

DT-AVR *Application Note*

AN229 – Hand-made Touch and Slide Sensor

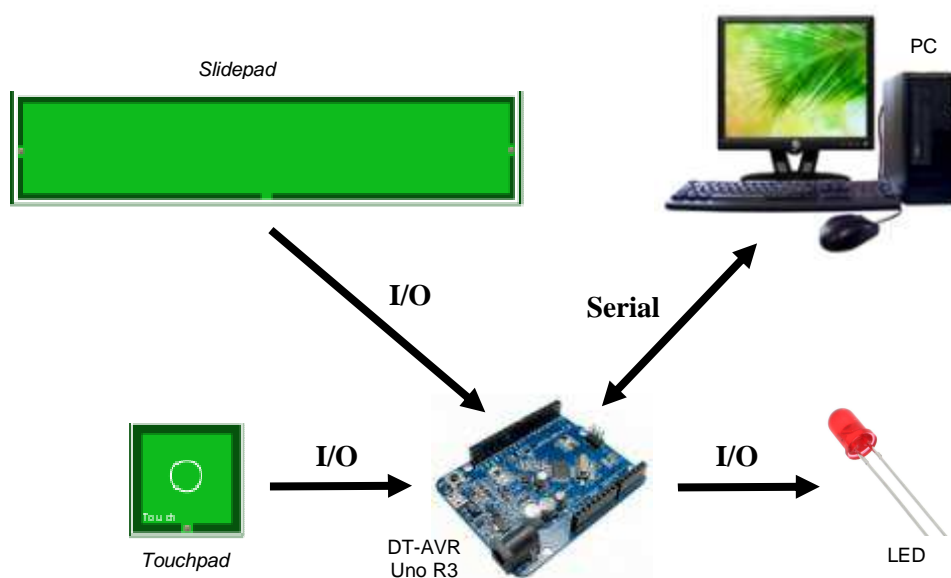
Oleh: Tim IE

Akhir-akhir ini kemajuan teknologi didominasi oleh perangkat keras layar sentuh yaitu ponsel pintar/*smartphone*. Munculnya teknologi tersebut secara signifikan telah memberikan sentuhan baru bagi dunia perangkat keras. Tak hanya *smartphone*, teknologi tersebut juga telah merambah pada industri manufaktur lainnya sebagai salah satu fitur antarmuka yang interaktif. Meski tergolong ke dalam level teknologi *high-end*, namun bukan berarti tidak dapat dibuat tiruannya. Bagi *developer* perangkat keras yang bermodalan oprek atau yang lebih kerennya dikenal dengan istilah *DIY (Do It Yourself)*, dengan melakukan sedikit trik dan modifikasi, membuat sensor sentuh bukanlah hal yang tidak mungkin. Oleh karena itu, pada aplikasi ini akan dikembangkan prototipe sederhana dari sensor sentuh (*touch sensor*) dan sensor geser (*slide sensor*).

Konsep dasar dari sensor ini sebenarnya meniru prinsip kerja dari kapasitor pada umumnya, yaitu proses *charge-discharge* muatan listrik. Yang mana, pada aplikasi ini, tubuh kita menjadi perumpamaan kapasitor itu sendiri. Sebab, tanpa kita sadari sebetulnya tubuh kita mampu menghantarkan listrik meski dalam rentang voltase yang sangat rendah. Dengan demikian, dapat dikatakan bahwa sensor ini bekerja berdasarkan prinsip kapasitif. Untuk membuat sensor ini tidaklah sulit, yang diperlukan hanyalah beberapa resistor dan plat tipis berbahan logam saja. Segala jenis logam yang mampu menghantarkan arus listrik bisa digunakan. Namun untuk mempermudah, lembaran kertas aluminium (*aluminium foil*) atau PCB adalah material yang cocok karena mudah diperoleh di sekitar kita. Dalam pengembangannya, aplikasi ini memerlukan beberapa perangkat sebagai berikut:

