

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL 9 – IF-THEN
ALGORITMA DAN PEMROGRAMAN 1**



Disusun oleh:

NAMA : PRIMATAMA SIGALINGGING

NIM : 109082500076

S1IF-13-07

Asisten Praktikum

Adithana dharma putra

Apri pandu wicaksono

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

LATIHAN KELAS – GUIDED

1. Guided 1

Source Code

```
package main

import "fmt"

func main() {

    var bilangan int

    fmt.Scan(&bilangan)

    if bilangan < 0 {

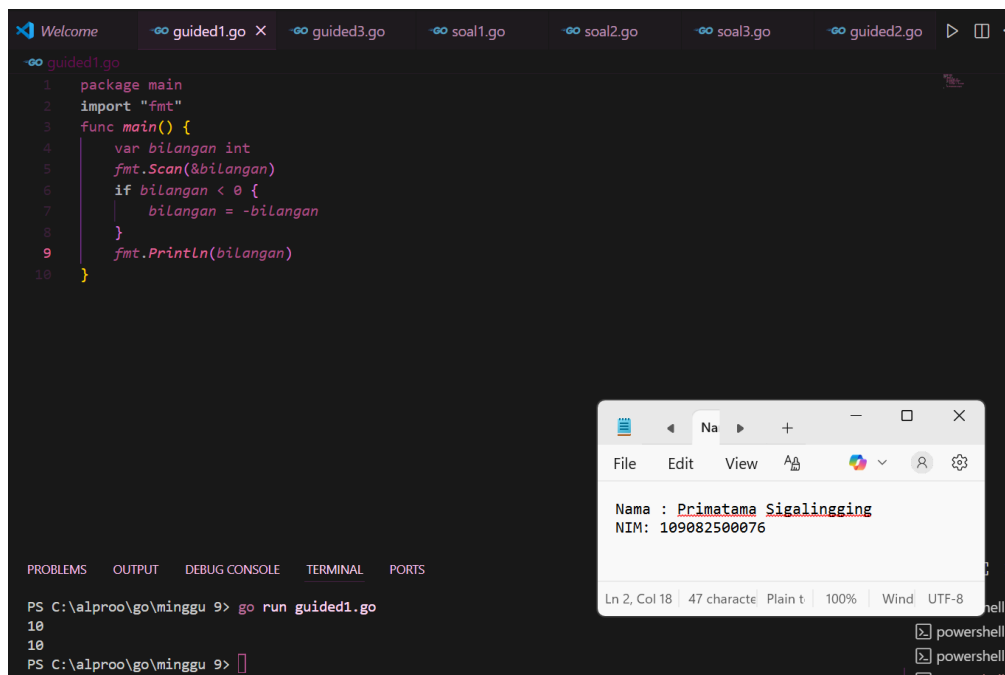
        bilangan = -bilangan

    }

    fmt.Println(bilangan)

}
```

Screenshoot program



Deskripsi program

➤ Tujuan

Program ini dibuat untuk menampilkan nilai absolut dari sebuah bilangan yang dimasukkan oleh pengguna. Tujuan pembuatannya adalah untuk melatih logika percabangan sederhana dalam bahasa Go serta memahami bagaimana komputer dapat memeriksa suatu kondisi dan memodifikasi nilai berdasarkan kondisi tersebut.

➤ Proses

Ketika program dijalankan, pengguna diminta memasukkan sebuah bilangan bulat. Program kemudian menyimpan bilangan tersebut ke dalam variabel dan memeriksa apakah bilangan tersebut kurang dari 0.

Jika bilangan negatif, program mengubahnya menjadi positif dengan cara mengalikan nilai tersebut dengan -1.

Jika bilangan positif atau nol, nilai tersebut langsung dicetak tanpa perubahan.

Proses ini memanfaatkan struktur percabangan (if statement) untuk menentukan tindakan yang harus dilakukan berdasarkan nilai input. Dengan demikian, program dapat memastikan bahwa hasil akhirnya selalu berupa bilangan positif.

Program ini dapat dikembangkan lebih lanjut untuk mengecek lebih banyak kondisi, seperti apakah bilangan tersebut juga ganjil, genap, atau termasuk kategori lainnya.

