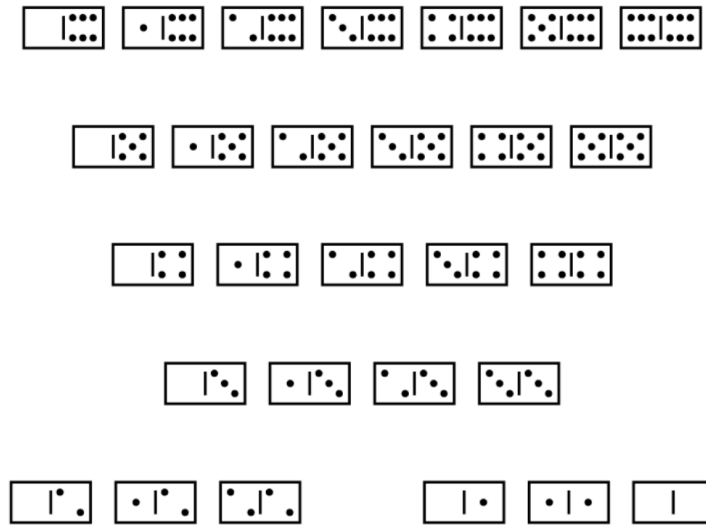


Jurnal Praktikum CII1F4

Modul 12 – Studi Kasus

Pada modul 12 ini kita akan mencoba membuat sebuah game sederhana dengan menggunakan bahasa Go. Asisten Praktikum akan membantu memberikan tutorial pada bagian tertentu dari template program (`template_game.go`) yang diberikan.

Domino western double-six terdiri dari **28 tiles** seperti ilustrasi pada gambar. Setiap **tile** terdiri dari dua bagian yang berisi titik dari 0 hingga 6. Double/balak adalah tile yang memiliki jumlah titik sama di kedua bagian. Nilai dari sebuah tile dihitung dari total titik yang terdapat di kedua bagian (0 sampai 12):



Gambar 1. Set domino western double-six

[Ceme] Domino diacak dengan posisi domino menghadap ke bawah untuk membentuk “**boneyard**” (domino yang sudah teracak). Pemain dapat mengambil **dua tile** dari boneyard dengan posisi menghadap ke atas, lawan juga dapat mengambil dua tile dari boneyard dengan kondisi menghadap ke bawah.

Pemenang ditentukan berdasarkan hal berikut :

- Dua tile yang dimiliki adalah double/balak semua sedangkan lawan tidak memiliki.
- Memiliki total nilai tile yang lebih tinggi dari pada lawan untuk kondisi lainnya

Game yang dibuat harus memiliki fitur berikut ini:

- Mengacak Domino
- Menghitung jumlah skor dan jumlah permainan
- Mengambil domino dan menerima permintaan pengambilan ulang.
- Menerima perintah user (0=done, 9=exit, 1=replace 1st tile, 2=replace 2nd tile)
- Menentukan pemenang dari permainan

Jurnal Praktikum CII1F4

Modul 12 – Studi Kasus

- Memperbarui Skor
- Menghentikan permainan dan menunjukkan statistic hasil permainan.

Contoh: Teks bergaris bawah (underline) adalah masukan dari user

```
Welcome to the Algorithm and Programming Practicum
Boneyard tiles: 28
Your score is 0/0

Dealing ...
Your tiles: (6,1) (5,2)
Boneyard tiles: 24
Decision? 1 // ganti tile pertama
Your tiles: (2,2) (5,2)
Boneyard tiles: 23
Decision? 2 // ganti tile kedua
Your tiles: (2,2) (1,0)
Boneyard tiles: 22
Decision? 0 // done atau lawankan dg dealer
Dealer tiles: (4,3) (2,2)
You lose
Your score is 0/1

Dealing ...
Your tiles: (4,4) (4,2)
Boneyard tiles: 18
Decision? 0 // done atau lawankan dg dealer
Dealer tiles: (4,0) (0,0)
You won
Your score is 1/2

Dealing ...
Your tiles: (3,1) (6,0)
Boneyard tiles: 14
Decision? 0 // done atau lawankan dg dealer
Dealer tiles: (5,3) (3,0)
You lose
Your score is 1/3

Dealing ...
Your tiles: (5,0) (5,5)
Boneyard tiles: 10
Decision? 1 // ganti tile pertama
Your tiles: (3,3) (5,5)
Boneyard tiles: 9
Decision? 0 // done atau lawankan dg dealer
Dealer tiles: (1,1) (3,3)
You won
Your score is 2/4

Dealing ...
Your tiles: (4,1) (2,0)
Boneyard tiles: 5
Decision? 0 // done atau lawankan dg dealer
```