- 1x DT-AVR Uno R3
- 4x Plat dengan ukuran 1cm x 1cm dari kertas aluminium atau PCB untuk *Touchpad*
- 1x Plat dengan ukuran 6cm x 1cm dari kertas aluminium atau PCB untuk *Slidepad*
- 8x Resistor 4.7 MOhm
- 6x Resistor 330 Ohm
- 6x LED warna merah
- Kabel data micro-USB
- Kabel jumper/konektor secukupnya

Adapun blok diagram dari aplikasi ini adalah sebagai berikut:



Gambar 1
Blok Diagram Aplikasi AN229

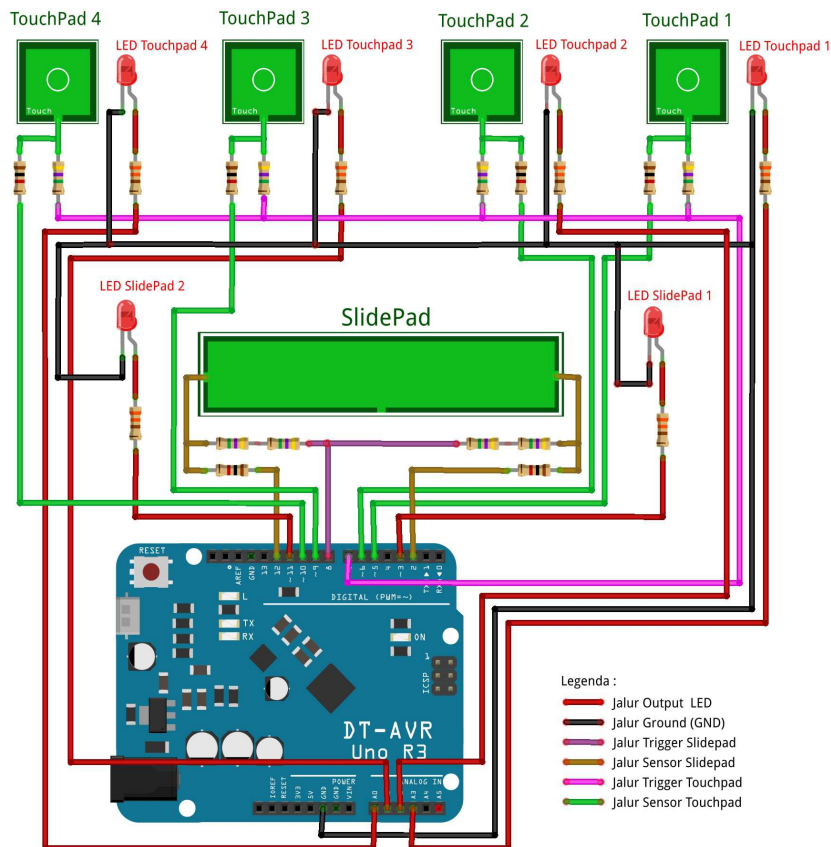
Hubungan antara DT-AVR Uno R3 dengan *Touchpad*, *Slidepad*, dan LED indikator adalah sebagai berikut:

DT-AVR Uno R3	<i>Touchpad</i>	<i>Slidepad</i>	LED Indikator
VCC (J4)	-	-	-
GND (J4)	-	-	Ground (GND)
Reset (J4)	-	-	-
Pin 0 (J2)		-	-
Pin 1 (J2)	-	-	-
Pin 2 (J2)	-	Sensor <i>Slidepad</i> 1	-
Pin 3 (J2)	-	-	LED indikator <i>Slidepad</i> 1
Pin 2 (J2)			
Pin 5 (J2)	Sensor <i>Touchpad</i> 1	-	-
Pin 6 (J2)	Sensor <i>Touchpad</i> 2	-	-
Pin 7 (J2)	Trigger <i>Touchpad</i>	-	-
Pin 8 (J2)	-	Trigger <i>Slidepad</i>	-
Pin 9 (J1)	Sensor <i>Touchpad</i> 3	-	-
Pin 10 (J1)	Sensor <i>Touchpad</i> 4	-	-
Pin 11 (J1)	-	-	LED indikator <i>Slidepad</i> 2
Pin 12 (J1)	-	Sensor <i>Slidepad</i> 2	-
Pin 13 (J1)	-	-	-
Pin A0 (J5)	-	-	LED indikator <i>Touchpad</i> 4
Pin A1 (J5)	-	-	LED indikator <i>Touchpad</i> 3
Pin A2 (J5)	-	-	LED indikator <i>Touchpad</i> 2
Pin A3 (J5)	-	-	LED indikator <i>Touchpad</i> 1
Pin A4 (J5)	-	-	-
Pin A5 (J5)	-	-	-

