

Nama : Ryan Erlando Supit

NIM : 17013047

Kode Kelas : 13108

Kelompok : -

Tanggal : 24 september 2020

Judul Praktikum : Caesar Cipher

Hasil

1. File Encrypt

Enter a message to encrypt : (masukkan kalimat yang akan di enkripsi)

>> tekan enter

Encrypted message : (hasil enkripsi kalimat)

File Decrypt

Enter a message to decrypt: (masukkan kalimat yang akan di dekripsi)

>>tekan enter

Decrypted message:(hasil dekripsi kalimat)

2. KEY 3

The screenshot displays a C++ IDE with two windows. The left window, titled 'encrypt.cpp', contains the source code for a Caesar cipher program. The code includes a header file 'ctype.h', defines a main function, and implements logic to encrypt and decrypt a message based on a key of 3. The right window, titled 'Select C:\Users\Ryan\Documents\encrypt.exe', shows the program's execution. It prompts the user to enter a message to encrypt, which is 'RYAN ERLANDO SUPIT'. The encrypted message is displayed as 'URDQ HODQQGK VXSJW'. The program then prompts for a message to decrypt, which is 'URDQ HODQQGK VXSJW'. The decrypted message is displayed as 'RYAN ERLANDO SUPIT'. The program exits after 4.4 seconds with a return value of 0.

```
1 #include <ctype.h>
2
3 int main()
4 {
5     char message[100], ch;
6     int i, key;
7
8     printf("Enter a message to encrypt: ");
9     gets(message);
10
11     key = 3;
12
13     for(i = 0; message[i] != '\0'; ++i){
14         ch = message[i];
15
16         if(ch >= 'a' && ch <= 'z'){
17             ch = ch + key;
18             if(ch > 'z'){
19                 ch = ch - 'z' + 'a' - 1;
20             }
21             message[i] = ch;
22         }
23         else if(ch >= 'A' && ch <= 'Z'){
24             ch = ch + key;
25             if(ch > 'Z'){
26                 ch = ch - 'Z' + 'A' - 1;
27             }
28             message[i] = ch;
29         }
30     }
31
32     printf("Encrypted message: ");
33     puts(message);
34
35     printf("Enter a message to decrypt: ");
36     gets(message);
37
38     key = 3;
39
40     for(i = 0; message[i] != '\0'; ++i){
41         ch = message[i];
42
43         if(ch >= 'a' && ch <= 'z'){
44             ch = ch - key;
45             if(ch < 'a'){
46                 ch = ch + 'z' - 'a' + 1;
47             }
48             message[i] = ch;
49         }
50         else if(ch >= 'A' && ch <= 'Z'){
51             ch = ch - key;
52             if(ch < 'A'){
53                 ch = ch + 'Z' - 'A' + 1;
54             }
55             message[i] = ch;
56         }
57     }
58
59     printf("Decrypted message: ");
60     puts(message);
61
62     return 0;
63 }
```

Compilation results...

- Errors: 0

- Warnings: 0

- Output Filename: C:\Users\Ryan\Documents\encrypt.exe

- Output Size: 120,100883075 Kib

- Compilation Time: 0.30s

Done parsing in 0.016 seconds

3. KEY 18

The screenshot shows a C++ IDE with the following code in `encrypt.cpp`:

```

1 #include <stdio.h>
2 #include <ctype.h>
3
4 int main()
5 {
6     char message[100], ch;
7     int i, key;
8
9     printf("Enter a message to encrypt: ");
10    gets(message);
11
12    key = 18;
13
14    for(i = 0; message[i] != '\0'; ++i){
15        ch = message[i];
16
17        if(ch >= 'a' && ch <= 'z'){
18            ch = ch + key;
19
20            if(ch > 'z'){
21                ch = ch - 'z' + 'a' - 1;
22            }
23
24            message[i] = ch;
25        }
26        else if(ch >= 'A' && ch <= 'Z'){
27            ch = ch + key;
28
29            if(ch > 'Z'){
30                ch = ch - 'Z' + 'A' - 1;
31            }
32
33            message[i] = ch;
34        }
35    }
36
37    printf("Encrypted message: ");
38    puts(message);
39
40    printf("Process exited after 13.49 seconds with return value 0\n");
41    printf("Press any key to continue . . .");
42    getch();
43    return 0;
44 }
```

