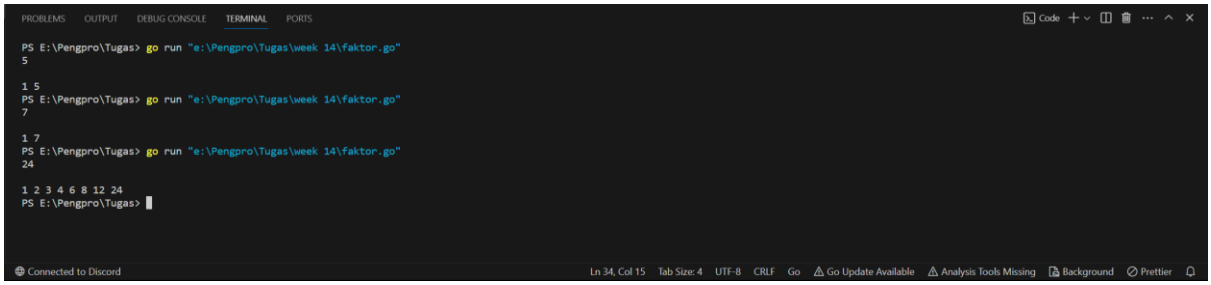
Reevanza Abel Desta Arifin_103012330104_Slide14

Faktor

```go
package main

import (
    "fmt"
)

func main()  {
    var x int

    fmt.Scan(&x)
    fmt.Println("")

    for i := 1; i <= x; i++ {
        if x % i == 0 {
            fmt.Print(i, " ")
        }
    }
}

// Program faktor

// Kamus
// x: Integer

// Algoritma
// Input(x)

// For i <- 1 to x do
//     If x mod i == 0 then
//         Output(i)
//     End If
// End For

// End Program
```



Prima

```go
package main

import (
    "fmt"
)

func main()  {
    var x, count int
    var prima bool

    fmt.Scan(&x)
    fmt.Println("")

    count = 0
    prima = true;

    for i := 1; i <= x; i++ {
        if x % i == 0 {
            count++
        }
    }

    if count != 2 {
        prima = false
    }

    fmt.Println(prima)
}

// Program Prima

// Kamus
// x, count: Integer
// prima: Boolean

// Algoritma
// Input(x)

// count <- 0
// prima <- true

// For i <- 1 to x do
//     If x mod i == 0 then
//         count + 1
//     End If
// End For
```

```
// If count != to 2 then
//     prima <- false
// End If

// Output(prima)

// End Program
```

Biner

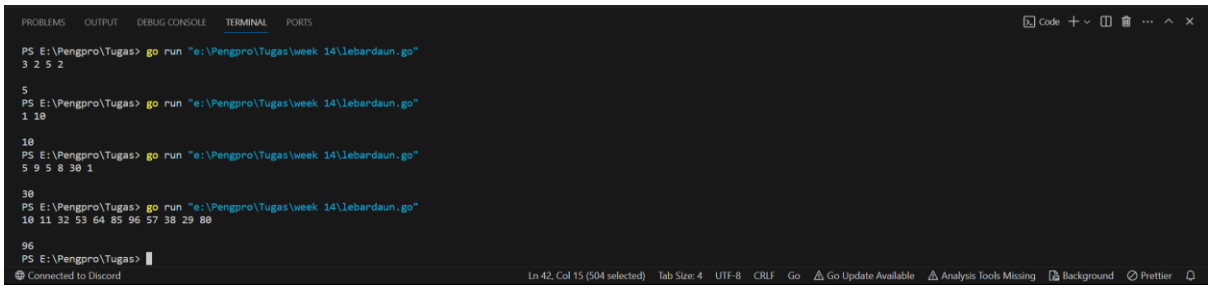
```
package main

import "fmt"

func main() {
    var x, y int
    var binaryString string

    fmt.Scan(&x)
    fmt.Println("")

    for x > 0  {
        y = x % 2
        x = x / 2
        binaryString = fmt.Sprint(y) + binaryString
    }

    fmt.Println(binaryString)
}

// Program biner

// Kamus
// x, y: Integer
// binaryString: String

// Algoritma
// Input(x)
// Output ("")
```

```
// while x > 0 Do
//     y <- x % 2
//     x <- x / 2
//     binaryString <- gabungkan(y, binaryString)
// End While

// Output(binaryString)
// End Program
```

Lebar Daun

```go
package main

import (
    "fmt"
);

func main()  {
    var n, x, hasil int

    fmt.Scan(&n)
    fmt.Println("")

    hasil = 0
    for i := 0; i < n; i++ {
        fmt.Scan(&x)
        if hasil < x {
            hasil = x
        }
    }

    fmt.Println(hasil)
}

// Prrogram lebardaun

// Kamus
// n, x, hasil: Integer
```