➤ Kesimpulan

Program ini memberikan pemahaman dasar tentang bagaimana komputer membuat keputusan menggunakan logika percabangan. Dengan langkah sederhana, program dapat mengubah bilangan negatif menjadi positif sehingga menghasilkan nilai absolut. Dari contoh ini, kita belajar bahwa manipulasi angka dalam pemrograman dapat dilakukan dengan mudah menggunakan logika sederhana dan terstruktur.

2. Guided 2

Source Code

```
package main

import "fmt"

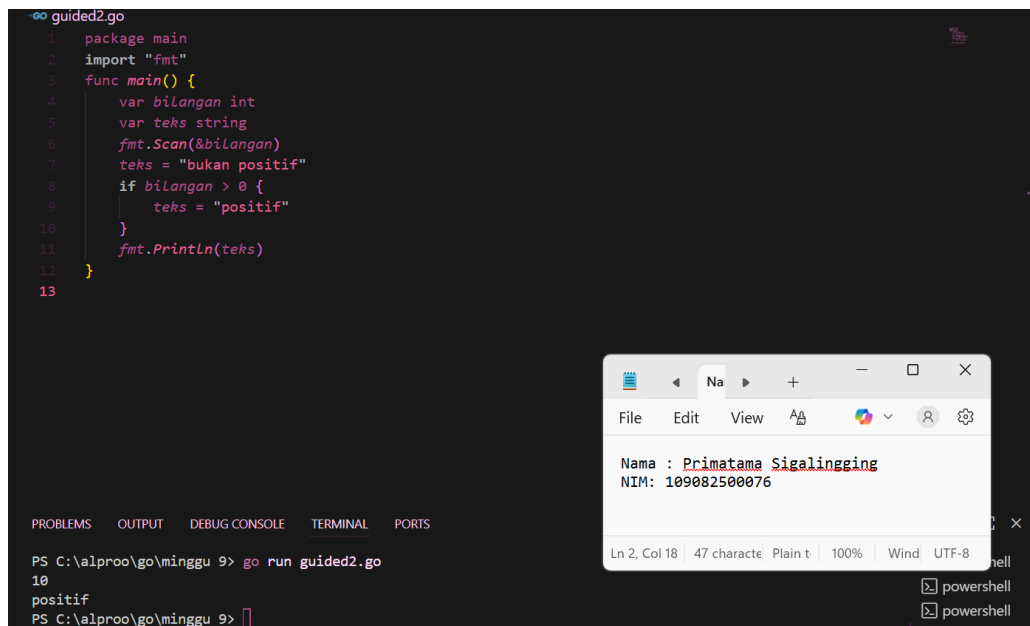
func main() {
    var bilangan int
    var teks string
    fmt.Scan(&bilangan)

    teks = "bukan positif"

    if bilangan > 0 {
        teks = "positif"
    }

    fmt.Println(teks)
}
```

