1a6;">int</span> dip[] = {<span style="color:
#7fcbcd;">2</span><span style="color: #7fcbcd;">3</span><span style="color:
#7fcbcd;">4</span><span style="color: #7fcbcd;">5</span><span style="color:
#7fcbcd;">6</span><span style="color: #7fcbcd;">7</span><span style="color:
#7fcbcd;">8</span><span style="color: #7fcbcd;">9</span>};</div><div><span
style="color: #0ca1a6;">int</span> dipvalue[] = {};</div><div><span style="color:
#0ca1a6;">void</span> <span style="color: #f39c12;">setup</span> ()</div><div><span
style="color: #f39c12;">Serial</span>.<span style="color: #f39c12;">begin</span>(<span
style="color: #7fcbcd;">9600</span>);<span style="color: #7f8c8d;"> //Starts Serial
Communication at Baud Rate 115200</span></div><div><span style="color:
#c586c0;">for</span>(<span style="color: #0ca1a6;">int</span> i = <span style="color:
#7fcbcd;">0</span>; i &lt; <span style="color: #7fcbcd;">8</span>; i++) {</div><div><span
style="color: #f39c12;">pinMode</span>(<span style="color: #f39c12;">dip</span>[i],
INPUT_PULLUP);</div><div>}</div><div><span style="color: #f39c12;">SPI</span>.<span
style="color: #f39c12;">begin</span>();<span style="color: #7f8c8d;"> //Begins the SPI
commnuication</span></div><div><span style="color: #f39c12;">SPI</span>.<span
style="color: #f39c12;">setClockDivider</span>(SPI_CLOCK_DIV8);<span style="color:
#7f8c8d;"> //Sets clock for SPI communication at 8 (16/8=2Mhz)</span></div><div><span
style="color: #f39c12;">digitalWrite</span>(SS,HIGH);<span style="color: #7f8c8d;"> //
Setting SlaveSelect as HIGH (So master doesnt connect with
slave)</span></div><div>}</div><br /><div><span style="color: #0ca1a6;">void</span>
<span style="color: #f39c12;">loop</span>(<span style="color: #0ca1a6;">void</span>)<
/><div><div>&nbsp; byte Mastersend;</div><div>&nbsp; <span style="color:
#0ca1a6;">int</span> x = <span style="color: #7fcbcd;">1</span>;</div><div>&nbsp; <span
style="color: #0ca1a6;">int</span> activeSwitches = <span style="color:
#7fcbcd;">0</span>;<span style="color: #7f8c8d;"> // variabel untuk menghitung jumlah
saklar yang aktif</span></div><br /><div><span style="color: #7f8c8d;">&nbsp; // Membaca
status saklar dan menghitung berapa yang aktif</span></div><div>&nbsp; <span
style="color: #c586c0;">for</span> (<span style="color: #0ca1a6;">int</span> i = <span
style="color: #7fcbcd;">0</span>; i &lt; <span style="color: #7fcbcd;">8</span>; i++)
{</div><div>&nbsp; &nbsp; <span style="color: #f39c12;">dipvalue</span>[i] = <span
style="color: #f39c12;">digitalRead</span>(<span style="color:
#f39c12;">dip</span>[i]);</div><div>&nbsp; &nbsp; <span style="color: #c586c0;">if</span>
(<span style="color: #f39c12;">dipvalue</span>[i] == LOW) {</div><div>&nbsp; &nbsp; &nbsp;
&nbsp; x = <span style="color: #f39c12;">dip</span>[i];</div><div>&nbsp; &nbsp; &nbsp; &nbsp;
activeSwitches++;</div><div>&nbsp; &nbsp; &nbsp; }</div><div>&nbsp; &nbsp; }</div><br /><div>&nbsp;
<span style="color: #c586c0;">if</span> (activeSwitches == <span style="color:
#7fcbcd;">2</span>) {</div><div>&nbsp; &nbsp; &nbsp; &nbsp; <span style="color:
#f39c12;">digitalWrite</span>(SS, LOW);</div><div>&nbsp; &nbsp; &nbsp; &nbsp; Mastersend =
<span style="color: #7fcbcd;">2</span>;</div><div>&nbsp; &nbsp; &nbsp; &nbsp; }</div><div>&nbsp; &nbsp; &nbsp; &nbsp;
<span style="color: #f39c12;">Serial</span>.<span style="color:

```

```

#f39c12;">println</span>(Mastersend);</div><div>&nbsp; &nbsp; &nbsp; <span
style="color: #f39c12;">SPI</span>.<span style="color:
#f39c12;">transfer</span>(Mastersend);</div><div>&nbsp; &nbsp; &nbsp; <span
style="color: #f39c12;">delay</span>(<span style="color:
#7fcbcd;">100</span>);</div><div>&nbsp; &nbsp; &nbsp; }</div><div><br
/></div></div></div></div></span></div><div style="color: #222222; font-family:
Georgia, Utopia, &quot;Palatino Linotype&quot;, Palatino, serif; font-size: 15.4px;"><br
/></div><div style="color: #222222; font-family: Georgia, Utopia, &quot;Palatino
Linotype&quot;, Palatino, serif; font-size: 15.4px;"><b>Slave</b></div><div style="color:
#222222; font-family: Georgia, Utopia, &quot;Palatino Linotype&quot;, Palatino, serif;
font-size: 15.4px;"><div style="background-color: #1f272a; color: #dae3e3; font-family:
Consolas, &quot;Courier New&quot;, monospace; font-size: 14px; line-height: 19px;
white-space: pre;"><div><span style="color: #c586c0;">#include</span> <span style="color:
#7fcbcd;">&lt;SPI.h&gt;</span></div><br /><div><span style="color:
#0ca1a6;">const</span> <span style="color: #0ca1a6;">int</span> segmentPins[] = {<span
style="color: #7fcbcd;">9</span>, <span style="color: #7fcbcd;">8</span>, <span
style="color: #7fcbcd;">7</span>, <span style="color: #7fcbcd;">6</span>, <span
style="color: #7fcbcd;">5</span>, <span style="color: #7fcbcd;">4</span>, <span
style="color: #7fcbcd;">3</span>, <span style="color: #7fcbcd;">2</span>, <span
style="color: #7fcbcd;">1</span>, <span style="color:
#7fcbcd;">0</span>};</div><div><span style="color: #0ca1a6;">volatile</span> boolean
received = <span style="color: #7fcbcd;">false</span>;</div><div><span style="color:
#0ca1a6;">volatile</span> byte Slaverceived;</div><div><span style="color:
#0ca1a6;">int</span> index;</div><div><span style="color: #0ca1a6;">int</span> dg1 =
A0;</div><div><span style="color: #0ca1a6;">int</span> dg2 = A1;</div><br /><div><span
style="color: #0ca1a6;">void</span> <span style="color:
#f39c12;">displayCharacter</span>(<span style="color: #0ca1a6;">int</span> ch)
{</div><div>&nbsp; byte <span style="color: #f39c12;">patterns</span>[<span style="color:
#7fcbcd;">10</span>][<span style="color: #7fcbcd;">7</span>] = {</div><div>&nbsp;
&nbsp; {<span style="color: #7fcbcd;">0</span>, <span style="color: #7fcbcd;">0</span>,
<span style="color: #7fcbcd;">0</span>, <span style="color: #7fcbcd;">0</span>, <span
style="color: #7fcbcd;">0</span>, <span style="color: #7fcbcd;">0</span>, <span
style="color: #7fcbcd;">1</span>},<span style="color: #7f8c8d;"> //
0</span></div><div>&nbsp; &nbsp; {<span style="color: #7fcbcd;">1</span>, <span
style="color: #7fcbcd;">0</span>, <span style="color: #7fcbcd;">0</span>, <span
style="color: #7fcbcd;">1</span>, <span style="color: #7fcbcd;">1</span>, <span
style="color: #7fcbcd;">1</span>, <span style="color: #7fcbcd;">1</span>},<span
style="color: #7f8c8d;"> // 1</span></div><div>&nbsp; &nbsp; {<span style="color:
#7fcbcd;">0</span>, <span style="color: #7fcbcd;">0</span>, <span style="color:
#7fcbcd;">1</span>, <span style="color: #7fcbcd;">0</span>, <span style="color:
#7fcbcd;">0</span>, <span style="color: #7fcbcd;">1</span>, <span style="color:
#7fcbcd;">0</span>},<span style="color: #7f8c8d;"> // 2</span></div><div>&nbsp; &nbsp;
