**Laporan Akhir Tugas Besar**   
**“The End of Injustice”**

# Mata Kuliah Algoritma dan Pemrograman 2



Oleh IF-04-05, dengan anggota:

Bertrand Lianto 103072400019

Bima Luthfi Nurhakim 103072400030

Ahmad Nur Fajri 103072430007

Program Studi Informatika

# Direktorat Kampus Surabaya Universitas Telkom 2025

**DAFTAR ISI**

[1. Latar Cerita Pemilihan *Project* 2](#_Toc199963901)

[2. Deskripsi dan Spesifikasi *Project* 3](#_Toc199963902)

[A. Deskripsi *Project* 4](#_Toc199963903)

[B. Spesifikasi *Project* 21](#_Toc199963904)

[3. *User Manual* 23](#_Toc199963905)

[4. Dokumentasi Hasil *Project* 52](#_Toc199963906)

## 1. Latar Cerita Pemilihan *Project*

Kami memilih proyek ini karena salah satu anggota kami yaitu Bertrand Lianto sudah memebuat program game kecil yang dimana program itu sudah di setujui oleh Bu Pima sebagai pengampu mata kuliah Algoritma dan Pemrograman 2 di kelas IF-04-05. Di dalam perkembangan teknologi saat ini hampir setiap orang gemar bermain game untuk menghilangkan rasa penat dan lelah mereka setelah bekerja ataupun sebagainya. Mereka merasakan berbagai tekanan setelah melakukan aktivitas di luar rumah maupun di dalam rumah. Supaya memiliki motivasi kembali dan juga semangat baru orang – orang menghilangkan tekanan tersebut dengan cara bermain game. Jadi kami membuat game kecil yang bisa dimainkan didalam VS Code.

Kelompok kami mengembangkan aplikasi atau games yang berjudul "The end of injustice". Game ini menceritakan seorang player yang harus terus berjuang melawan musuh musuhnya yang tiada habisnya hingga bossnya turun tangan, goals dari game ini adalah membunuh boss dari game ini dan harus bertahan dari musuh musuh yang terus bertambah kuat seiring waktu, namun dengan durasi yang cukup player juga memulihkan hp dan dapat memilih untuk meningkatkan kekuatannya atau membeli peralatan (equipment) untuk membantunya mengalahkan musuh.

Kelompok kami mengimplementasikan segala ilmu dari materi yang telah disampaikan untuk membuat game yang dapat mengurangi kebosanan, mekanisme yang simple tapi hasil tiap pertarungan tidak akan sama sehingga game ini berjalan sangat dinamis.

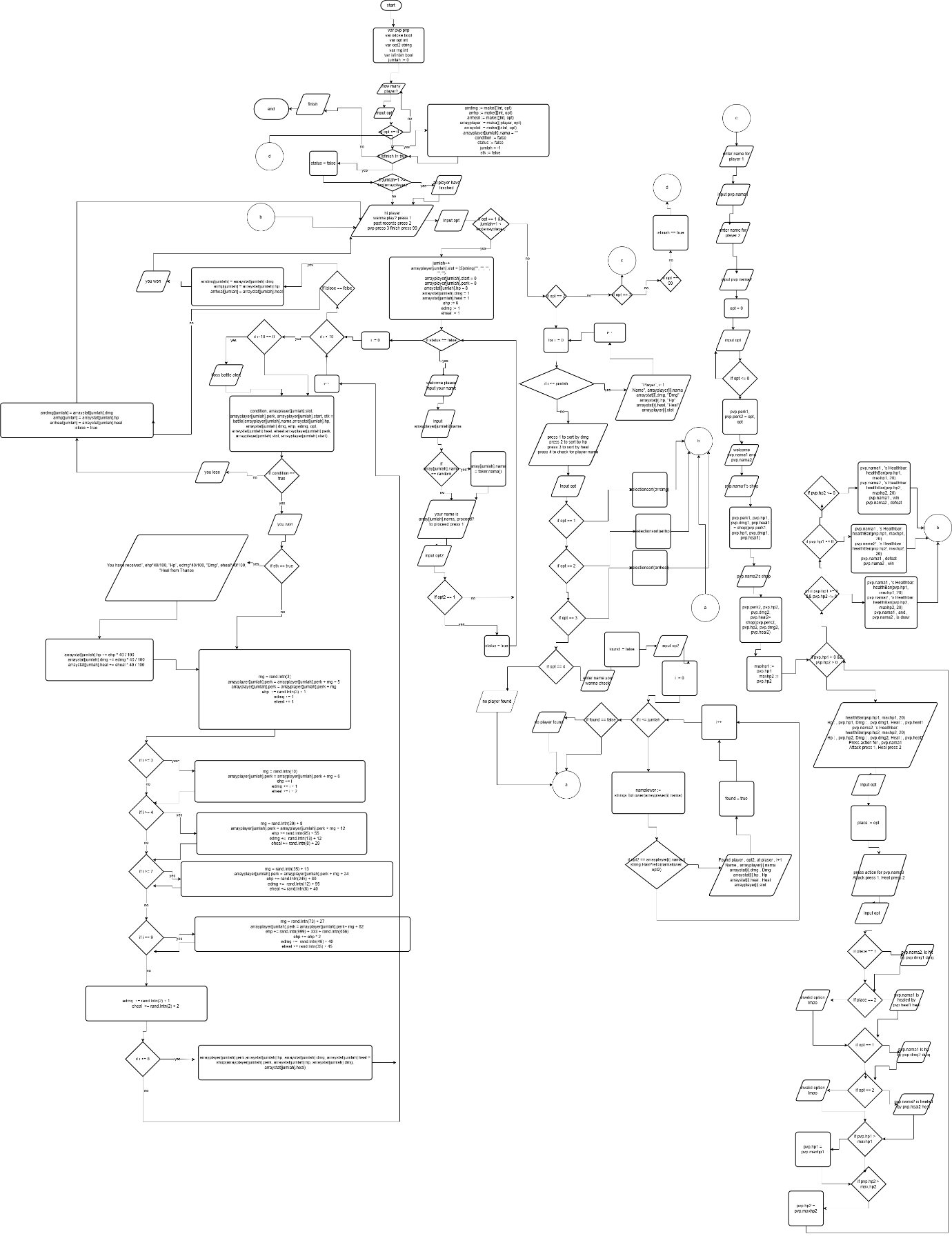
## 2. Deskripsi dan Spesifikasi *Project*

Di dalam implementasi game yang dibuat dengan judul "The end of injustice" menggunakan bahasa pemograman golang yang dikembangkan oleh google adapun fitur menunjang proses berjalannya game adalah sebagai berikut untuk gambarannya.

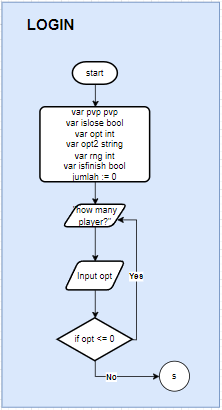
1. Pertarungan memiliki mekanisme seperti seorang player secara bergiliran untuk mengalahkan musuh tetapi tetap bertahan hidup hingga akhir.
2. Toko yang yang digunakan untuk meningkatkan kekuatan atau membeli equipment.
3. Dapat mendaftar lebih dari satu pemain sekaligus untuk mendapatkan lebih banyak kesempatan bermain, dan setiap stat dari pemain sebelumnya akan disimpan sehingga menciptakan leaderboard.
4. Dapat mencari sebuah player dan menampilkan statnya terhadap player itu, atau bahkan mengurutkan stat damage, hp, atau heal untuk melihat hasilnya.
5. Fitur menggandakan uang namun berisiko untuk kehilangan semuanya.

Fitur auto battle untuk terus melakukan penyerangan secara otomatis bila player sudah percaya diri untuk menang.

### **A. Deskripsi *Project***

Skema / Topologi flowchart yang digunakan untuk mengimplemntasikan program yang dibuat adalah sebagai berikut.

Gambar 1 1 Flowchart Pemograman

Dalam proses pembuatan program tugas besar dengan judul "*the end of justice*" telah dibuat beberapa topologi alur kerja program hingga tahap akhir. proyek ini juga telah memenuhi minimum requirement secara modular dengan menggunakan sub-program yang terdapat function dan juga procedure. berikut ini penjelasan alur program Utama yang dijalankan adalah sebagai berikut.

Program dimulai dengan mendeklarasikan variable – variable sederhana untuk membantu dalam menyimpan nilai – nilai pada saat perhitungan dalam peperangan dimulai seperti “*var pvp pvp*” mendeklrasikan variable pvp dengan menggunakan struct.

**type player struct** { // Struct yang digunakan untuk user

    nama string

    perk int

    slot [5] string

    start int

    hp,dmg,heal int

}

pvp ini berfungsi untuk menyimpan data dan status permainan di mode pvp. Kemudian ada variable islose digunakan untuk menentukan pengecekan player apabila player kalah. dan selain itu juga dibuat beberapa struct lagi seperti dibawah ini.

**type enemies struct{** // Struct ini digunakan untuk enemies atau musuh

ehp, edmg, eheal int

**}**

**type pvp struct{** // digunakan untuk mode pvp agar nilai yang dihasilkan tidak masuk kedalam pasrt record

    hp1, hp2, dmg1, dmg2, heal1, heal2, perk1, perk2 int

    nama1, nama2 string

**}**

Jika dilihat dari flowchart program akan berjalan dengan menampilkan beberapa outputan yang memerintahkan user untuk menginputkan jumlah user yang ingin bermain. Disini dibuat yaitu procedure safeScanInt untuk mengecek inputan user “*opt<= 0*”, jika hasil inputan tidak valid maka user akan mengulangi proses inputan jika inputan sesuai maka program akan lanjut berjalan ke step selanjutnya. Gambaran codenya seperti dibawah ini.

**func safeScanInt(opt \*int)** { // nilai dari variabel ini bisa diubah langsung di luar fungsi.

    var input string

    for { // memuat looping hingga user memasukkan inputan dengan benar

        fmt.Scan(&input)

        \_, err := fmt.Sscanf(input, "%d", opt) // mengubah input string menjadi angka

        if err == nil {

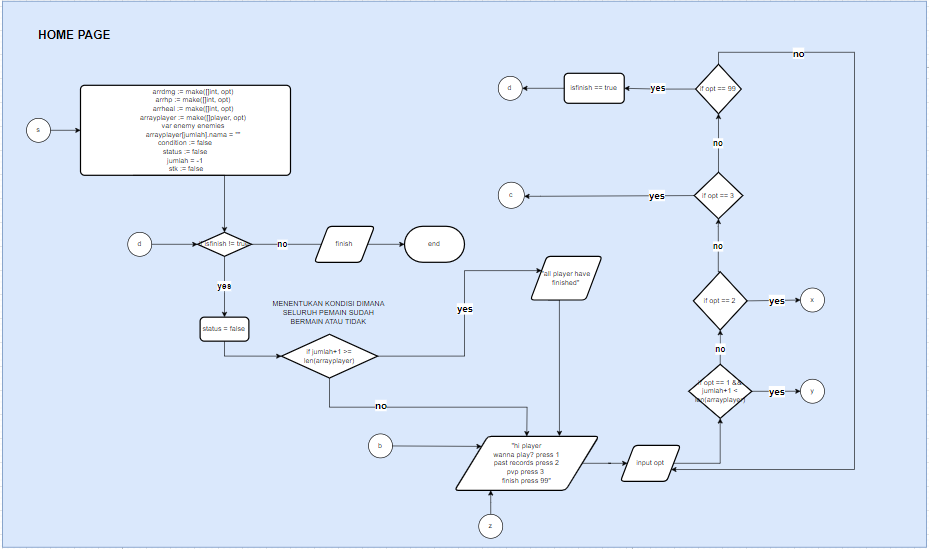
            break //menggunakan break karena ini proses repeat – until sampai kondisi terpenuhi

        }

        fmt.Println("Invalid input. Please input a number.")

    }

}



Di tahap selanjutnya masuk pada bagian homepage setelah user berhasil masuk di tampilan login. Beberapa code di deklarasikan untuk menyokong dan menyimpan beberapa nilai seperti jumlah demage setiap pemain, menyimpan Health Point setiap pemain, Heal, membuat array dari player struct sebanyak inputan user. Dengan beberapa variable yang sudah di buat ini, nantinya juga akan digunakan untuk mencatat dan juga sorting pada menu past record atau leaderboard setiap pemain.

**arrdmg := make([]int, opt)** // digunakan untuk menyimpan damage player

**arrhp := make([]int, opt)** // digunakan untuk menyimpan hp player

**arrheal := make([]int, opt)** // digunakan untuk menyimpan heal player

**arrayplayer := make([]player, opt)** // membuat slice untuk menyimpan data player

**var enemy enemies** // membuat struct untuk menyimpan data musuh

**arrayplayer[jumlah].nama = ""** // Menginisialisasi nama player pertama

**condition := false** // menyimpan kondisi apakah player menang atau kalah

**status := false** // menyimpan status apakah player sudah memasukkan nama

**jumlah = -1**  // menginisialisasi jumlah player

**stk := false** // digunakan untuk menyimpan apakah player mendapatkan stat dari thanos

Proses flowchart diatas dimulai dari melakukan deklarasi variable lalu melakukan looping untuk selalu memunculkan homepage selama user masih belum menginputkan “*opt = “99*” “ yang berarti program selesai.

**for isfinish** != true {

**status** = false // memastikan bahwa menyelesaikan input data dengan benar

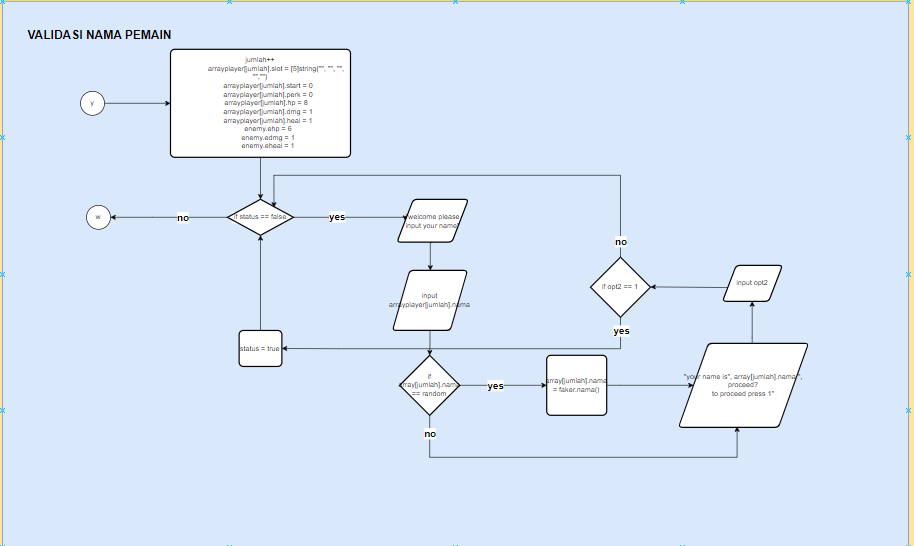
Selanjutnya menambahkan kondisi **if jumlah+1 >= len(arrayplayer) {** di mana ini digunakan untuk mengecek apakah semua pemain sudah selesai bermain atau belum, jika sudah maka program akan menampilkan outputan pada terminal “ *All player have finished* ”dan lalu akan di redirect pada main menu utama. Dan jika belum, player akan langsung redirect pada main utama. Lalu selanjutnya dibuat 4 opsi pilihan untuk user yaitu PVE, Past Record, PVP, dan Finish program. Masing – masing memiliki kondisi nilai yang berbeda misal untuk PVE ia memiliki nilai “1”, untuk Past Record “2”, dan PVP adalah “3”, sedangkan Finish program adalah “4”. Jika player memilih angka selain kondisi diatas maka yang terjadi adalah “**Invalid Option**”

        } **else** { // kondisi terakhir dari semua kondisi

            fmt.Println("**Invalid option**")

            refresh()

        }



Pada bagian flowchart PVE melakukan beberapa deklarasi variable seperti jumlah++ untuk melakukan pergantian pemain selanjutnya. Dan juga beberapa array lainnya.

