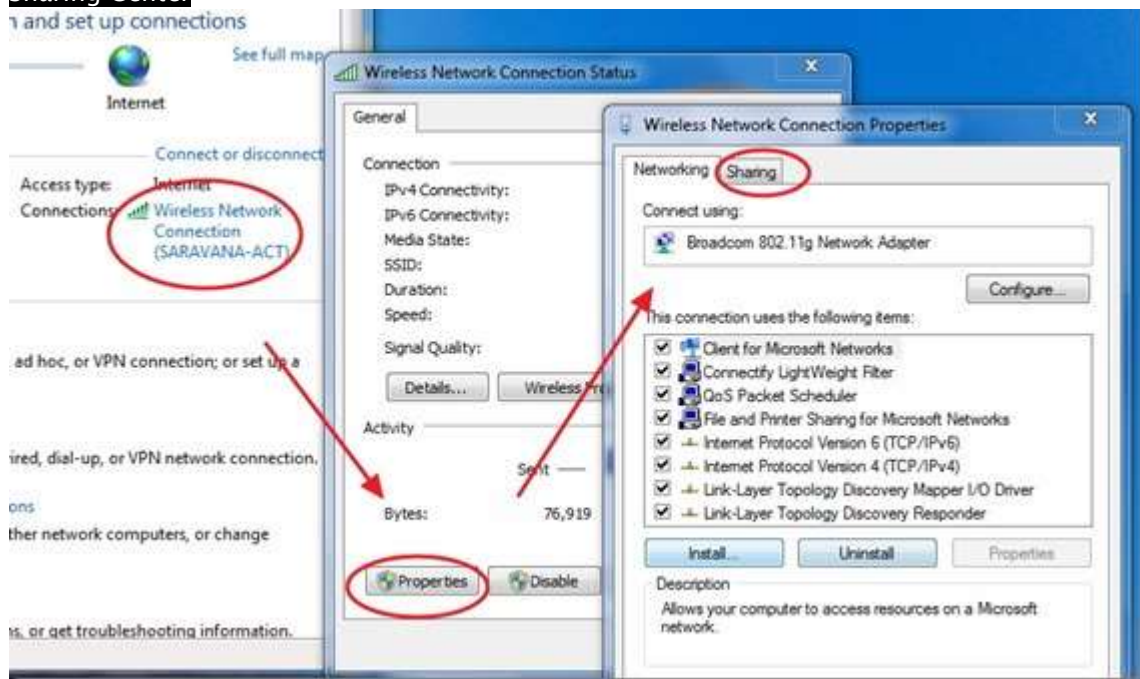


**MODUL**  
**ARDUINO ETHERNET SHIELD (3)**  
**Web Client**

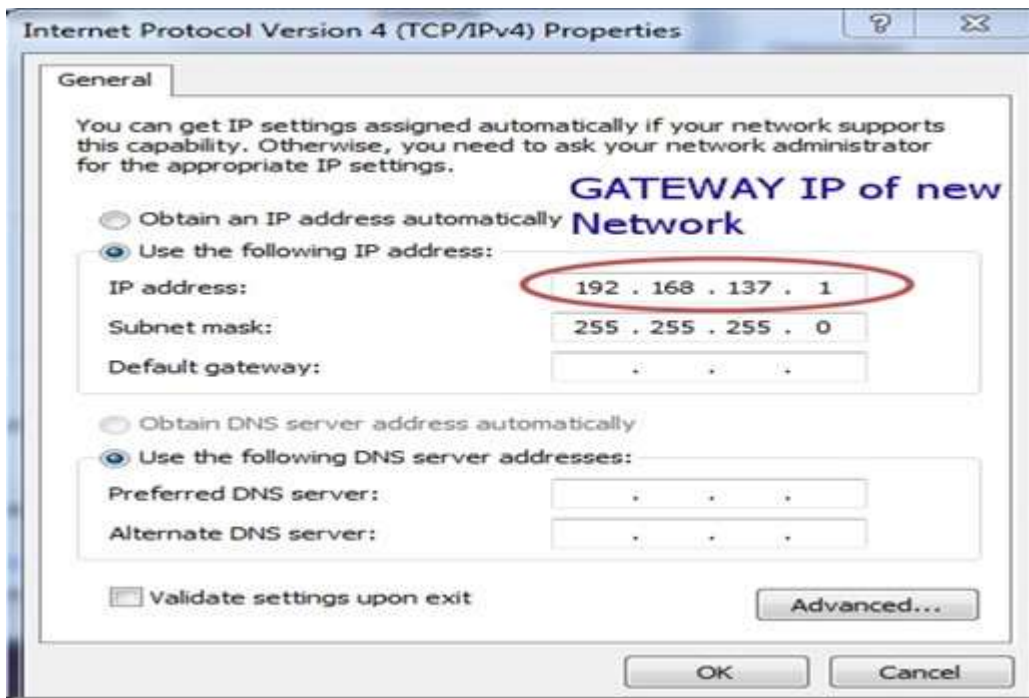
**PRAKTIKUM**

1. Untuk menggunakan Arduino UNO sebagai web client, maka Arduino harus terkoneksi ke server. Jika server berada di Internet (WAN), maka Arduino harus tersambung ke WAN.
2. Untuk menghubungkan Arduino UNO ke Internet menggunakan Ethernet Shield, maka Ethernet Shield bisa langsung dihubungkan dengan router. Jika tidak terdapat Router, opsinya adalah dengan menggunakan koneksi wireless yang ada di Laptop untuk terhubung ke Internet.
3. Share koneksi Internet dengan membuka **Control Panel -> Network & Internet -> Network & Sharing Center**





4. Ganti IP Address untuk LAN



5. Coba program Arduino We Client. Source code bisa diperoleh di **File > Examples > Ethernet > WebClient.**

Source Code

```
#include <SPI.h>
#include <Ethernet.h>

// Enter a MAC address for your controller below.
```

```

// Newer Ethernet shields have a MAC address printed on a sticker on the shield
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
int result;
// if you don't want to use DNS (and reduce your sketch size)
// use the numeric IP instead of the name for the server:
//IPAddress server(74,125,232,128); // numeric IP for Google (no DNS)
char server[] = "www.google.com"; // name address for Google (using DNS)

// Set the static IP address to use if the DHCP fails to assign
IPAddress ip(192, 168, 1, 177);

// Initialize the Ethernet client library