Screenshoot program



The screenshot shows a Go program named `guided2.go` in a code editor. The program defines a `main` function that takes an integer `bilangan` and a string `teks`. It uses `fmt.Scan(&bilangan)` to read input. If `bilangan > 0`, it sets `teks = "positif"`; otherwise, it sets `teks = "bukan positif"`. Finally, it prints `teks` using `fmt.Println(teks)`. Below the code, the terminal shows the command `go run guided2.go` being executed, with the output `10` and `positif`. An inset window shows the user's name `Primatama Sigalingging` and NIM `109082500076`.

```
1 package main
2 import "fmt"
3 func main() {
4     var bilangan int
5     var teks string
6     fmt.Scan(&bilangan)
7     teks = "bukan positif"
8     if bilangan > 0 {
9         teks = "positif"
10    }
11    fmt.Println(teks)
12 }
13
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\alproo\go\minggu 9> go run guided2.go

10

positif

PS C:\alproo\go\minggu 9>

File Edit View A+ 100% Wind UTF-8

Nama : Primatama Sigalingging

NIM: 109082500076

Deskripsi program

➤ Tujuan

Program ini dibuat untuk menentukan apakah sebuah bilangan termasuk bilangan positif atau bukan. Tujuan pembuatannya adalah untuk melatih kemampuan dalam menggunakan struktur kontrol if pada bahasa Go serta memahami bagaimana komputer mengambil keputusan berdasarkan kondisi tertentu.

➤ Proses

Ketika program dijalankan, pengguna diminta memasukkan sebuah bilangan bulat melalui keyboard. Program kemudian menyimpan bilangan tersebut ke dalam variabel dan langsung melakukan pengecekan:

Jika bilangan yang dimasukkan lebih besar dari 0, maka program menampilkan teks “positif”.

Jika bilangan kurang dari atau sama dengan 0, maka program menampilkan teks “bukan positif”.

Proses ini memanfaatkan percabangan sederhana (if statement) untuk menentukan output yang sesuai. Pemisahan logika ini membantu komputer memilih tindakan berdasarkan nilai yang diberikan oleh pengguna.

Program serupa dapat dikembangkan lebih lanjut agar tidak hanya mengecek bilangan positif, tetapi juga menentukan apakah suatu bilangan negatif, nol, ganjil, genap, atau bahkan melakukan klasifikasi lebih kompleks.

➤ Kesimpulan

Program ini menunjukkan bahwa pengambilan keputusan dalam pemrograman dapat dilakukan dengan mudah menggunakan percabangan. Dari program sederhana ini, kita dapat memahami dasar bagaimana komputer mengevaluasi kondisi dan memberikan output yang sesuai secara konsisten dan akurat.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {

    var bilangan int

    var hasil bool

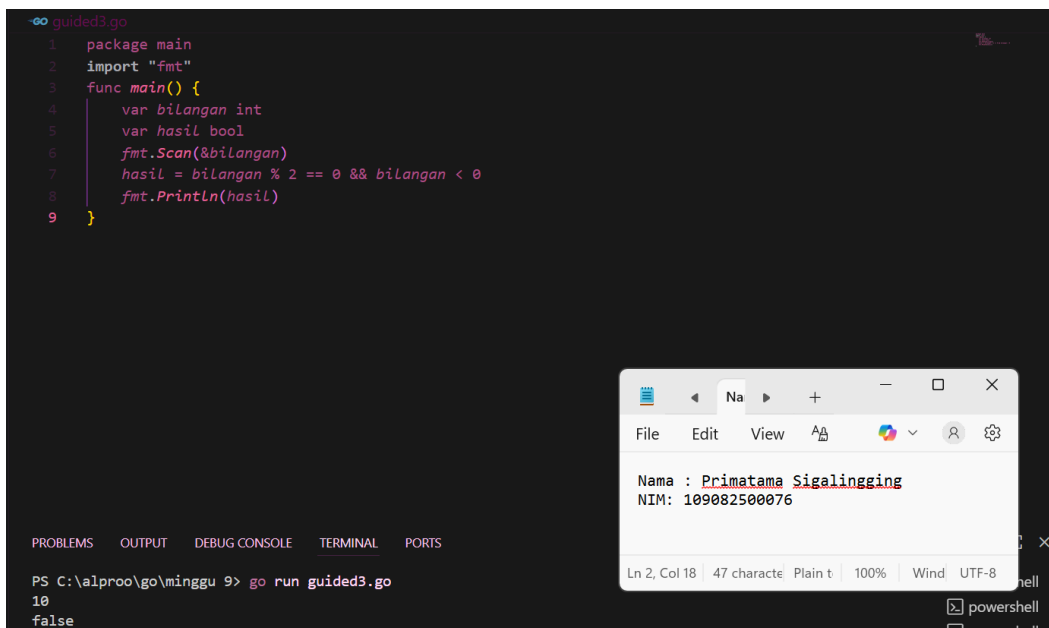
    fmt.Scan(&bilangan)

    hasil = bilangan % 2 == 0 && bilangan < 0

    fmt.Println(hasil)

}
```

Screenshoot program



```
guided3.go
1 package main
2 import "fmt"
3 func main() {
4     var bilangan int
5     var hasil bool
6     fmt.Scan(&bilangan)
7     hasil = bilangan % 2 == 0 && bilangan < 0
8     fmt.Println(hasil)
9 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\alproo\go\minggu 9> go run guided3.go
10
false
```

Nama : Primatama Sigalingging
NIM: 10908250076

Ln 2, Col 18 | 47 character Plain t | 100% | Wind UTF-8

powershell

Deskripsi program

➤ Tujuan

Program ini dibuat untuk mengecek apakah sebuah bilangan termasuk kategori genap negatif. Tujuan dari pembuatan program ini adalah untuk melatih penggunaan operator logika (logical operator) dalam bahasa Go, serta memahami bagaimana komputer melakukan evaluasi terhadap lebih dari satu kondisi secara bersamaan.

