Listing program pengukur damage samsak tinju:

```
#include <LiquidCrystal_I2C.h>
#include <NewPing.h>

#define FSR1 A0
#define FSR2 A1
#define FSR3 A2
#define FSR4 A3
#define FSR5 A6

#define TRIG_1 2
#define ECHO_1 3
#define TRIG_2 4
#define ECHO_2 5
#define TRIG_3 6
#define ECHO_3 7
#define TRIG_4 8
#define ECHO_4 9

#define BUTTON_PIN_1 10
#define BUTTON_PIN_2 11
#define BUTTON_PIN_3 12

LiquidCrystal_I2C lcd(0x27, 20, 4);

NewPing sonar1(TRIG_1, ECHO_1, 200);
NewPing sonar2(TRIG_2, ECHO_2, 200);
NewPing sonar3(TRIG_3, ECHO_3, 200);
NewPing sonar4(TRIG_4, ECHO_4, 200);
```

```
double jarak_ultrasonik = 5;
double previousTime, currentTime, stopTime;

bool button1Active = false, button2Active = false,  button3Active = false;
bool hasilNilai = false;
bool HC1,HC2,HC3,HC4;

bool a=true;
int x = 0;
int pukulan = 1, tendangan = 1;

static double
PUKUL_KIRI=0,PUKUL_KANAN=0,TENDANG_KIRI=0,TENDANG_KANAN=0,TENDANG_TENGAH=0;
static double damage[3], speed[3];

void displayLCD(int level=0, int punch=0, int kick=0){
 if(level==1 || level==2 || level==3){
   lcd.setCursor(0,0); lcd.print("===== LEVEL "+String(level)+" =====");
   lcd.setCursor(0,1); lcd.print("Lakukan :          ");
   lcd.setCursor(0,2); lcd.print("    "+String(punch)+" Pukulan       ");
   lcd.setCursor(0,3); lcd.print("    "+String(kick)+" Tendangan      ");
 } else if(level==0){
   lcd.setCursor(0,0); lcd.print("===== UNKNOWN =====");
   lcd.setCursor(0,1); lcd.print("   Silahkan Pilih  ");
   lcd.setCursor(0,2); lcd.print("   Level Terlebih  ");
   lcd.setCursor(0,3); lcd.print("      Dahulu       ");
 } else if(level==10){
   lcd.setCursor(0,0); lcd.print("===== RESULTS =====");
   lcd.setCursor(0,1); lcd.print("1.dmg:"+String(damage[0])+" sp:"+String(speed[0])+"   ");
   lcd.setCursor(0,2); lcd.print("2.dmg:"+String(damage[1])+" sp:"+String(speed[1])+"   ");
   lcd.setCursor(0,3); lcd.print("3.dmg:"+String(damage[2])+" sp:"+String(speed[2])+"   ");
```

```
  }
}

void setup() {
  Serial.begin(115200);
  lcd.init(); lcd.backlight(); lcd.clear();

  pinMode(FSR1, INPUT); pinMode(FSR2, INPUT);
  pinMode(FSR3, INPUT); pinMode(FSR4, INPUT); pinMode(FSR5, INPUT);
  pinMode(BUTTON_PIN_1, INPUT_PULLUP);
  pinMode(BUTTON_PIN_2, INPUT_PULLUP);
  pinMode(BUTTON_PIN_3, INPUT_PULLUP);
  pinMode(13,OUTPUT);

  lcd.setCursor(0,0); lcd.print("                ");
  lcd.setCursor(0,1); lcd.print("     WELCOME     ");
  lcd.setCursor(0,2); lcd.print("                ");
  lcd.setCursor(0,3); lcd.print("                ");
  delay(1000);
}

void loop() {

  bool level1 = digitalRead(BUTTON_PIN_1);
  bool level2 = digitalRead(BUTTON_PIN_2);
  bool level3 = digitalRead(BUTTON_PIN_3);

  long DISTANCE_1 = sonar1.ping_cm();
  long DISTANCE_2 = sonar2.ping_cm();
  long DISTANCE_3 = sonar3.ping_cm();
  long DISTANCE_4 = sonar4.ping_cm();
```

```
double pukulKiri = analogRead(FSR1);
double pukulKanan = analogRead(FSR2);
double tendangKiri = analogRead(FSR3);
double tendangKanan  = analogRead(FSR4);
double tendangTengah = analogRead(FSR5);

pukulKiri = pukulKiri*100/1023;
pukulKanan = pukulKanan*100/1023;
tendangKiri = tendangKiri*100/1023;
tendangKanan = tendangKanan*100/1023;
tendangTengah = tendangTengah*100/1023;

Serial.print(DISTANCE_1); Serial.print(" ");
Serial.print(DISTANCE_2); Serial.print(" ");
Serial.print(DISTANCE_3); Serial.print(" ");
Serial.print(DISTANCE_4); Serial.print(" | ");

Serial.print(pukulKiri); Serial.print(" ");
Serial.print(pukulKanan); Serial.print(" ");
Serial.print(tendangKiri); Serial.print(" ");
Serial.print(tendangKanan); Serial.print(" ");
Serial.print(tendangTengah); Serial.println(" ");


