LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
“LATIHAN SOAL”

DISUSUN OLEH:

RAIHAN ADI ARBA

103112400071

S1 IF-12-01

DOSEN:

Dimas Fanny Hebrasianto Permadi, S.ST., M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024/2025

SOAL TIPE A

1. **Source code :**

```go
// Raihan Adi Arba_103112400071
package main

import "fmt"

// Fungsi untuk menghitung biaya sewa
func hitungBiayaSewa(DurasiJam int, DurasiMenit int, INIMember bool, nomorVoucher string) float64 {
    tarifPerJam := 5000.0
    if INIMember {
        tarifPerJam = 3500.0
    }

    // Total jam sebagai float
    totalJam := float64(DurasiJam) + float64(DurasiMenit)/60.0

    // Minimal durasi adalah 1 jam
    if totalJam < 1.0 {
        totalJam = 1.0
    }

    biayaTotal := totalJam * tarifPerJam

    // Diskon hanya jika durasi > 3 jam dan voucher valid
    if totalJam > 3.0 && isVoucherValid(nomorVoucher) {
        biayaTotal *= 0.9
    }

    return biayaTotal
}

// Fungsi validasi voucher (5 atau 6 digit angka)
func isVoucherValid(voucher string) bool {
    jumlahDigit := 0
    for _, ch := range voucher {
        if ch < '0' || ch > '9' {
            return false
        }
        jumlahDigit++
    }
    return jumlahDigit == 5 || jumlahDigit == 6
}

func main() {
    var DurasiJam, DurasiMenit int
    var INIMember bool
```

```
    var nomorVoucher string

    fmt.Print("Masukkan durasi (jam): ")
    fmt.Scan(&DurasiJam)

    fmt.Print("Masukkan durasi (menit): ")
    fmt.Scan(&DurasiMenit)

    fmt.Print("Apakah member? (true/false): ")
    fmt.Scan(&INIMember)

    fmt.Print("Masukkan nomor voucher (jika ada): ")
    fmt.Scan(&nomorVoucher)

    biayaSewa := hitungBiayaSewa(DurasiJam, DurasiMenit, INIMember, nomorVoucher)
    fmt.Printf("Biaya sewa setelah diskon (jika memenuhi syarat): Rp %.2f\n", biayaSewa)
}
```

Output :

```
PS D:\raihan\103112400071_LAT2\A> go run 1.go
Masukkan durasi (jam): 2
Masukkan durasi (menit): 30
Apakah member? (true/false): true
Masukkan nomor voucher (jika ada): 123456
Biaya sewa setelah diskon (jika memenuhi syarat): Rp 8750.00
PS D:\raihan\103112400071_LAT2\A>
```

Deskripsi :

Program Latihan soal pertama ini dibuat untuk menghitung biaya sewa berdasarkan durasi pemakaian, status keanggotaan, dan penggunaan voucher diskon. Setelah menerima input dari pengguna, program akan menghitung total biaya sewa dengan mempertimbangkan tarif yang berbeda antara member dan non-member.

Program menggunakan dua fungsi utama:

1. Fungsi hitungBiayaSewa yang bertugas melakukan perhitungan biaya berdasarkan parameter yang diberikan
2. Fungsi isVoucherValid yang berfungsi memvalidasi apakah voucher yang dimasukkan pengguna memenuhi syarat (5 atau 6 digit angka)

Tarif sewa dibedakan berdasarkan status keanggotaan dimana member Rp.3500/jam sementara non-member Rp.5000/jam. Perhitungan durasi dilakukan dengan menjumlahkan jam dan menit (yang dikonversi ke dalam bentuk jam), dengan minimal durasi adalah 1 jam.Program juga menerapkan sistem diskon sebesar 10% yang hanya berlaku jika durasi sewa melebihi 3 jam dan pengguna memasukkan nomor voucher

yang valid. Validasi voucher dilakukan dengan memeriksa apakah voucher tersebut terdiri dari 5 atau 6 digit angka. Dalam fungsi main, program meminta input dari pengguna berupa durasi jam, durasi menit, status member (true/false), dan nomor voucher. Setelah semua data diperoleh, program memanggil fungsi hitungBiayaSewa untuk melakukan perhitungan dan menampilkan hasil perhitungan biaya sewa dalam format rupiah dengan dua angka desimal.