**jumlah++ /**/ digunakan untuk menambah jumlah player supaya bisa memasukkan data player selanjutnya

**arrayplayer[jumlah].slot = [5]string{"", "", "", "",""}** // menginisialisasi slot player

**arrayplayer[jumlah].start = 0** // menginisialisasi start player

**arrayplayer[jumlah].perk = 0** // menginisialisasi perk player

**arrayplayer[jumlah].hp = 8** // menginisialisasi hp player

**arrayplayer[jumlah].dmg = 1** // menginisialisasi damage player

**arrayplayer[jumlah].heal = 1** // menginisialisasi heal player

**enemy.ehp = 6** // menginisialisasi hp musuh

**enemy.edmg = 1** // menginisialisasi damage musuh

**enemy.eheal = 1** // menginisialisasi heal musuh

setelah variable terdeklarasi terlihat flowchart selanjutnya adalah decision dengan kondisi “if status == false” ini digunakan untuk memastikan player mengisi dan confirm nama mereka untuk akun mereka masing - masing. Lalu jika kondisi tersebut benar maka akan di perintahkan untuk memasukkan nama input oleh program. Membuat kondisi juga jika apabila seorang user ingin menggunakan nama random maka dibuatkan program yaitu

if **arrayplayer**[jumlah].nama == "random" {

                    arrayplayer[jumlah].nama = faker.Name()

                }

Code diatas membuat nama secara random dengan bantuan library dari github untuk menampilkan nama secara acak kepada user [ke - ]apabila user menginputkan data “*random*”. Proses selanjutnya ada konfirmasi untuk menyetujui nama yang sudah dibuat sebelumnya dengan menginputkan data dengan nilai 1 maka status akan berubah menjadi true dan masuk ketahap selanjutnya yaitu decision untuk kondisi if == false adalah false. Code yang digunakan adalah kurang lebih seperti dibawah ini.

**fmt.Println**("Your name is", arrayplayer[jumlah].nama+",", "proceed?")

**fmt.Println**("To proceed please press 1")

**fmt.Scan**(&opt2)

if opt2 == "1" {

status = true

refresh()

        }

Setiap kali akan masuk pada sesi program selanjutnya ia akan memanggil function refresh untuk membersihkan halaman dengan menggunakan procedure.

**func** **refresh()** {

**time.Sleep**(500 \* time.Millisecond)

**screen.Clear**()

**screen.MoveTopLeft**()

    var i int

    i = rand.Intn(40)

    fmt.Println("Cleaning in progress...", i, "%")

    time.Sleep(100 \* time.Millisecond)

    i = 40 + rand.Intn(30)

    fmt.Println("Cleaning in progress...", i, "%")

    time.Sleep(100 \* time.Millisecond)

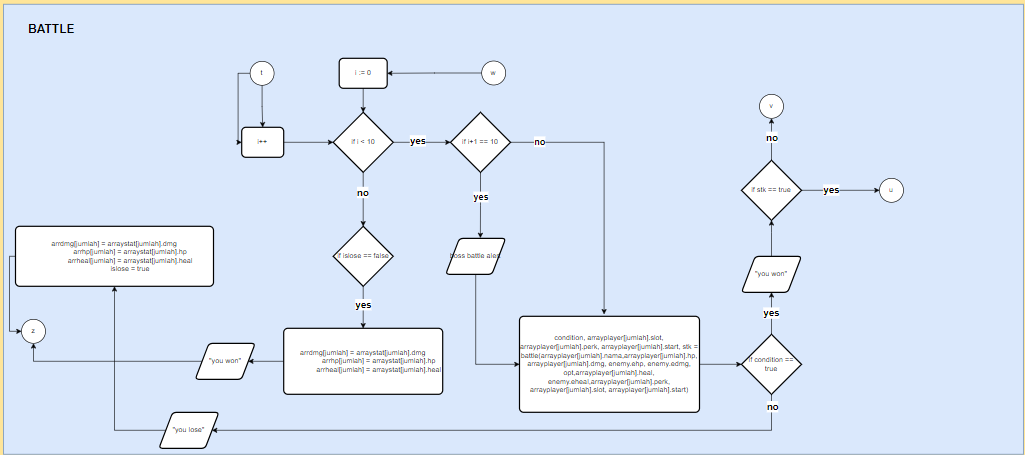
    i = 70 + rand.Intn(30)

    fmt.Println("Cleaning in progress...", i, "%")

    time.Sleep(500 \* time.Millisecond)

    screen.Clear()

    screen.MoveTopLeft()

}

Pada tahap selanjutnya tepatnya di room PVE yaitu battle program melakukan perulangan dengan kapasitas hingga 10 level menggunakan kondisi if i < 10; untuk setiap player, for i := 0; i < 10; i++ {…} dan untuk menentukan level saat ini code yang digunakan adalah fmt.Println("Current level is", i+1) dimana setiap kali ia mengoutputkan nilai I + 1 seiring dia kali memenangkan pertempuran. Dan untuk menentukan kapan ia akan bertemu raja terakhir program dibuat dengan mengeluarkan kondisi

**if i+1 == 10** { // kondisi level user saat ini

                    fmt.Println("Boss battle alert") //memberi kabar kepada user

                }

Ditahap program selanjutnya adalah proses battle dimulai dimana program memanggil function battle dimana ia akan mengambil beberapa nilai dari variabel yaitu.

        condition, arrayplayer[jumlah].slot, arrayplayer[jumlah].perk, arrayplayer[jumlah].start, stk = battle(arrayplayer[jumlah].nama,arrayplayer[jumlah].hp, arrayplayer[jumlah].dmg, enemy.ehp, enemy.edmg, opt, arrayplayer[jumlah].heal, enemy.eheal,arrayplayer[jumlah].perk, arrayplayer[jumlah].slot, arrayplayer[jumlah].start)

Function battle akan memproses nilai variabel tersebut. Berikut adalah spesifikasi yang digunakan didalam functionya.

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Type** | **Fungsi yang digunakan** |
| Name | string | Nama pemain |
| Hp | Integer | Health pemain |
| Dmg | Integer | Effect Demage yang diberikan oleh pemain |
| Ehp | Integer | Health Musuh |
| Edmg | Integer | Effect Demage yang diberikan oleh musuh |
| Opt | Integer | Input pemain |
| Heal | Int | Jumlah heal yang diberikan saat pemain memilih opsi heal |
| Eheal | Integer | Heal musuh |
| Perk | Integer | Jumlah koin yang dimiliki oleh pemain |
| Slot | [5]string (array) | Slot untuk menyimpan equitment pemain yang maximal 5 slot |
| Start | integer | Index slot equitment yang sedang digunakan |

Lalu selanjutnya didalam function battle setelah melakukan deklarasi dan memanggil variabel yang dibutuhkan melakukan pengecekan dan pemilihan equitment dengan menggunakan perulangan. Berikut dibawah ini adalah implementasi code yang dibuat.

opt = 0 // mereset nilai option sebelumnya menjadi 0

**for i := 0; i < len(slot); i**++ { // pengecekan jika slot kosong

if slot[i] != "" { // pengecekan apabila slot tidak kosong maka

fmt.Println("Keep equipment unchanged? 1 for keep, 0 to Edit")

safeScanInt(&opt) // user memilih akan tetap mempertahankan equit atau tidak

}

Break //

}

Jika ada item di slot, user akan ditanya apakah ingin mempertahankan atau mengedit item. Berikut adalah kondisi code yang digunakan untuk menentukan hal tersebut.

**if opt** == 1 { // jika user ingin mempertahankan equitment saat ini yang dimiliki

status = true

} **else** { // jika user ingin mengganti atau mengubah equitment saat ini

status = false

}

selanjutnya terdapat pembelian equitment jika user tidak memilih *opt == “ 1 “* maka selanjutnya adalah akan menampilkan shop equitment Dimana code di implementasikan sebagai berikut.

**for start < len(slot)** { // Loop sampai semua slot terisi atau user memutuskan untuk melewatkan.

if status == true || perk <= 0 { // Jika pengguna ingin menyimpan peralatan atau tidak ada yang tersisa

break

}

**fmt.Println**("Welcome to the equipment store")

...

**safeScanInt**(&opt) // memanggil procedure untuk menjaga agar inputan agar valid sesuai aturan

Berdasarkan code yang dibuat ia melakukan looping untuk memilih equitment di toko, selama masih ada slot kosong dengan kondisi *start < 5* dan koin perk yang mencukupi. Lalu berdasarkan input user, jika item belum dimiliki dan poin perk user mencukupi, item akan ditandai true atau kondisi seperti “*orc = true*” kemudian poin perk user akan berkuran dan item masuk ke dalam slot.

Selanjutnya setelah melakukan pengaturan equitment, effect equitment ini akan langsung aktif setelah melakukan pembelian berikut adalah implementasi code yang dibuat.

**for i := 0; i < len(slot); i++** { // memeriksa slot untuk setiap equitment

if slot[i] == "Oracle" { // exam, Jika Oracle dipilih maka

orc = true // effect oracle diberikan kepada user

maxhp += (hp / 2) // Menghitung untuk meningkatkan Max HP sebesar 50%

hp += (hp / 2) // menghitung Boost HP sebesar 150%

fmt.Println("Oracle granted")

...

}

...

}

Contoh code yang di implementasikan diatas ia sebagai oracle akan melakukan boost terhadap HP 150%, lalu juga ada equitment infinity ia memberikan demage lebih setiap turn. Jadi pada setiap equitment ia memiliki logika perhitungannya masing – masing agar dapat memberikan effect yang berbeda – beda terhapat player.

Selanjutnya untuk melakukan battle loop terhadap game PVE ini code yang dijalankan menggunakan looping for adalah sebagai berikut.

**enemy** = faker.Name() // membuat nama random untuk musuh saat ini dengan bantuan library

...

**for hp > 0 && ehp > 0** { // battle loop

...

fmt.Println("To Attack press 1", "to Heal press 2") // proses pemilihan pada saat battle

if thanos == true && doom == 0 { // Jika equitment Thanos tersedia

fmt.Println("Thanos snap press 3") // tekan angka 3 untuk melakukan aksi terhadap enemy

}

dari salah satu contoh implementasi code dibuat diatas yang digunakan untuk battle loop dan selama kondisi “*HP Player > 0 && HP enemy > 0* ” player akan diberikan opsi nilai yang dapat digunakan untuk menyerang “ 1 ”, “ 2 ” heal, dan “ 3 ” hanya muncul (cooldown session) jika pakai thanos dan belum aktif doom.

Lalu didalam battle dibuat beberapa opsi untuk player yaitu mode auto dan input opsi dimana ini digunakan untuk mempermudah player dalam implementasi dan juga kenyaman pemain didalam game. Dan untuk itu implementasi code yang bisa dibuat adalah sebagai berikut ini.

**if auto == true** { // Jika otomatis benar, maka lewati inputan (kondisi didalam for)

opt = 1 // Jika tidak otomatis, meminta inputan

} **else** {

safeScanInt(&opt)

}

**if opt == 99** { // digunakan untuk memulai auto battle

auto = true

}

**if opt == 1 || auto == true** {

...

if dsc {

// Discipline Whip: critical random

}

...

}

Kode diatas menjelaskan untuk aksi pemain yaitu melakukan penyerangan Dimana jika “*opt == 1 \\ auto == true”* maka akan melakukan eksekusi program yang ada didalamnya. Sebagai contoh lanjutannya adalah jika player pemain memiliki Discipline Whipe atau dalam code adalah *if dsc {…}* maka equitment tersebut akan memberikan effect terhadap player untuk menghasilkan demage critical hit “*dsc == true*” .

Selanjutnya didalam battle juga terdapat kondisi player untuk melakukan heal dengan logika code adalah sebagai berikut.

} **else if opt == 2** { // melakukan heal

Hp = hp + heal // menyimpan nilai heal kedalam HP

fmt.Println(name+"'s", "healed", heal, "Hp") // menampilkan pesan heal

salah satu equitment yang mungkin digunakan oleh player ia akan memiliki cooldown sebanyak 3 turn Dimana logika program berjalan adalah sebagai berikut.

} **else if opt == 3 && thanos == true && doom == 0** { // jika user ingin melakukan snap dengan opt 3

doom += 1 // menambah nilai doom counter selanjutnya

selanjutnya untuk bagian kondisi musuh menyerang dibuat beberapa logika code adalah sebagai berikut.

**if stun == false** { // Jika musuh tidak dalam kondisi stun

**if snap == false** { // Jika efek Thanos Snap belum aktif

opt = rand.Intn(5) // Random nilai 0 sampai 4 untuk menentukan aksi musuh

dari penjelasan code diatas musuh melakukan penyerangan dengan acak dengan melihat nilai isi dari opt Dimana “*opt == 0 , 1 , 2 , 3* ” ia akan melakukan penyerangan dan jika “*opt ==4*” ia akan melakukan heal. Selanjutnya masuk pada logika musuh menyerang, beberapa code di implementasikan adalah sebagai berikut.

**if opt == 0 || (opt >= 1 && opt <= 3)** { // Jika musuh memutuskan untuk menyerang

if dp == true { // Jika nilai dp saat ini memiliki Dodge Potion

opt = rand.Intn(5) // maka akan melakukan pengacakan nilai 0 sampai 4 untuk peluang dodge

if opt == 0 { // jika nilai opt == 0 maka

fmt.Println("You dodged opponent's attack") // mencetak “dodge berhasil”

} else {

if fs == true { // Jika punya Fragile Shield

fmt.Println("Fragile shield took", edmg, "Dmg")

fmt.Println("Fragile shield is now broken")

fs = false // Fragile Shield hanya sekali pakai

} else {

hp = hp - edmg // Jika tidak dodge dan tidak ada shield → terkena damage

fmt.Println(enemy, " hit you", edmg, "Dmg")

}

}

} **else** { // Jika tidak punya Dodge Potion

if fs == true {

fmt.Println("Fragile shield took", edmg, "Dmg")

fmt.Println("Fragile shield is now broken")

fs = false

} else {

hp = hp - edmg // terkena serangan langsung

fmt.Println(enemy, " hit you", edmg, "Dmg")

}

}

Jika digambarkan penjelasan logika serangan dari code diatas adalah maka jika pemain memiliki dodge potion dengan variable “*dp*” maka akan melakukan acak peluang sebesar 20% untuk menghindari serangan. Dan jika gagal maka akan mengecek apakah pemain memiliki fragile shield dengan variable “*fs*”. Jika ya maka serangan akan di serap, dan shield yang dimiliki akan hancur dan jika tidak maka pemain akan menerima demage yaitu “*hp = hp - edmg*”. Kemudian masuk pada logika musuh melakukan heal Dimana code dapat dibuat adalah sebagai berikut.

} **else if opt == 4** { // Musuh memilih untuk heal

if vnm == true { // Jika pemain memiliki Venomous Blade

fmt.Println("Opponent suffered from serious bleed")

ehp = ehp + eheal\*20/100 // Heal hanya 20% (efek bleed)

fmt.Println(enemy, "healed", eheal\*20/100, "Hp")

} **else** {

ehp = ehp + eheal // Heal normal

fmt.Println(enemy, "healed", eheal, "Hp")

}

}

Cara code diatas berjalan adalah sebagai berikut jika nilai “*opt==4*” maka musuh memilih untuk heal. Lalu apabila pemain menggunakan Venomous atau variable “*vnm*” maka musuh akan terkena efek bleed Dimana jika heal hanya “*20%*” dari normal. Jika tidak maka musuh heal dengan nilai penuh disimpan pada “*ehp = ehp + eheal*”.

Selanjutnya yang terakhir adalah akhir dari battle yang dilakukan dimana logika code yang dibuat adalah sebagai berikut.

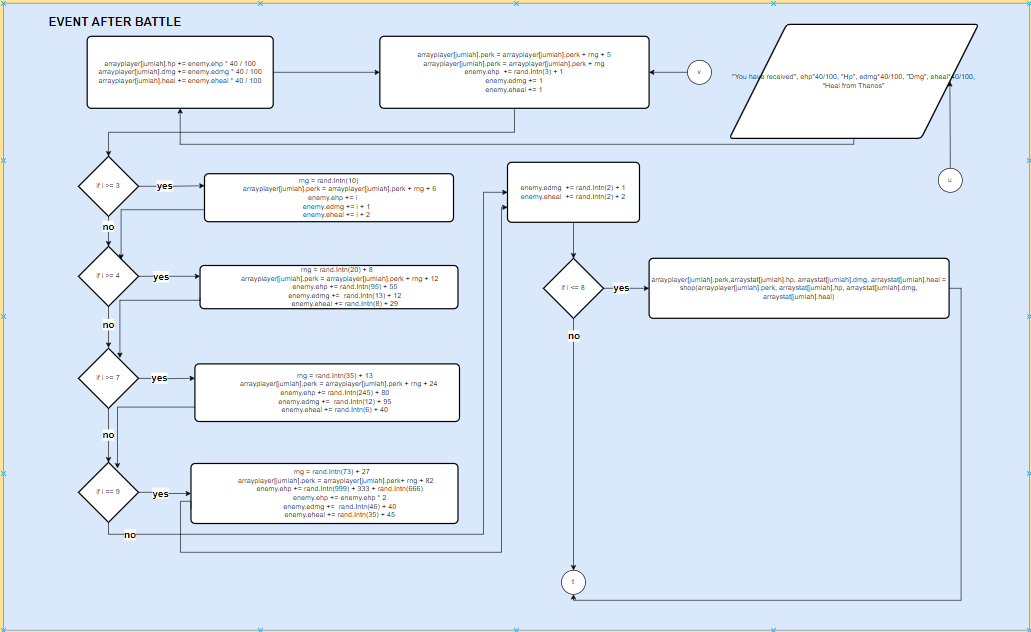
**if hp > 0 && ehp <= 0** { // kondisi jika untuk player menang

return true, slot, perk, start, stk // mengembalikan nilai variabel

}

return false, slot, perk, start, stk // mengembalikan nilai variable

Penjelasan logika untuk code diatas adalah jika pemain menang maka nilai return akan di set sebagai “*true*” dan jika pemain kalah maka nilai return akan di set sebagai “*false*”. Return ini juga akan membawa variable seperti slot, perk, start, stk untuk melakukan proses perhitungan selanjutnya di function main.