➤ **Proses**

Ketika program dijalankan, pengguna diminta memasukkan sebuah bilangan bulat. Program menyimpan bilangan tersebut dalam variabel, kemudian melakukan pengecekan dengan menggunakan dua kondisi:

1. $\text{bilangan} \% 2 == 0 \rightarrow$ mengecek apakah bilangan tersebut genap.
2. $\text{bilangan} < 0 \rightarrow$ mengecek apakah bilangan tersebut negatif.

Kedua kondisi ini digabungkan menggunakan operator logika AND (&&). Artinya, nilai variabel hasil akan bernilai:

true jika bilangan tersebut genap dan negatif sekaligus,
false jika salah satu atau kedua kondisi tidak terpenuhi.

Setelah proses evaluasi selesai, program mencetak nilai boolean tersebut sebagai output.

Program ini merupakan contoh dasar bagaimana menggunakan operator logika untuk membuat pengecekan yang lebih kompleks dan akurat.

➤ **Kesimpulan**

Program ini mengajarkan bahwa komputer mampu mengevaluasi lebih dari satu kondisi secara bersamaan dengan bantuan operator logika. Dengan memadukan logika matematika dan percabangan, kita dapat membuat program yang dapat menentukan kategori suatu bilangan dengan lebih spesifik. Program sederhana ini menjadi pondasi penting dalam memahami konsep evaluasi kondisi dalam pemrograman.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {

    var orang int

    fmt.Print("Masukkan jumlah orang")

    fmt.Scan(&orang)

    var motor intif orang%2 == 0 {

        motor = orang / 2

    } else {

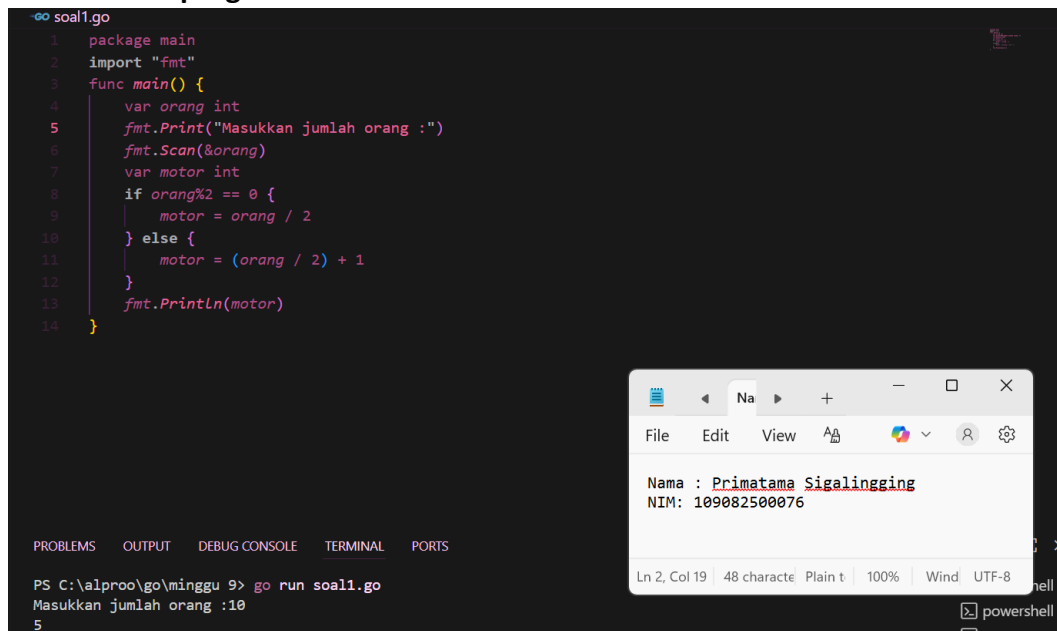
        motor = (orang / 2) + 1

    }

    fmt.Println(motor)

