

LAPORAN TUGAS BESAR

IF2111/Algoritma dan Struktur Data

Mobitangga


Dipersiapkan oleh:

Kelompok 20

Amelia Wahyu Dewanti	18220043
Natanael Dias	18220051
Rahadyanino Maheswara	18220055
Khafifanisa	18220056
Ray Clement Juwanda	18220098

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		IF2111-TB-20		15
		Revisi	1	28 November 2021

Daftar Isi

Ringkasan	2
Penjelasan Tambahan Spesifikasi Tugas	3
Intro	3
Outro	4
Jumlah Player (Bonus)	4
Struktur Data (ADT)	4
ADT Array	4
ADT List double pointer	4
ADT Map	5
ADT Mesin Karakter	5
ADT Mesin Kata	5
ADT Player	6
ADT Stack	6
Program Utama	6
Algoritma-Algoritma Menarik	7
Algoritma Undo	7
Algoritma Player	8
Pembagian Kerja dalam Kelompok	8
Lampiran	8
Deskripsi Tugas Besar	8
Notulen Rapat	9
Log Activity Anggota Kelompok	13

1 Ringkasan

Program ini merupakan sebuah permainan berbasis CLI (command-line interface) yang dibuat dalam bahasa C dengan menggunakan implementasi dari beberapa struktur data, diantaranya ADT Array, ADT List, ADT Stack, ADT Map, ADT Mesin Karakter, ADT Mesin Kata dan lainnya.

Permainan ini merupakan sebuah permainan yang serupa dengan ular tangga, namun dalam bentuk satu dimensi dengan jumlah pemain sebanyak dua sampai empat orang. Mereka akan berlomba untuk mencapai garis finish yaitu petak N dengan menggunakan dadu. Pemain juga mendapatkan berbagai skill yang dapat digunakan untuk menjadi buff selama permainan berlangsung. Skill diperoleh secara random setiap awal permainan dimulai dan setiap satu ronde permainan selesai. Ronde permainan dihitung setelah tiap pemain menyelesaikan masing-masing turn-nya. Di map permainan juga terdapat beberapa teleporter yang diletakkan secara acak sehingga player dapat berpindah ke bagian tertentu pada map.

Laporan ini memuat berbagai penjelasan tambahan di luar spesifikasi yang telah diberikan untuk memperjelas alur dari game ini. Laporan ini mencakup penjelasan tambahan spesifikasi tugas, penjelasan tentang struktur data yang digunakan, penjelasan mengenai cara berjalannya program utama, pembagian kerja dalam kelompok, serta notulensi rapat dan lampiran lainnya.

Program ini dibuat menggunakan bahasa C dengan memanfaatkan ADT yang sudah dipelajari pada mata kuliah IF2110 Algoritma dan Struktur Data. Game ini dimulai dengan menampilkan main menu yang memberikan pilihan pada pemain untuk start game atau quit. Setelah permainan dimulai, para pemain dapat memainkan permainan secara bergantian. Command yang valid pada game ini adalah *SKILL*, *MAP*, *BUFF*, *INSPECT*, *ROLL*, *ENDTURN*, dan *UNDO*. Pemain yang menyentuh petak terakhir terlebih dahulu akan menjadi pemenangnya.

2 Penjelasan Tambahan Spesifikasi Tugas

2.1 Intro

Intro merupakan tampilan awal ketika membuka permainan Mobitangga. Fungsi intro diimplementasikan dalam file header “pre.h” dan file body “pre.c”. Pada bagian Intro, diperlihatkan judul permainan, yaitu “Mobitangga” dalam ukuran besar seperti pada gambar berikut:



2.2 Outro

Fungsi outro merupakan tampilan akhir game. Fungsi outro akan tampil di layar setelah permainan berakhir, atau ketika mengetik command EXIT. Fungsi outro diimplementasikan dalam file header “post.h” dan file body “post.c”. Tampilan fungsi outro pada layar adalah sebagai berikut:

```
.o88b.      db      db      d888888b d8b      db      d888888b      d8b      db d888888b db      db d888888b      d888b      .d8b.      .88b      d88.      d888888b
d8P  Y8      88      88      `88'      888o      88      ~~~88~~~ 88      88      88'      888o      88      88'      `8b      d8'      ~~~88~~~ 88'      Y8b      d8'      8b      88'YbdP' 88      88'
8P          88      88      88      88V8o      88      88      88ooo88 88ooo88      88V8o      88      88ooo88      `8bd8'      88      88      88ooo88      88      88      88ooo88
8b          88      88      88      88      88V8o88      88      88~~~88 88~~~88      88      V8o88      88~~~88      .dPYb.      88      88      88~~~88      88      88      88~~~88
Y8b      d8      88b      d88      .88.      88      V888      88      88      88      88.      88      V888      88.      .8P      Y8.      88      88.      ~8~      88      88      88      88      88.
`Y88P'      ~Y8888P'      Y888888P VP      V8P      YP      YP      YP      Y88888P      VP      V8P      Y88888P YP      YP      YP      Y888P      YP      YP      YP      YP      Y88888P

SAYONARA~~~~~ ~
~)
```

2.3 Jumlah Player (Bonus)

Permainan Mobitangga ini dapat dimainkan minimal 2 hingga maksimal 4 pemain. Pemain akan bergantian untuk bermain sesuai dengan urutan input pemain pada awal permainan. Setiap pemain akan mendapatkan 1 skill pada setiap round, namun apabila pemain mendapatkan “Teknologi Gagal”, berarti pemain gagal mendapatkan skill pada round tersebut. Saat bermain, pemain dapat memasukkan command-command untuk melakukan aksi dalam permainan.

3 Struktur Data (ADT)

3.1 ADT Array

- Sketsa struktur data : struktur array yang terdiri atas nilai Neff dan TI yang merupakan integer. TI merupakan memori tempat penyimpanan elemen (container) yang berisi 100 elemen. Neff merupakan banyaknya elemen efektif. Disini konstanta IdxMax adalah 100, IdxMin adalah 1, dan IdxUndef adalah -999.
- Persoalan yang diselesaikan : Pemanfaatan ini dapat dijumpai pada ADT Map dan Player untuk merepresentasikan ataupun menyimpan sesuatu dengan panjang yang tetap.
- Alasan pemilihan : ADT Array dapat digunakan untuk menyimpan suatu informasi dengan jumlah maksimal yang ditampungnya kita ketahui sehingga menjadi lebih mudah dan efisien tanpa perlu adanya alokasi/dealokasi memori.
- Diimplementasikan sebagai ADT Array dengan nama file header “array.h” dan file body “array.c”

3.2 ADT List double pointer

- Sketsa struktur data : Variasi ADT list yang digunakan berupa list dengan pointer ganda. ADT list pointer ganda tersebut terdiri dari dua tipe bentukan utama. Tipe bentukan pertama yaitu elemen list yang berisi info list (nilai yang disimpannya), address dari *next* (menunjuk ke elemen list berikutnya), dan address dari *prev* (menunjuk ke elemen list sebelumnya). Tipe bentukan kedua yaitu list itu sendiri yang berisi address *First*

(menunjuk elemen awal list) dan address *Last* (menunjuk elemen akhir list). Disini konstanta Nil berupa null.

- Persoalan yang diselesaikan : ADT tersebut digunakan untuk penyimpanan *skill* serta pengelolaannya (tambah *skill*, print *skill*, buang *skill*, menjalankan *skill*, *get skill*). Selain itu, ADT list ini juga dapat dijumpai pada ADT player sebagai penyimpanan *skill* pada masing-masing *player*.
- Alasan pemilihan : Variasi ADT list dengan pointer ganda dipilih karena dirasa pengaksesannya lebih mudah dan efektif walaupun memakan lebih banyak memori dari variasi list biasa.
- Diimplementasikan sebagai ADT List dengan pointer ganda dengan nama file header “listdp.h” dan file body “listdp.c”