Selanjutnya adalah penjelasan pada flowchart pasca battle / setelah perang berakhir Dimana terdapat event yang bisa dilakukan. Setelah battle selesai maka akan ada beberapa bonus yang mungkin akan di dapat. Di samping itu untuk melakukan eksekusi bonus ada code yang harus dijalankan untuk menentukan setiap kondisi. Berikut implementasi code yang bisa dijalankan adalah berikut ini.

**if condition == true** {

// (blok code bagi player yang menang)

} **else** {

// (blok code bagi player yang kalah)

}

Code ini akan dieksekusi setiap kali pemain menyelesaikan satu level yang dijalankan di procedure **battle()**. Dan bagian condition itu menunjukan apakah pemain saat ini menang/true atau kalah/false.

Masuk pada code dibagian yang menang/true, code pertama yang akan dieksekusi adalah apakah menang karena “Thanos” jika ia maka akan mendapatkan 40% dari status musuh yaitu dari variable “hp”,”dmg”,dan ”heal” atau condition “*stk == true*”. Untuk menentukannya berikut code yang di implementasikan.

**if stk == true** {

fmt.Println("You have received", ehp\*40/100, "Hp", edmg\*40/100, "Dmg", eheal\*40/100, "Heal from Thanos")

arraystat[jumlah].hp += ehp \* 40 / 100 // 40% stat musuh

arraystat[jumlah].dmg += edmg \* 40 / 100

arraystat[jumlah].heal += eheal \* 40 / 100

}

Selanjutnya membuat penambahan perk dan scalling musuh dimana pemain akan mendapatkan perk tambahan antara 5 hingga 7 dan ditambah lagi dengan rng, jadi total antara 5 + 0 + 0 hingga 5 + 2 + 2 = 9 perk.

**rng** = rand.Intn(3) // menyimpan nilai acak untuk perk

**arrayplayer[jumlah].perk** = arrayplayer[jumlah].perk + rng + 5 // menyimpan perk yang didapat sebanyak rng + 5

**arrayplayer[jumlah].perk** = arrayplayer[jumlah].perk + rng

setelah membuat code diatas selanjutnya adalah membuat scalling status musuh untuk level selanjutnya dimana logika code yang dibuat adalah sebagai berikut.

**enemy.ehp  += rand.Intn(3) + 1** // menjumlahkan nilai untuk ehp dengan tujuan memperkuat level musuh

**enemy.edmg += 1** // menyimpan nilai 1 untuk meningkatkan kerusakan yang dihasilkan musuh

**enemy.eheal += 1** // menyimpan nilai 1 untuk penyembuhan

Selanjutnya setelah proses ini skala lanjutan akan dibuat seiiring level tinggi Tingkat kesusah musuh akan semakin sulit dimana nyawa dan kerusakan yang ditimbulkan semakin besar. Seperti pada level yaitu “ *i >= 3* “, “*i >= 4*”, “*i >= 7* ”,” *i >= 9* ”.

Di akhir ronde dibuat kondisi untuk random scalling akhir Dimana penambahan acak demage dan heal yang dihasilkan oleh musuh akan meningkat. Untuk membuat random scalling ini code logic yang dibuat adalah sebagai berikut ini.

**enemy.ehp += i**

**enemy.edmg += i + 1** // mendapatkan 2 point extra

Selanjutnya membuat akses untuk menuju shop setiap pemain dengan ketentuan hanya terjadi jika level pemain saat ini adalah “< 10” karena shop akan ditiadakan setelah menang melawan boss. Untuk membuat kondisi seperti ini code yang bisa diterapkan adalah sebagai berikut.

**if i <= 8** {

arrayplayer[jumlah].perk, arraystat[jumlah].hp, arraystat[jumlah].dmg, arraystat[jumlah].heal =

shop(arrayplayer[jumlah].perk, arraystat[jumlah].hp, arraystat[jumlah].dmg, arraystat[jumlah].heal)

}

Setelah masuk pada menu bagi player yang menang terakhir adalah masuk pada kondisi untuk menu code bagi pemain yang kalah dimana status pemain tersebut akan disimpan ke array dan direkap pada variable “i”,”*arrhp*”,”*arrheal*”. Dimana code dapat dibuat adalah sebagai berikut.

fmt.Println("You lose") // Mencetak untuk pernyataan bahwa anda kalah saat ini

**arrdmg[jumlah]** = arraystat[jumlah].dmg // menyimpan stat player untuk keperluan past record

**arrhp[jumlah]** = arraystat[jumlah].hp

**arrheal[jumlah]** = arraystat[jumlah].heal

islose = true // menge-set variable islose sebagai true untuk kekalahan

time.Sleep(2 \* time.Second)

refresh()

break

Untuk membuat kondisi yang sama untuk bagi player yang menang maka logika yang dapat digunakan juga bisa sama seperti yang kalah dimana tujuannya adalah untuk menyimpan hasil rekap pemain setelah bermain untuk di peringkatkan di past record menu yang ada di homepage.

if islose == false { // menge-set variable islose sebagai false untuk kemenangan

**arrdmg[jumlah]** = arraystat[jumlah].dmg

**arrhp[jumlah]** = arraystat[jumlah].hp

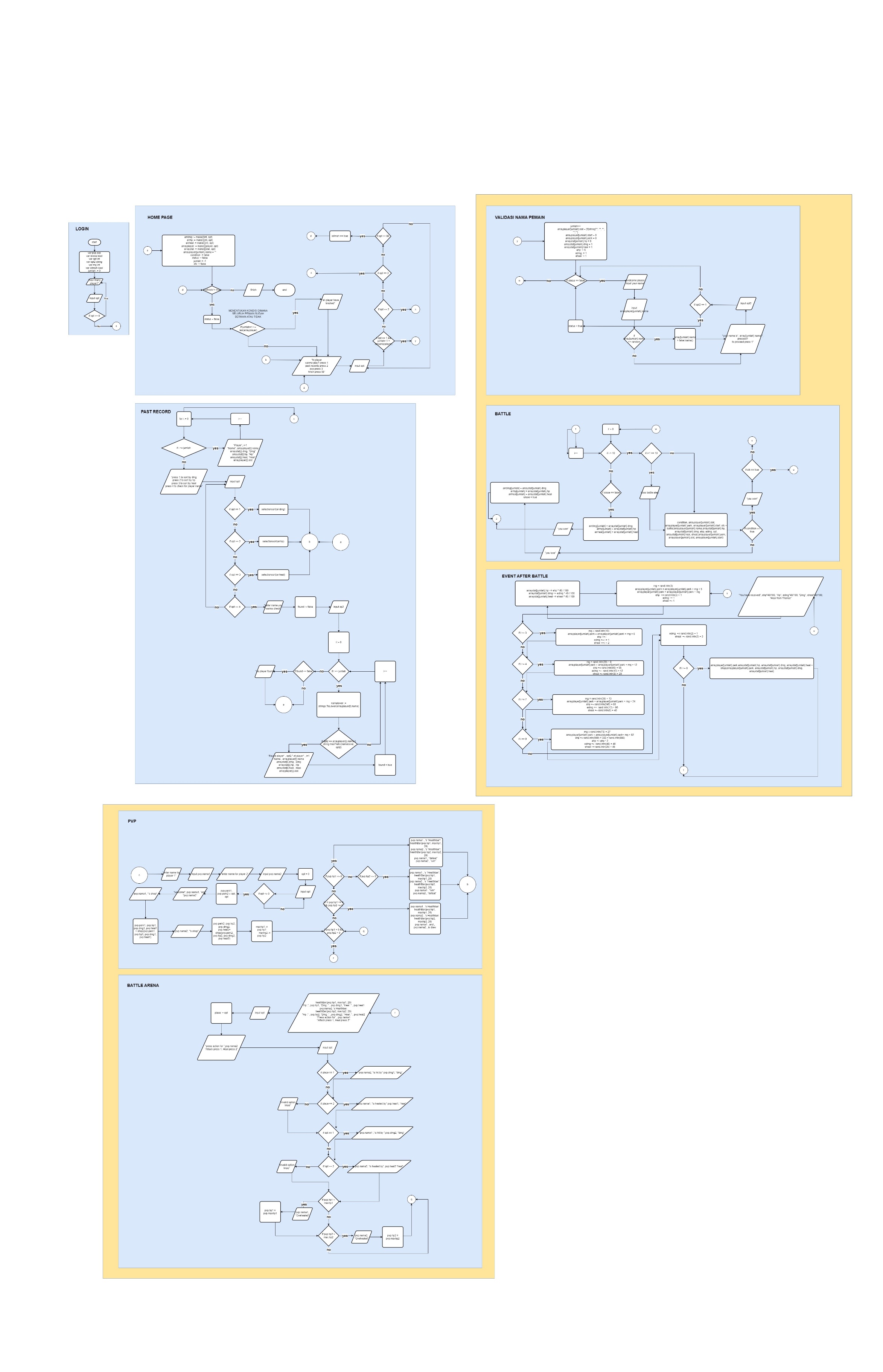
**arrheal[jumlah]** = arraystat[jumlah].heal

fmt.Println("Congratulations you won")

refresh()

}

Jika seluruh permainan sudah bermain maka kembali kondisi seperti pada flowchart yang kedua diatas bahwa semua pemain telah bermain dan kemabli pada tampilan homepage. Untuk saat ini di mode PVE kurang lebih seperti itu untuk logika yang digunakan dalam program.



Selanjutnya adalah masuk pada table past record Dimana ini terdapat pada flowchart yang kedua jika memilih “*opt==2*”. Untuk menjelaskan alur pada past record ini terdapat beberapa function yang digunakan. Berikut adalah jika berhasil masuk pada tab past record.

Fitur ini menampilkan rekap data pemain yang telah bermain, dan menyediakan opsi untuk mengurutkan atau mencari berdasarkan atribut tertentu maka bisa dibilang ini menggunakan searching dan sorting. Data yang disediakan untuk dicari bisa berupa Damage, HP, Heal, Nama pemain. Sekaligus data tersebut dapat diurutkan dari terkecil ke terbesar. Bagaimana fitur ini dapat menyimpan nilai ?, jadi saat pemain menyelesaikan permainan, program akan menyimpan nilai variable seperti dibawah ini.

**arrayplayer[i].nama** = Nama pemain

**arraystat[i].dmg** = Total damage pemain

**arraystat[i].hp** = Total HP pemain

**arraystat[i].heal** = Total heal pemain

**arrayplayer[i].slot** = Daftar equipment yang dimiliki

Jadi pada variable arrayplayer ini merupakan slice yang berisi struct player yang didalamnya berisi “*nama*” ,”*perk*” , “*slot*”, dan “*start*”. Lalu ada juga arraystat yang memiliki struct stat untuk “*hp*”, “*dmg*”, “*heal*”. Jadi masing – masing arrdmg, arrhp dan arrheal merupakan array terpisah untuk menyimpan nilai statistika yang akan disorting. Berikut merupakan alur jika program dijalankan melalui past record.

**for i := 0; i <= jumlah; i++** {

fmt.Println("Player", i+1)

fmt.Println("Name", arrayplayer[i].nama)

fmt.Println(arraystat[i].dmg, "Dmg")

fmt.Println(arraystat[i].hp, "Hp")

fmt.Println(arraystat[i].heal, "Heal")

fmt.Println(arrayplayer[i].slot)

}

Code diatas menampilkan semua data pemain dan jika untuk melakukan sorting berdasarkan damage, HP, dan juga heal masing masing memiliki opsi tersendiri untuk proses sortingnya sendiri digunakan yaitu fungsi **selectionsort**() untuk mengurutkan data array. Berikut implementasinya.

**func selectionsort(arr []int)** {

n := len(arr)

for i := 0; i < n-1; i++ {

min := i

for j := i + 1; j < n; j++ {

if arr[j] < arr[min] {

min = j

}

}

if min != i {

arr[i], arr[min] = arr[min], arr[i]

}

}

}

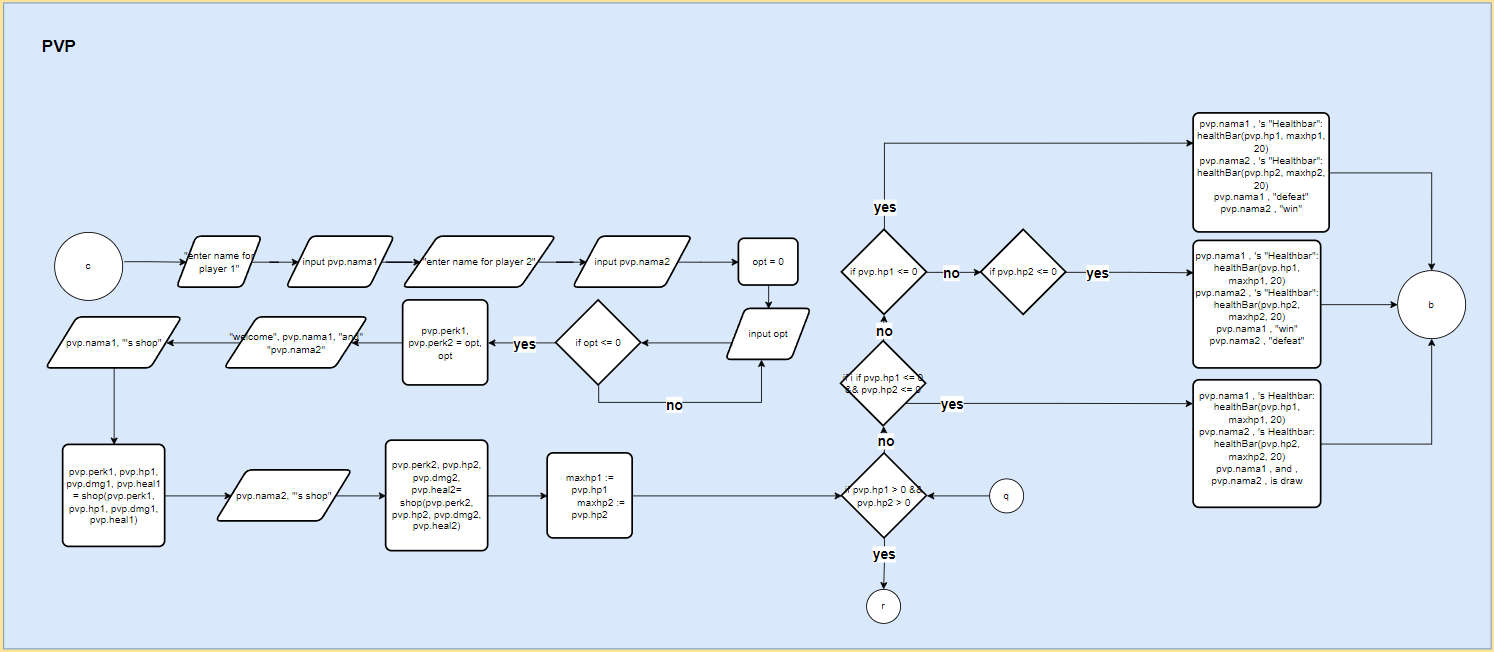
Untuk mencari nama pemain sendiri menggunakan perulangan dimana kondisi untuk mencari nama player ini menggunakan kondisi pencocokan nama atau awalan nama player. Jadi untuk implementasi logika yang bisa digunakan adalah sebagai berikut.

i**f opt2 == arrayplayer[i].nama || strings.HasPrefix(namalower, opt2)**

Jadi code diatas mengambil nilai opt2 atau inputan user, lalu arrayplayer[i].nama merupakan nama pemain ke-i dalam data pemain. Lalu namalower adalah versi huruf kecil semua dari arrayplayer[i].nama. jadi artinya nama yang dimasukkan pengguna persis sama dengan nama pemain atau nama pemain diawali dengan string opt2.

|  |  |
| --- | --- |
| **Penggunaan fungsi** | |
| safeScanInt(&opt) | Membaca input integer dari user menghindari kesalahan. |
| refresh() | Membersihkan layar dan memberikan animasi progress screen agar kesan tidak kaku. |
| strings.HasPrefix() | Mengecek apakah nama pemain diawali dengan string tertentu. |
| strings.ToLower() | Membantu pencocokan nama tanpa case sensitive. |

Jadi tujuan adanya past record ini adalah untuk melihat performa semua pemain yang telah bermain sebelumnya, Membandingkan statistik antar pemain lewat fungsi sortir, Dan mencari pemain tertentu dengan pencarian nama atau menggunakan searching.