2. Source code :

```
//Raihan Adi Arba_103112400071
package main

import "fmt"

// cekk apakah sebuah bilangan adalah perfect number
func isPerfectNumber(n int) bool {
    sum := 0
    for i := 1; i < n; i++ {
        if n%i == 0 {
            sum += i
        }
    }
    return sum == n
}

func main() {
    var a, b int

    fmt.Print("Masukkan nilai a: ")
    fmt.Scan(&a)

    fmt.Print("Masukkan nilai b: ")
    fmt.Scan(&b)

    // Validasi input
    if a <= 0 || b <= 0 || a > b {
        fmt.Println("Error: a dan b harus bilangan bulat positif dan a <= b")
        return
    }

    // Cetak perfect number dalam rentang a hingga b
    fmt.Printf("Perfect numbers antara %d dan %d: ", a, b)
    found := false
    for i := a; i <= b; i++ {
        if isPerfectNumber(i) {
            fmt.Printf("%d ", i)
            found = true
        }
```

```
    }
    if !found {
        fmt.Print("Tidak ada")
    }
    fmt.Println()
}
```

Output :

```
PS D:\raihan\103112400071_LAT2\A> go run 2.go
Masukkan nilai a: 3
Masukkan nilai b: 13
Perfect numbers antara 3 dan 13: 6
PS D:\raihan\103112400071_LAT2\A> go run 2.go
Masukkan nilai a: 4
Masukkan nilai b: 14
Perfect numbers antara 4 dan 14: 6
```

Deskripsi :

Program dimulai dengan meminta input pengguna untuk memasukkan dua nilai bilangan bulat positif, yaitu a dan b, yang mewakili batas bawah dan batas atas rentang pencarian. Setelah menerima input dari pengguna, program melakukan validasi untuk memastikan bahwa a dan b bernilai positif dan a < b. Jika validasi gagal, program akan menampilkan pesan error dan berhenti. Program menggunakan fungsi isPerfectNumber() yang menerima parameter bilangan bulat dan mengembalikan nilai boolean. Cara kerja fungsi ini yaitu menghitung jumlah semua pembagi dari bilangan tersebut (kecuali bilangan itu sendiri) menggunakan perulangan, kemudian membandingkan jumlah tersebut dengan bilangan aslinya. Jika jumlahnya sama, maka bilangan tersebut adalah perfect number dan fungsi mengembalikan nilai true. Pada fungsi utama main(), setelah validasi input berhasil, program akan melakukan perulangan dari a hingga b untuk memeriksa setiap bilangan dalam rentang tersebut menggunakan fungsi isPerfectNumber(). Jika ditemukan perfect number, program akan menampilkannya. Program juga menggunakan variabel found untuk melacak apakah terdapat perfect number dalam rentang tersebut. Jika tidak ada satupun perfect number yang ditemukan, maka program akan menampilkan pesan "Tidak ada". Output yang ditampilkan mencakup semua perfect number yang ditemukan dalam rentang a hingga b, atau pesan "Tidak ada" jika tidak ditemukan perfect number dalam rentang tersebut.

3. **Source code :**

```go
//Raihan Adi Arba_103112400071
package main

import "fmt"

func main() {
    var x, y int

    fmt.Print("Masukkan nilai x: ")
    fmt.Scan(&x)

    fmt.Print("Masukkan nilai y: ")
    fmt.Scan(&y)

    if x <= 0 || y <= 0 {
        fmt.Println("Error: Nilai x dan y harus bilangan bulat positif")
        return
    }

    // menghitung jumlah pertemuan langsung dari kelipatan x
    jumlahPertemuan := 0
    for hari := x; hari <= 365; hari += x {
        if hari%y != 0 {
            jumlahPertemuan++
        }
    }

    fmt.Println("Jumlah pertemuan dalam setahun:", jumlahPertemuan)
}
```

Output :

```
PS D:\raihan\103112400071_LAT2\A> go run 3.go
Masukkan nilai x: 2
Masukkan nilai y: 5
Jumlah pertemuan dalam setahun: 146
```

Deskripsi :

Program dimulai dengan meminta pengguna untuk memasukkan nilai x dan y menggunakan fungsi fmt.Print dan fmt.Scan. Setelah menerima input dari pengguna, program akan melakukan validasi untuk memastikan bahwa kedua nilai yang dimasukkan adalah bilangan bulat positif. Jika salah satu atau kedua nilai tidak positif, program akan menampilkan pesan kesalahan dan langsung berhenti.