The compiler output shows no errors or warnings. The output file is `C:\Users\Ryan\Documents\encrypt.exe` with a size of 128,100,859,975 KiB and a compilation time of 0.26s.

The terminal output shows the execution of the program:

```

Enter a message to encrypt: RYAN ERLANDO SUPIT
Encrypted message: JQSF WJDSFV6 KPMAL
Process exited after 13.49 seconds with return value 0
Press any key to continue . . .

```

4. KEY 0

The screenshot shows the same C++ IDE with the following code in `encrypt.cpp`:

```

1 #include <stdio.h>
2 #include <ctype.h>
3
4 int main()
5 {
6     char message[100], ch;
7     int i, key;
8
9     printf("Enter a message to encrypt: ");
10    gets(message);
11
12    key = 0;
13
14    for(i = 0; message[i] != '\0'; ++i){
15        ch = message[i];
16
17        if(ch >= 'a' && ch <= 'z'){
18            ch = ch + key;
19
20            if(ch > 'z'){
21                ch = ch - 'z' + 'a' - 1;
22            }
23
24            message[i] = ch;
25        }
26        else if(ch >= 'A' && ch <= 'Z'){
27            ch = ch + key;
28
29            if(ch > 'Z'){
30                ch = ch - 'Z' + 'A' - 1;
31            }
32
33            message[i] = ch;
34        }
35    }
36
37    printf("Encrypted message: ");
38    puts(message);
39
40    printf("Process exited after 34.2 seconds with return value 0\n");
41    printf("Press any key to continue . . .");
42    getch();
43    return 0;
44 }
```

The compiler output shows no errors or warnings. The output file is `C:\Users\Ryan\Documents\encrypt.exe` with a size of 128,100,859,975 KiB and a compilation time of 0.33s.

The terminal output shows the execution of the program:

```

Enter a message to encrypt: RYAN ERLANDO SUPIT
Encrypted message: RYAN ERLANDO SUPIT
Process exited after 34.2 seconds with return value 0
Press any key to continue . . .

```

5. KEY 3

The screenshot shows a C++ IDE with a file named `encrypt.cpp`. The code implements a Caesar cipher encryption with a key of 3. The `main` function prompts the user to enter a message, reads it, and then iterates through each character, shifting it forward by the key value. The output window shows the encrypted message: `Enter a message to encrypt: IRLDQ MHKDGGR VXSILH` and the decrypted message: `Decrypted message: RYAN ERLANDO SUPIT`. The process exited after 11.09 seconds with a return value of 0.

```

1 #include <stdio.h>
2 #include <ctype.h>
3
4 int main()
5 {
6     char message[100], ch;
7     int i, key;
8
9     printf("Enter a message to encrypt: ");
10    gets(message);
11
12    key = 3;
13
14    for(i = 0; message[i] != '\0'; ++i){
15        ch = message[i];
16
17        if(ch >= 'a' && ch <= 'z'){
18            ch = ch - key;
19
20            if(ch < 'a'){
21                ch = ch + 'z' - 'a' + 1;
22            }
23
24            message[i] = ch;
25        }
26        else if(ch >= 'A' && ch <= 'Z'){
27            ch = ch - key;
28
29            if(ch < 'A'){
30                ch = ch + 'Z' - 'A' + 1;
31            }
32
33            message[i] = ch;
34        }
35    }
36}