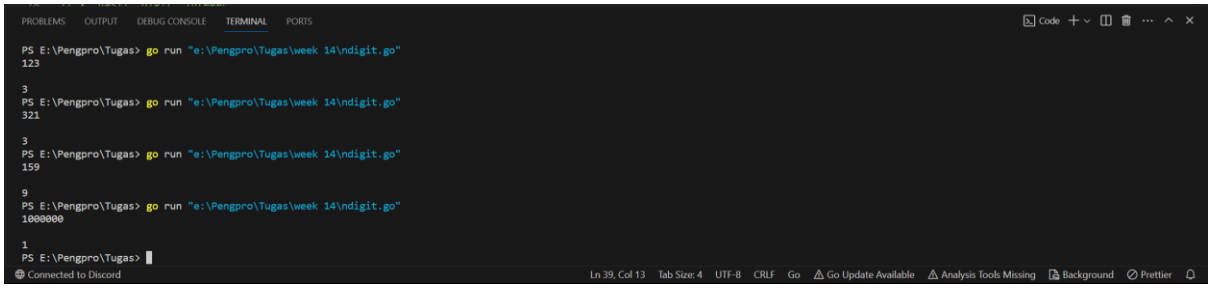
```
// Algoritma
// Input(n)
// hasil <- 0

// For i <- 0 to n - 1 do
//     Input(x)
//     If hasil < x then
//         hasil <- x
//     End If
// End For

// Output(hasil)

// End Program
```



N Digit

```go
package main

import (
    "fmt"
)

func main()  {
    var x, hasil, digit int

    fmt.Scan(&x)
    fmt.Println("")

    hasil = x % 10
    for x > 0 {
        digit = x % 10
        if digit > hasil {
            hasil = digit
        }
        x /= 10
    }

    fmt.Println(hasil)
}
```

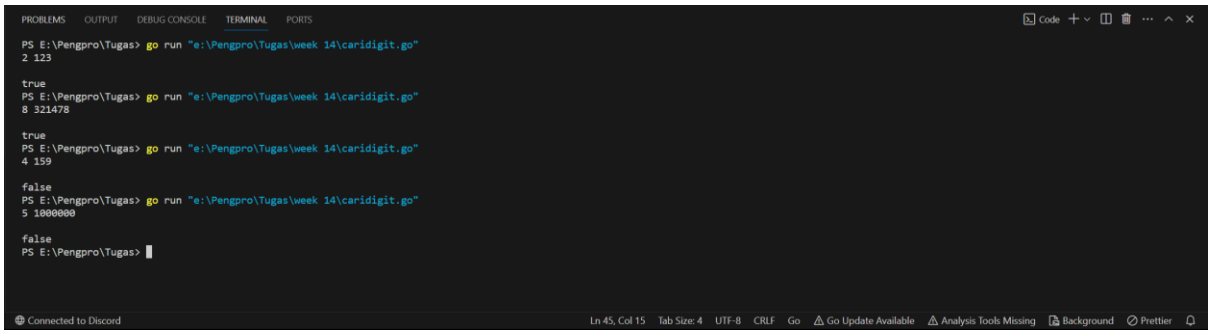
```
// Program ndigit

// Kamus
// x, hasil, digit: Integer

// Algoritma
// Input(x)
// hasil <- x % 10
// While x > 0 do
//     digit <- x % 10
//     If digit > hasil then
//         hasil <- digit
//     End If
//     x <- x / 10
// End While

// Output(hasil)

// End Program
```



Cari Digit

```
package main

import (
    "fmt"
)

func main()  {
    var x, y, digit int
    var hasil bool

    fmt.Scan(&x, &y)
    fmt.Println("")

    hasil = false
    for y > 0 {
        digit = y % 10
        if digit == x {
```

```
            hasil = true
        }
        y /= 10
    }

    fmt.Println(hasil)
}

// Program caridigit

// Kamus
// x, y, digit: Integer
// hasil: Boolean

// Algoritma
// Input(x, y)

// hasil <- false
// while y > 0 do
//     digit <- y mod 10
//     If digit == x then
//         hasil <- true
//     End If
//     y <- y div 10
// End while

// Output(hasil)
// End Program
```