Selanjutnya masuk di mode PVP dimana pada saat di homepage atau flowchart yang kedua ia memiliki “*opt == 3*”. Masuk pada flowchart yang ketiga spesifikasi yang digunakan juga banyak karena PVP ini juga penggunaannya hampir sama seperti PVE, Cuma perbedaannya adalah mode ini melawan Human. Berikut contoh beberapa code yang digunakan diawal game.

fmt.Println("Enter name for player 1")

fmt.Scan(&pvp.nama1) // menginputkan data oleh user

fmt.Println("Enter name for player 2")

fmt.Scan(&pvp.nama2) // menginputkan data oleh user

Menginputkan nama setiap pemain yang akan digunakan tracking pada saat battle dan juga past record. Selanjutnya didalam PVP ini ada penggunaan struct dimana ia memiliki fungsi untuk menyimpan semua data pemain. Berikut adalah implementasinya

**type pvp struct** {

hp1, hp2, dmg1, dmg2, heal1, heal2, perk1, perk2 int

nama1, nama2 string

}

Selanjutnya ada function Shop dimana mode PVP ini memanfaatkan shop(), yaitu fungsi umum yang digunakan juga pada mode PVE. Berikut adalah implementasi untuk memanggil function shop didalam mode ini.

**func shop**(perk, hp, dmg, heal int) (int, int, int, int)

di dalam function tersebut selain shop juga terdapat juga opsi untuk bermain minigames & gamble untuk menambahkan poin perk untuk membeli equitment. Dan satu hal lagi didalam minigame ini dibuat sebuah insertion sort untuk mengurutkan array dari angka secara ascending menggunakan insertion sort. Berikut ini adalah implementasi code yang diterapkan.

**func insertionSort(a []int)** {

n := len(a)

for i := 1; i < n; i++ {

k := a[i] // menyimpan elemen saat ini

j := i - 1

for j >= 0 && a[j] > k {

a[j+1] = a[j] // menggeser elemen lebih besar ke kanan

j--

}

a[j+1] = k // menempatkan elemen di posisi yang benar

}

}

Selanjutnya menambahkan function bar untuk meningkat visual didalam UI agar terlihat lebih rapi. Maka dibuat function untuk *healthbar()* berikut code implementasi yang dibuat untuk visualisasi.

**func healthBar**(current, max, barLength int) string {

    temphp := current

    if current >= max {

        current = max

    }

    if current <= 0 {

        current = 0

    }

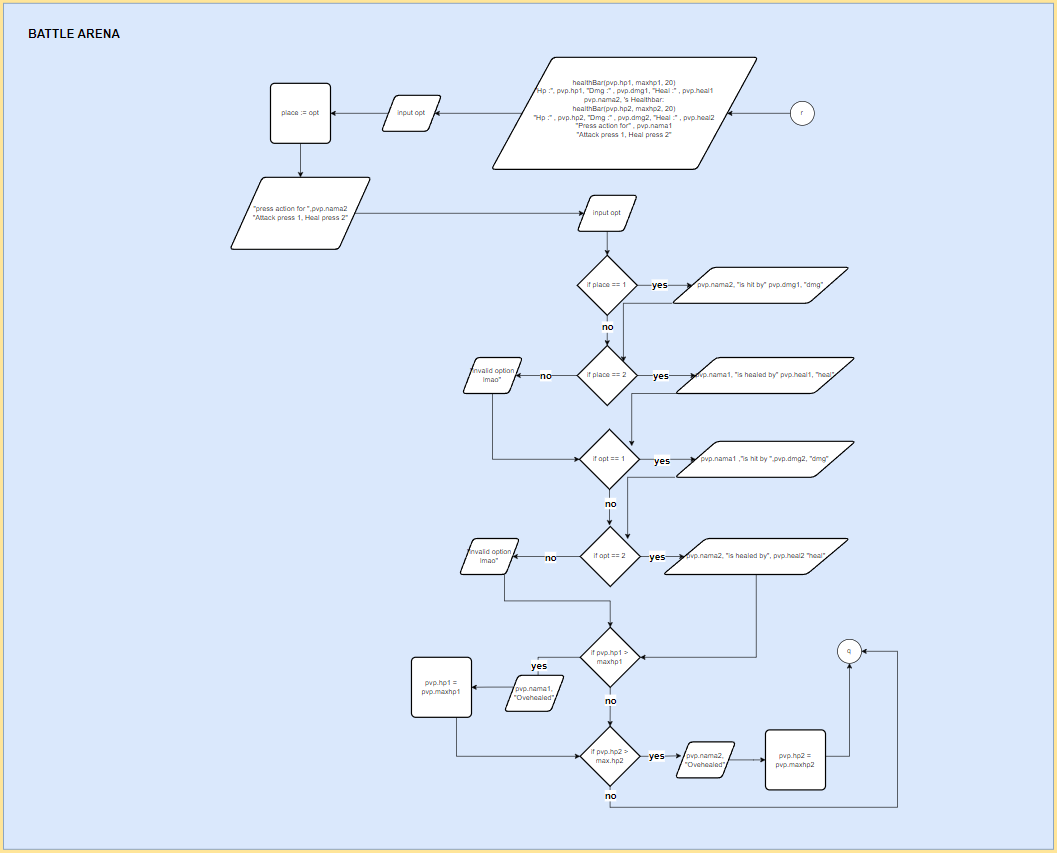
    healthRatio := float64(current) / float64(max)

    filledLength := int(healthRatio \* float64(barLength))

    bar := strings.Repeat("█", filledLength) + strings.Repeat("░", barLength-filledLength)

    return fmt.Sprintf("HP: [%s] %d/%d", bar, temphp, max)

}

Selanjutnya terdapat pula perulangan for dimana ini sebagai loop utama dalam pertarungan yang berjalan terus hingga HP Player <= 0. Berikut adalah implementasi untuk membuat kondisi tersebut adalah sebagai berikut.

**For pvp.hp1 > 0 && pvp.hp2 > 0 {**

Lalu pada PVP ini ia menggunakan percabangan juga yang digunakan untuk mengecek aksi seperti “*opt==1*” dan “*opt == 2*” untuk menyerang dan juga heal, lalu ada juga untuk menentukan hasil pertarungan menang atau kalah dan bisa juga draw, dan ini juga bisa sebagai pembatas untuk HP tidak melebihi Max HP. Berikut adalah salah satu contoh implementasi yang di terapkan adalah sebagai berikut.

**if pvp.hp1 > maxhp1** {

pvp.hp1 = maxhp1

}

Jadi secara singkat untuk alur programnya adalah sebagai berikut user memilih PVP “*opt==3*” kemudian user menginputkan nama player 1 dan player 2. Kemudian menginputkan perkyang diperlukan dalam battle untuk membeli equitment shop. Kemudian masing – masing pemain masuk kedalam shop untuk memilih equitment yang dibuthkan. Lalu melakukan pertarungan, dimana pada saat pertarungan kedua pemain akan bertukar giliran atau melakukan aksi. Kemudian damage atau heal dilakukan sesuai pilihan, dan ditampilkan bar HP untuk track statistics nyawa yang dimiliki. Jika salah satu pemain memili HP <= 0 maka akan muncul hasil akhir yaitu kemenangan, kekalahan, atau draw, dan terakhir loop akan selesai.

### **B. Spesifikasi *Project***

Adapun data yang dapat di tampilkan untuk proyek besar yang digunakan adalah seperti dibawah ini.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Fungsi** | **Jumlah Loop** | **Jumlah If Statement** |
| 1 | refresh() | 0 | 0 |
| 2 | qrefresh() | 0 | 0 |
| 3 | safeScanInt(opt \*int) | 1 | 1 |
| 4 | healthBar(current, max, ...) | 0 | 2 |
| 5 | battle(...) | 4 | 38 |
| 6 | shop(...) | 6 | 14 |
| 7 | main() | 6 | 16 |
| 8 | insertionSort(a […]int) | 2 | 1 |
| 9 | selectionsort(arr […]int) | 2 | 1 |
|  | **Total** | **21** | **73** |

Untuk membangun sebuah program jalannya game yang telah dibuat kami memuat banyak sekali syntax yang digunakan untuk menyokong struktur jalannya game. Berikut ini adalah beberapa data juga yang bisa ditampilkan adalah sebagai berikut.

1. **Struct Player**

**type player struct** {

nama string

perk int

slot [5]string

start int

}

Ini merupakan sebagai tempat untuk table pemain yang menyimpan beberapa nilai yaitu nama, perk, slot, dan juga start dimana nama digunakan untuk menyimpan nama pemain, perk digunakan untuk jumlah perk poin yang dimiliki, slot yang digunakan untuk tempat penyimpanan equitment pemain dengan batas maximal 5, dan yang terakhir adalah start digunakan untuk sebagai penanda index slot yang sudah terisi.

1. **Struct Stat**

**type enemies struct** {

ehp, edmg, eheal int

}

Struct ini digunakan untuk player musuh untuk mewakili semua atribut yang dimilikinya seperti hp,dmg, heal dan yang nantinya data tersebut akan di kirim menuju past record.

1. **Struct PVP**

**type pvp struct** {

hp1, hp2, dmg1, dmg2, heal1, heal2, perk1, perk2 int

nama1, nama2 string

}

Struct PVP ini digunakan untuk hanya mode PVP ia menyimpan nama dan status dua pemain yang sedang bertarung, dan menyimpan semua informasi yang dibutuhkan untuk battle antar pemain (HP, Damage, Heal, Perk). Dan struct ini tidak digunakan di luar ruang lingkup PVP.

1. **Array (arrplayer, arrstat)**

arrayplayer := make([]player, opt)

arraystat := make([]stat, opt)

arrplayer disini digunakan untuk menyimpan seluruh data pemain seperti nama dan perlengkapan. Lalu ada juga arrstat dimana tugasnya untuk menyimpan statistik masing-masing pemain.

1. **Array (arrdmg, arrhp, arrheal)**

arrdmg := make([]int, opt)

arrhp := make([]int, opt)

arrheal := make([]int, opt)

Mendeklarasikan array dimana ia akan digunakan untuk menyimpan statistic pemain terakhhir dari mode PVE dan digunakan untuk pada bagian past record atau menyimpan nilai yang nantinya bisa digunakan untuk sorting berdasarkan salah satu kriteria yang sudah disediakan seperi HP, Damage, Heal dengan menggunakan selection sort.

1. **Array (slot [5]string)**

**slot [5]string{"", "", "", "", ""}**

Mendeklarasikan array yang akan digunakan untuk sebagai slot perlengkapan equitment yang akan digunakan oleh player seperti Oracle, thanos, dan lain sebagainya. Selain itu array ini digunakan pada function battle() untuk menerapkan effect perlengkapan selama pertarungan berlangsung.

1. **Sorting**

**func insertionSort**(a []int) string {

    done := "Minigame Started"

    n := len(a)

    for i := 1; i < n; i++ {

        k := a[i]

        j := i - 1

        for j >= 0 && a[j] > k {

            a[j+1] = a[j]

            j--

        }

        a[j+1] = k

    }

**return done**

}

Menggunakan dua sorting salah satunya adalah insertion sort yang digunakan untuk minigames setelah battle untuk mengurutkan angka secara ascending.

**func selectionsort**(arr []int) string {

    n := len(arr)

    done:= "Stat sorted."

    for i := 0; i < n-1; i++ {

        min := i

        for j := i + 1; j < n; j++ {

            if arr[j] < arr[min] {

                min = j

            }

        }

        if min != i {

            arr[i], arr[min] = arr[min], arr[i]

        }

    }

**return done**

}

Fungsi selection sort digunakan untuk mengurutkan statistic seperti damage, hp, dan heal pada past record.

1. **Searching**

**for i := 0; i < len(arr); i++** {

                fmt.Scan(&guess[i])

            }

            same := true

            for i := 0; i < len(arr); i++ {

                if guess[i] != arr[i] {

                same = false

                break

                }

            }

Digunakan oleh system untuk mencocokan hasil inputan user dengan hasil program terkait. Jika salah maka program akan berhenti.

**for i := 0; i < len(slot); i++** {

        if slot[i] == "Oracle"{

            orc = true

            maxhp += (hp/2)

            hp += (hp/2)

            fmt.Println("Oracle granted")

            fmt.Println("You've received blessing", hp/2 ," hp from oracle")

        }

        if slot[i] == "Infinity power"{

            inf = true

            fmt.Println("Infinity power granted")

        }

Digunakan untuk mengecek equitment apa saja yang dimiliki oleh user atau pemain. Dan diatas merupakan contoh potongan code saat ini yang di implementasikan. lalu setiap equitment yang dimiliki mereka akan mengeluarkan effect yang berbeda – beda yang akan diaktifkan jika terdapat pada slot pemain dan kapasitas maximal yang dimiliki adalah berjumlah 5.

1. **Function Utama**

**func main(){**

**… }**

Merupakan fungsi utama dari program ini karena ia fungsi khusus titik awal eksekusi program yang dibuat dan juga function ini tidak menerima input parameter maupun mengembalikan output seperti return value.

**func healthBar**(current, max, barLength int) string {

    temphp := current

    if current >= max {

        current = max

    }

    if current <= 0 {

        current = 0

    }

    healthRatio := float64(current) / float64(max)

    filledLength := int(healthRatio \* float64(barLength))

    bar := strings.Repeat("█", filledLength) + strings.Repeat("░", barLength-filledLength)

    return fmt.Sprintf("HP: [%s] %d/%d", bar, temphp, max)

}

Function ini digunakan untuk membuat representasi visual untuk nyawa setiap pemain pada saat battle dimulai. Ia menampilkan presentase HP yang dibandingkan dengan HP Maximal. Visual ini diperlukan untuk mempermudah pemain dalam tracking nyawa yang mereka miliki sehingga dapat memperhitungkan untuk menghindari kekalahan.

**func insertionSort(a []int) string** {

    done := "Minigame Started"

    n := len(a)

    for i := 1; i < n; i++ {

        k := a[i]

        j := i - 1

        for j >= 0 && a[j] > k {

            a[j+1] = a[j]

            j--

        }

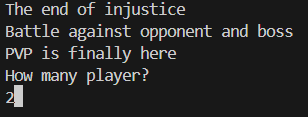
        a[j+1] = k

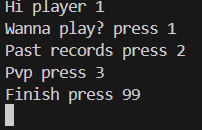
    }

    return done

}

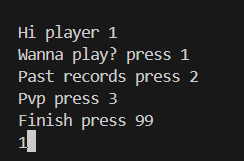
## 3. *User Manual*

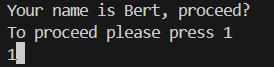
Pertama kita bisa memulai program dengan mengetik “go run battleFinal2.go” yang akan mengoutput tampilan sebagai berikut. Lalu kita masukkan input berapa banyak pemain yang kita inginkan sebagai contoh 2 pemain.

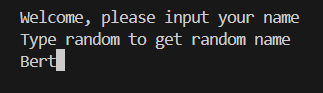
Lalu tampilan akan berubah menjadi loading yang berjalan selama beberapa detik(tidak bisa di screenshoot karena terlalu cepat). Setalah itu akan muncul output lain sebagai berikut.

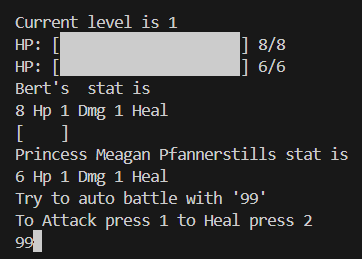
Disini kita bisa memilih input antara “1”, “2”, “3”, dan “99”.