* pin ini tidak mutlak dan dapat diganti pin lain tetapi harus mengubah konfigurasi pin pada program

Tabel 1
Hubungan DT-AVR Uno R3 dengan *Touchpad*, *Slidepad*, dan LED Indikator

Hal pertama yang perlu dilakukan adalah melakukan koneksi antara *Touchpad*, *Slidepad*, dan LED indikator dengan DT-AVR Uno R3 seperti pada **Tabel 1**. Ilustrasi koneksi tersebut adalah seperti pada **Gambar 2**.

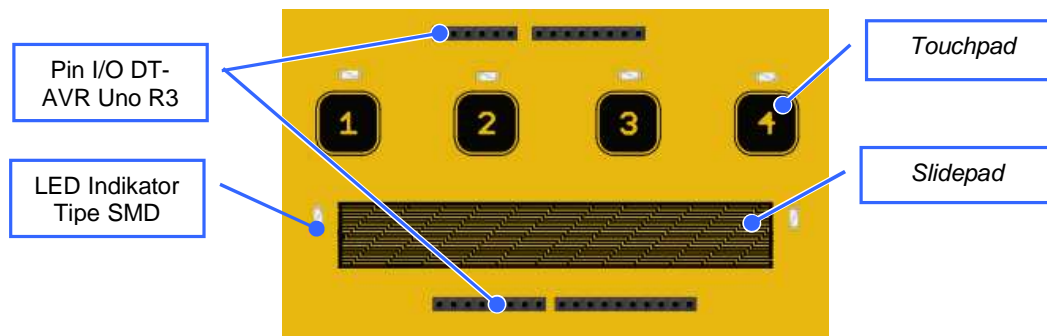


* Ilustrasi gambar dibuat melalui aplikasi [Fritzing](#)

Gambar 2
Ilustrasi Koneksi antara *Touchpad*, *Slidepad*, dan LED Indikator dengan DT-AVR Uno R3


Agar lebih memudahkan, seperti yang terlihat pada **Gambar 2**, istilah sensor sentuh disebut **Touchpad** sedangkan sensor geser disebut **Slidepad**. Kedua istilah tersebut akan sering dijumpai baik pada *application note* ini maupun pada kode program. Pada aplikasi ini akan dibuat 4 buah *Touchpad* dan 1 buah *Slidepad*. Tiap *Touchpad* memerlukan 2 buah resistor yang masing-masing nilainya 1 kOhm dan 4,7 MOhm. Sedangkan *Slidepad* memerlukan resistor 1 kOhm dan 9,4 MOhm (2 x 4,7 MOhm dipasang seri). Kedua resistor tersebut terhubung pada plat sensor dan pin DT-AVR Uno R3. Satu pin berfungsi sebagai *trigger* dan pin lainnya sebagai *input* sensor. Untuk memperjelas, masing-masing pin *trigger* dan pin sensor diberi nama sesuai jenisnya. Pada aplikasi ini juga memakai LED sebagai indikator. Untuk lebih jelasnya dapat dilihat pada **Gambar 2 & Gambar 3**.