3.3 ADT Map

- Sketsa struktur data : ADT map terdiri dari 2 tipe bentukan. Bentuk pertama yaitu Petak yang berisi alamat berupa integer, IsiPetak berupa character, dan penanda teleporter berupa integer. Bentuk kedua yaitu Map itu sendiri yang berisi PanjangMap (integer), MaxRoll (nilai maksimal untuk melakukan roll berupa integer), JumlahTeleporter (integer), dan sebuah TabMap yang bertipekan bentukan Petak.
- Persoalan yang diselesaikan : ADT Map digunakan untuk menyimpan informasi dari konfigurasi map yang nantinya digunakan sebagai medan bermain.
- Alasan pemilihan : Pemilihan ADT map berdasarkan kebutuhan kami dalam menyimpan suatu informasi berupa map yang dibaca dari suatu konfigurasi menggunakan fungsi/prosedur yang ada pada ADT mesin karakter.
- Diimplementasikan sebagai ADT Map dengan nama file header “map.h” dan file body “map.c”

3.4 ADT Mesin Karakter

- Sketsa struktur data : ADT Mesin Karakter terdiri dari state mesin CC berupa character dan suatu boolean EOP dengan MARK berupa *newline*.
- Persoalan yang diselesaikan : ADT Mesin Karakter digunakan untuk membaca suatu konfigurasi yang berupa kumpulan karakter.
- Alasan pemilihan : ADT ini sangat cocok untuk melakukan suatu pembacaan terhadap file konfigurasi.
- Diimplementasikan sebagai ADT Mesin Karakter dengan nama file header “mesinkar.h” dan file body “mesinkar.c”

3.5 ADT Mesin Kata

- Sketsa struktur data : ADT Mesin Kata terdiri dari TabKata (array dengan jumlah maksimalnya NMax+1) bertipekan character dan Length berupa integer. Disini konstanta BLANK berupa ‘ ’ (spasi) dan NMax adalah 110. State dari mesin ada EndKata berupa boolean dan CKata berupa bentukan Kata.
- Persoalan yang diselesaikan : Bersama dengan ADT Mesin Karakter membaca suatu konfigurasi Map yang nantinya akan digunakan ada ADT MAP.

- Alasan pemilihan : ADT Mesin Kata cocok digunakan untuk melakukan suatu pembacaan terhadap file konfigurasi yang memiliki suatu pemisah/BLANK..
- Diimplementasikan sebagai ADT Mesin Kata dengan nama file header “mesinkata.h” dan file body “mesinkata.c”

3.6 ADT Player

- Sketsa struktur data : ADT *Player* terdiri dari 2 tipe bentukan. Bentukan pertama adalah *Player* yang berisi Nama bertipekan character, *CPosition* berupa integer, dan *Skill* (berupa list). Bentuk kedua yaitu *TabPlayer* yang berisi informasi *Player* (berupa array bertipekan bentukan *Player*) dan sebuah integer *Neff* yang merupakan sebuah nilai efektif.
- Persoalan yang diselesaikan : ADT *Player* dimanfaatkan untuk membuat suatu penyimpanan informasi yang digunakan nantinya saat bermain seperti nama, skill, dan posisi pemain.
- Alasan pemilihan : Dengan membuat ADT yang spesifik untuk player maka pengaksesan informasi pada player menjadi lebih mudah.
- Diimplementasikan sebagai ADT *Player* dengan pointer dengan nama file header “player.h” dan file body “player.c”

3.7 ADT Stack

- Sketsa struktur data : ADT *Stack* terdiri dari *playerstate* yang bertipekan bentukan *TabPlayer* dari ADT *Player* dan *round* berupa integer sebagai elemen puncak stack (TOP)
- Persoalan yang diselesaikan : ADT *Stack* dimanfaatkan untuk menyimpan suatu *history* bermain yang nantinya dapat digunakan untuk meng-undo langkah pada permainan dengan fungsi pop.
- Alasan pemilihan : ADT *Stack* cocok digunakan untuk menyimpan suatu *history*/langkah dalam permainan karena sifatnya yang *Last In First Out*.
- Diimplementasikan sebagai ADT *Stack* dengan pointer dengan nama file header “stack.h” dan file body “stack.c”