```

Compilation results:
 - Errors: 0
 - Warnings: 0
 - Output Filename: C:\Users\Ryan\Documents\decrypt.exe
 - Output Size: 128,100,859,975 B
 - Compilation Time: 0.33s

6. KEY 18

The screenshot shows the same C++ IDE with the `encrypt.cpp` file. The key value has been changed from 3 to 18. The `main` function prompts the user to enter a message, reads it, and then iterates through each character, shifting it forward by the key value of 18. The output window shows the encrypted message: `Enter a message to encrypt: IRLDQ MHKDGGR VXSILH` and the decrypted message: `Decrypted message: RYAN ERLANDO SUPIT`. The process exited after 11.09 seconds with a return value of 0.

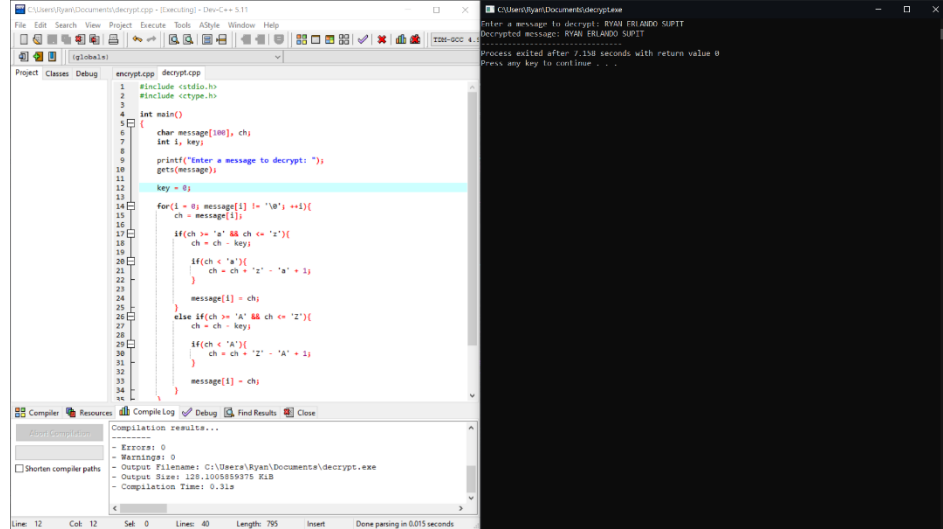
```

1 #include <stdio.h>
2 #include <ctype.h>
3
4 int main()
5 {
6     char message[100], ch;
7     int i, key;
8
9     printf("Enter a message to encrypt: ");
10    gets(message);
11
12    key = 18;
13
14    for(i = 0; message[i] != '\0'; ++i){
15        ch = message[i];
16
17        if(ch >= 'a' && ch <= 'z'){
18            ch = ch - key;
19
20            if(ch < 'a'){
21                ch = ch + 'z' - 'a' + 1;
22            }
23
24            message[i] = ch;
25        }
26        else if(ch >= 'A' && ch <= 'Z'){
27            ch = ch - key;
28
29            if(ch < 'A'){
30                ch = ch + 'Z' - 'A' + 1;
31            }
32
33            message[i] = ch;
34        }
35    }
36}

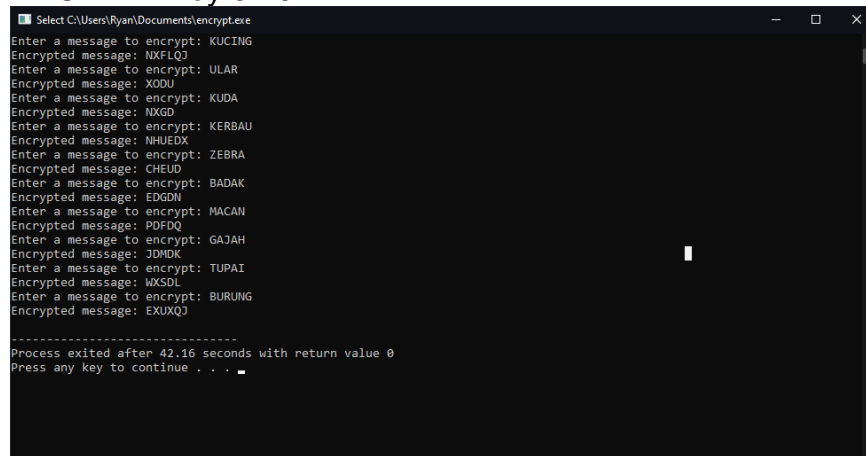
```