Jika kedua nilai valid, program akan menghitung jumlah pertemuan dalam setahun berdasarkan aturan bahwa pertemuan terjadi setiap kelipatan x hari, kecuali jika hari tersebut juga merupakan kelipatan y (yaitu, pertemuan tidak diadakan pada hari

yang merupakan kelipatan dari y). Perhitungan ini dilakukan dengan menggunakan perulangan for yang dimulai dari hari ke-x sampai hari ke-365, dengan penambahan sebesar x pada setiap iterasi. Pada setiap iterasi, program memeriksa apakah hari tersebut bukan merupakan kelipatan y menggunakan operator modulo (%). Jika hari tersebut bukan kelipatan y, maka jumlah pertemuan bertambah satu.

Setelah perhitungan selesai, program akan menampilkan jumlah total pertemuan dalam setahun menggunakan fmt.Println. Program ini berguna untuk menjadwalkan pertemuan berkala dengan pola tertentu dan menghindari jadwal yang bertabrakan dengan kegiatan lain yang juga memiliki pola teratur

SOAL TIPE B

1. **Source code :**

```go
// Raihan Adi Arb

package main

import (
    "fmt"
)

func main() {
    var a, b int
    fmt.Print("Masukkan nilai a: ")
    fmt.Scan(&a)
    fmt.Print("Masukkan nilai b: ")
    fmt.Scan(&b)

    fmt.Printf("Banyaknya angka ganjil: %d\n", hitungJumlahGanjil(a, b))
}

func hitungJumlahGanjil(min, max int) int {
    jumlah := 0
    for angka := min; angka <= max; angka++ {
        if isGanjil(angka) {
            jumlah++
        }
    }
    return jumlah
}

func isGanjil(n int) bool {
    return n%2 != 0
}
```

Source Code:

```
PS D:\raihan\103go run 1.goAT2\A\B>
Masukkan nilai a: 1
Masukkan nilai b: 1000
Banyaknya angka ganjil: 500
PS D:\raihan\103112400071_LAT2\A\B>
```

Output:

Penjelasan Program :
Program ini meminta pengguna untuk memasukkan dua nilai bilangan bulat (a dan b) sebagai batas rentang angka. Setelah menerima input dari pengguna, program akan menghitung dan menampilkan jumlah angka ganjil yang ada dalam rentang tersebut (termasuk nilai batas). Program ini menggunakan fungsi fmt.Print dan fmt.Scan untuk menerima input dari pengguna. Setelah kedua nilai a dan b diperoleh, program akan memanggil fungsi hitungJumlahGanjil yang bertugas menghitung jumlah bilangan ganjil dalam rentang tersebut. Fungsi hitungJumlahGanjil menerima dua parameter, yaitu nilai minimum (min) dan nilai maksimum (max). Di dalam fungsi ini, program melakukan iterasi dari nilai min hingga max, dan untuk setiap angka dalam rentang tersebut, program memanggil fungsi isGanjil untuk memeriksa apakah angka tersebut merupakan bilangan ganjil. Jika angka tersebut ganjil, maka jumlah bilangan ganjil akan bertambah. Fungsi isGanjil bertugas memeriksa apakah suatu bilangan adalah ganjil atau bukan dengan memeriksa sisa pembagian bilangan tersebut dengan 2. Jika sisa pembagiannya tidak sama dengan 0 (n%2 != 0), berarti bilangan tersebut ganjil dan fungsi akan mengembalikan nilai true. Jika tidak, fungsi akan mengembalikan nilai false.