4 Program Utama

Program utama dengan nama *game_flow.c* akan berjalan apabila kita meng-include semua file header dari ADT beserta file konfigurasi. Program utama akan menampilkan *intro* dan *main menu* dari permainan yaitu *NEW GAME* dan *EXIT*. Command *NEW GAME* dapat digunakan apabila pengguna akan memulai permainan sedangkan *command EXIT* digunakan untuk keluar dari program permainan. Apabila pengguna memasukan *command NEW GAME*, program akan lanjut dengan meminta kepada pengguna untuk memasukan banyak pemain yang akan bermain. Jumlah pemain yang dapat bermain adalah 2 sampai 4 pemain. Setelah program menerima *input* banyak pemain yang bermain, program akan melakukan *looping* sebanyak jumlah pemain untuk menerima *input* nama dari setiap pemain. Setelah itu program akan membuat *TabPlayer* yang berisi data dari setiap pemain yang berupa nama, posisi, *list* dari *skill* yang dimiliki

pemain. Program kemudian akan membaca *file* konfigurasi yang telah disediakan dengan menggunakan ADT Mesin Karakter dan *ADT* Mesin Kata.

Setelah itu, program akan menampilkan *map* permainan lalu permainan dimulai dengan giliran pada pemain pertama. Game akan me-*random* secara otomatis *skill* yang akan didapat oleh pemain pada setiap *round* dan memasukkannya ke *List skill* pada *TabPlayer*. Kemudian, program akan menunggu *input command* dari pengguna. Command yang valid adalah *SKILL*, *MAP*, *BUFF*, *INSPECT*, *ROLL*, *UNDO*, dan *ENDTURN*. Apabila *command* yang dimasukan tidak valid, pembacaan *input* akan terus diulangi hingga pengguna memasukan *command* yang valid.

Jika player meng-*input*-kan *SKILL*, maka program akan mengeluarkan *skill* apa saja yang dimiliki oleh pemain dengan memanggil fungsi pada ADT Player dan ADT List dengan pointer. Pemain kemudian dapat menggunakan *skill* tersebut dengan mengetikan nomor pada skill, atau pengguna juga dapat membuang *skill* yang ada dengan mengetikan angka negatif dari nomor *skill* tersebut. Angka nol dapat digunakan untuk keluar dari tampilan *skill*. Jika pemain menginputkan command *MAP*, maka program akan mengeluarkan gambaran *MAP* dari tiap pemain dengan memanggil fungsi pada ADT Map untuk membaca map dengan menggunakan ADT Mesin Karakter dan ADT Mesin Kata. Command *BUFF* dapat digunakan untuk melihat efek dari *skill* yang telah digunakan oleh pemain. Command *INSPECT* dapat digunakan untuk melihat petak dari *map* apakah petak yang dipilih tersebut adalah petak biasa, petak terlarang, atau petak dengan teleporter. Command lainnya yaitu command *ROLL* digunakan untuk mengocok dadu agar pemain dapat bergerak ke petak lainnya dengan menggunakan fungsi *roll*. Command *UNDO* dapat digunakan untuk me-*reset* kondisi permainan ke *round* sebelumnya dengan melakukan *Pop round state* pada *round* sebelumnya di *Stack* kemudian me-*replace TabPlayer* di program utama. Command terakhir yaitu *ENDTURN* digunakan ketika pemain mengakhiri *turn*-nya. Setelah *turn* seorang pemain selesai, *turn* akan dilanjutkan ke pemain berikutnya. Apabila pemain terakhir telah mengakhiri *turn*-nya, maka *round* tersebut berakhir dan berganti ke *round* berikutnya. Permainan akan berhenti apabila salah seorang pemain memenangkan permainan dengan menyentuh petak ke 20 yang merupakan petak terakhir atau ada pemain yang memilih untuk *EXIT*.