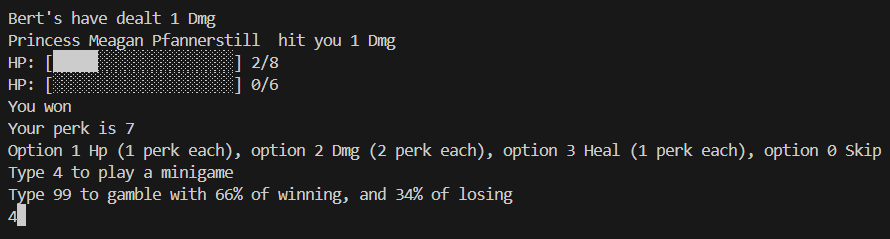
1. : berfungsi untuk memulai permainan solo. Yang dimana sebelumnya kita menginput berapa banyak pemain yang akan diubah secara otomatis menjadi 1 pemain dan akan melawan komputer.
2. : berfungsi untuk melihat ringkasan setelah bermain.
3. : berfungsi untuk melakukan permainan Pvp(duel), yang dimana sebelumnya kita menginput beberapa pemain.

**99** : berfungsi untuk keluar dari program.

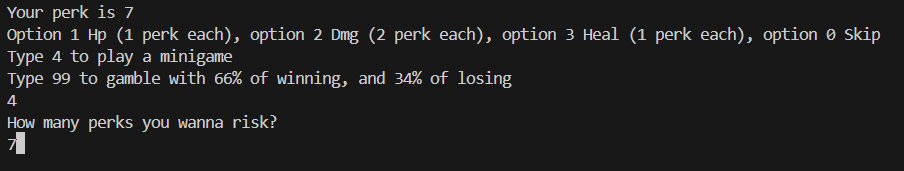
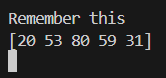
Jika kita menginput “1” maka akan muncul output sebagai berikut.

Disini kita diminta untuk menginput nama atau kita bisa mendapatakan nama random dengan menginput “random”. Kita akan mencoba menginput “Bert” yang akan mnegoutput sebagai berikut.

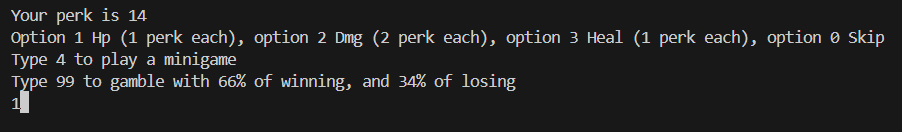
Disini ditanyai apakah ingin lanjut dengan nama “Bert” atau tidak. Jika tidak maka input selain “1” yang akan kembali ke output sebelumnya(menginput nama secara manual atau random). Disini kita akan menginput “1” untuk lanjut.

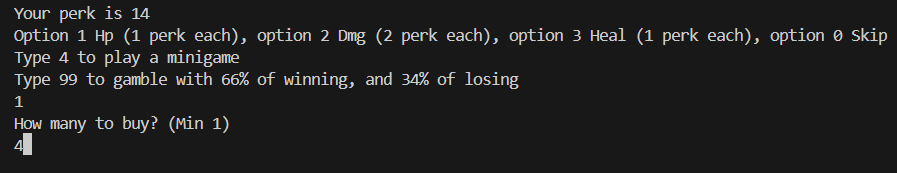
Disini akan muncul output status player kita dan status lawan. Disini terdapat tiga pilihan input yaitu “99” untuk tanding secara otomatis, “1” untuk menyerang lawan, “2” untuk mengobati player kita. Disini kita akan auto-battle dengan menginput “99”.

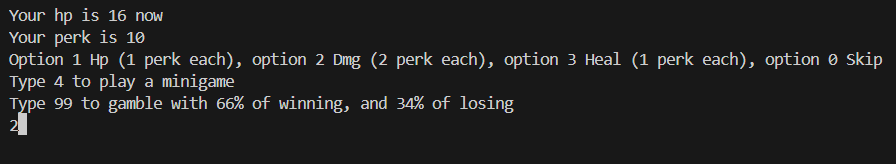
Disini terdapat tampilan setelah menginput “99”. Terdapat status pemain kita dan lawan setelah bertarung. Disni kita menang. Lalu kita mendapatkan “7 perk” yang dimana perk ini gunakan untuk meng-upgrade status pemain kita, melakukan gamble(judi), dan membeli item. Terdapat beberapa opsi yaitu “1” untuk menambah bar hp, “2” untuk menambah dmg, “3” untuk mengobati pemain kita, “0” untuk melewati sesi ini, “4” untuk bermain mini game(untuk menggandakan perk atau kehilangan perk), dan “99” untuk gambling menang atau kalah untuk mendapatkan tambahan perk. Disini kita input “4”.

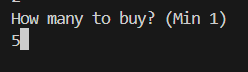
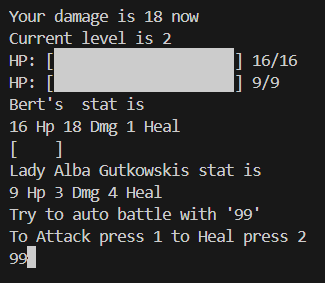
Lalu akan muncul output seperti ini, kita akan merisikokan 7 perk dengan menginput “7”.

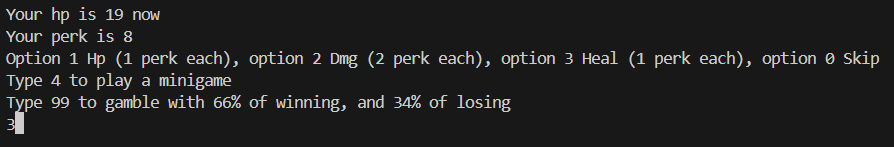
Lalu akan mengoutputkan seperti ini, yang dimana akan ada 5 angka acak yang harus diurutkan dari terkecil-terbesar. Dan output ini hanya akan bertahan 2 detik.

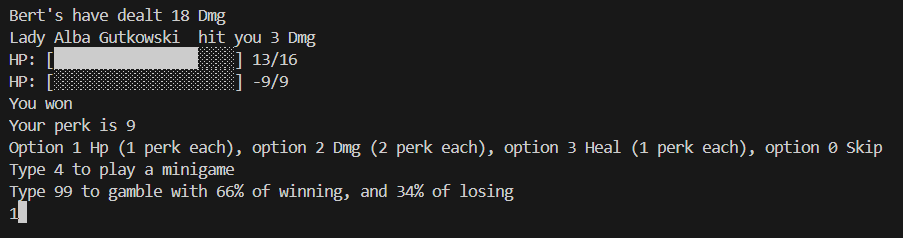
Lalu kita input angka-angka tersebut.

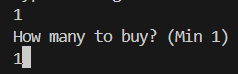
Akan muncul output seperti ini. Karena kita sudah mengurutkan angka-angka dengan benar, maka perk kita berlipat ganda menjadi 4 perk. Disini kita akan membeli hp dengan menginput “1”.

Akan muncul output string seperti ini. Kita akan membeli 4 hp dengan menginput “4”.

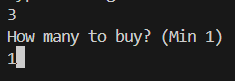
Lalu muncul tampilan seperti ini, hp kita menjadi 16, disini kita bisa membeli dmg dengan menginput “2”.

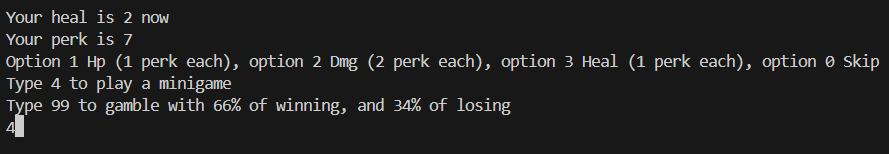
Akan muncul output string seperti ini. Kita akan membeli 5 dmg dengan menginput “5”.

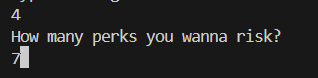
Akan muncul output seperti ini, dmg kita menjadi 18. Karena harga 1 dmg adalah 2 perk dan perk kita tadi 10 lalu kita membeli 5 dmg, maka perk kita habis. Ketika habis akan langsung menuju ke level berikutnya. Disini ditampilkan status player kita dan status lawan. Kita input “99” untuk auto-battle.

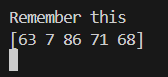
Akan muncul output seperti ini, kita menang dan mendapatkan 9 perk, kita akan membli hp dengan input “1”.

Akan muncul tampilan seperti ini, kita akan membeli 1 hp dengan input “1”.

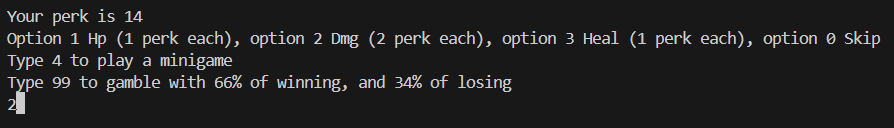
Akan muncul tampilan seperti sebelumnya, hp kita menjadi 19 dan kita akan mengobati player kita dengan input “3”.

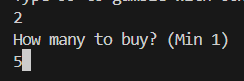
Akan muncul tampilan seperti ini, kita akan membeli 1 heal dengan input “1”.

Akan muncul output seperti ini, heal kita sekarang adalah 2. Kita input “4” untuk bermain mini game.

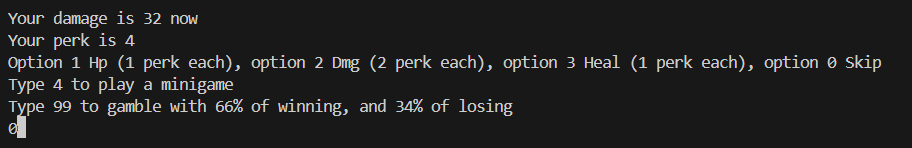
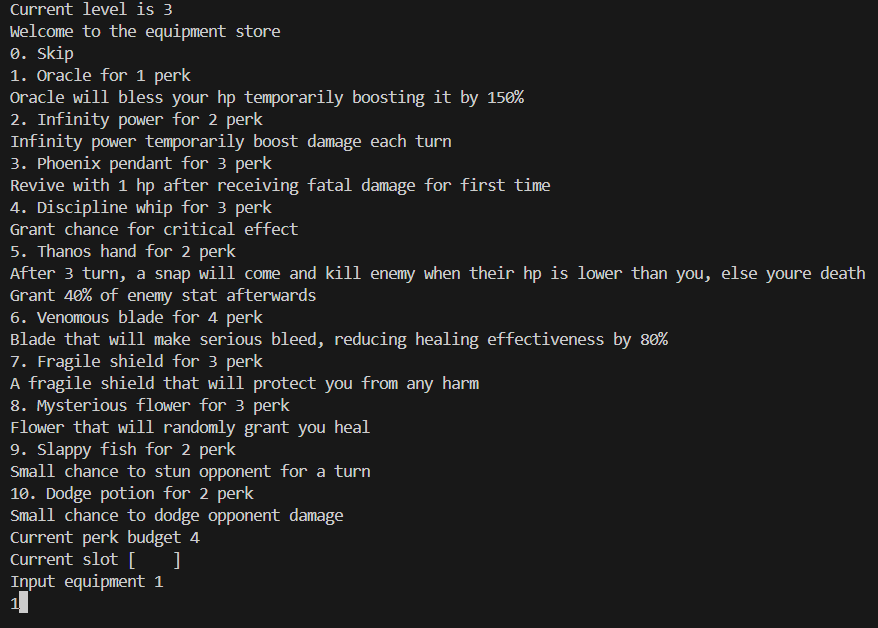
Akan muncul tampilan seperti ini, kita akan merisikokan semua perk kita. Dengan input “7” karena perk kita tersisa 7.

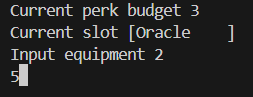
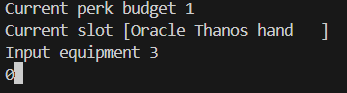
Lalu akan mengoutputkan seperti ini, yang dimana akan ada 5 angka acak yang harus diurutkan dari terkecil-terbesar. Dan output ini hanya akan bertahan 2 detik.

Lalu kita akan menginput angka-angka yang sudah kita urutkan.

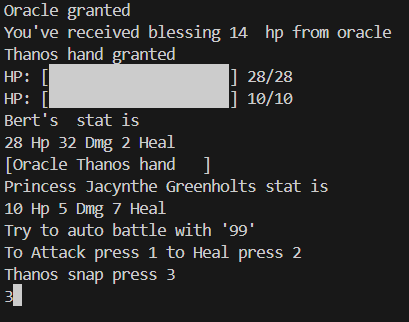
Lalu akan muncul tampilan seperti ini, perk kita menjadi 14. Kita input “2” untuk membeli dmg.

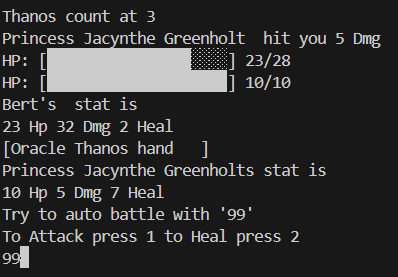
Lalu kita beli sebanyak 5 dengan input “5”.

Disini dmg kita menjadi 32 dan perk kita tersisa 4, kita akan pergi ke shop dengan menginput “0”.

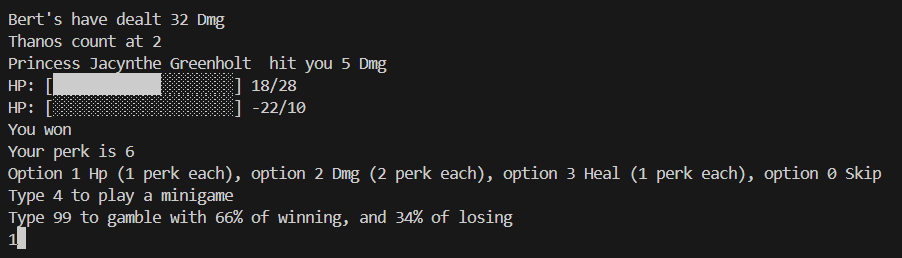
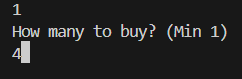
Akan muncul tampilan level saat ini, semua item, dan perk kita. Kita akan membeli item oracle dengan input “1”.

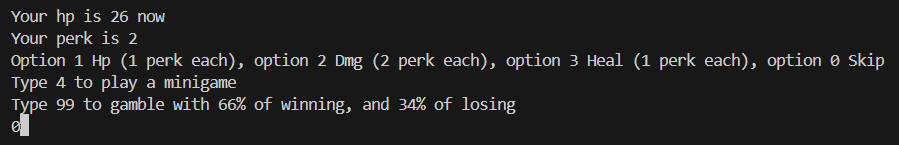
Akan muncul tampilan di Curernt slot bahwa item oracle sudah dibeli, kita akan membeli item thanos snap dengan menginput “5”.

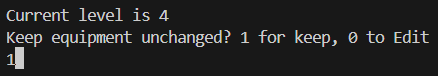
Disini ditampilkan bahwa item thanos snap sudah kita beli dan sisa perk kita ada 1, kita input “0” untuk skip.

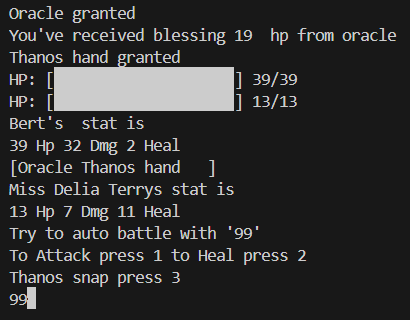
Disini muncul tampilan status kita dan status lawan. Kita input “3” untuk menggunakan item thanos snap.

disini muncul tampilan hp bar kita berkurang, karena kita sudah menggunakan Langkah kita untuk memakai item. Disini kita input “99” untuk auto-battle.

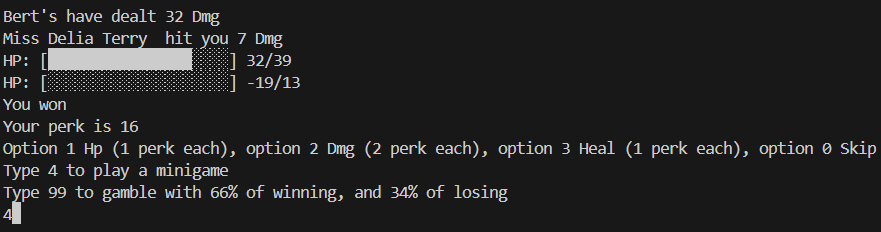
Setelah iut, akan muncl tampilan seperti ini dan kita menang dan perk kita menjadi 6. Kita akan mebeli hp dengan input “1”.

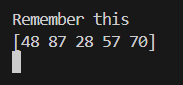
kita akan membeli hp sebanyak 4 dengan input “4”.

Hp kita menjadi 26, lalu kita input “0” untuk skip.