if(!level1){
   button1Active = true;
   button2Active = false;
   button3Active = false;
   tendangan = 1;
   pukulan = 1;
```

```
    x=0;
    damage[0]=0; damage[1]=0; damage[2]=0;
    speed[0]=0; speed[1]=0; speed[2]=0;
  } else if(!level2){
    button1Active = false;
    button2Active = true;
    button3Active = false;
    tendangan = 1;
    pukulan = 2;
    x=0;
    damage[0]=0; damage[1]=0; damage[2]=0;
    speed[0]=0; speed[1]=0; speed[2]=0;
  } else if(!level3){
    button1Active = false;
    button2Active = false;
    button3Active = true;
    tendangan = 2;
    pukulan = 1;
    x=0;
    damage[0]=0; damage[1]=0; damage[2]=0;
    speed[0]=0; speed[1]=0; speed[2]=0;
  }

  if((DISTANCE_1!=0 && DISTANCE_1<=40) || (DISTANCE_2!=0 && DISTANCE_2<=40) ||
(DISTANCE_3!=0 && DISTANCE_3<=40) || (DISTANCE_4!=0 && DISTANCE_4<=40)){
    digitalWrite(13,HIGH);
    if(a){ previousTime = millis(); a=false;}
    else{}
  } else { digitalWrite(13,LOW); a=true;}

  if(pukulKiri>10){                    //Lakukan Pukulan di Kiri
```

```
  delay(11);
  currentTime = millis();
  if(pukulKiri>PUKUL_KIRI){
    PUKUL_KIRI = pukulKiri;
  } else if(pukulKiri<PUKUL_KIRI){
     if(pukulan != 0){
       damage[x] = PUKUL_KIRI;
       speed[x] = jarak_ultrasonik*1000/((currentTime-previousTime));
       x++;
    }
    delay(500);
    pukulan--;
  }
}
if(pukulKanan>10){                  // Lakukan tendangan di kanan
  delay(9);
  currentTime = millis();
  if(pukulKanan>PUKUL_KANAN){
    PUKUL_KANAN = pukulKanan;
  } else if(pukulKanan<PUKUL_KANAN){
     if(pukulan != 0){
       damage[x] = PUKUL_KANAN;
       speed[x] = jarak_ultrasonik*1000/((currentTime-previousTime));
       x++;
    }
    delay(500);
    pukulan--;
  }
}
if(tendangKanan>10){                  // Lakukan Tendangan di Kanan
  delay(9);
```

```
  currentTime = millis();
  if(tendangKanan>TENDANG_KANAN){
    TENDANG_KANAN = tendangKanan;
  } else if(tendangKanan<TENDANG_KANAN){
     if(tendangan != 0){
       damage[x] = TENDANG_KANAN;
       speed[x] = jarak_ultrasonik*1000/((currentTime-previousTime));
       x++;
    }
    delay(500);
    tendangan--;
  }
}
if(tendangKiri>10){                  // Lakukan tendangan di Kiri
  delay(11);
  currentTime = millis();
  if(tendangKiri>TENDANG_KIRI){
    TENDANG_KIRI = tendangKiri;
  } else if(tendangKiri<TENDANG_KIRI){
     if(pukulan != 0){
       damage[x] = TENDANG_KIRI;
       speed[x] = jarak_ultrasonik*1000/((currentTime-previousTime));
       x++;
    }
    delay(500);
    tendangan--;
  }
}
if(tendangTengah>10){                // Lakukan Tendangan di Tengah
  delay(10);
  currentTime = millis();
```

```
   if(tendangTengah>TENDANG_TENGAH){
     TENDANG_TENGAH = tendangTengah;
   } else if(tendangTengah<TENDANG_TENGAH){
      if(pukulan != 0){
        damage[x] = TENDANG_TENGAH;
        speed[x] = jarak_ultrasonik*1000/((currentTime-previousTime));
        x++;
     }
     delay(500);
     tendangan--;
   }
 }

x = (button1Active && x==2) ? 0:x ;
x = (button2Active && x==3) ? 0:x ;
x = (button3Active && x==3) ? 0:x ;

tendangan = (tendangan<=0) ? 0 : tendangan;
pukulan = (pukulan<=0) ? 0 : pukulan;

 if(tendangan<=0 && pukulan<=0){
   hasilNilai=true;
   button1Active=false;
   button2Active=false;
   button3Active=false;
 } else {
   hasilNilai=false;
 }

  if(button1Active){
     displayLCD(1, pukulan, tendangan);
```

```
  } else if(button2Active){
     displayLCD(2, pukulan, tendangan);
  } else if(button3Active){
     displayLCD(3, pukulan, tendangan);
  } else if(hasilNilai){
    displayLCD(10);
  } else {
    displayLCD(0);
  }

}
```

