STEP BY STEP CODE EXPLANATION

```
#include <LiquidCrystal.h>                    // Include the LiquidCrystal library
```

This line includes the LiquidCrystal library, which is used for controlling LCD displays.

```
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);        // Create an LCD object
```

Here, an LiquidCrystal object named lcd is created and configured to work with your LCD display. The numbers 2, 3, 4, 5, 6, 7 represent the pins used for connecting to the LCD.

```
#define sensor_pin A0

int adc_value;

int percent_value;
```

These lines define variables and constants for reading and storing sensor data. sensor_pin represents the analog pin (A0) connected to the rain sensor. adc_value will store the raw analog reading from the sensor, and percent_value will store the mapped percentage of rain level.

```
#define bt_silent     A5

int silent = 0;
```

Here, a constant bt_silent is defined, representing a button pin (A5) used to activate silent mode. The silent variable is used to track whether silent mode is activated (1) or not (0).

```
#define G_led 8

#define R_led 9

#define buzzer 13
```

These lines define constants for the pins connected to LEDs (G_led for green LED and R_led for red LED) and a buzzer. These pins will be used to provide visual and audible feedback based on the rain sensor readings.

```
void setup() {            // Setup code that runs once when the Arduino starts
```

The setup() function is where you put the setup code that runs only once when the Arduino is powered on or reset.

```
pinMode(sensor_pin, INPUT);
```

This line sets the sensor_pin as an input, as it's connected to the rain sensor.

```
pinMode(bt_silent, INPUT_PULLUP);
```

This line sets the bt_silent pin as an input with a pull-up resistor. The pull-up resistor ensures that the input is high when the button is not pressed.

```
pinMode(R_led, OUTPUT);
pinMode(G_led, OUTPUT);
pinMode(buzzer, OUTPUT);
```

These lines set the pins for the red LED, green LED, and buzzer as outputs, indicating that these pins will be used to send signals to the LEDs and buzzer.

```
lcd.begin(16, 2);         // Initialize the LCD display with 16 columns and 2 rows
lcd.clear();
```

This code initializes the LCD display with the specified number of columns and rows. In this case, it's a 16x2 character LCD display. It also clears the LCD display.

```
lcd.setCursor (0,0);
lcd.print("   Welcome To   ");
lcd.setCursor (0,1);
lcd.print(" Rain  Detector ");
delay(2000);
```

```
lcd.clear();
```

This section prints a welcome message on the LCD for 2 seconds and then clears the LCD.

```
void loop() {             // Main loop code that runs repeatedly
```

The loop() function is where you put the main code that runs repeatedly while the Arduino is powered on.

```
adc_value = analogRead(sensor_pin);
percent_value = map(adc_value, 0, 1023, 100, 0);
```

These lines read the analog value from the rain sensor connected to sensor_pin, and then they map this value from the range 0-1023 to the range 100-0 (percent_value) to represent the rain level as a percentage.

```
if (digitalRead(bt_silent) == 0) {
  silent = 1;
  delay(100);
}
```

This code checks if the button connected to bt_silent is pressed (its input is low). If pressed, it sets the silent variable to 1 (silent mode activated) and adds a short delay to debounce the button.

```
lcd.setCursor(0, 0);
lcd.print("Rain Level: ");
lcd.print(percent_value);
lcd.print("%    ");
```

Here, it displays the rain level on the first line of the LCD, followed by the percentage value.

```
lcd.setCursor(0, 1);
```

This line moves the cursor to the beginning of the second line of the LCD.

```
if (percent_value > 30) {
    if (silent == 0) {
        digitalWrite(buzzer, HIGH);
    }
    lcd.print("Rain Alert...!!!");
    digitalWrite(G_led, LOW);
    digitalWrite(R_led, HIGH);
    delay(300);
} else {
    silent = 0;
    lcd.print(".....Normal.....");
    digitalWrite(G_led, HIGH);
    digitalWrite(R_led, LOW);
}
```

This section checks if the rain level (percent_value) is greater than 30%. If it is, and the system is not in silent mode (silent == 0), it activates the buzzer and displays a "Rain Alert...!!!" message on the LCD. It also turns off the green LED (G_led) and turns on the red LED (R_led) for visual indication. After a brief delay, it returns to normal mode.

If the rain level is 30% or lower, it sets silent to 0, displays ".....Normal....." on the LCD, turns on the green LED, and turns off the red LED.

```
digitalWrite(buzzer, LOW);
delay(100);
```

Finally, it turns off the buzzer and adds a short delay before the loop repeats.

This code essentially reads the rain sensor's analog value, maps it to a percentage, provides visual and audible feedback based on the rain level, and allows for a silent mode using a button.