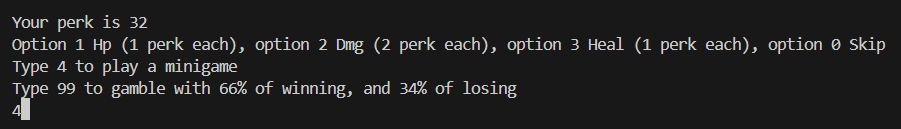
Lalu akan muncul tampilan seperti ini, kita input “1” untuk tidak merubah item kita.

Lalu akan muncul tampilan seperti ini, terlihat status player kita dan status lawan. Kita input “99” untuk auto battle.

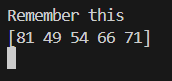
Kita menang dan perk kita menjadi 16, kita input “4” untuk bermain mini game.

Kita akan merisikokan semua perk kita dengan input “16”.

Akan muncul 5 angka acak yang akan kita urutkan.

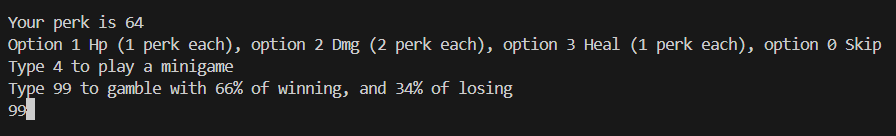
Kita input angka-angka yg sudah kita urutkan.

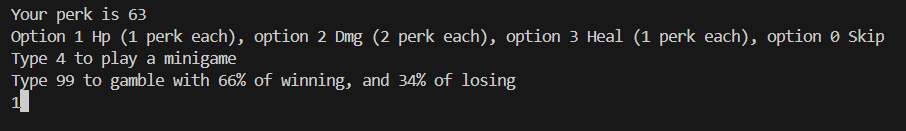
Akan muncul tampilan seperti ini dan perk kita menjadi 32. Kita akan bermain mini game lain dengan input “4”.

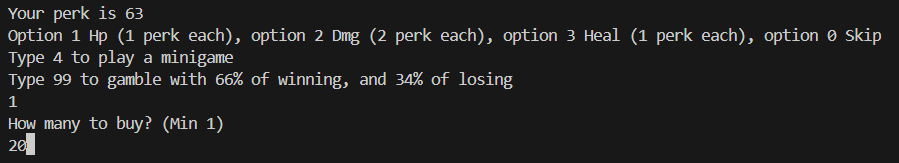
Kita akan merisikokan semua perk kita dengan input “32”.

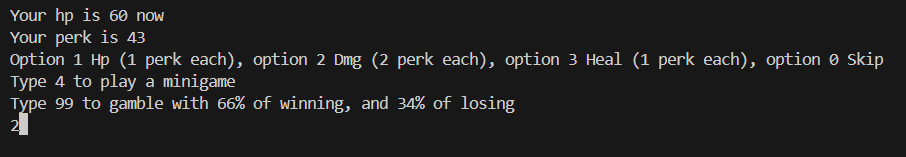
Akan muncul 5 angka acak yang akan kita urutkan.

Kita input angka-angka yg sudah kita urutkan.

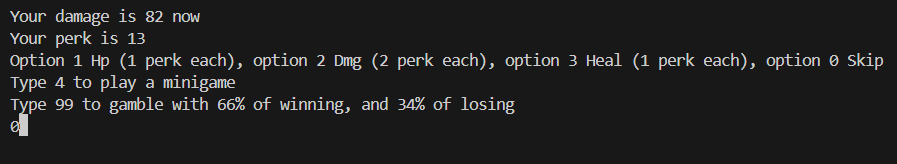
maka muncul tampilan seperti ini, perk kia menjadi 64. Kita input “99” untuk melakukan gambling.

Kita akan merisikokan 1 perk dengan input “1”.

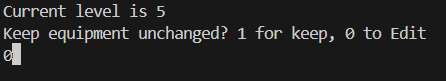
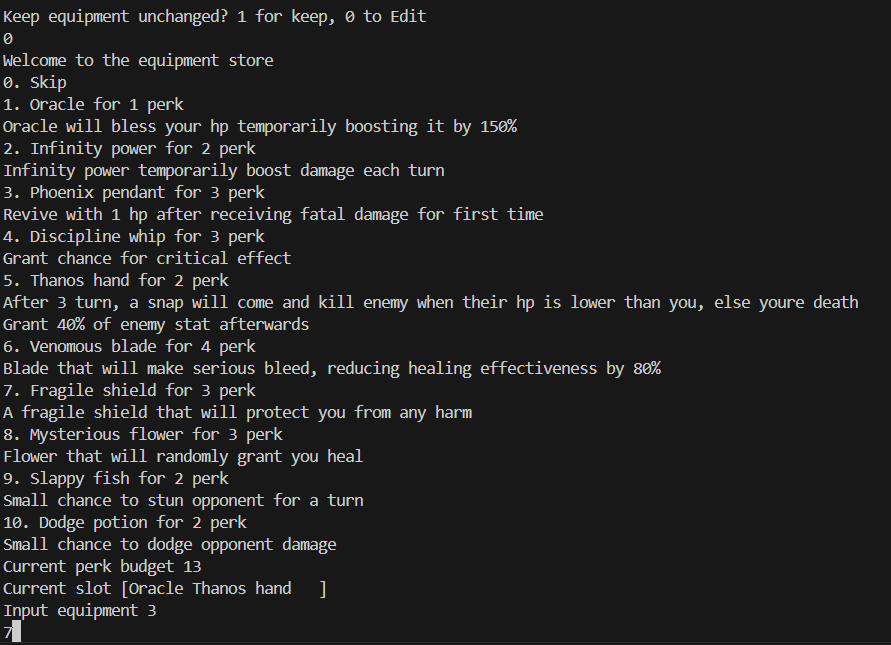
Disini ditampilkan bahwa kita kalah gambling dan perk kita menjadi 63, kita akan membeli hp dengan input “1”.

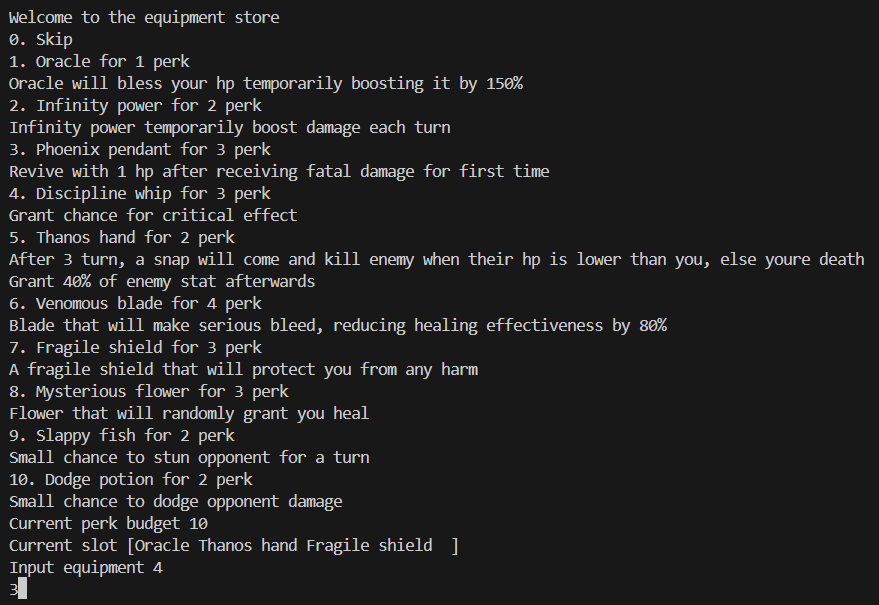
Kita akan membeli 20 hp dengan input “20”.

Akan muncul tampilan seperti ini, hp kita menjadi 60 dan perk kita tersisa 43. Kita akan membeli dmg dengan input “2”.

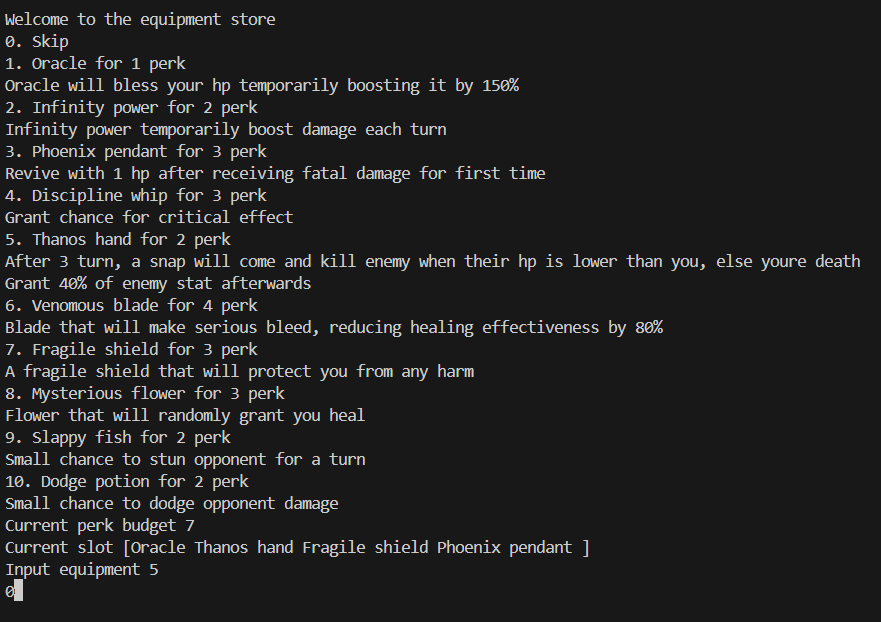
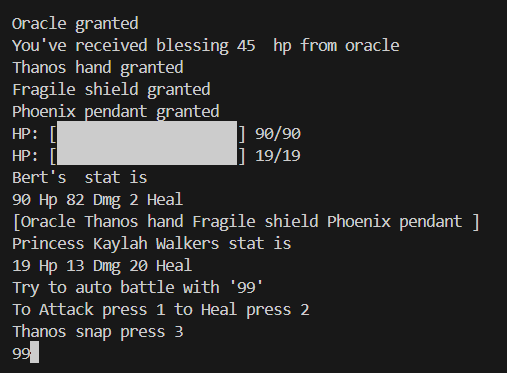
Kita akan membeli 15 dmg yang berjumlah 30 perk. Kita input “15”.

Disini ditampilkan otput seperti ini, dmg kita menjadi 82 dan perk kita tersisa 13, kita akan skip dengan input “0”.

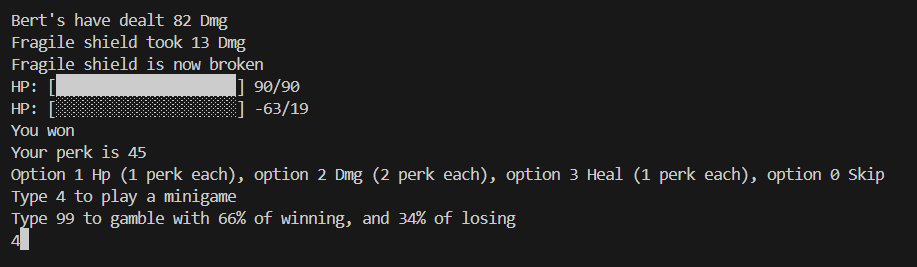
Akan muncul tampilan seperti ini. Kita input “0” untuk mengedit item kita.

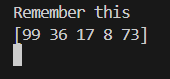
disini muncul tampilan shop, kita akan membeli fragile shield dengan input “7”.

Disini muncul tampilan seperti sebelunya dengan item fragile shield sudah masuk ke slot item kita. Lalu kita akan membeli phoenix pendant dengan input “3”.

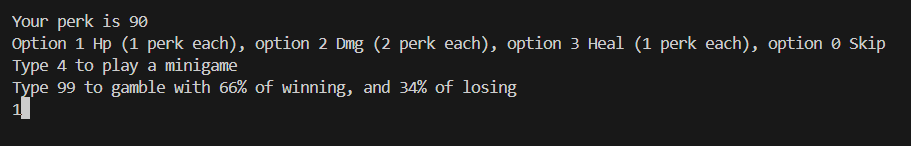
Disini muncul tampilan seperti sebelunya dengan item phoenix pendant sudah masuk ke slot item kita. Lalu kita akan skip dan melanjutkan ke level berikutnya dengan input “0”.

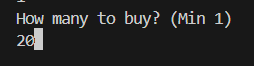
Disini akan muncul status player kita dan status lawan. Karena efek item oracle yang membuat hp kita menjadi 150% maka hp yang awalnya 60 menjadi 90. Kita input “99” untuk auto-battle.

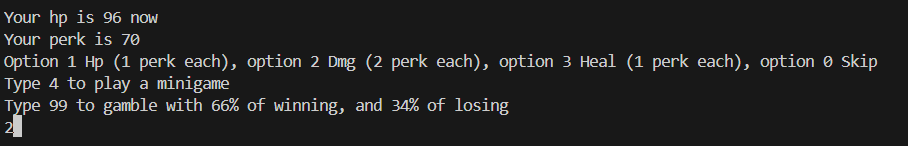
Akan muncul tampilan seperti ini dan kita menang serta perk kita menjadi 45. Kita akan bermain mini game dengan input “4”.

Kita akan merisikokan semua perk kita.

Lalu akan muncul 5 angka acak yang akan kita urutkan.

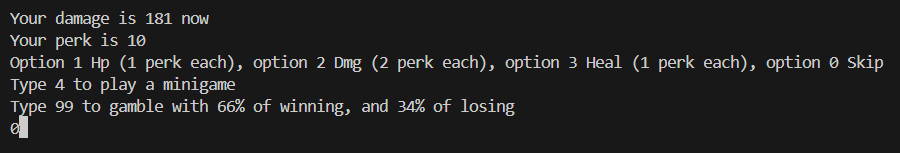
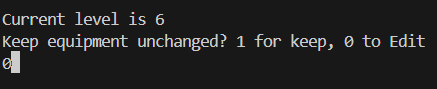
Kita input angka yang sudah kita urutkan.

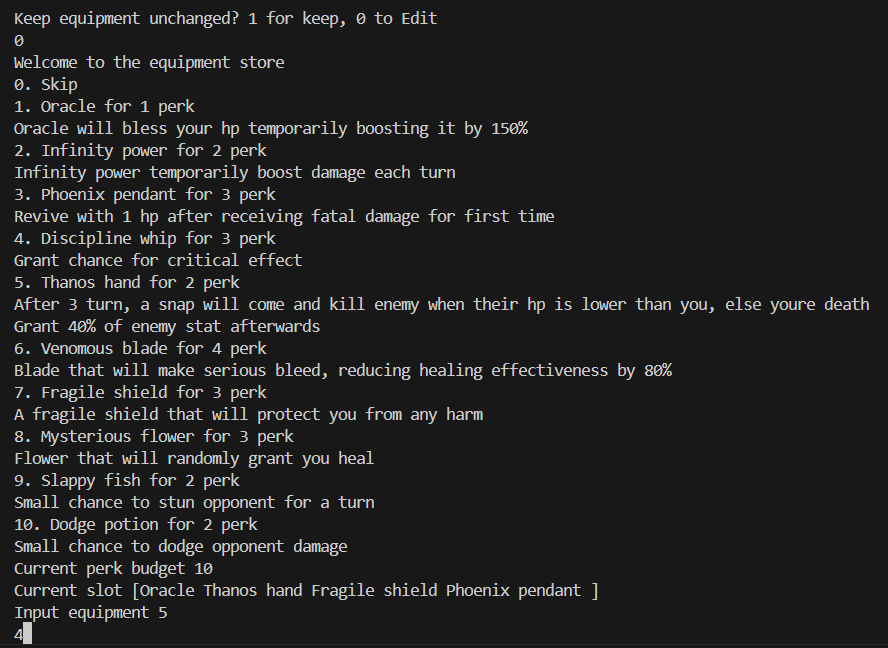
Kita benar dan perk kita menjadi dua kali lipat. Kita input “1” untuk membeli hp.

Kita akan membeli 20 hp dengan input “20”.