{<span style="color: #7fcbcd;">0</span>, <span style="color: #7fcbcd;">0</span>, <span
style="color: #7fcbcd;">0</span>, <span style="color: #7fcbcd;">0</span>, <span
style="color: #7fcbcd;">1</span>, <span style="color: #7fcbcd;">1</span>, <span
style="color: #7fcbcd;">0</span>},<span style="color: #7f8c8d;"> //
3</span></div><div>&nbsp; &nbsp; {<span style="color: #7fcbcd;">1</span>, <span
style="color: #7fcbcd;">0</span>, <span style="color: #7fcbcd;">0</span>, <span
style="color: #7fcbcd;">0</span>, <span style="color: #7fcbcd;">1</span>, <span
style="color: #7fcbcd;">1</span>, <span style="color: #7fcbcd;">1</span>},<span
style="color: #7f8c8d;"> //

```

```

style="color: #7fcbcd;">1</span>, <span style="color: #7fcbcd;">1</span>, <span
style="color: #7fcbcd;">0</span>, <span style="color: #7fcbcd;">0</span>},<span
style="color: #7f8c8d;"> // 4</span></div><div>&nbsp; &nbsp; {<span style="color:
#7fcbcd;">0</span>, <span style="color: #7fcbcd;">1</span>, <span style="color:
#7fcbcd;">0</span>, <span style="color: #7fcbcd;">0</span>, <span style="color:
#7fcbcd;">1</span>, <span style="color: #7fcbcd;">0</span>, <span style="color:
#7fcbcd;">0</span>},<span style="color: #7f8c8d;"> // 5</span></div><div>&nbsp; &nbsp;
{<span style="color: #7fcbcd;">0</span>, <span style="color: #7fcbcd;">1</span>, <span
style="color: #7fcbcd;">0</span>, <span style="color: #7fcbcd;">0</span>, <span
style="color: #7fcbcd;">0</span>, <span style="color: #7fcbcd;">0</span>, <span
style="color: #7fcbcd;">0</span>},<span style="color: #7f8c8d;"> //
6</span></div><div>&nbsp; &nbsp; {<span style="color: #7fcbcd;">0</span>, <span
style="color: #7fcbcd;">0</span>, <span style="color: #7fcbcd;">0</span>, <span
style="color: #7fcbcd;">1</span>, <span style="color: #7fcbcd;">1</span>, <span
style="color: #7fcbcd;">1</span>, <span style="color: #7fcbcd;">1</span>},<span
style="color: #7f8c8d;"> // 7</span></div><div>&nbsp; &nbsp; {<span style="color:
#7fcbcd;">0</span>, <span style="color: #7fcbcd;">0</span>, <span style="color:
#7fcbcd;">0</span>, <span style="color: #7fcbcd;">0</span>, <span style="color:
#7fcbcd;">0</span>, <span style="color: #7fcbcd;">0</span>, <span style="color:
#7fcbcd;">0</span>},<span style="color: #7f8c8d;"> // 8</span></div><div>&nbsp; &nbsp;
{<span style="color: #7fcbcd;">0</span>, <span style="color: #7fcbcd;">0</span>, <span
style="color: #7fcbcd;">0</span>, <span style="color: #7fcbcd;">0</span>, <span
style="color: #7fcbcd;">1</span>, <span style="color: #7fcbcd;">0</span>, <span
style="color: #7fcbcd;">0</span>}<span style="color: #7f8c8d;"> &nbsp; &nbsp; //
9</span></div><div>&nbsp; &nbsp; };</div><br /><div>&nbsp; &nbsp; <span style="color:
#c586c0;">if</span> ((ch &gt;= <span style="color: #7fcbcd;">0</span> & & ch
&lt;= <span style="color: #7fcbcd;">9</span>)) {</div><div><span style="color:
#7f8c8d;">&nbsp; &nbsp; &nbsp; // Get the digit index (0-9) from the
character</span></div><div>&nbsp; &nbsp; <span style="color: #0ca1a6;">int</span> index
= ch;</div><div><span style="color: #7f8c8d;">&nbsp; &nbsp; &nbsp; // Write the pattern to the
segment pins</span></div><div>&nbsp; &nbsp; <span style="color: #c586c0;">for</span>
(<span style="color: #0ca1a6;">int</span> i = <span style="color: #7fcbcd;">0</span>; i &lt;
<span style="color: #7fcbcd;">7</span>; i++) {</div><div>&nbsp; &nbsp; &nbsp; <span