Compilation results:
 - Errors: 0
 - Warnings: 0
 - Output Filename: C:\Users\Ryan\Documents\decrypt.exe
 - Output Size: 128,100,859,975 B
 - Compilation Time: 0.33s

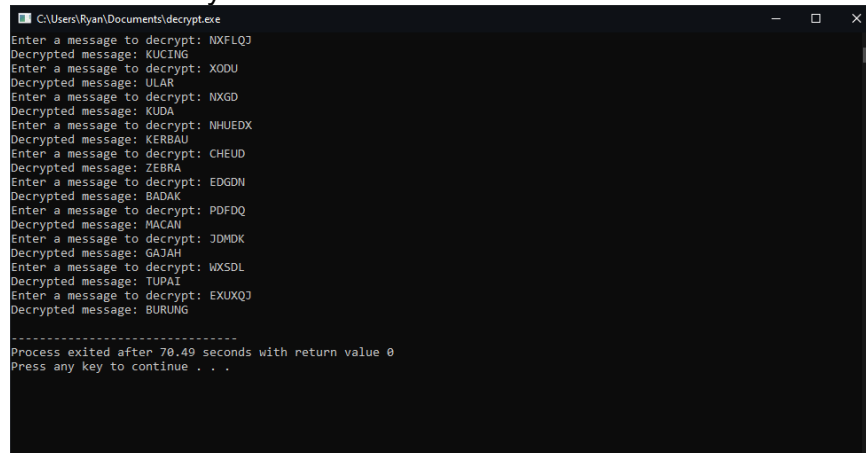
7. KEY 0



8. ENCRYPT key 3 10x



DECRYPT key 3 10x



ENCRYPT key 5 15x

```

C:\Users\Ryan\Documents\encrypt.exe
Enter a message to encrypt: DURIAN
Encrypted message: GXULDQ
Enter a message to encrypt: PEPAYA
Encrypted message: SHSDBD
Enter a message to encrypt: TIMUN
Encrypted message: WLPXQ
Enter a message to encrypt: ANGGUR
Encrypted message: DQJXU
Enter a message to encrypt: PISANG
Encrypted message: SLVDQJ
Enter a message to encrypt: LECHI
Encrypted message: OHFKL
Enter a message to encrypt: PIR
Encrypted message: SLU
Enter a message to encrypt: RAMBUTAN
Encrypted message: UDPEXMDQ
Enter a message to encrypt: KIMI
Encrypted message: NLZL
Enter a message to encrypt: TOMAT
Encrypted message: WRPDM
Enter a message to encrypt: JERUK
Encrypted message: MHUXN
Enter a message to encrypt: LEMON
Encrypted message: OHPRQ
Enter a message to encrypt: SALAK
Encrypted message: VDDN
Enter a message to encrypt: APEL
Encrypted message: DSHO
Enter a message to encrypt: MANGGA
Encrypted message: PDQJJD
-----
Process exited after 90.99 seconds with return value 0
Press any key to continue . . .

