

NAMA : Nabil Ulil Albab  
NPM : 23670010



```
1 package main
2
3 import (
4     "fmt"
5     "strings"
6 )
7
8 func encryptCaesar(plaintext string, key int) string {
9     var ciphertext strings.Builder
10    for _, char := range plaintext {
11        if char >= 'A' && char <= 'Z' {
12            newChar := 'A' + (char-'A'+rune(key))%26
13            ciphertext.WriteRune(newChar)
14        } else {
15            ciphertext.WriteRune(char)
16        }
17    }
18    return ciphertext.String()
19 }
20
21 func encryptRailFence(plaintext string, key int) string {
22     rails := make([][]rune, key)
23     railNum := 0
24     direction := 1
25
26     for _, char := range plaintext {
27         rails[railNum] = append(rails[railNum], char)
28         if railNum == 0 {
29             direction = 1
30         } else if railNum == key-1 {
31             direction = -1
32         }
33         railNum += direction
34     }
35
36     var ciphertext strings.Builder
37     for _, rail := range rails {
38         for _, char := range rail {
39             ciphertext.WriteRune(char)
40         }
41     }
42     return ciphertext.String()
43 }
44
45 func pause() {
46     var dummy string
47     fmt.Print("\nTekan Enter untuk kembali ke menu...")
48 }
```

===== MENU KRIPTOGRAFI =====  
1. Enkripsi Caesar Cipher  
2. Enkripsi Rail Fence Cipher  
3. Keluar  
Pilih menu (1-3): 1  
Masukkan teks: NABILULILALBAB  
Masukkan kunci Caesar (angka): 3  
  
Hasil Caesar Cipher: QDELOXOLOODEDE  
  
Tekan Enter untuk kembali ke menu...

===== MENU KRIPTOGRAFI =====  
1. Enkripsi Caesar Cipher  
2. Enkripsi Rail Fence Cipher  
3. Keluar  
Pilih menu (1-3): 2  
Masukkan teks: NABILULILALBAB  
Masukkan jumlah rails: 3  
  
Hasil Rail Fence Cipher: NLLAAIUIABBBLL  
  
Tekan Enter untuk kembali ke menu...

===== MENU KRIPTOGRAFI =====  
1. Enkripsi Caesar Cipher  
2. Enkripsi Rail Fence Cipher  
3. Keluar  
Pilih menu (1-3):

package main

```
import (  
    "fmt"  
    "strings"  
)
```

```
func encryptCaesar(plaintext string, key int) string {  
    var ciphertext strings.Builder  
    for _, char := range plaintext {  
        if char >= 'A' && char <= 'Z' {  
            newChar := 'A' + (char-'A'+rune(key))%26  
            ciphertext.WriteRune(newChar)  
        } else {  
            ciphertext.WriteRune(char)  
        }  
    }  
    return ciphertext.String()  
}
```

```

}

func encryptRailFence(plaintext string, key int) string {
    rails := make([][]rune, key)
    railNum := 0
    direction := 1

    for _, char := range plaintext {
        rails[railNum] = append(rails[railNum], char)
        if railNum == 0 {
            direction = 1
        } else if railNum == key-1 {
            direction = -1
        }
        railNum += direction
    }

    var ciphertext strings.Builder
    for _, rail := range rails {
        for _, char := range rail {
            ciphertext.WriteRune(char)
        }
    }
    return ciphertext.String()
}

func pause() {
    var dummy string
    fmt.Print("\nTekan Enter untuk kembali ke menu...")
    fmt.Scanln(&dummy)
}

func main() {
    for {
        fmt.Println("\n===== MENU KRIPTOGRAFI =====")
        fmt.Println("1. Enkripsi Caesar Cipher")
        fmt.Println("2. Enkripsi Rail Fence Cipher")
        fmt.Println("3. Keluar")
        fmt.Print("Pilih menu (1-3): ")

        var pilihan int
        fmt.Scanln(&pilihan)

        if pilihan == 3 {
            fmt.Println("Terima kasih, program selesai.")
            break
        }

        var teks string
    }
}

```

```

fmt.Print("Masukkan teks: ")
fmt.Scanln(&teks)
plaintext := strings.ToUpper(teks)

switch pilihan {
case 1:
    var key int
    fmt.Print("Masukkan kunci Caesar (angka): ")
    fmt.Scanln(&key)
    cipher := encryptCaesar(plaintext, key)
    fmt.Printf("\nHasil Caesar Cipher: %s\n", cipher)
    pause()

case 2:
    var key int
    fmt.Print("Masukkan jumlah rails: ")
    fmt.Scanln(&key)
    cipher := encryptRailFence(plaintext, key)
    fmt.Printf("\nHasil Rail Fence Cipher: %s\n", cipher)
    pause()

default:
    fmt.Println("Pilihan tidak valid.")
    pause()
}
}
}

```