}
```

Screenshoot program



```
soal1.go
1 package main
2 import "fmt"
3 func main() {
4     var orang int
5     fmt.Print("Masukkan jumlah orang :")
6     fmt.Scan(&orang)
7     var motor int
8     if orang%2 == 0 {
9         motor = orang / 2
10    } else {
11        motor = (orang / 2) + 1
12    }
13    fmt.Println(motor)
14 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\alproo\go\minggu 9> go run soal1.go
Masukkan jumlah orang :10
5

Nama : Primatama Sigalingging
NIM: 109082500076

Ln 2, Col 19 | 48 character Plain t | 100% Wind UTF-8

Deskripsi program

➤ **Tujuan**

Program ini dibuat untuk menghitung jumlah motor yang dibutuhkan untuk mengangkut sejumlah orang. Tujuan dari pembuatan program ini adalah untuk melatih penggunaan percabangan (if-else) dalam bahasa Go, sekaligus memahami bagaimana komputer menentukan hasil berdasarkan kondisi tertentu. Program ini juga membantu memahami konsep pembagian dan penanganan kasus bilangan ganjil dalam logika pemrograman.

➤ **Proses**

Ketika program dijalankan, pengguna diminta memasukkan jumlah orang. Nilai tersebut disimpan dalam variabel, kemudian program melakukan proses perhitungan sebagai berikut:

1. Program mengecek apakah jumlah orang merupakan bilangan genap menggunakan ekspresi: $\text{orang} \% 2 == 0$

Jika jumlah orang genap, maka jumlah motor yang diperlukan adalah:

$\text{motor} = \text{orang} / 2$ (karena satu motor dapat membawa dua orang)

2. Jika jumlah orang ganjil, program menghitung jumlah motor dengan:

$\text{motor} = (\text{orang} / 2) + 1$

Tambahan 1 motor diperlukan karena ada satu orang yang tidak memiliki pasangan.

Setelah proses perhitungan selesai, program mencetak jumlah motor yang harus disediakan.

Program ini menunjukkan bagaimana percabangan membantu menentukan hasil yang berbeda berdasarkan kondisi tertentu.

➤ **Kesimpulan**

Program ini memberikan pemahaman bahwa komputer dapat mengambil keputusan berdasarkan kondisi matematis yang diberikan. Dengan menggabungkan operator aritmetika dan percabangan, kita dapat membuat logika yang mampu menyelesaikan masalah dunia nyata, seperti menentukan jumlah kendaraan yang dibutuhkan. Program sederhana ini menjadi dasar penting dalam memahami cara kerja kontrol alur (flow control) dalam pemrograman.

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {
    var bilangan int
    var teks string
    fmt.Scan(&bilangan)
    teks = "bukan"
    if bilangan < 0 {
        teks = "genap positif"
    }
    fmt.Println(teks)
}
```

Screenshoot program

The screenshot shows a Go program in a dark-themed editor. The code defines a `main` function that reads an integer `bilangan` and a string `teks`. It checks if `bilangan` is less than 0; if so, it sets `teks` to "genap positif", otherwise it remains "bukan". The program then prints `teks`. Below the editor, a terminal window shows the command `go run soal2.go` being executed, with the output `10` and `bukan`. An inset window shows a form with the name `Primatama Sigalingging` and NIM `10908250076`.

```
soal2.go
1 package main
2 import "fmt"
3 func main() {
4     var bilangan int
5     var teks string
6     fmt.Scan(&bilangan)
7     teks = "bukan"
8     if bilangan < 0 {
9         teks = "genap positif"
10    }
11    fmt.Println(teks)
12 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\alproo\go\minggu 9> go run soal2.go
10
bukan
PS C:\alproo\go\minggu 9>

File Edit View Aa 100% Wind UTF-8

Nama : Primatama Sigalingging
NIM: 10908250076

Ln 2, Col 19 | 48 character Plain t 100% Wind UTF-8

Deskripsi program

➤ Tujuan

Program ini dibuat untuk menentukan apakah sebuah bilangan termasuk kategori “genap positif” atau bukan. Tujuan pembuatannya adalah untuk melatih penggunaan struktur percabangan dalam bahasa Go serta memahami bagaimana komputer mengevaluasi suatu kondisi berdasarkan nilai input.

➤ Proses

Saat program dijalankan, pengguna diminta memasukkan sebuah bilangan bulat. Program kemudian menyimpan input tersebut ke dalam variabel dan memberikan nilai awal pada variabel teks, yaitu "bukan".

