Java Crypto Example with PBKDF2

```java
import javax.crypto.*;
import javax.crypto.spec.PBEKeySpec;
import javax.crypto.spec.SecretKeySpec;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.spec.InvalidKeySpecException;
import java.security.spec.KeySpec;
import java.util.Base64;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) throws NoSuchPaddingException, NoSuchAlgorithmException, InvalidKeyException, IllegalBlockSizeException, BadPaddingException, InvalidKeySpecException {
        SecureRandom random = new SecureRandom();
        byte [] salt = new byte[16];
        random.nextBytes(salt);
        //String saltString = Base64.getEncoder().encodeToString(salt);
        String saltString = "ZY1kxrD843mGRVORU58JLA==";
        salt = Base64.getDecoder().decode(saltString.getBytes());
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter password: ");
        String keyString = scanner.nextLine();

        KeySpec spec = new PBEKeySpec(keyString.toCharArray(), salt, 1024, 128);
        SecretKeyFactory factory = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA256");
        SecretKey privateKey = factory.generateSecret(spec);

        System.out.print("Do you want to encrypt or decrypt (e|d): ");
        String option = scanner.nextLine();
        Cipher cipher = Cipher.getInstance("AES");
        SecretKeySpec key = new SecretKeySpec(privateKey.getEncoded(), "AES");

        if (option.equals("d")) {
            System.out.print("Enter message to decrypt: ");
            String message = scanner.nextLine();
            cipher.init(Cipher.DECRYPT_MODE, key);

            byte [] encryptedData = Base64.getDecoder().decode(message);
            byte [] decryptedData = cipher.doFinal(encryptedData);
            String decryptedMessage = new String(decryptedData);

            System.out.println("Message is " + decryptedMessage);
        }
        else if (option.equals("e")) {
            System.out.print("Enter message to encrypt: ");
            String message = scanner.nextLine();
            cipher.init(Cipher.ENCRYPT_MODE, key);
            byte [] encryptedData = cipher.doFinal(message.getBytes());
            String messageString = new String(Base64.getEncoder().encode(encryptedData));
            System.out.println(messageString);
        }
        else {
            System.err.println("Invalid option");
            System.exit(1);
        }

    }
}
```