

CARA MENJALANKAN PROGRAM

1. Buka program IDE Arduino



The screenshot shows the Arduino IDE interface. The title bar says "sketch_dec01a | Arduino 1.8.6". The menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar has icons for upload, download, and other functions. The code editor window contains the following code:

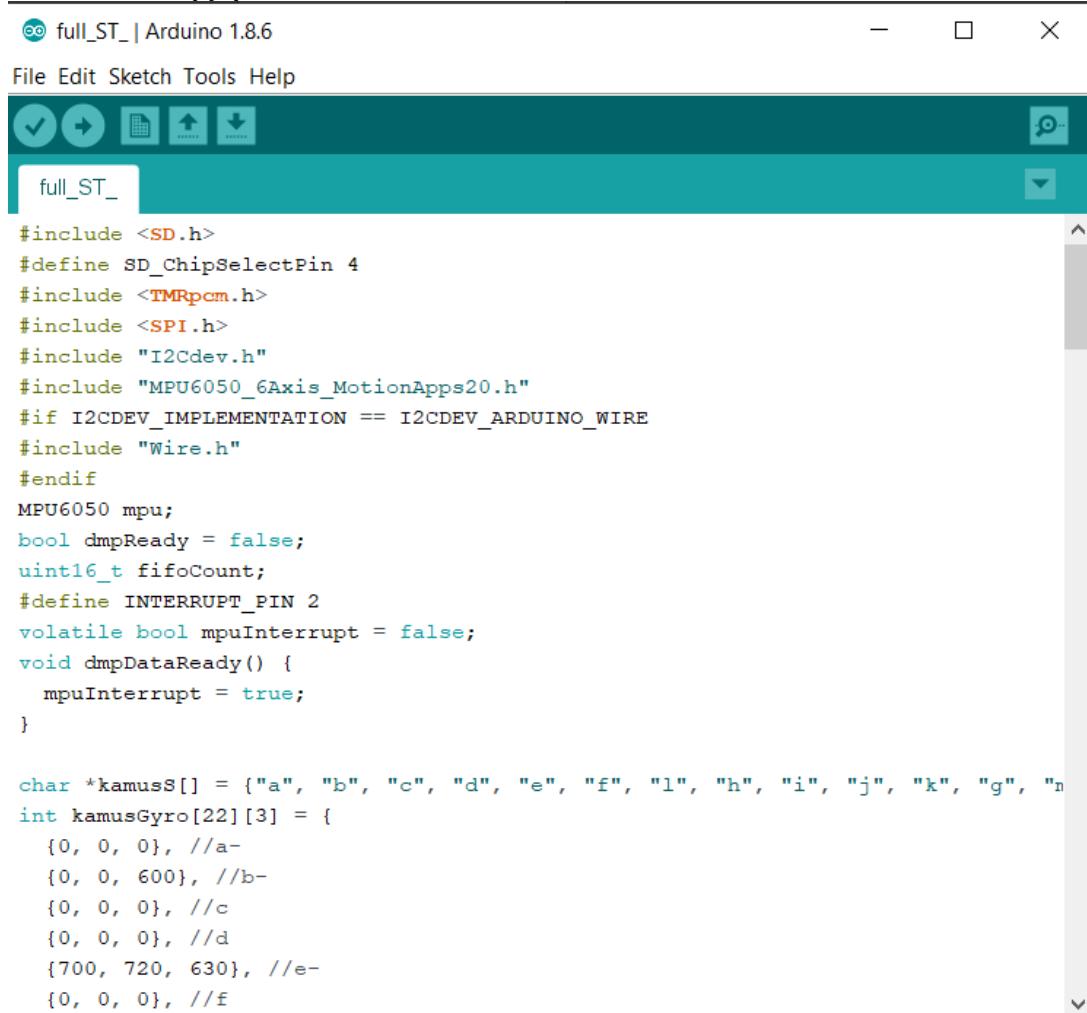
```
void setup() {
    // put your setup code here, to run once:

}

void loop() {
    // put your main code here, to run repeatedly:

}
```

2. Kemudian *copy-paste source code* kedalam *tool* IDE Arduino



The screenshot shows the Arduino IDE interface with a sketch named "full_ST_". The title bar says "full_ST_ | Arduino 1.8.6". The code editor window contains a large amount of C++ code related to interfacing with an MPU6050 sensor via SPI and I2C, including definitions for SD card access and motion detection.

```
#include <SD.h>
#define SD_ChipSelectPin 4
#include <TMRpcm.h>
#include <SPI.h>
#include "I2Cdev.h"
#include "MPU6050_6Axis_MotionApps20.h"
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
#include "Wire.h"
#endif
MPU6050 mpu;
bool dmpReady = false;
uint16_t fifoCount;
#define INTERRUPT_PIN 2
volatile bool mpuInterrupt = false;
void dmpDataReady() {
    mpuInterrupt = true;
}

char *kamusS[] = {"a", "b", "c", "d", "e", "f", "l", "h", "i", "j", "k", "g", "n"};
int kamusGyro[22][3] = {
    {0, 0, 0}, //a-
    {0, 0, 600}, //b-
    {0, 0, 0}, //c
    {0, 0, 0}, //d
    {700, 720, 630}, //e-
    {0, 0, 0}, //f
    {0, 0, 0}, //g
    {0, 0, 0}, //h
    {0, 0, 0}, //i
    {0, 0, 0}, //j
    {0, 0, 0}, //k
    {0, 0, 0}, //l
    {0, 0, 0}, //m
    {0, 0, 0}, //n
    {0, 0, 0}, //o
    {0, 0, 0}, //p
    {0, 0, 0}, //q
    {0, 0, 0}, //r
    {0, 0, 0}, //s
    {0, 0, 0}, //t
    {0, 0, 0}, //u
    {0, 0, 0}, //v
    {0, 0, 0}, //w
    {0, 0, 0}, //x
    {0, 0, 0}, //y
    {0, 0, 0}, //z
}
```

3. Kemudian tes program dengan menekan tanda centang pojok kiri (*Verify*).



4. Lihat pesan notifikasi. Apabila hasil sama seperti gambar dibawah berarti berhasil.

```
Done compiling.  
  
Sketch uses 27428 bytes (89%) of program storage space. Maximum is 30720 bytes.  
Global variables use 1624 bytes (79%) of dynamic memory, leaving 424 bytes for local variables. Maximum is 2048 bytes.  
Low memory available, stability problems may occur.  
  
60  
Arduino Nano, ATmega328P (Old Bootloader) on COM3
```

5. Selanjutnya upload program kedalam arduinu nano dengan menekan tombol panah kanan.



6. Sebelum menjalankan program keseluruhan alat perlu dirakit terlebih dahulu sehingga siap untuk digunakan dan di ujicoba