Untuk bahan sensornya digunakan plat tembaga dari PCB. PCB tersebut didesain menggunakan aplikasi *Eagle* membentuk 4 buah *Touchpad* dan 1 buah *Slidepad* lalu di-*etching*. Tata peletakkannya juga disesuaikan dengan pin-pin I/O dari DT-AVR Uno R3 sehingga bisa membentuk modul/*shield* yang kompatibel dengan pin Arduino. Dalam percobaannya, modul tersebut dipasang langsung di atas DT-AVR Uno R3. Bentuk desain modul/*shield* tersebut adalah seperti pada **Gambar 3**.

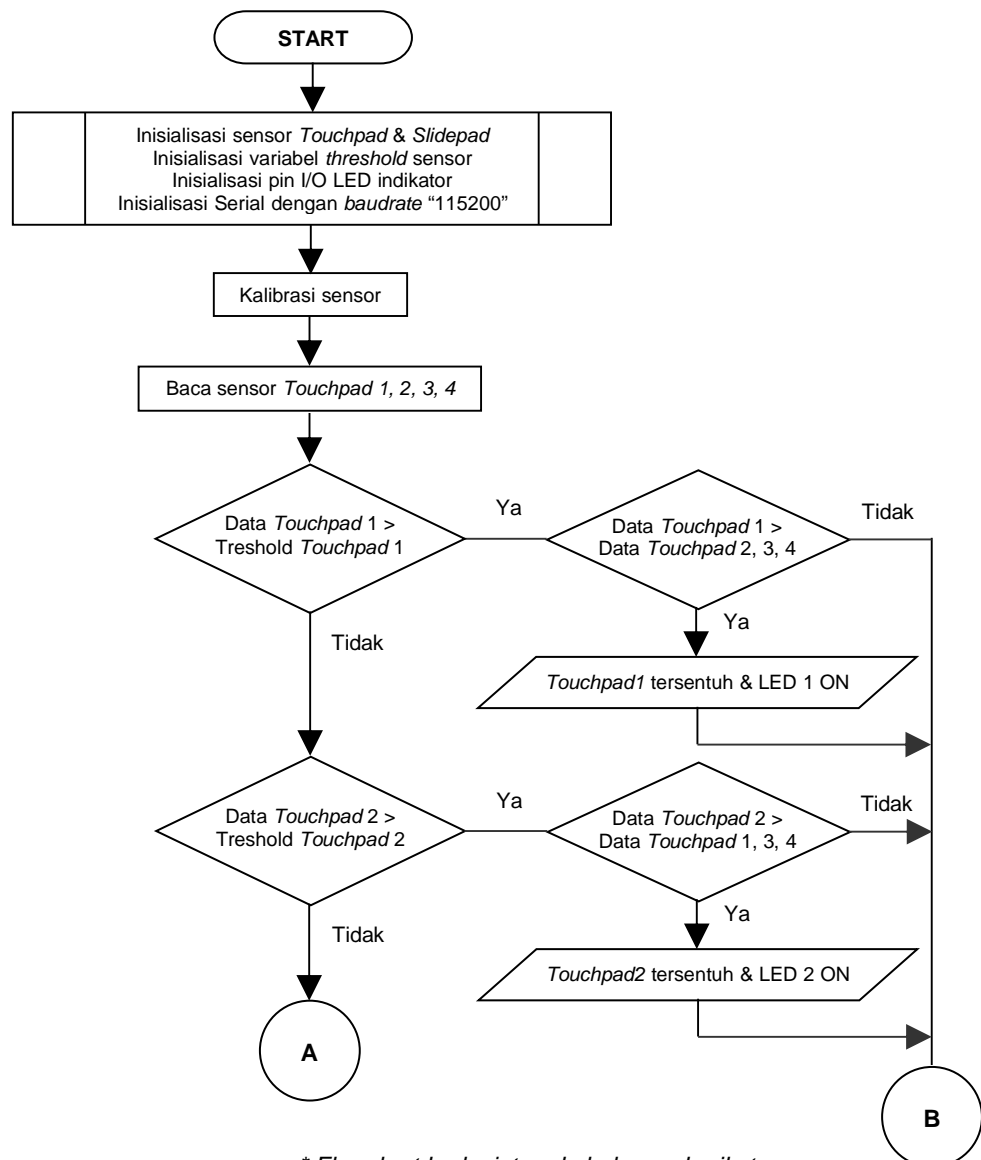