Selanjutnya, program melakukan pengecekan:

Jika bilangan yang dimasukkan kurang dari 0, maka program akan mengubah nilai variabel teks menjadi "genap positif".

Jika kondisi tersebut tidak terpenuhi, maka teks tetap bernilai "bukan".

Kemudian, program mencetak hasil dari variabel teks tersebut.

Walaupun kondisi yang digunakan dalam program (`bilangan < 0` menghasilkan "genap positif") tidak sesuai dengan logika matematika sebenarnya, program ini tetap menunjukkan bagaimana percabangan bekerja untuk menentukan output berdasarkan kondisi tertentu.

Program dapat dikembangkan lebih lanjut agar melakukan pengecekan yang lebih tepat, seperti:

- menentukan apakah bilangan tersebut positif atau negatif,
- mengecek apakah bilangan genap atau ganjil,

- menggabungkan kedua pengecekan tersebut.

➤ Kesimpulan

Program ini memberikan pemahaman dasar tentang penggunaan percabangan (if statement) untuk menentukan suatu kategori berdasarkan nilai input. Dengan logika percabangan, komputer dapat memilih output yang sesuai dengan kondisi yang diberikan. Meskipun sederhana, program ini menjadi dasar untuk memahami bagaimana struktur kontrol digunakan dalam pemrograman.

3. Tugas 3

Source code

```
package main
import "fmt"
func main() {
    var x, y int
    fmt.Print("Masukkan x dan y")
    fmt.Scan(&x, &y)
    var xFaktorY bool
    var yFaktorX bool
    if y % x == 0 {
        xFaktorY = true
    } else {
        xFaktorY = False
    }
    if x % y == 0 {
        yFaktorX = true
    } else {
        yFaktorX = false
    }
    fmt.Println(xFaktorY)
    fmt.Println(yFaktorX)
}
```

Screenshoot program

```
1 package main
2 import "fmt"
3 func main() {
4     var bilangan int
5     fmt.Scan(&bilangan)
6     if bilangan < 0 {
7         bilangan = -bilangan
8     }
9     fmt.Println(bilangan)
10 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\alproo\go\minggu 9> go run soal3.go
Masukkan x dan y :10 5
false
true
PS C:\alproo\go\minggu 9>

Deskripsi program

➤ Tujuan

Program ini dibuat untuk menghitung jumlah motor yang dibutuhkan untuk mengangkut sejumlah orang. Tujuan dari pembuatan program ini adalah untuk melatih penggunaan percabangan (if-else) dalam bahasa Go, sekaligus memahami bagaimana komputer menentukan hasil berdasarkan kondisi tertentu. Program ini juga membantu memahami konsep pembagian dan penanganan kasus bilangan ganjil dalam logika pemrograman.

➤ Proses

Ketika program dijalankan, pengguna diminta memasukkan jumlah orang. Nilai tersebut disimpan dalam variabel, kemudian program melakukan proses perhitungan sebagai berikut:

1. Program mengecek apakah jumlah orang merupakan bilangan genap menggunakan ekspresi:

$\text{orang} \% 2 == 0$

Jika jumlah orang genap, maka jumlah motor yang diperlukan adalah:

$\text{motor} = \text{orang} / 2$ (karena satu motor dapat membawa dua orang)

2. Jika jumlah orang ganjil, program menghitung jumlah motor dengan:

$\text{motor} = (\text{orang} / 2) + 1$

Tambahan 1 motor diperlukan karena ada satu orang yang tidak memiliki pasangan.

Setelah proses perhitungan selesai, program mencetak jumlah motor yang harus disediakan.

Program ini menunjukkan bagaimana percabangan membantu menentukan hasil yang berbeda berdasarkan kondisi tertentu.

➤ **Kesimpulan**

Program ini memberikan pemahaman bahwa komputer dapat mengambil keputusan berdasarkan kondisi matematis yang diberikan. Dengan menggabungkan operator aritmetika dan percabangan, kita dapat membuat logika yang mampu menyelesaikan masalah dunia nyata, seperti menentukan jumlah kendaraan yang dibutuhkan. Program sederhana ini menjadi dasar penting dalam memahami cara kerja kontrol alur (flow control) dalam pemrograman.