5 Algoritma-Algoritma Menarik

5.1 Algoritma Undo

Algoritma undo didefinisikan sebagai *UNDO* pada command. Algoritma ini digunakan pada saat program utama dijalankan dan pemain memasukkan command “UNDO”. Alasan algoritma ini disebut menarik karena algoritma ini tidak akan membiarkan stack kosong, melainkan selalu memiliki *playerstate* pada awal ronde 1. Selain itu, fungsi Push pada stack juga tidak hanya langsung menumpuk List Skill, melainkan menggunakan iterasi for loop untuk meng-*copy* setiap skill di list yang ada. Hal ini dikarenakan apabila langsung menumpuk list skill ke stack, dapat terjadi error pada *address* di setiap skill pada list.

5.2 Algoritma Player

Algoritma Player terdapat pada ADT Player. Algoritma ini dianggap menarik oleh kelompok kami karena algoritma ini berhubungan dengan banyak ADT lainnya. Selain itu, algoritma ini menyimpan data ADT lain ke dalam player masing-masing yang bermain. Di dalam proses membuat *game_flow*, kami mengalami beberapa kendala karena state player harus disambungkan dengan ADT lain. Hal tersebut cukup merangsang rasa penasaran kami dan itu juga menjadi salah satu alasan algoritma ini sangat menarik.

6 Pembagian Kerja dalam Kelompok

No.	Fitur/ADT	NIM Coder	NIM Tester
1.	ADT Array	18220056	18220056
2.	ADT List dengan double pointer	18220051	18220051
3.	ADT Map	18220056 18220051	18220056
4.	ADT Mesin Karakter	18220043 18220051	18220043
5.	ADT Mesin Kata	18220043 18220051	18220043
6.	ADT Player	18220051	18220051
7.	ADT Stack	18220098	18220098
8.	ADT Game Flow	18220051 18220098	18220051 18220098 18220043 18220056 18220055
9.	Fungsi Roll	18220051	18220051
10.	Fungsi Pre (Intro)	18220055	18220055
11.	Fungsi Post (Outro)	18220055	18220055

7 Lampiran

7.1 Deskripsi Tugas Besar

Sebuah Institut Teknologi tertentu sedang mengadakan lomba *game dev* dengan tema membuat board game digital terbaik se-kecamatan. Mendengar hal tersebut, Borakemon dan Mobita bersekongkol untuk membuat sebuah board game digital terasyik. Mereka kepikiran untuk menggabungkan game ular tangga dengan modifikasi-modifikasi yang dapat mengganggu lawan sehingga lahirlah ide Mobitangga.

Namun sayangnya, Mobita tidak memiliki kemampuan maupun niat memprogram Mobitangga. Borakemon, kucing robot Mobita, juga belum memiliki kemampuan untuk memprogram karena belum belajar terlalu *deep*. Oleh karena itu, Borakemon menculik sekelompok *programmer* dari dimensi lain agar dapat membantu mereka membuat program Mobitangga agar dapat memenangkan lomba *game dev* itu.






7.2 Notulen Rapat

**Form Asistensi Tugas Besar
IF2110/Algoritma dan Struktur Data
Sem. 1 2021/2022**

No. Kelompok/Kelas : 20 / K01 & K02
Nama Kelompok : PSG (Pejuang Semester Ganjil)
Anggota Kelompok (Nama/NIM) :
1. Amelia Wahyu Dewanti / 18220043
2. Natanael Dias / 18220051
3. Rahadyanino Maheswara / 18220055
4. Khafifanisa / 18220056
5. Ray Clement Juwanda / 18220098