```

DECRYPT key 5 15x

```

Select C:\Users\Ryan\Documents\decrypt.exe
Enter a message to decrypt: GXULDQ
Decrypted message: DURIAN
Enter a message to decrypt: SHSDBD
Decrypted message: PEPAYA
Enter a message to decrypt: WLPXQ
Decrypted message: TIMUN
Enter a message to decrypt: DQJXU
Decrypted message: ANGGUR
Enter a message to decrypt: SLVDQJ
Decrypted message: PISANG
Enter a message to decrypt: OHFKL
Decrypted message: LECHI
Enter a message to decrypt: SLU
Decrypted message: PIR
Enter a message to decrypt: UDPEXMDQ
Decrypted message: RAMBUTAN
Enter a message to decrypt: NLZL
Decrypted message: KIMI
Enter a message to decrypt: WRPDM
Decrypted message: TOMAT
Enter a message to decrypt: MHUXN
Decrypted message: JERUK
Enter a message to decrypt: OHPRQ
Decrypted message: LEMON
Enter a message to decrypt: VDDN
Decrypted message: SALAK
Enter a message to decrypt: DSHO
Decrypted message: APEL
Enter a message to decrypt: PDQJJD
Decrypted message: MANGGA
-----
Process exited after 64.74 seconds with return value 0
Press any key to continue . . .

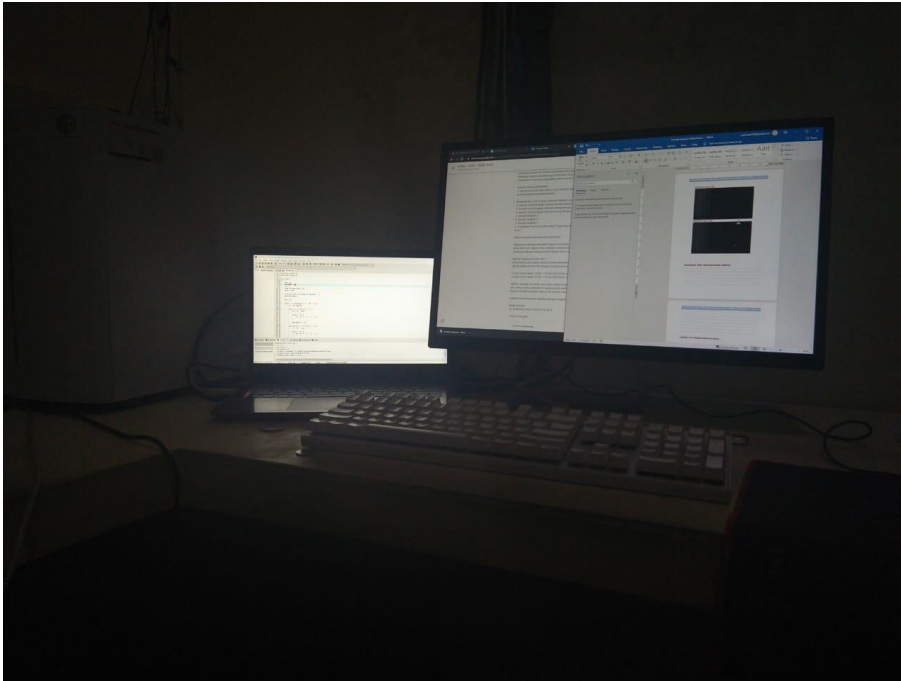
```

Caesar Cipher

Merupakan sistem persandian atau algoritma substitusi sederhana yang digunakan untuk pengamanan data, caesar cipher menggunakan operasi *shift* dengan mensubstitusikan suatu huruf menjadi huruf urutan pada daftar alfabet.

Dengan begitu metode caesar cipher sering digunakan sebagai metode atau proses enkripsi dimana data atau informasi diubah menjadi bentuk yang hampir tidak dikenali sebagai informasi awalnya, dan metode caesar cipher juga dapat menterjemahkan kode kembali dalam bentuk awalnya yang dapat dimengerti (dekripsi).

Dokumentasi (Foto) selama percobaan praktikum



Kendala yang dihadapi selama percobaan

Sulit menemukan motivasi dalam membuat tugas jika di rumah.