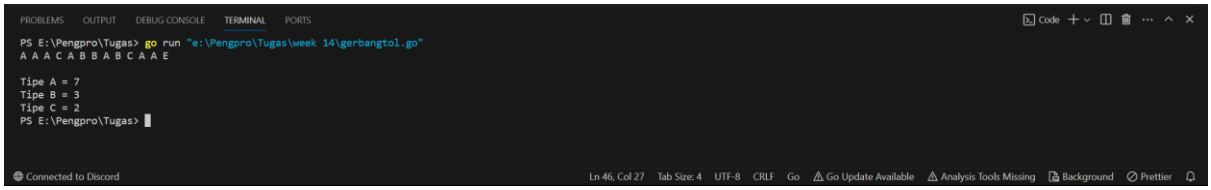


Gerbang Tol

```
package main

import (
    "fmt"
)

func main()  {
    var x string
```

```
    var a, b, c int
    var isLoop bool

    isLoop = true
    a, b, c = 0, 0, 0


    for isLoop == true{
        fmt.Scan(&x)
        if x == "A" {
            a++
        } else if x == "B" {
            b++
        } else if x == "C" {
            c++
        } else {
            isLoop = false
        }
    }

    fmt.Println("")
    fmt.Println("Tipe A =", a)
    fmt.Println("Tipe B =", b)
    fmt.Println("Tipe C =", c)
}

// Progra gerbangtol

// Kamus
// x: String
// a, b, c: Integer
// isLoop: Boolean

// Algoritma
// isLoop <- True
// a, b, c <- 0, 0, 0

// While isLoop == True do
//     Input(x)
//     If x == "A" then
//         a++
//     Else If x == "B" then
//         b++
//     Else If x == "C" then
//         c++
//     Else
//         isLoop == False
//     End If
// End While
```

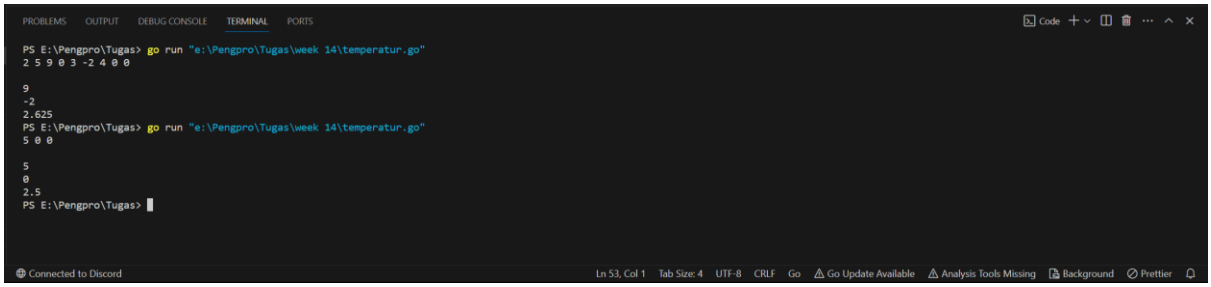
```
// Output("Tipe A =", a)
// Output("Tipe B =", b)
// Output("Tipe C =", c)

// End Program
```



Temperatur

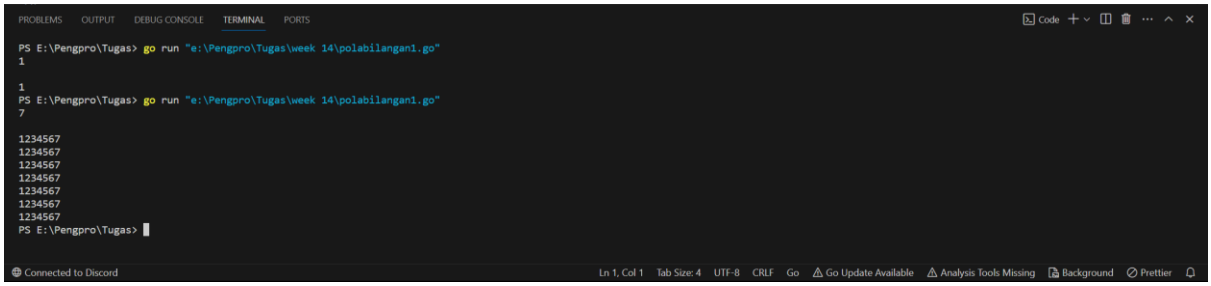
```go
package main

import (
    "fmt"
)

func main()  {
    var x, min, max, lastX, sum, n int
    var average float64
    var isLoop bool

    isLoop = true
    average, min, max, lastX, sum, n = 0, 0, 0, 0, 0, 0;

    for isLoop == true {
        fmt.Scan(&x)
        if lastX == 0 && x == 0 {
            isLoop = false;
        }

        if min > x {
            min = x
        }

        if max < x {
            max = x
        }

        sum += x
        lastX = x
        n++
    }
```

```
    average = float64(sum) / float64(n-1)

    fmt.Println("")
    fmt.Println(max)
    fmt.Println(min)
    fmt.Println(average)
}


// Program temperatur

// Kamus
// x, min, max, lastX, sum, n: Integer
// average: Real
// isLoop: Boolean

// Algoritma
// isLoop <- true
// average, min, max, lastX, sum, n <- 0

// While isLoop is true do
//     Input(x)
//     If lastX == 0 && x == 0 then
//         isLoop <- false
//     End If
//
//     If min > x then
//         min <- x
//     End If
//
//     If max < x then
//         max <- x
//     End If
//
//     sum <- sum + x
//     lastX <- x
//     n <- n + 1
// End While

// average <- sum / (n - 1)

// Output(max)
// Output(min)
// Output(average)

// End Program
```