Asisten Pembimbing : Jonathan Yudi Gunawan / 13518084

Asistensi I






Tanggal : Minggu / 7 November 2021	Catatan Asistensi:
Tempat : https://meet.google.com/fnf-hado-ziv	
<p>Kehadiran Anggota Kelompok:</p> <p>No NIM Tanda tangan</p> <p>1 18220043</p>  <p>2 18220051</p>  <p>3 18220055</p>  <p>4 18220056</p>  <p>5 18220098</p> 	<p>1. Gambaran singkat tentang tubes? Game flownya itu ular tangga 1 dimensi, ada beberapa petak yang gak bisa dilewat. Di dalemnya ada skill dan buff. Bisa pake beberapa skill, kalo skillnya dipake, bakal aktifin buff. Nanti buff bakal ada kondisi tertentu biar buffnya kepake. Bisa ngecek map, nge-inspect, save, load dan undo.</p> <p>2. Persen chance? Di awal dapet skill random sesuai dengan chance. Implementasi chance cari sendiri, pake import random c++. Output random 0-1, 0-0,1 dapet apa, 0,1-0,2 dapet apa dan seterusnya sampe nyampe 1.</p> <p>3. Masalah bonus, apakah ada yg ditentukan di awal? Yg ga terkait save sama load. Design lebih dari 2 orang bisa nentu di awal atau ubah di tengah-tengah. Yang agak ribet bonus skill, harus ditentukan di awal dulu biar ga ngubah banyak di flow game nya.</p> <p>4. Saran pembagian tugas? Coba bagi fitur yang kecil-kecil, baru bisa dilihat bobot trus disebar. Yang pertama harus dibikin tuh yg input komen. Untuk fitur yang game mekanik bisa satu orang. Untuk skill agak banyak bisa dua orang. Bisa dibagi satu orang satu ADT biar ga konflik waktu pengerjaan.</p> <p>5. Asistensi kedua harus siap apa aja? Ga ada ketentuannya dan belum disampein tanggalnya. Nanti bakal disampein di google chat.</p> <p>6. File konfigurasi gimana? Baris satu panjang peta, baris dua peta nya, baris tiga roll ($1 \leq \text{Roll} \leq N-1$), baris keempat ($1 \leq M \leq N-2$), baris selanjutnya isi pijakan dimana dan arah teleporternya loop sebanyak M.</p> <p>7. Untuk repository github apakah sudah aman?</p>


	<p>Sudah cukup, jangan lupa di private</p> <p>8. Tips dan trik</p> <p>Speknya agak banyak jadi bisa dimulai segera, trus buat ngetes skill gabisa cepet. Sarannya, bikin save sama load supaya ngetes skillnya gampang. Bisa masukin skill di tengah. ADTnya wajib pakai yang di kelas, tapi kalo perlu properti yang lain misalnya list untuk menyimpan skill, boleh ditambah-tambah ADTnya.</p>
	Tanda Tangan Asisten:



Jonathan Yudi Gunawan

Asistensi II

Tanggal : Minggu / 21 November 2021	Catatan Asistensi:
Tempat : https://meet.google.com/rgm-anpm-gbb	
Kehadiran Anggota Kelompok:	
No NIM	
Tanda tangan	
1	<ol style="list-style-type: none"> 1. Kalo buff bisa dipake 2 secara bersamaan, atau bakal ke replace kalo pake skill? Boleh dibaca di FAQ. 2. Kalo buffnya mirip bisa ditimpa gitu ga kak? bebas, boleh diganti, atau boleh ga jadi dimasukin ke buff. 3. Gimana nandain 2 skill dihapus saat melakukan undo? di push ke stack semua dulu 4. Misalnya pake cermin pengganda, itu kan dapet 2 skill random. Misal dapet lagi itu dibuang skillnya atau ngga? Boleh nanti dijelasin aja waktu demo, nanti kalian tambahin fitur buat buang skill
18220043	
	