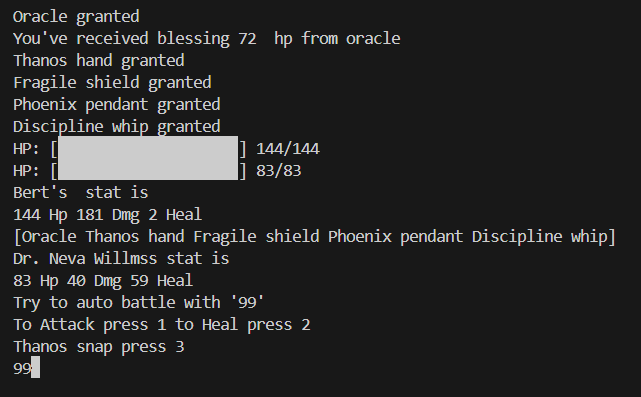
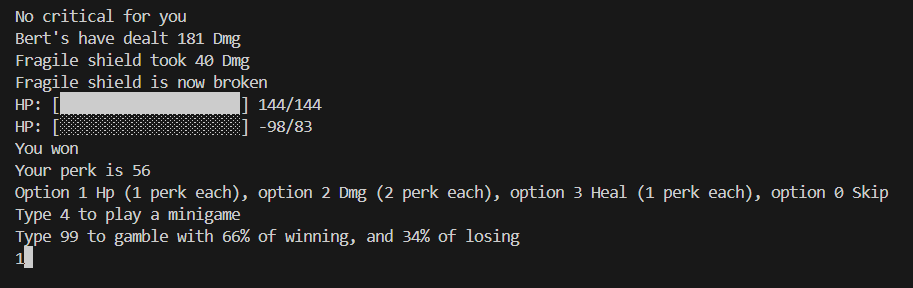
Lalu akan muncul tampilan seperti ini, hp kita mejadi 96 dan kita akan membeli dmg dengan input “2”.

kita akan membeli 30 dmg dengan input “30”.

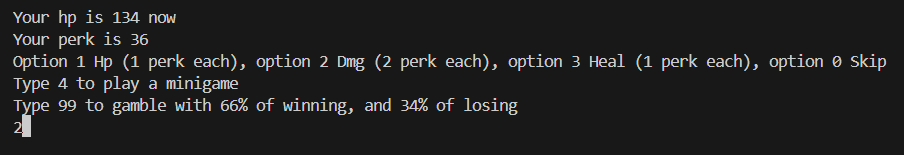
Lalu akan muncul tampilan seperti ini, dmg kita mejadi 181 dan kita akan skip dengan input “0”.

Akan muncul tampilan seperti ini, kita berada di level 6. Kita akan input “0” untuk membeli item.

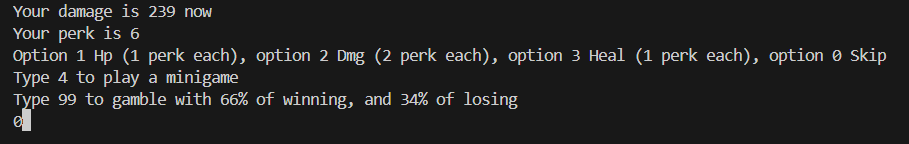
Muncul tampilan shop, kita akan membeli item discipline whip dengan input “4”.

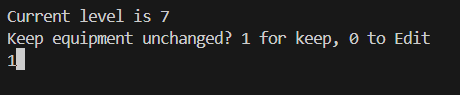
Lalu akan muncul status player kita dan status lawan. Muncul tampilan ini dikarenakan maksimal batas item adalah 5. Jadi langsung ke sesi berikutnya. Kita input “99” untuk auto-battle.

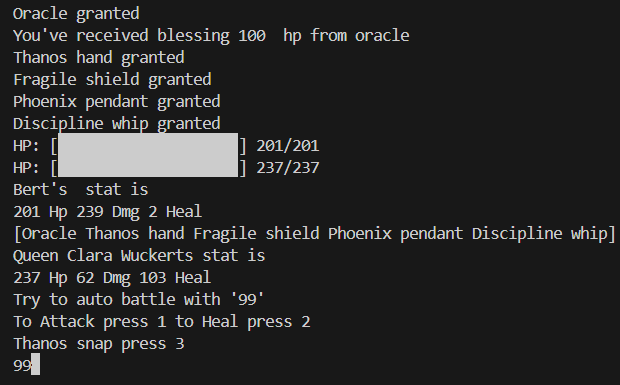
Akan muncul tampilan seperti ini, kita menang dan perk kita menjadi 56. Kita akan membeli hp dengan input “1”.

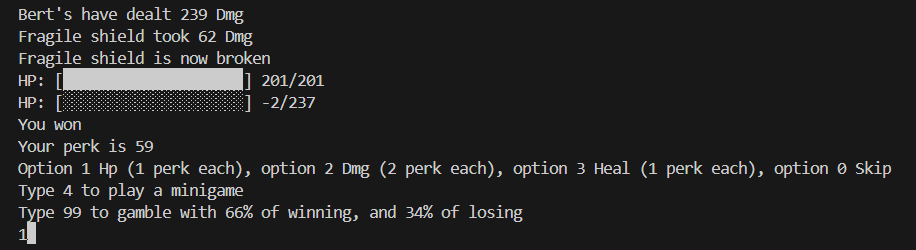
Kita akan membeli 20 hp dengan input “20”.

Lalu akan muncul tampilan seperti ini, dimana hp kita menjadi 134 dan perk kita tersisa 36. Kita akan membeli dmg dengan input “2”.

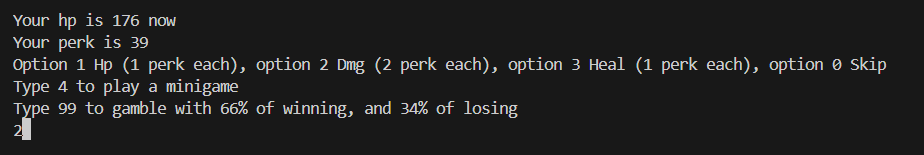
Kita akan membeli 15 dmg dengan input “15”.

Lalu akan muncul tampilan seperti ini, dimana dmg kita menjadi 239 dan perk kita tersisa 6. Kita akan skip dengan input “0”.

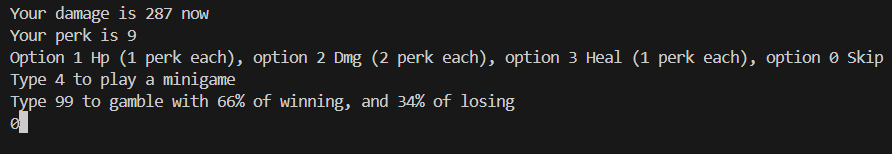
Akan muncul tampilan seperti ini, kita berada di level 7. Kita akan input “1” untuk tidak merubah item.

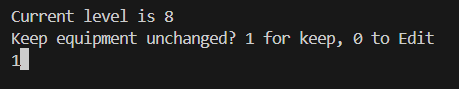
Akan muncul tampilan status player kita dan status lawan kita. Kita input “99” untuk auto-battle.

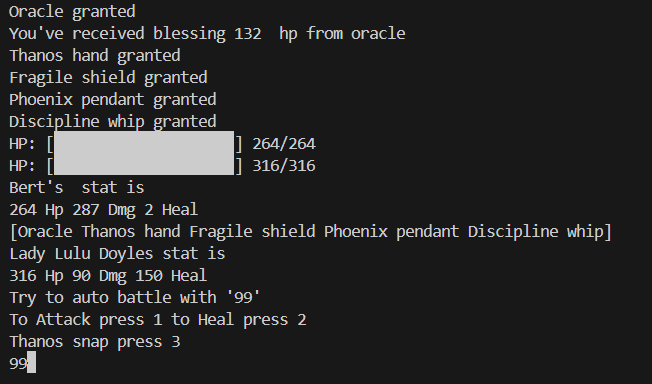
Akan muncul tampilan seperti ini, kita menang dan perk kita menjadi 59. Kita akan membeli hp dengan input “1”.

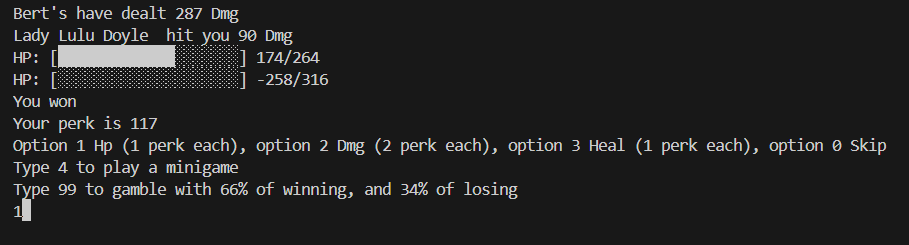
Kita akan membeli 20 hp dengan input “20”.

Lalu akan muncul tampilan seperti ini, dimana hp kita menjadi 176 dan perk kita tersisa 36. Kita akan membeli dmg dengan input “2”.

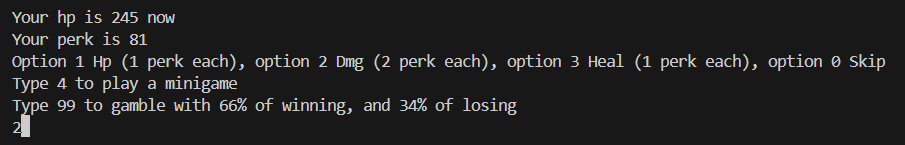
Kita akan membeli 15 dmg dengan input “15”.

Lalu akan muncul tampilan seperti ini, dimana dmg kita menjadi 287 dan perk kita tersisa 9. Kita akan skip dengan input “0”.

Lalu akan muncul output seperti ini dan kita berada di level 8. Kita akan menginput “1” untuk tidak mengganti item.

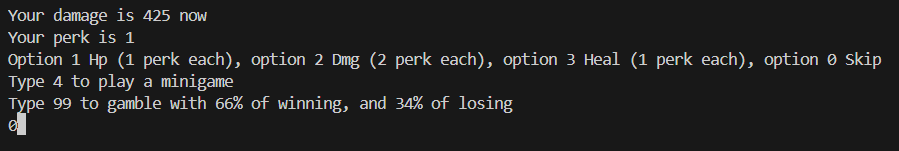
Akan muncul tampilan status player kita dan status lawan kita. Kita input “99” untuk auto-battle.

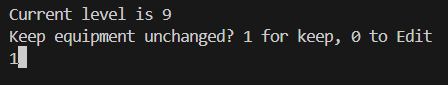
Akan muncul tampilan seperti ini, kita menang dan perk kita menjadi 117. Kita akan membeli hp dengan input “1”.

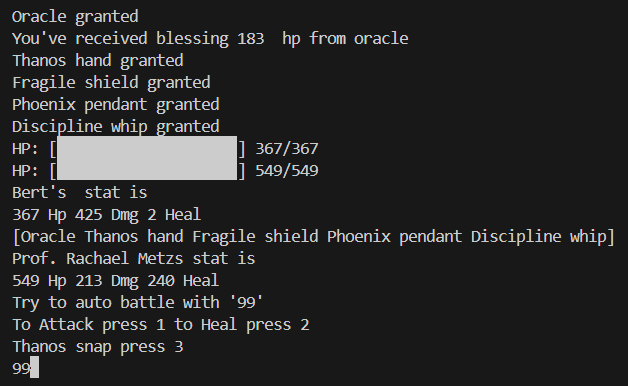
Kita akan membeli 36 hp dengan input “36”.

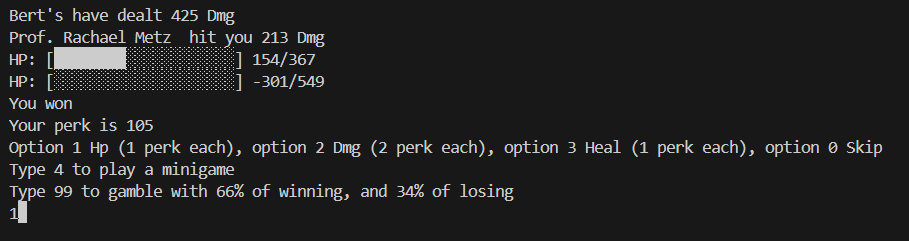
Lalu akan muncul tampilan seperti ini, dimana hp kita menjadi 245 dan perk kita tersisa

81. Kita akan membeli dmg dengan input “2”.

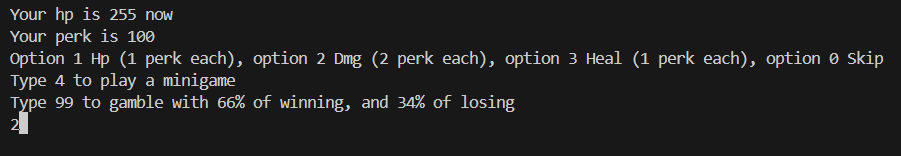
Kita akan membeli 40 dmg dengan input “40”.

Lalu akan muncul tampilan seperti ini, dimana dmg kita menjadi 425 dan perk kita tersisa 1. Kita akan skip dengan input “0”.

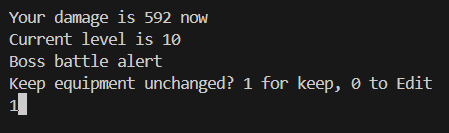
Lalu akan muncul output seperti ini dan kita berada di level 9. Kita akan menginput “1” untuk tidak mengganti item.

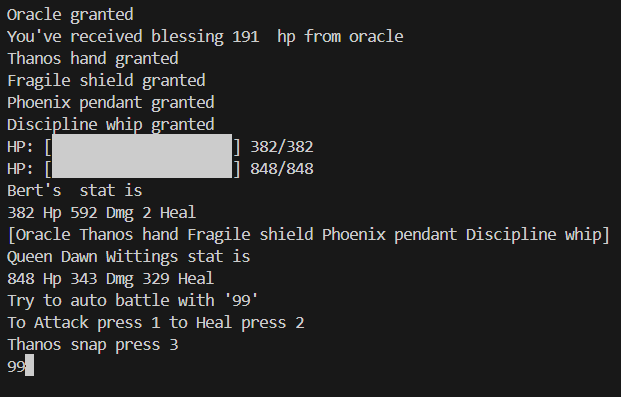
Akan muncul tampilan status player kita dan status lawan kita. Kita input “99” untuk auto-battle.

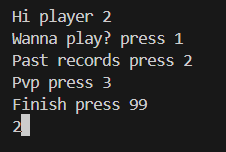
Akan muncul tampilan seperti ini, kita menang dan perk kita menjadi 105. Kita akan membeli hp dengan input “1”.

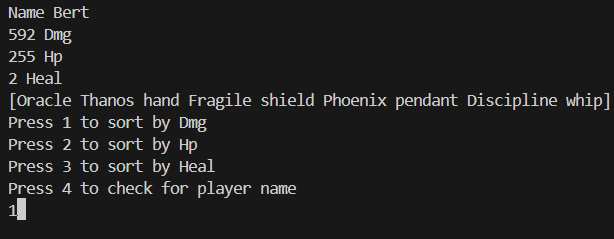
Kita akan membeli 5 hp dengan input “5”.

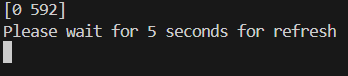
Lalu akan muncul tampilan seperti ini, dimana hp kita menjadi 255 dan perk kita tersisa 100. Kita akan membeli dmg dengan input “2”.

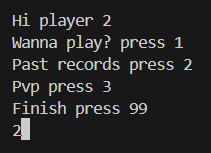
Kita akan membeli 50 dmg dengan input “50”.

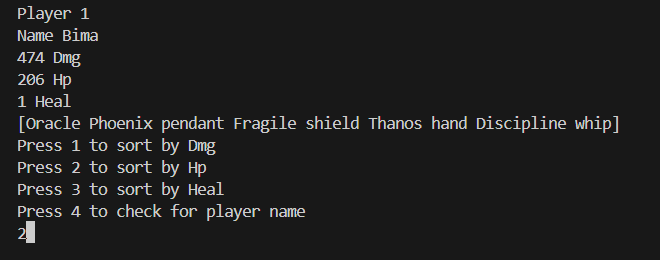
Lalu akan muncul tampilan seperti ini, dmg kita menjadi 592, karena kita mengahbiskan perk kita. Kita otomatis dikiriim ke sesi berikutnya. Kita berada di level 10, kitta input “1” untuk tidak merubah item kita.

Akan muncul tampilan status player kita dan status lawan kita. Kita input “99” untuk auto-battle.

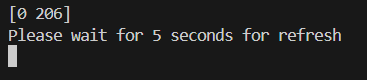
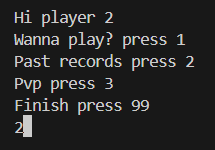
Kita menang dan akan muncul tampilan utama, kita input “2” untuk melihat rekam jejak kita.

