Algorithm 1 Gambar Fungsi Main

```
1: function MAIN
2:     T ← INPUT
3:     while T ≠ 0 do
4:         T = T − 1
5:         m ← input                          ▷ masukkan batas atas dari kunci
6:         message[ ] ← input                 ▷ masukkan plaintext
7:         cipher[ ] ← input                  ▷ masukkan ciphertext
8:         SOLVE(message,chiper,m)
9:     end while
10: end function
```

Algorithm 2 Gambar Fungsi SOLVE

```
1: function SOLVE(message,chiper,m)
2:     counter_diketahui ← 0
3:     counter_yang_ingin_diketahui ← 0
4:     diketahui[ ]
5:     Selisih_diketahui[ ]
6:     ingin_diketahui[ ]
7:     Key[ ]
8:     for i = 0 to message[i] ≠ 0 ; i+ = 1 do
9:         if message[i] ≠′ ∗′ dan cipher[i] ≠′ ∗′ then
10:            diketahui[counter_diketahui] = i
11:            Selisih_diketahui[i] = (message[i] − cipher[i] + 26)%26
12:            counter_diketahui = counter_diketahui + 1
13:        else if message[i] =′ ∗′ dan cipher[i] ≠′ ∗′ then
14:            ingin_diketahui[counter_yang_ingin_diketahui] = i
15:            counter_yang_ingin_diketahui + 1
16:        end if
17:    end for
18:    m = min(m, panjang message)
19:    for n = m/2 + 1 to n ≤ m;n+ = 1 do
20:        if VALIDITY(Key, counter_diketahui, diketahui, Selisih_diketahui, n) = True
   then
21:            counter ← 0
22:            while counter ≠ sizeof(ingin_diketahui) do
23:                if Key[ingin_diketahui[counter]%n] = null then
24:                    message[ingin_diketahui[counter]] =′ ∗′
25:                    remove element index i in ingin_diketahui
26:                else if message[ingin_diketahui[counter]] =′ ∗′ then
27:                    message[ingin_diketahui[counter]] = (ciphertext[ingin_diketahui[counter]]−
   Key[ingin_diketahui[counter] + 26)%26
28:                    counter = counter + 1
29:                else if message[ingin_diketahui[counter]] ≠
    (ciphertext[ingin_diketahui[counter]] − Key[ingin_diketahui[counter] + 26) %26) then
30:                    message[ingin_diketahui[counter]] =′ ∗′
31:                    remove element index i in ingin_diketahui
32:                else
33:                    counter = counter + 1
34:                end if
35:            end while
36:        end if
37:    end for
38: end function
```

Algorithm 3 Gambar Fungsi VALIDITY

```
1: function VALIDITY(Key,counter_diketahui,diketahui,Selisih_diketahui,n)
2:     Intialize(Key, −1)
3:     for i = 0 to i < counter_diketahui; i+=1 do
4:         temp = diketahui[i]
5:         if Key[temp%n] = −1 then
6:             Key[temp%n] = Selisih_diketahui[temp]
7:         else if Key[temp%n] ≠ Selisih_diketahui[temp] then return False
8:         end if
9:     end for
10:    return True
11: end function
```