2	
18220051	
	
3	
18220055	
	
4	
18220056	
	
5	
18220098	
	

	Tanda Tangan Asisten:  Jonathan Yudi Gunawan
--	---

7.3 Log Activity Anggota Kelompok

Nama Lengkap - NIM	Activity
Amelia Wahyu Dewanti - 18220043	<ul style="list-style-type: none"> • 20 November 2021 : Upload mesinkata.c dan mesinkata.h • 20 November 2021 : Upload mesinkar.c dan mesinkar.h • 27 November 2021 : Mengedit laporan bagian lampiran, menambahkan bagian ADT pada laporan yang dirasa kurang, merapikan laporan • 28 November 2021 : Menambahkan laporan bagian Algoritma menarik "Algoritma Player"
Natanael Dias - 18220051	<ul style="list-style-type: none"> • 16 November 2021 : Upload listdp.c dan listdp.h • 17 November 2021 : Upload roll.c dan roll.h • 19 November 2021 : Update listdp.c karena ada yang kurang • 20 November 2021 : Update roll.c roll.h, listdp.c dan list.h (menyesuaikan spesifikasi) serta upload player.c dan player.h • 21 November 2021 : Update map.c, map.h, mesinkar.c, mesinkar.h, mesinkata.c, mesinkata.h, listdp.c agar lebih mudah dipakai dan dipahami • 22 November 2021 : Upload main_driver.c, game_flow.c, dan

	<p>game_flow.h. Update player.c, player.h, dan listdp.c (agar lebih mudah dipakai)</p> <ul style="list-style-type: none"> • 24 November 2021 : Update game_flow.c (menyesuaikan spesifikasi) • 25 November 2021 : Update game_flow.c (terdapat bug, penyesuaian dengan spesifikasi, dan test di linux) • 26 November 2021 : Update map.c (terdapat salah logic) dan game_flow.c (penyesuaian dengan fitur undo yang ditambah, penyesuaian dengan spesifikasi, dan perbaikan bug)
Rahadyanino Maheswara - 18220055	<ul style="list-style-type: none"> • 5 November 2021 : Membuat repository 'Mobitangga' • 20 November : Merapikan repository • 20 November : Fix include tiap file • 20 November : Upload pre.c dan pre.h • 25 November: test main_driver.c • 25 November : Mengisi laporan bagian ADT dan memperbaiki beberapa yang kurang • 27 November : Update pre.c dan upload post.c + post.h
Khafifanisa - 18220056	<ul style="list-style-type: none"> • 20 November 2021 : Upload array.h dan array.c • 20 November 2021 : Upload map.h dan map.c • 20 November 2021 : Mengedit laporan bagian awal laporan, memperbaiki typo dan susunan kalimat bagian ringkasan. • 21 November 2021 : Mengedit kalimat dan merapikan laporan. • 25 November 2021 : Membuat laporan bagian Penjelasan Tambahan Spesifikasi Tugas "Fungsi Intro" dan

	<p>“Jumlah Player (bonus)</p> <ul style="list-style-type: none"> • 25 November 2021 : Membuat laporan bagian “Pembagian Kerja dalam Kelompok”. • 28 November 2021 : Membuat laporan bagian Penjelasan Tambahan Spesifikasi Tugas “Fungsi Outro”.
Ray Clement Juwanda - 18220098	<ul style="list-style-type: none"> • 18 November 2021 : Upload stack.c dan stack.h • 19 November 2021 : Modifikasi stack untuk fungsi undo • 21 November 2021 : Membuat kerangka dasar laporan dan membuat bagian ringkasan • 22 November 2021 : Minor Update ADT Stack • 23 November 2021 : Membuat laporan bagian penjelasan program utama • 23 November 2021 : Penambahan fungsi UNDO pada game_flow, rapihin print di terminal • 24 November 2021 : Minor Update stack dan fungsi undo di game_flow • 25 November 2021 : Major update ADT Stack, Player dan game_flow • 26 November 2021 : Melengkapi laporan bagian algoritma menarik “Algoritma Undo” dan memasukkan lampiran notulensi asistensi • 28 November 2021 : Minor Update Mesin Kata, Mesin Kar, game_flow tambah makefile