// with the IP address and port of the server
// that you want to connect to (port 80 is default for HTTP):
EthernetClient client;

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  pinMode(4, OUTPUT);
  digitalWrite(4, HIGH);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }

  // start the Ethernet connection:
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
    // try to configure using IP address instead of DHCP:
    Ethernet.begin(mac, ip);
  }
  // give the Ethernet shield a second to initialize:

  Serial.println("connecting...");
  delay(2000);
  Serial.println(Ethernet.localIP());

  // if you get a connection, report back via serial:
  if (result = client.connect(server, 80)) {
    Serial.print(result);
    Serial.println();
    Serial.println("connected");
    // Make a HTTP request:
    client.println("GET /search?q=arduino HTTP/1.1");
    client.println("Host: www.google.com");
    client.println("Connection: close");
    client.println();
  } else {
    // if you didn't get a connection to the server:
    Serial.println("connection failed");
  }
}

void loop() {
  // if there are incoming bytes available

```

```
// from the server, read them and print them:
if (client.available()) {
    char c = client.read();
    Serial.print(c);
}

// if the server's disconnected, stop the client:
if (!client.connected()) {
    Serial.println();
    Serial.println("disconnecting.");
    client.stop();

    // do nothing forevermore:
    while (true);
}
}
```

6. Amati Outputnya.