Penjelasan :

1. libarary untuk menggunakan LCD dan Sensor ultrasonik

```
#include <LiquidCrystal_I2C.h>
#include <NewPing.h>
```

2.Mendeklarasikan Pin-pin arduino nano yang digunakan

```
#define FSR1 A0
#define FSR2 A1
#define FSR3 A2
#define FSR4 A3
#define FSR5 A6

#define TRIG_1 2
#define ECHO_1 3
#define TRIG_2 4
#define ECHO_2 5
#define TRIG_3 6
#define ECHO_3 7
#define TRIG_4 8
#define ECHO_4 9

#define BUTTON_PIN_1 10
#define BUTTON_PIN_2 11
#define BUTTON_PIN_3 12
```

3. Inisialisasi LCD

```
LiquidCrystal_I2C lcd(0x27, 20, 4);
```

4. Inisialisi Sensor Ultrasonik

```
NewPing sonar1(TRIG_1, ECHO_1, 200);
NewPing sonar2(TRIG_2, ECHO_2, 200);
NewPing sonar3(TRIG_3, ECHO_3, 200);
NewPing sonar4(TRIG_4, ECHO_4, 200);
```

5. Deklarasi Variabel program yang diperlukan

```
double jarak_ultrasonik = 5;

double previousTime, currentTime, stopTime;


bool button1Active = false, button2Active = false,
button3Active = false;

bool hasilNilai = false;

bool HC1,HC2,HC3,HC4;


bool a=true;

int x = 0;

int pukulan = 1, tendangan = 1;


static double
PUKUL_KIRI=0,PUKUL_KANAN=0,TENDANG_KIRI=0,
TENDANG_KANAN=0,TENDANG_TENGAH=0;

static double damage[3], speed[3];
```

6. Fungsi menampilkan Tulisan Display LCD

```
void displayLCD(int level=0, int punch=0, int kick=0){
 if(level==1 || level==2 || level==3){
  lcd.setCursor(0,0); lcd.print("===== LEVEL "+String(level)+" =====");
  lcd.setCursor(0,1); lcd.print("Lakukan :          ");
  lcd.setCursor(0,2); lcd.print("   "+String(punch)+" Pukulan       ");
  lcd.setCursor(0,3); lcd.print("   "+String(kick)+" Tendangan      ");
 } else if(level==0){
  lcd.setCursor(0,0); lcd.print("===== UNKNOWN =====");
  lcd.setCursor(0,1); lcd.print("   Silahkan Pilih  ");
  lcd.setCursor(0,2); lcd.print("   Level Terlebih  ");
  lcd.setCursor(0,3); lcd.print("     Dahulu       ");
 } else if(level==10){
  lcd.setCursor(0,0); lcd.print("===== RESULTS =====");
  lcd.setCursor(0,1); lcd.print("1.dmg:"+String(damage[0])+" sp:"+String(speed[0])+"   ");
  lcd.setCursor(0,2); lcd.print("2.dmg:"+String(damage[1])+" sp:"+String(speed[1])+"   ");
  lcd.setCursor(0,3); lcd.print("3.dmg:"+String(damage[2])+" sp:"+String(speed[2])+"   ");
 }
}
```