Gambar 3
Bentuk Desain dari Modul/*Shield* *Touchpad* dan *Slidepad*

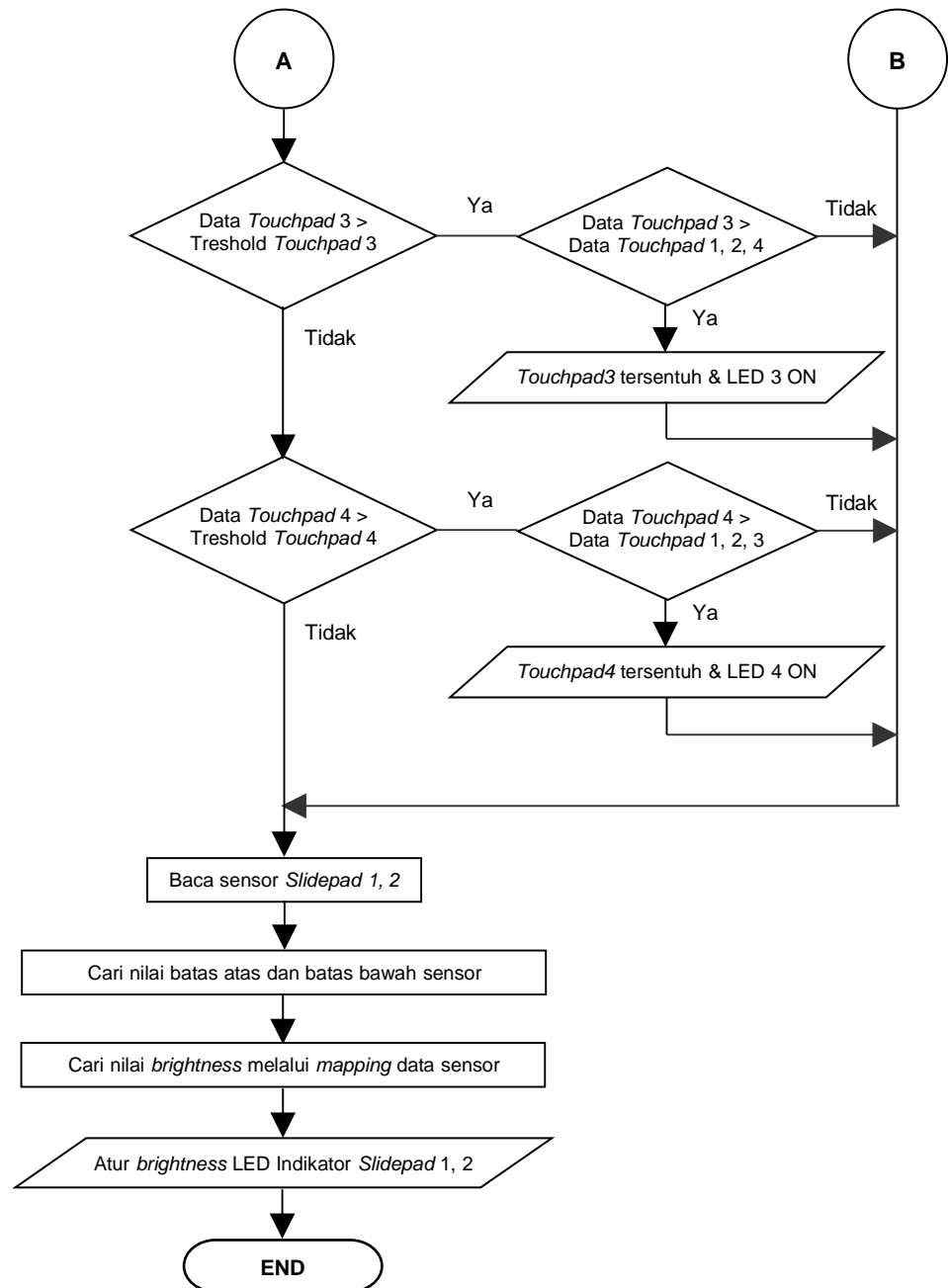
Tujuan dari aplikasi ini adalah bagaimana caranya membuat sensor kapasitif sederhana yang murah dan mampu difungsikan sebagai sensor sentuh dan sensor geser. Yang dimaksud *Slidepad* atau sensor geser disini adalah sensor yang cara kerjanya seperti *mousepad* laptop untuk navigasi arah horisontal atau vertikal. Untuk melakukan percobaan ini, ikuti langkah-langkahnya sebagai berikut:

1. *Etching* desain jalur PCB menggunakan referensi desain “/Eagle File/Shield TouchPad and SlidePad.brd”.
2. Pasang *shield* pada DT-AVR Uno R3 kemudian hubungkan dengan PC/Laptop menggunakan kabel USB.
3. Tambahkan *library CapacitiveSensor* pada folder *library* “/My documents/Arduino/libraries/”.
4. Jalankan program Arduino IDE.
5. Atur *board* yang akan digunakan dengan memilih **Tools – Board – Arduino Uno**.
6. Atur *port* serial yang akan digunakan pada **Tools – Port – COMx** (COMx disesuaikan dengan *port serial* DT-AVR Uno R3 yang muncul pada *device manager*).
7. Buka *file* program **AN229_Test.ino**. Lalu lakukan perintah *upload sketch* dengan menekan tombol  (Jika terjadi kendala saat proses *upload*, coba lakukan *troubleshooting* dengan bantuan yang merujuk pada laman <https://www.arduino.cc/en/guide/troubleshooting>).
8. Selanjutnya buka *terminal/serial monitor* dan atur *baudrate* pada nilai 115200.
9. Sentuh *Touchpad* secara bergantian kemudian amati LED indikator yang terdapat pada tiap *Touchpad*.
10. Sentuh dan geser *Slidepad* ke kanan dan ke kiri dan amati nyala LED indikator.

Flowchart sistem dari program “AN229_Test.ino” adalah sebagai berikut.



* Flowchart berlanjut pada halaman berikutnya



Gambar 4
Flowchart Program AN229_Test.ino

Penjelasan singkat mengenai *flowchart* tersebut adalah sebagai berikut:

1. Inisialisasi sensor *Touchpad* dan *Slidepad*. Inisialisasi ini bertujuan untuk melakukan konfigurasi pin DT-AVR Uno R3 agar bisa digunakan sebagai sensor *Touchpad* dan *Slidepad*.

```

// inisialisasi pin yg dipakai untuk TouchPad
const int pinTrigger_TouchPad = 7;
const int pinSensor_TouchPad1 = 5;
const int pinSensor_TouchPad2 = 6;
const int pinSensor_TouchPad3 = 9;
const int pinSensor_TouchPad4 = 10;
// inisialisasi pin yg dipakai untuk Slidepad
const int pinTrigger_SlidePad = 8;
const int pinSensor_SlidePad1 = 2;
const int pinSensor_SlidePad2 = 12;

```

```

// inisialisasi sensor capacitiveSensor
CapacitiveSensor TouchPad1 = CapacitiveSensor(pinTrigger_TouchPad, pinSensor_TouchPad1);
CapacitiveSensor TouchPad2 = CapacitiveSensor(pinTrigger_TouchPad, pinSensor_TouchPad2);
CapacitiveSensor TouchPad3 = CapacitiveSensor(pinTrigger_TouchPad, pinSensor_TouchPad3);
CapacitiveSensor TouchPad4 = CapacitiveSensor(pinTrigger_TouchPad, pinSensor_TouchPad4);
CapacitiveSensor SlidePad1 = CapacitiveSensor(pinTrigger_SlidePad, pinSensor_SlidePad1);
CapacitiveSensor SlidePad2 = CapacitiveSensor(pinTrigger_SlidePad, pinSensor_SlidePad2);