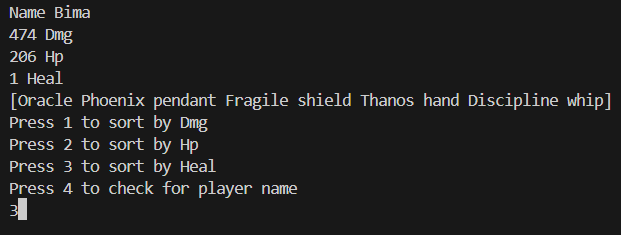
Disini ditampilkan status player kita dan item-itemnya. Disini juga terdapat 4 opsi yaitu “1” untuk mengurutkan berdasarkan dmg, “2” untuk mengurutkan berdasarkan hp, “3” untuk mengurutkan berdasarkan heal, dan “4” untuk mengecek nama player. Kita input “1” untuk mengururkan berdasarkan dmg.

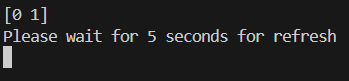
Lalu akan muncul tampilan seperti ini dan alan merefresh dalam 5 detik.

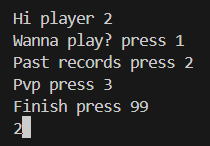
Lalu akan kembali ke menu utama, kita input “2” untuk melihat rekam jejak kita.

Lalu akan muncul seperti sebelumnya, kita input “2” untuk melihat berdasarkan hp. (status player berbeda karena data player hanya disimpan saat program dijalankan).

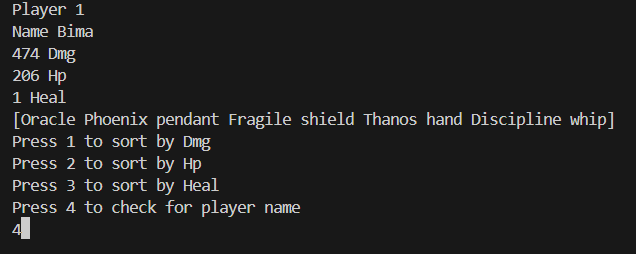
Ini adalah output tampilannya, dimana hp kita adalah 206 dan akan kembali ke menu utama dalam 5 detik.

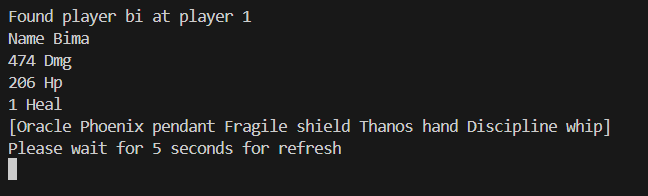
kita input “2” untuk melihat rekam jejak kita.

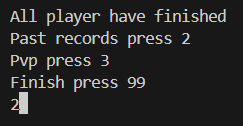
Lalu akan muncul seperti sebelumnya, kita input “3” untuk melihat berdasarkan heal.

Ini adalah output tampilannya, dimana heal kita adalah 1 dan akan kembali ke menu utama dalam 5 detik.

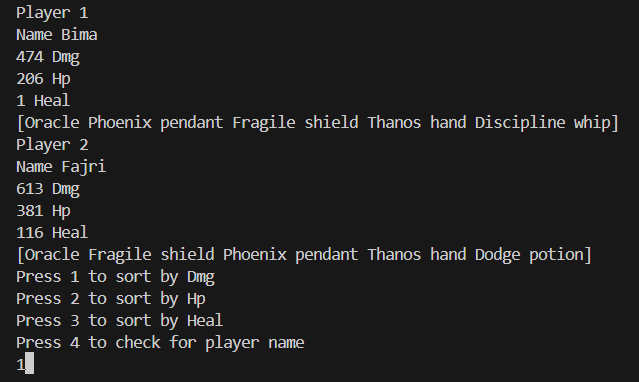
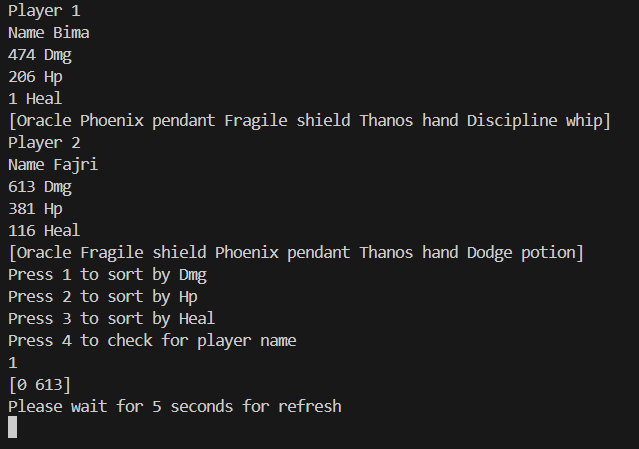
kita input “2” untuk melihat rekam jejak kita.

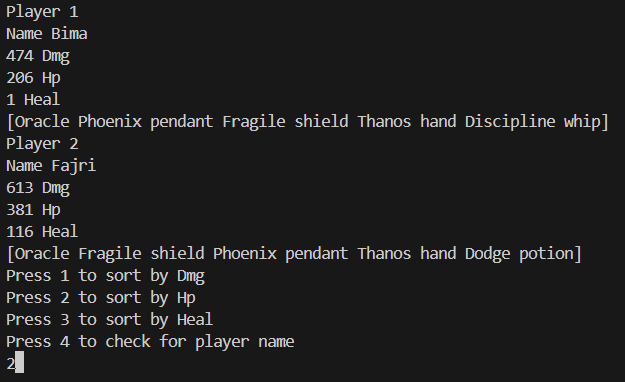
Lalu akan muncul seperti sebelumnya, kita input “4” untuk melihat berdasarkan nama.

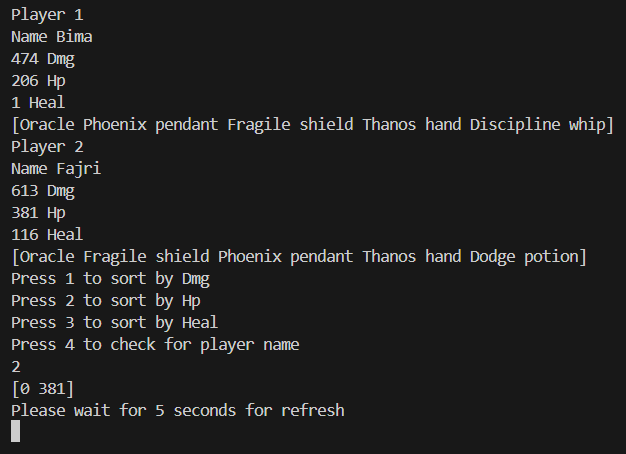
Lalu akan muncul tampilan seperti ini. Untuk fitur pencarian nama player, kitab isa hanya menginput sebagian huruf lalu akan mucl nama yang mengandung huruf yang kita inputkan. Disini kita input “bi”.

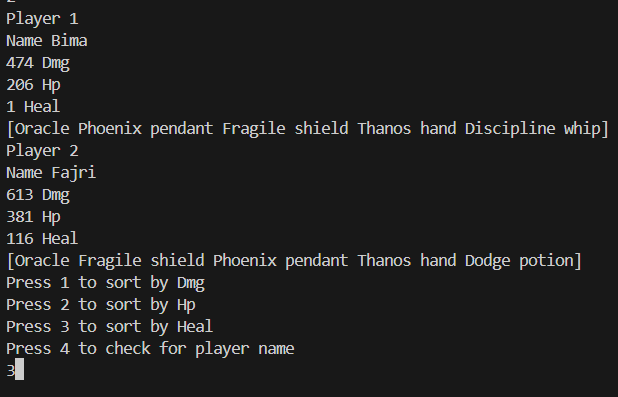
Lalu akan muncul tampilan nama playernya dan akan kemabli ke menu utama dalam 5 detik.

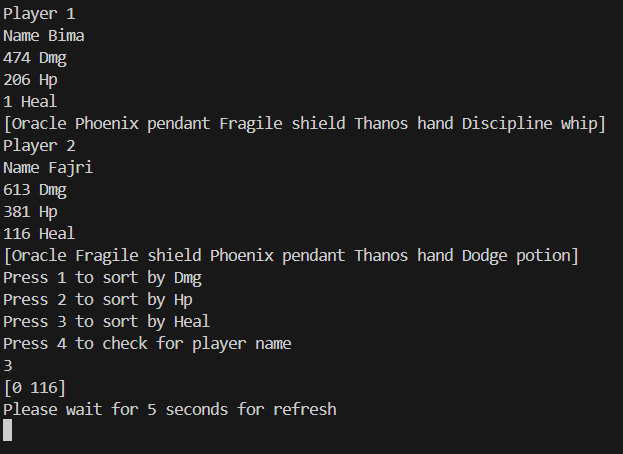
Kita bisa bermain terus-menerus, lalu kita isa cek semua player kita(selama program masih jalan) dengan menginput “2”.

Lalu akan muncul tampilan seperti sebelunya, disini ada tambahan status player jika kita bermian solo player lagi, kita input “ 1” untuk mengurutkan berdasarkan dmg di semua player.

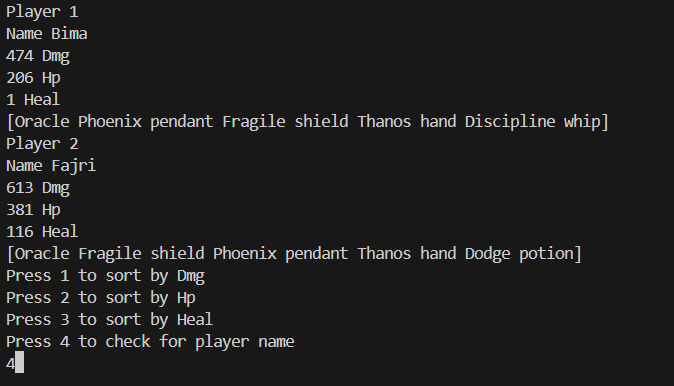
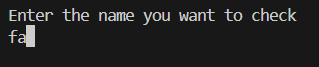
Lalu

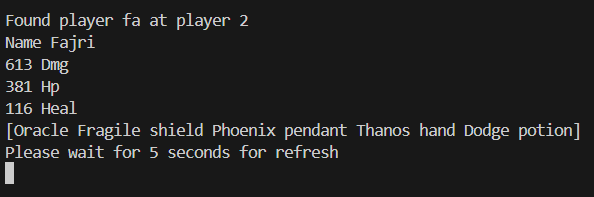
Lalu

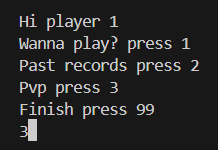
Lalu

Lalu

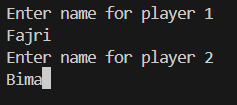
Lalu

Lalu

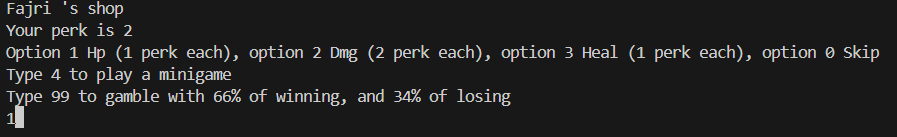
Lalu kita input “fa” untuk mencari nama yang mengandung huruf ‘fa’.

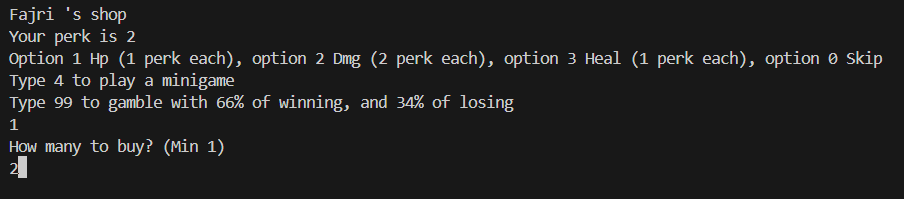
Lalu akan muncul tampilan nama playernya dan akan kemabli ke menu utama dalam 5 detik.

Setelah merefresh akan kemabli ke menu utama. Kita input”3” untuk bermain pvp.

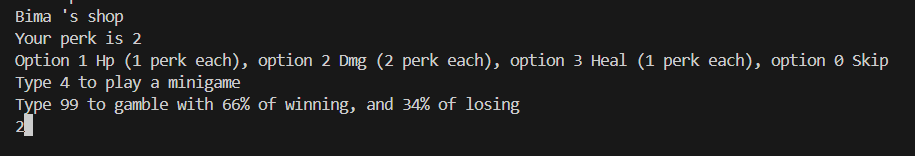
Akan muncul tampilan seperti ini dan kita akan dperintahkan untuk menginput nama player 1. Kita akan menginput “Fajri”.

Lalu akan muncul tampilan seperti ini dan kita akan diperintahkan untuk menginput nama player 2. Kita akan menginput “Bima”.

Lalu akan muncul tampilan seperti ini, kita diperintahkan untuk menginput berapa jumlah perk para player. Kita input “2”.

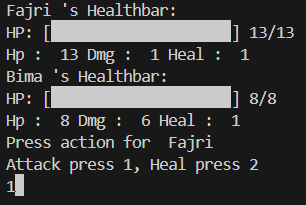
Lalu akan muncul untuk player 1 melakukan pembelian, kita input “1” untuk membeli hp.

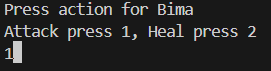
Lalu kita membeli 2 hp dengan input “2”.

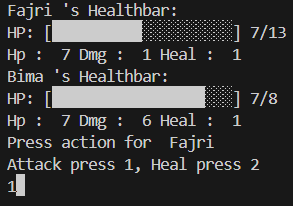
Hp Ppayer 1 ‘Fajri’ menjadi 13. Karena perk player 1 sudah habis, akan otomatis beralih ke player 2.

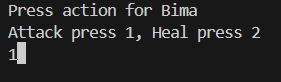
Lalu akan muncul untuk player 2 melakukan pembelian, kita input “2” untuk membeli dmg.

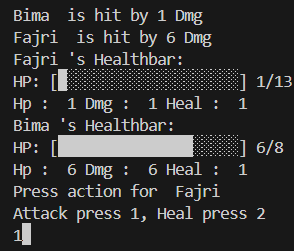
Lalu kita membeli 2 dmg dengan input “2”.

Lalu akan muncul bahwa dmg player 2 ‘Bima’ menjadi 5. Karena perk player 2 sudah habis, akan otomatis beralih ke sesi berikutnya.

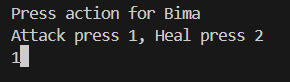
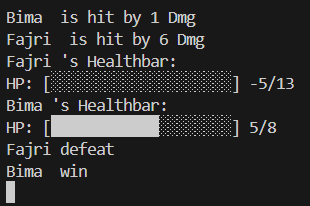
Akan muncul tampilan status Fajri dan status Bima. Disini adalah aksi Fajri jadi kita menginput ”1” untuk menyerang.

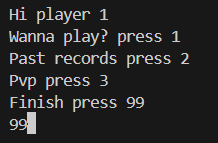
Dilanjut muncul tampilan seperti ini, untuk aksi Bima. Kita input “1” untuk menyerang juga.

Lalu akan muncul status Fajri dan status Bima, dan seperti sebelumnya kita akan emmulai aksi daari aksi Fajri terlebih dahulu. Kita input “1” untuk menyerang.

Dilanjut muncul tampilan seperti ini, untuk aksi Bima. Kita input “1” untuk menyerang juga.

Lalu akan muncul status Fajri dan status Bima, dan seperti sebelumnya kita akan emmulai aksi daari aksi Fajri terlebih dahulu. Kita input “1” untuk menyerang.

Dilanjut muncul tampilan seperti ini, untuk aksi Bima. Kita input “1” untuk menyerang juga.

Lalu akan muncul status Fajri dan status Bima, dan Fajri kalah sedangkan Bima menang. Tampilan ini akan langsung ebrubah menajdi menu utama.

Ini adalah tampilan menu utama, karena kita sudah selesai, kitab isa input “99” untuk menutup game.

Akan muncul tampilan seperti ini setelah menginput “99”.

## 4. Dokumentasi Hasil *Project*

Pada sesi ini ditunjukkan dokumentasi proyek yang tidak sempat dijelaskan pada sesi-sesi sebelumnya. Kalian bisa memasukkan tangkapan layar yang tidak terjelaskan pada sesi sebelumnya. Kalian juga bisa menyantumkan foto Ketika berkegiatan Bersama. Kalian juga bisa mencantumkan cuplikan salindia atau cuplikan Ketika melakukan presentasi. Gambar dan table semua diberi caption.

  
Responsi Proyek

|  |  |
| --- | --- |
| Kontribusi Proyek | |
| Bertrand Lianto | Flowchart, Main Developer program, Design PPT |
| Bima Luthfi Nurhakim | Fixing problem solving & Progress report |
| Ahmad Nur Fajri | Design Flowchart & Progress report |