style="color: #f39c12;">digitalWrite</span>(<span style="color:
#f39c12;">segmentPins</span>[i], <span style="color:
#f39c12;">patterns</span>[index][i]);</div><div>&nbsp; &nbsp; &nbsp; }</div><div>&nbsp; &nbsp;
}</div><div></div><br /><div><span style="color: #0ca1a6;">void</span> <span
style="color: #f39c12;">setup</span>() {</div><div>&nbsp; &nbsp; <span style="color:
#f39c12;">Serial</span>.<span style="color: #f39c12;">begin</span>(<span style="color:
#7fcbcd;">9600</span>);</div><div>&nbsp; &nbsp; <span style="color: #c586c0;">for</span>
(<span style="color: #0ca1a6;">int</span> i = <span style="color: #7fcbcd;">0</span>; i &lt;
<span style="color: #7fcbcd;">8</span>; i++) {</div><div>&nbsp; &nbsp; <span style="color:
#f39c12;">pinMode</span>(<span style="color: #f39c12;">segmentPins</span>[i],
OUTPUT);</div><div>&nbsp; &nbsp; }</div><div>&nbsp; &nbsp; <span style="color:
#f39c12;">pinMode</span>(dg1, OUTPUT);</div><div>&nbsp; &nbsp; <span style="color:
#f39c12;">pinMode</span>(dg2, OUTPUT);</div><div>&nbsp; &nbsp; SPCR |= <span style="color:
#f39c12;">_BV</span>(SPE);<span style="color: #7f8c8d;"> // Turn on SPI in Slave
Mode</span></div><div>&nbsp; &nbsp; <span style="color: #f39c12;">SPI</span>.<span

```

```

style="color: #f39c12;">attachInterrupt</span>());<span style="color: #7f8c8d;"> // Interrupt
ON is set for SPI communication</span></div><div></div><br /><div><span style="color:
#f39c12;">ISR</span> (SPI_STC_vect) {<span style="color: #7f8c8d;"> // Interrupt routine
function</span></div><div>&nbsp; Slavereceived = SPDR;<span style="color: #7f8c8d;"> //
Value received from master is stored in variable Slavereceived</span></div><div>&nbsp;
received = <span style="color: #7fcbcd;">true</span>;<span style="color: #7f8c8d;"> // Sets
received as True</span></div><div></div><br /><div><span style="color:
#0ca1a6;">void</span> <span style="color: #f39c12;">loop</span>() {</div><div>&nbsp;
<span style="color: #c586c0;">if</span> (received) {<span style="color: #7f8c8d;"> // Logic
to SET LED ON OR OFF depending upon the value received from
master</span></div><div>&nbsp; &nbsp; <span style="color:
#f39c12;">Serial</span>.<span style="color:
#f39c12;">println</span>(Slavereceived);</div><div>&nbsp; &nbsp; <span style="color:
#c586c0;">if</span> (Slavereceived == <span style="color: #7fcbcd;">2</span></div>
{</div><div>&nbsp; &nbsp; &nbsp; <span style="color: #c586c0;">for</span>(<span
style="color: #0ca1a6;">int</span> i=<span style="color: #7fcbcd;">0</span>;i<span style="color:
#7fcbcd;">9</span>;i++){</div><div>&nbsp; &nbsp; &nbsp; <span style="color: #c586c0;">if</span>(i % <span style="color: #7fcbcd;">2</span> == <span
style="color: #7fcbcd;">0</span> )</div><div>&nbsp; &nbsp; &nbsp; &nbsp; <span
style="color: #f39c12;">digitalWrite</span>(dg1,HIGH);</div><div>&nbsp; &nbsp; &nbsp;
&nbsp; &nbsp; <span style="color:
#f39c12;">digitalWrite</span>(dg2,LOW);</div><div>&nbsp; &nbsp; &nbsp; &nbsp; &nbsp;
<span style="color: #f39c12;">displayCharacter</span>(i);</div><div>&nbsp; &nbsp; &nbsp;
&nbsp; &nbsp; <span style="color: #f39c12;">delay</span> (<span style="color:
#7fcbcd;">100</span>);</div><div>&nbsp; &nbsp; &nbsp; &nbsp; }</div><div>&nbsp;
&nbsp; &nbsp; <span style="color: #c586c0;">else</span>{</div><div>&nbsp;
&nbsp; &nbsp; &nbsp; <span style="color: #f39c12;">digitalWrite</span>(dg1,