2. **Source code :**

```
//Raihan Adi Arba_103112400071
package main

import "fmt"

func main() {
   var jumlahRombongan int

   fmt.Print("Masukkan jumlah rombongan: ")
   fmt.Scan(&jumlahRombongan)

   for i := 1; i <= jumlahRombongan; i++ {
      var menu, orang, sisa int
      fmt.Print("Masukkan jumlah menu, jumlah orang, dan status sisa makanan (0 untuk tidak, 1 untuk
iya): ")
```

```
        fmt.Scan(&menu, &orang, &sisa)

        totalBiaya := hitungTotalBiaya(menu, orang, sisa)
        fmt.Printf("Total biaya untuk rombongan %d: Rp %d\n", i, totalBiaya)
    }
}

func hitungTotalBiaya(menu, orang, sisa int) int {
    biaya := biayaMenu(menu)
    if sisa == 1 {
        return biaya * orang
    }
    return biaya
}

func biayaMenu(menu int) int {
    if menu > 50 {
        return 100000
    } else if menu <= 3 {
        return 10000
    }
    return 10000 + (menu-3)*2500
}
```

Output:

```
PS D:\raihan\103112400071_LAT2\A\B> go run 2.go
Masukkan jumlah rombongan: 2
Masukkan jumlah menu, jumlah orang, dan status sisa makanan (0 untuk tidak, 1 untuk iya): 3 12 0
Total biaya untuk rombongan 1: Rp 10000
Masukkan jumlah menu, jumlah orang, dan status sisa makanan (0 untuk tidak, 1 untuk iya): 2 15 1
Total biaya untuk rombongan 2: Rp 150000
PS D:\raihan\103112400071_LAT2\A\B>
```

Deskripsi Program:

Program ini menghitung total biaya makanan untuk sejumlah rombongan berdasarkan parameter yang dimasukkan pengguna. Pertama, program meminta jumlah rombongan, kemudian untuk setiap rombongan, pengguna diminta memasukkan jumlah menu, jumlah orang, dan status sisa makanan. Perhitungan biaya dilakukan melalui fungsi biayaMenu yang menentukan biaya berdasarkan jumlah menu yang dipilih: Rp100.000 untuk menu lebih dari 50, Rp10.000 untuk menu tidak lebih dari 3, atau Rp10.000 ditambah Rp2.500 untuk setiap menu tambahan di luar 3 menu pertama jika jumlah menu antara 4-50. Fungsi hitungTotalBiaya kemudian memperhitungkan apakah ada sisa makanan; jika ada (status 1), biaya dikalikan dengan jumlah orang, jika tidak (status 0), biaya tetap berdasarkan menu saja. Hasil perhitungan ditampilkan untuk setiap rombongan dengan format yang jelas

3. **Source Code:**

```go
//Raihan Adi Arba_103112400071

package main


import "fmt"

func main() {
    fmt.Println("Masukkan bilangan (negatif untuk berhenti):")
    fmt.Println("Jumlah bilangan kelipatan 4:", kelipatanEmpat(0))
}

func kelipatanEmpat(jumlah int) int {
    var x int
    fmt.Scan(&x)

    if x < 0 {
        return jumlah
    }

    if x > 0 && x%4 == 0 {
        return kelipatanEmpat(jumlah + x)
    }

    return kelipatanEmpat(jumlah)
}
```

Output:

```
PS D:\raihan\103112400071_LAT2\A\B> go run 3.go
Masukkan bilangan (negatif untuk berhenti):
2 3 4 5 6 -1
Jumlah bilangan kelipatan 4: 4
PS D:\raihan\103112400071_LAT2\A\B>
```

Deskripsi Program:

Program ini menghitung total biaya makanan untuk sejumlah rombongan berdasarkan parameter yang dimasukkan pengguna. Pertama, program meminta jumlah rombongan, kemudian untuk setiap rombongan, pengguna diminta memasukkan jumlah menu, jumlah orang, dan status sisa makanan. Perhitungan biaya dilakukan melalui fungsi biayaMenu yang menentukan biaya berdasarkan jumlah menu yang dipilih: Rp100.000 untuk menu lebih dari 50, Rp10.000 untuk menu tidak lebih dari 3, atau Rp10.000 ditambah Rp2.500 untuk setiap menu tambahan di luar 3 menu pertama jika jumlah menu antara 4-50. Fungsi hitungTotalBiaya kemudian memperhitungkan apakah ada

sisa makanan; jika ada (status 1), biaya dikalikan dengan jumlah orang, jika tidak (status 0), biaya tetap berdasarkan menu saja. Hasil perhitungan ditampilkan untuk setiap rombongan dengan format yang jelas. Program tidak memiliki validasi input, sehingga pengguna perlu memasukkan data dengan format yang sesuai untuk mendapatkan hasil perhitungan yang benar.