```

2. Inisialisasi variabel data sensor dan variabel *threshold* sensor. Variabel *threshold* di sini sangat penting, karena dipakai sebagai pembanding data sensor. Tanpa variabel ini kita akan kesulitan dalam menentukan apakah *Touchpad* tersebut dalam kondisi tersentuh atau tidak.

```

// deklarasi variabel TouchPad
long dataThreshold_TouchPad1, dataThreshold_TouchPad2,
    dataThreshold_TouchPad3, dataThreshold_TouchPad4;
long dataSensor_TouchPad1, dataSensor_TouchPad2,
    dataSensor_TouchPad3, dataSensor_TouchPad4;
// deklarasi variabel SlidePad
long dataSensor_SlidePad1, dataSensor_SlidePad2;
long a0=0, b0=0, a1=0, b1=0;

```

3. Inisialisasi pin I/O untuk LED indikator. Selain sensor, kita juga memerlukan LED sebagai indikator yang memudahkan kita dalam mengamati perubahan sensor yang kita sentuh. Pada aplikasi ini dipakai 6 buah LED dengan masing-masing 4 buah untuk *Touchpad* dan 2 buah lagi untuk *Slidepad*.

```

// inisialisasi pin untuk LED indikator
const int pinLED_TouchPad1 = A3;
const int pinLED_TouchPad2 = A2;
const int pinLED_TouchPad3 = A1;
const int pinLED_TouchPad4 = A0;
const int pinLED_SlidePad1 = 3;
const int pinLED_SlidePad2 = 11;
pinMode(pinLED_TouchPad1, OUTPUT); digitalWrite(pinLED_TouchPad1, LOW);
pinMode(pinLED_TouchPad2, OUTPUT); digitalWrite(pinLED_TouchPad2, LOW);
pinMode(pinLED_TouchPad3, OUTPUT); digitalWrite(pinLED_TouchPad3, LOW);
pinMode(pinLED_TouchPad4, OUTPUT); digitalWrite(pinLED_TouchPad4, LOW);

```

4. Inisialisasi Serial. Untuk mengetahui lebih detail tentang data dari masing-masing sensor, kita dapat memanfaatkan serial monitor. Sehingga akan lebih mudah untuk dipahami dan dipelajari.

```
Serial.begin(115200);
```

5. Baca data sensor *Touchpad1*, *Touchpad2*, *Touchpad3*, dan *Touchpad4*. Masing-masing data sensor tersebut akan dibaca oleh mikrokontroler dan dimasukkan ke dalam variabel *dataSensor_Touchpad**.

```

const int n_sampling = 10;
// pembacaan sensor Touchpad
dataSensor_TouchPad1 = TouchPad1.capacitiveSensor(n_sampling);
dataSensor_TouchPad2 = TouchPad2.capacitiveSensor(n_sampling);
dataSensor_TouchPad3 = TouchPad3.capacitiveSensor(n_sampling);
dataSensor_TouchPad4 = TouchPad4.capacitiveSensor(n_sampling);

```

6. Selanjutnya program akan menentukan apakah di antara keempat sensor tersebut ada yang tersentuh atau tidak. Data sensor pertama akan dibandingkan dengan data *threshold*-nya dan data sensor lainnya. Jika nilai datanya lebih besar dari data *threshold*-nya dan juga lebih besar dari data sensor lainnya maka dapat dipastikan bahwa sensor pertama (*Touchpad 1*) tersentuh oleh jari. Proses yang sama akan terus berlanjut mulai dari sensor *Touchpad 1* sampai sensor *Touchpad 4* dan akan selalu berulang dari awal.

```

// Pemrosesan data Touchpad untuk menentukan apakah ada Touchpad yang disentuh
if(dataSensor_TouchPad1 > dataThreshold_TouchPad1){
    if( dataSensor_TouchPad1 > dataSensor_TouchPad2
    && dataSensor_TouchPad1 > dataSensor_TouchPad3
    && dataSensor_TouchPad1 > dataSensor_TouchPad4 ){
        Serial.println("##### TouchPad1 Touched\r\n");
        digitalWrite(pinLED_TouchPad1, HIGH); // LED_TouchPad1 ON
    }
}