LOW);</div><div>&nbsp; &nbsp; &nbsp; &nbsp; <span style="color:
#f39c12;">digitalWrite</span>(dg2, HIGH);</div><div>&nbsp; &nbsp; &nbsp; &nbsp;
<span style="color: #f39c12;">displayCharacter</span>(i);</div><div>&nbsp; &nbsp; &nbsp;
&nbsp; &nbsp; <span style="color: #f39c12;">delay</span> (<span style="color:
#7fcbcd;">100</span>);</div><div>&nbsp; &nbsp; &nbsp; &nbsp; }</div><div>&nbsp;
&nbsp; &nbsp; }</div><div>&nbsp; &nbsp; }</div><div>&nbsp; &nbsp; received = <span
style="color: #7fcbcd;">false</span>;<span style="color: #7f8c8d;"> // Reset received
flag</span></div><div>&nbsp; }</div><div></div><br /></div></div><div style="color:
#222222; font-family: Georgia, Utopia, &quot;Palatino Linotype&quot;, Palatino, serif;
font-size: 15.4px;"><br /></div><div style="color: #222222; font-family: Georgia, Utopia,
&quot;Palatino Linotype&quot;, Palatino, serif; font-size: 15.4px;"><span style="font-family:
inherit; font-weight: bold;">b. Flowchart</span></div><div style="color: #222222; font-family:
Georgia, Utopia, &quot;Palatino Linotype&quot;, Palatino, serif; font-size: 15.4px;"><span
style="font-family: inherit; font-weight: bold;"><br /></span></div><div style="color: #222222;
font-family: Georgia, Utopia, &quot;Palatino Linotype&quot;, Palatino, serif; font-size:
15.4px;"><span style="font-family: inherit; font-weight: bold;">Arduino
Master</span></div><div style="color: #222222; font-family: Georgia, Utopia, &quot;Palatino
Linotype&quot;, Palatino, serif; font-size: 15.4px;"><span style="font-family: inherit;
font-weight: bold;"><br /></span></div><div style="color: #222222; font-family: Georgia,
Utopia, &quot;Palatino Linotype&quot;, Palatino, serif; font-size: 15.4px;"><span
style="font-family: inherit; font-weight: bold;"><div class="separator" style="clear: both;

```

[illegible]

switch pull-down dan 4 switch pull up mengaktifkan led,4 switch pull down dan 1 switch pull up mengaktifkan blinking led</div><div style="color: #222222; font-family: Georgia, Utopia, "Palatino Linotype", Palatino, serif; font-size: 15.4px;">6.<b style="color: #38761d;">Simulasi[Kembali]</div><div style="color: #222222; font-family: Georgia, Utopia, "Palatino Linotype", Palatino, serif; font-size: 15.4px;">
</div><div style="color: #222222; font-family: Georgia, Utopia, "Palatino Linotype", Palatino, serif; font-size: 15.4px;"><div class="separator" style="clear: both; text-align: center;"><object class="BLOG_video_class" contentid="8a3585b4581070a4" height="266" id="BLOG_video-8a3585b4581070a4" width="320"></object></div>

<div class="separator" style="clear: both; text-align: center;"> </div></div><div style="color: #222222; font-family: Georgia, Utopia, "Palatino Linotype", Palatino, serif; font-size: 15.4px;">7. Download File[Kembali]</div><div style="color: #222222; font-family: Georgia, Utopia, "Palatino Linotype", Palatino, serif; font-size: 15.4px;"><div>Download HMTLKlik disini</div><div>Download Simulasi RangkaianKlik disini</div><div>Download Program Arduino MasterKlik disini</div><div>Download Program Arduino SlaveKlik disini</div><div>Download Video SimulasiKlik disini</div><div>Download Datasheet ARDUINO UNOklik disini</div><div>Download Datasheet Dipswithklik disini</div><div>Download Datasheet Resistorklik disini</div><div>Download Datasheet LEDklik disini</div>

[illegible]