SOAL TIPE C

4. Source code :

```
// Raihan Adi Arba_103112400071

package main

import "fmt"

// hitung jumlah digit
func countDigits(n int) int {
    if n == 0 {
        return 0
    }
    return 1 + countDigits(n/10)
}

// hitung pangkat 10
func power10(exp int) int {
    if exp == 0 {
        return 1
    }
    return 10 * power10(exp-1)
}

func main() {
    var n int
    fmt.Print("Masukkan bilangan bulat positif (>10): ")
    fmt.Scan(&n)

    if n <= 10 {
        fmt.Println("Error: Bilangan harus lebih besar dari 10")
        return
    }

    digit := countDigits(n)

    // Menentukan titik bagi
    split := digit / 2
    if digit%2 != 0 {
        split++
```

```
    }

    divisor := power10(digit - split)

    bil1 := n / divisor
    bil2 := n % divisor

    fmt.Println("Bilangan 1:", bil1)
    fmt.Println("Bilangan 2:", bil2)
    fmt.Println("Hasil penjumlahan:", bil1+bil2)
}
```

Output :

```
PS D:\raihan\103112400071_LAT2\A\B\C> go run 1.go
Masukkan bilangan bulat positif (>10): 12345
Bilangan 1: 123
Bilangan 2: 45
Hasil penjumlahan: 168
PS D:\raihan\103112400071_LAT2\A\B\C>
```

Deskripsi :

Program ini meminta pengguna memasukkan bilangan bulat positif lebih besar dari 10, kemudian memisahkannya menjadi dua bagian dan menampilkan hasil penjumlahannya. Kode ini menggunakan pendekatan rekursif untuk menghitung jumlah digit dalam bilangan dengan fungsi countDigits yang membagi bilangan dengan 10 hingga mencapai 0. Fungsi power10 juga bekerja secara rekursif untuk menghitung nilai 10 pangkat n, yang digunakan sebagai pembagi untuk memisahkan bilangan. Pada fungsi main, program memvalidasi input untuk memastikan bilangan lebih besar dari 10, kemudian menentukan titik pemisahan digit berdasarkan jumlah digit bilangan. Untuk jumlah digit ganjil, titik pemisah dibulatkan ke atas menggunakan perhitungan split++. Setelah mendapatkan pembagi yang sesuai, program memisahkan bilangan asli menjadi dua bagian dengan operasi pembagian dan modulo, lalu menampilkan kedua bilangan hasil pemisahan beserta hasil penjumlahannya

5. **Source code**

```
// Raihan Adi Arba_103112400071
package main

import "fmt"

//  cek apakah semua digit sama
func semuaDigitSama(n, prev int) bool {
    if n == 0 {
        return true
    }
    curr := n % 10
```

```
    if curr != prev {
        return false
    }
    return semuaDigitSama(n/10, prev)
}

// berfungsi embandingkan dua angka dan cek apakah ada digit yang sama
func adaDigitSama(a, b int) bool {
    if a == 0 {
        return false
    }
    digitA := a % 10
    if cekDigit(b, digitA) {
        return true
    }
    return adaDigitSama(a/10, b)
}

//cek kalau ada digit tertentu ada di angka
func cekDigit(n, target int) bool {
    if n == 0 {
        return false
    }
    if n%10 == target {
        return true
    }
    return cekDigit(n/10, target)
}

//  cek apakah semua digit berbeda
func semuaDigitBerbeda(n int) bool {
    if n < 10 {
        return true
    }
    lastDigit := n % 10
    sisa := n / 10
    if cekDigit(sisa, lastDigit) {
        return false
    }
    return semuaDigitBerbeda(sisa)
}

func main() {
    var N int
    fmt.Print("Masukkan jumlah peserta: ")
    fmt.Scan(&N)

    countA, countB, countC := 0, 0, 0
```

```
    for i := 1; i <= N; i++ {
        var nomor int
        fmt.Printf("Masukkan nomor kartu peserta ke-%d: ", i)
        fmt.Scan(&nomor)

        last := nomor % 10
        if semuaDigitSama(nomor, last) {
            fmt.Println("Hadiah A")
            countA++
        } else if semuaDigitBerbeda(nomor) {
            fmt.Println("Hadiah B")
            countB++
        } else {
            fmt.Println("Hadiah C")
            countC++
        }
    }

    fmt.Printf("\nJumlah yang memperoleh Hadiah A: %d\n", countA)
    fmt.Printf("Jumlah yang memperoleh Hadiah B: %d\n", countB)
    fmt.Printf("Jumlah yang memperoleh Hadiah C: %d\n", countC)
}
```