```

```

}
else if(dataSensor_TouchPad2 > dataThreshold_TouchPad2){
  if( dataSensor_TouchPad2 > dataSensor_TouchPad1
    && dataSensor_TouchPad2 > dataSensor_TouchPad3
    && dataSensor_TouchPad2 > dataSensor_TouchPad4 ){
    Serial.println("##### TouchPad2 Touched\r\n");
    digitalWrite(pinLED_TouchPad2, HIGH); // LED_TouchPad2 ON
  }
}
else if(dataSensor_TouchPad3 > dataThreshold_TouchPad3){
  if( dataSensor_TouchPad3 > dataSensor_TouchPad1
    && dataSensor_TouchPad3 > dataSensor_TouchPad2
    && dataSensor_TouchPad3 > dataSensor_TouchPad4 ){
    Serial.println("##### TouchPad3 Touched\r\n");
    digitalWrite(pinLED_TouchPad3, HIGH); // LED_TouchPad3 ON
  }
}
else if(dataSensor_TouchPad4 > dataThreshold_TouchPad4){
  if( dataSensor_TouchPad4 > dataSensor_TouchPad1
    && dataSensor_TouchPad4 > dataSensor_TouchPad2
    && dataSensor_TouchPad4 > dataSensor_TouchPad3 ){
    Serial.println("##### TouchPad4 Touched\r\n");
    digitalWrite(pinLED_TouchPad4, HIGH); // LED_TouchPad4 ON
  }
}
}
}

```

7. Baca data sensor *slidepad1* dan *slidepad2*.

```

// pembacaan sensor Slidepad
dataSensor_SlidePad1 = SlidePad1.capacitiveSensor(n_sampling);
dataSensor_SlidePad2 = SlidePad2.capacitiveSensor(n_sampling);

```

8. Mencari nilai batas atas dan batas batas bawah. Prinsipnya hampir sama dengan variabel *threshold* pada *Touchpad*, namun pada *Slidepad* terdapat dua jenis variabel batas, yaitu variabel *a0* dan *b0* untuk batas bawah dan variabel *a1* dan *b1* untuk batas atas. Variabel-variabel tersebut diperlukan sebagai batas *mapping* nilai *brightness* LED indikator untuk sensor *Slidepad*.

```

if(dataSensor_SlidePad1 <= a0) dataSensor_SlidePad1 = a0;
if(dataSensor_SlidePad2 <= b0) dataSensor_SlidePad2 = b0;
if(dataSensor_SlidePad1 > a1) a1 = dataSensor_SlidePad1;
if(dataSensor_SlidePad2 > b1) b1 = dataSensor_SlidePad2;

```

9. *Mapping* data sensor untuk mendapatkan nilai *brightness*. Fungsi *map()* pada cuplikan program berikut adalah mengubah nilai sensor *Slidepad* yang semula antara *a0* sampai *a1* dan *b0* sampai *b1* menjadi 0 sampai 255.

```

byte brightness_Value_1 = map(dataSensor_SlidePad1, a0, a1, 0, 255);
byte brightness_Value_2 = map(dataSensor_SlidePad2, b0, b1, 0, 255);

```

10. Mengatur *brightness* LED indikator dari *slidepad1* dan *slidepad2*.

```

analogWrite(pinLED_SlidePad1, brightness_Value_1);
analogWrite(pinLED_SlidePad2, brightness_Value_2);

```

Demikian penjelasan aplikasi ini, tentunya masih banyak lagi yang bisa dikembangkan dari *board DT-AVR Uno R3* ini.

Listing program aplikasi ini terdapat pada **AN229.ZIP**.

Selamat berinovasi!

*All trademarks, company names, product names and trade names are the property of their respective owners.
All softwares are copyright by their respective creators and/or software publishers.*