7. Inisialisi Mode Pin pada Arduino

```
void setup() {
 Serial.begin(115200);
 lcd.init(); lcd.backlight(); lcd.clear();


 pinMode(FSR1, INPUT); pinMode(FSR2, INPUT);
 pinMode(FSR3, INPUT); pinMode(FSR4, INPUT); pinMode(FSR5, INPUT);
 pinMode(BUTTON_PIN_1, INPUT_PULLUP);
 pinMode(BUTTON_PIN_2, INPUT_PULLUP);
 pinMode(BUTTON_PIN_3, INPUT_PULLUP);
 pinMode(13,OUTPUT);
```

8. Inisialisasi Sistem dengan Menampilkan Tulisan Welcome Pada LCD

```
lcd.setCursor(0,0); lcd.print("                ");
 lcd.setCursor(0,1); lcd.print("     WELCOME     ");
 lcd.setCursor(0,2); lcd.print("                ");
 lcd.setCursor(0,3); lcd.print("                ");
 delay(1000);
```

void loop()

9. Membaca nilai pada button level

```
 bool level1 = digitalRead(BUTTON_PIN_1);
 bool level2 = digitalRead(BUTTON_PIN_2);
 bool level3 = digitalRead(BUTTON_PIN_3);
```

10. Membaca nilai Sonsor Ultrasonik

```
 long DISTANCE_1 = sonar1.ping_cm();
 long DISTANCE_2 = sonar2.ping_cm();
 long DISTANCE_3 = sonar3.ping_cm();
 long DISTANCE_4 = sonar4.ping_cm();
```

11. Membaca Nilai pukulan pada sensor Tekanan

```
double pukulKiri = analogRead(FSR1);
double pukulKanan = analogRead(FSR2);
double tendangKiri = analogRead(FSR3);
double tendangKanan  = analogRead(FSR4);
double tendangTengah = analogRead(FSR5);
```

12. kalibrasi rentang nilai pukulan pada sensor tekanan

```
pukulKiri = pukulKiri*100/1023;
pukulKanan = pukulKanan*100/1023;
tendangKiri = tendangKiri*100/1023;
tendangKanan = tendangKanan*100/1023;
tendangTengah = tendangTengah*100/1023;
```

13. Menampilkan nilai sensor pada serial monitor arduino IDE

```
Serial.print(DISTANCE_1); Serial.print(" ");
Serial.print(DISTANCE_2); Serial.print(" ");
Serial.print(DISTANCE_3); Serial.print(" ");
Serial.print(DISTANCE_4); Serial.print(" | ");

Serial.print(pukulKiri); Serial.print(" ");
Serial.print(pukulKanan); Serial.print(" ");
Serial.print(tendangKiri); Serial.print(" ");
Serial.print(tendangKanan); Serial.print(" ");
Serial.print(tendangTengah); Serial.println(" ");
```

14. jika tombol level 1 ditekan maka lakukan 1 pukulan dan 1 tendangan, jika tombol level 2 ditekan maka lakukan 2 pukulan dan 1 tendangan, jika tombol level 3 ditekan maka lakukan 1 pukulan dan 2 tendangan

```
if(!level1){
    button1Active = true;
    button2Active = false;
    button3Active = false;
    tendangan = 1;
    pukulan = 1;
    x=0;
    damage[0]=0; damage[1]=0; damage[2]=0;
    speed[0]=0; speed[1]=0; speed[2]=0;
} else if(!level2){
    button1Active = false;
    button2Active = true;
    button3Active = false;
    tendangan = 1;
    pukulan = 2;
    x=0;
    damage[0]=0; damage[1]=0; damage[2]=0;
    speed[0]=0; speed[1]=0; speed[2]=0;
} else if(!level3){
    button1Active = false;
    button2Active = false;
    button3Active = true;
    tendangan = 2;
    pukulan = 1;
    x=0;
    damage[0]=0; damage[1]=0; damage[2]=0;
    speed[0]=0; speed[1]=0; speed[2]=0;
}
```

15.Jika terdeteksi jarak 40 cm maka akan menghasilkan indikator bunyi pada buzzer

```
 if((DISTANCE_1!=0 && DISTANCE_1<=40) || (DISTANCE_2!=0 && DISTANCE_2<=40) ||
(DISTANCE_3!=0 && DISTANCE_3<=40) || (DISTANCE_4!=0 && DISTANCE_4<=40)){
   digitalWrite(13,HIGH);
    if(a){ previousTime = millis(); a=false;}
    else{}
 } else { digitalWrite(13,LOW); a=true;}
```