Jurnal Praktikum CII1F4

Modul 12 – Studi Kasus

```
Dealer tiles: (6,4) (6,2)
You lose
Your score is 2/5

Dealing ...
Your tiles: (2,1) (6,5)
Boneyard tiles: 1           // tile di boneyard tinggal 1
Decision? 0                 // done atau lawankan dg dealer
Dealer tiles: (5,4) (5,1)
You lose
Your last score is 2/6       // game berakhir karena tile tinggal 1
Thank you for playing with us.
Your winning rate is 33 %
```

File template_game.go

```
package main
import "fmt"
import "math/rand" // modul random
import "time"      // meminjam modul time untuk membangkitkan seed

type domino struct {
    left,right int
}
const NMAX = 28
type arrDomino [NMAX]domino
type arrInt [NMAX]int

func main(){
    // KERJAKAN MANDIRI dan BOLEH BERTANYA KE ASISTEN PRAKTIKUM
    var playerScore int = 0; var dealerScore int = 0
    var boneyard, playerTiles, dealerTiles arrDomino
    var decision, nBoneyard, pSkor, dSkor, total int
    fmt.Println("Welcome to the Algorithm and Programming Practicum")
    rand.Seed(time.Now().UnixNano())
    // membuat 28 Tile ke dalam boneyard
    buatTile(&boneyard)
    // inisialisasi jumlah Tile di boneyard
    nBoneyard = NMAX
    // total pertandingan masih 0
    total = 0
    // menampilkan jumlah Tile di boneyard
    fmt.Println("Boneyard tiles:",nBoneyard)
    // lakukan loop selama decision adalah 0 dan di boneyard masih cukup Tile
    for decision == 0 && nBoneyard >= 4 {
        // menampilkan skor
        fmt.Printf("Your score is %d/%d \n\n",playerScore,total)
        // dealing atau melakukan pengacakan Tile
        ...
        // membagikan 2 Tile dari boneyard kepada player
        bagiTile(&boneyard, &playerTiles, &nBoneyard)
        // membagikan 2 Tile dari boneyard kepada dealer
        ...
    }
```

Jurnal Praktikum CII1F4

Modul 12 – Studi Kasus

```
// tampilkan Tile player
...
// menampilkan jumlah Tile di boneyard
...
// minta decision dengan memanggil getDecision
...
// lakukan loop selama pilihan adalah 1 atau 2 yaitu replace tile
for decision == 1 || decision == 2 {
    // ganti satu kartu pemain berdasarkan nilai decision
    ...
    // tampilkan Tile player
    printTile(playerTiles,"player")
    // menampilkan jumlah Tile di boneyard
    ...
    // minta decision dengan memanggil getDecision
    getDecision(&decision)
}
// apabila pilihan adalah 0 maka hitung skor dan pemenang
if decision == 0 {
    // tampilan Tile dealer
    ...
    // cari pemenang dengan memanggil subprogram menang
    ...
    // tampilkan pemenang putaran ini
    if pSkor == 1 {
        fmt.Println("You won")
        playerScore++
    }else if dSkor == 1 {
        fmt.Println("You lose")
        dealerScore++
    }
    total++
}
}
fmt.Printf("Your last score is %d/%d \n",playerScore,total)
fmt.Println("Thank you for playing with us.")
fmt.Println("Your winning rate is",playerScore*100/total,"%")
}

func getDecision(decision *int){
    /* I.S. decision telah siap pada piranti masukan
       F.S. decision berisi nilai "0", "1", "2", atau "9", di luar itu minta masukan
       kembali hingga valid*/
    // KERJAKAN MANDIRI
    ...
}

func buatTile(T *arrDomino){
    /* I.S. -
       F.S. T berisi 28 Tile domino */
    var kiri,kanan,i int
    i = 0
    for kiri = 0; kiri <= 6; kiri++ {
        for kanan = 0; kanan <= kiri; kanan++ {
            T[i].left = kiri
```