Output :

```
PS D:\raihan\103112400071_LAT2\A\B\C> go run 2.go
Masukkan jumlah peserta: 3
Masukkan nomor kartu peserta ke-1: 333
Hadiah A
Masukkan nomor kartu peserta ke-2: 123
Hadiah B
Masukkan nomor kartu peserta ke-3: 898
Hadiah C

Jumlah yang memperoleh Hadiah A: 1
Jumlah yang memperoleh Hadiah B: 1
Jumlah yang memperoleh Hadiah C: 1
PS D:\raihan\103112400071_LAT2\A\B\C>
```

Deskripsi :

Program ini dirancang untuk mengolah data peserta undian dan membagi hadiah berdasarkan pola angka pada nomor kartu peserta. Kode ini menggunakan pendekatan rekursif dalam beberapa fungsi untuk memeriksa karakteristik digit dalam nomor kartu. Fungsi semuaDigitSama memeriksa apakah semua digit dalam nomor identik dengan cara membandingkan setiap digit dengan digit sebelumnya secara rekursif. Fungsi

adaDigitSama dan cekDigit bekerja sama untuk memeriksa apakah ada digit yang sama antara dua angka yang berbeda, dengan cekDigit secara khusus memeriksa keberadaan digit target dalam suatu angka. Sementara itu, fungsi semuaDigitBerbeda memeriksa apakah semua digit dalam nomor berbeda satu sama lain dengan mencocokkan digit terakhir dengan digit-digit sebelumnya. Pada fungsi main, program meminta jumlah peserta dan nomor kartu masing-masing, kemudian memberikan hadiah berdasarkan kriteria: Hadiah A untuk nomor dengan semua digit sama, Hadiah B untuk nomor dengan semua digit berbeda, dan Hadiah C untuk kasus lainnya. Program mencatat jumlah penerima masing-masing hadiah dan menampilkannya di akhir proses.

6. Source code

```go
// Raihan Adi Arba_103112400071
package main

import "fmt"

func perkalian(n, m int) int {
    if n == 0 || m == 0 {
        return 0
    }
    if m < 0 {
        return -perkalian(n, -m)
    }
    return n + perkalian(n, m-1)
}

func main() {
    var n, m int
    fmt.Print("Masukkan bilangan n: ")
    fmt.Scan(&n)
    fmt.Print("Masukkan bilangan m: ")
    fmt.Scan(&m)

    hasil := perkalian(n, m)
    fmt.Printf("Hasil dari %d x %d = %d\n", n, m, hasil)
}
```

Output :

```
PS D:\raihan\103112400071_LAT2\A\B\C> go run 3.go
Masukkan bilangan n: 5
Masukkan bilangan m: 6
Hasil dari 5 x 6 = 30
PS D:\raihan\103112400071_LAT2\A\B\C>
```

Deskripsi :

Program ini mengimplementasikan operasi perkalian menggunakan pendekatan rekursif tanpa menggunakan operator perkalian bawaan. Kode ini memanfaatkan konsep matematika dimana perkalian dapat dipandang sebagai penjumlahan berulang. Fungsi perkalian menerima dua parameter bilangan bulat n dan m, kemudian menggunakan rekursi untuk menyelesaikan operasi perkalian. Fungsi ini memiliki tiga kasus: jika salah satu bilangan adalah 0, hasilnya adalah 0 (kasus dasar); jika m negatif, hasilnya adalah negatif dari perkalian n dengan m yang dimutlakkan; dan untuk kasus umum, fungsi mengembalikan n ditambah dengan hasil perkalian n dengan (m-1). Pada fungsi main, program meminta pengguna memasukkan dua bilangan n dan m, memanggil fungsi perkalian untuk mendapatkan hasil, dan menampilkan hasil perkalian kedua bilangan tersebut.