Pola Bilangan 1

```go
package main

import (
    "fmt"
)

func main()  {
    var x int;

    fmt.Scan(&x)
    fmt.Println("")

    for i := 1; i <= x; i++ {
        for j := 1; j <= x; j++ {
            fmt.Print(j)
        }
        fmt.Println("")
    }
}

// Program polabilangan1

// Kamus
// x, i, j: Integer

// Algoritma
// Input(x)

// For i <- 1 to x do
//     For j <- 1 to x do
//         Output(j)
//     End For
//     Output("")
// End For

// End Program
```



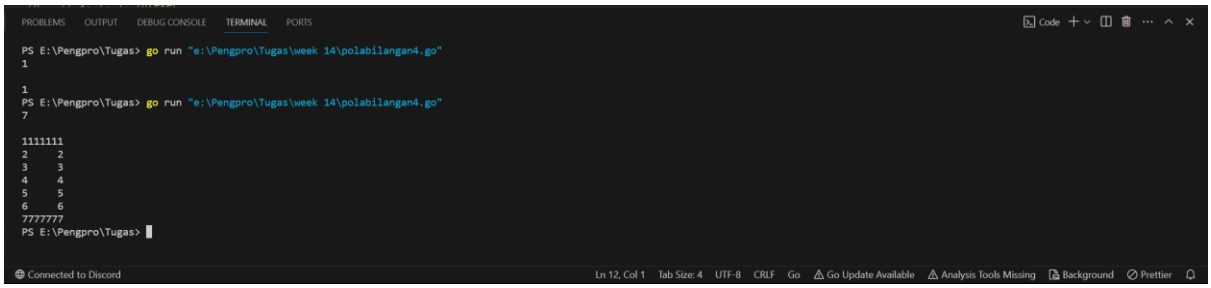
Pola Bilangan 2

```go
package main

import (
    "fmt"
)

func main()  {
    var x int

    fmt.Scan(&x)
    fmt.Println("")

    for i := 1; i <= x; i++ {
        for j := 1; j <= x; j++ {
            fmt.Print(i)
        }
        fmt.Println("")
    }
}

// Program polabilangan2

// Kamus
// x, i, j: Integer

// Algoritma
// Input(x)

// For i <- 1 to x do
//     For j <- 1 to x do
//         Output(i)
//     End For
//     Output("")
// End For

// End Program
```

Pola Bilangan 4

```go
package main

import (
    "fmt"
);

func main()  {
    var x int;

    fmt.Scan(&x)
    fmt.Println("")

    for i := 1; i <= x; i++ {
        for j := 1; j <= x; j++ {
            if j == 1 || j == x || i == 1 || i == x {
                fmt.Print(i)
            } else {
                fmt.Print(" ")
            }
        }
        fmt.Println("")
    }
}

// Program polabilangan4

// Kamus
// x, i, j: Integer

// Algoritma
// Input(x)

// For i <- 1 to x do
//     For j <- 1 to x do
//         If j == 1 OR j == x OR i == 1 OR i == x then
//             Output(i)
//         Else
//             Output(" ")
//         End If
```

```
//     End For
//     Output("")
// End For

// End Program
```



Pola Bilangan 3

```go
package main

import (
    "fmt"
);

func main()  {
    var x int;

    fmt.Scan(&x)
    fmt.Println("")

    for i := 1; i <= x; i++ {
        for j := 1; j <= x; j++ {
            if i == j || j == x - i + 1 {
                fmt.Print(i)
            } else {
                fmt.Print(" ")
            }
        }
        fmt.Println("")
    }
}

// Program polabilangan3

// Kamus
// x, i, j: Integer

// Algoritma
// Input(x)
```

```
// For i <- 1 to x do
//     For j <- 1 to x do
//         If i == j OR j == x - i + 1 then
//             Output(i)
//         Else
//             Output(" ")
//         End If
//     End For
//     Output("")
// End For

// End Program
```