16. Menghitung nilai pukulan pada samsak kiri

```
 if(pukulKiri>10){                  //Lakukan Pukulan di Kiri
  delay(11);
  currentTime = millis();
  if(pukulKiri>PUKUL_KIRI){
   PUKUL_KIRI = pukulKiri;
  } else if(pukulKiri<PUKUL_KIRI){
     if(pukulan != 0){
       damage[x] = PUKUL_KIRI;
       speed[x] = jarak_ultrasonik*1000/((currentTime-previousTime));
       x++;
    }
    delay(500);
    pukulan--;
  }
 }
```

17. Menghitung nilai Pukulan pada samsak kanan

```
if(pukulKanan>10){                  // Lakukan Pukulan di kanan
   delay(9);
   currentTime = millis();
   if(pukulKanan>PUKUL_KANAN){
     PUKUL_KANAN = pukulKanan;
   } else if(pukulKanan<PUKUL_KANAN){
      if(pukulan != 0){
        damage[x] = PUKUL_KANAN;
        speed[x] = jarak_ultrasonik*1000/((currentTime-previousTime));
        x++;
     }
     delay(500);
     pukulan--;
   }
 }
```

18. Menghitung nilai tendangan pada samsak kanan

```
if(tendangKanan>10){                    // Lakukan Tendangan di Kanan
  delay(9);
  currentTime = millis();
  if(tendangKanan>TENDANG_KANAN){
    TENDANG_KANAN = tendangKanan;
  } else if(tendangKanan<TENDANG_KANAN){
     if(tendangan != 0){
       damage[x] = TENDANG_KANAN;
       speed[x] = jarak_ultrasonik*1000/((currentTime-previousTime));
       x++;
    }
    delay(500);
    tendangan--;
  }
}
```

19. Menghitung nilai tendangan pada samsak kiri

```
if(tendangKiri>10){                  // Lakukan tendangan di Kiri
  delay(11);
  currentTime = millis();
  if(tendangKiri>TENDANG_KIRI){
    TENDANG_KIRI = tendangKiri;
  } else if(tendangKiri<TENDANG_KIRI){
     if(pukulan != 0){
       damage[x] = TENDANG_KIRI;
       speed[x] = jarak_ultrasonik*1000/((currentTime-previousTime));
       x++;
    }
    delay(500);
    tendangan--;
  }
}
```

20. Menghitung nilai tendangan pada samsak tengah

```
if(tendangTengah>10){                // Lakukan Tendangan di Tengah
  delay(10);
  currentTime = millis();
  if(tendangTengah>TENDANG_TENGAH){
    TENDANG_TENGAH = tendangTengah;
  } else if(tendangTengah<TENDANG_TENGAH){
     if(pukulan != 0){
       damage[x] = TENDANG_TENGAH;
       speed[x] = jarak_ultrasonik*1000/((currentTime-previousTime));
       x++;
    }
    delay(500);
    tendangan--;
  }
}
```

21. reset nilai pukulan dan tendangan jika syarat level terpenuhi

```
 if(tendangTengah>10){                // Lakukan Tendangan di Tengah
  delay(10);
  currentTime = millis();
  if(tendangTengah>TENDANG_TENGAH){
   TENDANG_TENGAH = tendangTengah;
  } else if(tendangTengah<TENDANG_TENGAH){
     if(pukulan != 0){
      damage[x] = TENDANG_TENGAH;
      speed[x] = jarak_ultrasonik*1000/((currentTime-previousTime));
      x++;
    }
    delay(500);
    tendangan--;
  }
 }


if(tendangan<=0 && pukulan<=0){
  hasilNilai=true;
  button1Active=false;
  button2Active=false;
  button3Active=false;
 } else {
  hasilNilai=false;
 }
```

22. menampilkan Hasil nilai damage pukulan dan tendangan pada LCD

```
if(button1Active){
   displayLCD(1, pukulan, tendangan);
} else if(button2Active){
   displayLCD(2, pukulan, tendangan);
} else if(button3Active){
   displayLCD(3, pukulan, tendangan);
} else if(hasilNilai){
  displayLCD(10);
} else {
  displayLCD(0);
}
```