Jurnal Praktikum CII1F4

Modul 12 – Studi Kasus

```
        T[i].right = kanan
        i++
    }
}

func searchInt(T arrInt, n,x int) int{
    /* mengembalikan posisi x pada array T yang berisi n bilangan bulat, -1 apabila tidak
    ditemukan */
    // KERJAKAN MANDIRI
    ...
}

func acakTile(T *arrDomino, n int){
    /* I.S. terdefinisi array T berisi sejumlah n Tile domino
    F.S. array T tersusun acak */
    // BOLEH DIBANTU ASISTEN PRAKTIKUM
    fmt.Println("Dealing ...")
    var temp arrInt
    var temp2 arrDomino = *T
    // 1. buat array of integer (temp) yang berisi bilangan acak dari 1 hingga n
    // 2. bilangan acak yang dibuat hanya akan ditambahkan ke array (temp) apabila
    belum ada di array
    // 3. temp akan berisi bilangan 1 hingga n dengan susunan acak dan tidak ada
    duplikat
    ...
    // 4. gunakan temp sebagai indeks dari temp2 sehingga tile pada T akan tersusun
    acak
    ...
}

func bagiTile(T,p *arrDomino, n *int){
    /* I.S. terdefinisi array T yang berisi sejumlah n Tile domino
    F.S. pemain p memperoleh 2 Tile terakhir dari T, nilai n berkurang
    catatan: ambil Tile sejumlah m yang terakhir dari T, nilai n berkurang*/
    // KERJAKAN MANDIRI
    ...
}

func replaceTile(T,p *arrDomino, n *int, posisi int){
    /* I.S. terdefinisi array T yang berisi sejumlah n Tile domino, dan array p yang berisi
    sejumlah Tile pemain
    F.S. mengambil satu Tile dari T dan mengganti sesuai nomor posisi Tile pada array
    p*/
    // KERJAKAN MANDIRI
    ...
}

func haveBalak(p domino) bool {
    /* Mengembalikan true apabila p adalah balak, false untuk kondisi sebaliknya */
    // KERJAKAN MANDIRI
    ...
}

func tilePoin(p domino) int {
```

Jurnal Praktikum CII1F4

Modul 12 – Studi Kasus

```
/* Mengembalikan total poin sisi left dan right dari p*/
// KERJAKAN MANDIRI
...
}

func haveTwoBalak(p arrDomino) bool {
/* Mengembalikan true apabila pemain p memiliki 2 Tile balak. */
// KERJAKAN MANDIRI
...
}

func menang(p1, p2 arrDomino, p1Skor, p2Skor *int){
/* I.S. terdefinisi 2 Tile yang masing-masing dimiliki oleh pemain p1 dan p2
F.S. p1Skor bernilai 1 dan p2Skor bernilai 0 apabila p1 menang, atau p1Skor bernilai
0 dan p2Skor bernilai 1 apabila p2 menang */
// KERJAKAN MANDIRI dan BOLEH BERTANYA KE ASISTEN PRAKTIKUM
var poin1, poin2 int
*p1Skor = 0; *p2Skor = 0
// hitung total poin setiap pemain
poin1 = ...
poin2 = ...
// hanya p1 yang memiliki 2 balak
if ... {
...
// hanya p2 yang memiliki 2 balak
}else if ... {
...
// untuk kondisi yang lain
}else{
if poin1 > poin2 {
...
}else{
...
}
}
}

func printTile(p arrDomino, s string){
/* I.S. terdefinisi 2 Tile pada array p, dan sting s yang berisi nama pemain
F.S. menampilkan isi dari p dan s sesuai contoh masukan dan keluaran */
var teks string
if s == "player" {
teks = "Your tiles"
}else{ // s == "dealer"
teks = "Dealer tiles"
}
fmt.Printf("%s: (%d,%d) (%d,%d)\n", teks, p[0].left, p[0].right, p[1].left,
p[1].right)
}
```

Bonus Challenge:

- Modifikasi program sehingga pemain hanya boleh mengganti tile sebanyak 2 kali saja
- Perbaiki/menambahkan error handling untuk mengurangi bug yang mungkin terjadi.