

# Spesifikasi Tugas Besar

## IF2010 Pemrograman Berorientasi Objek STI

**Tenggat Waktu *Milestone 1*: 27 April 2025 Pukul 20.10**

**Tenggat Waktu *Milestone 2*: 1 Juni 2025 Pukul 01.02**

Versi 1.0

### Change Log:

14 April 2025 - *Initiation*

16 April 2025 - Revisi waktu pengumpulan

19 April 2025 - Penambahan aturan repository

24 April 2025 - Koreksi “Village Map” menjadi “World Map”

24 April 2025 - Revisi Maksimal Tanggal asistensi

24 April 2025 - Revisi lore NPC Caroline dan Emily

25 April 2025 - Koreksi “friendship status” menjadi “relationship status”

26 April 2025 - Koreksi deskripsi heartPoints NPC

27 April 2025 - Penambahan Form Pengumpulan M1

13 Mei 2025 - Penulisan Deadline M2 di awal Spek yang telah tertulis di bagian Milestone

18 Mei 2025 - Batas akhir asistensi M2

29 Mei 2025 - Pengunduran Deadline Milestone 2

29 Mei 2025 - Mekanisme Pengumpulan Milestone 2

Form Pengumpulan: <https://bit.ly/SubmissionM1TubesOOP2425>

# Daftar Isi

Daftar Isi	1
Glosarium	3
Pedoman	4
Deskripsi Persoalan	5
Spesifikasi Sistem	7
Ketentuan Umum	7
Ketentuan Teknis	7
Entitas Permainan	8
Player	8
Farm	9
House	9
NPC	9
Items	13
Maps	19
Farm Map	19
World Map	21
Gold	21
Inventory	21
Shipping Bin	21
Action	23
Fishing	27
Cooking	28
Time, Season, and Weather	31
Time	31
Season	31
Weather	31
End Game	32
Menu Game	33
Flow Permainan	34
Bonus	35
Furnitures (+1 per furnitur)	35
GUI (+3 - +10)	36
Keyboard and Mouse Input (+1 - +3)	36
Free Market (+3)	36
Lets Go Gambling! (+2)	37
Save dan Load (+3)	39
NPC Easter Egg (+1 per NPC)	39
Kelompok	40

<b>QnA</b>	<b>40</b>
<b>Asistensi</b>	<b>40</b>
<b>Pengumpulan</b>	<b>41</b>
Milestone 1 - 25/04/2025	41
Milestone 2 - 30/04/2025 - 01/06/2025	41
Source Code	41
Buklet	41
<b>Demo</b>	<b>42</b>
<b>Extras</b>	<b>43</b>

## Glosarium

Land	Tile yang berbentuk tanah
Soil	Land yang sudah dibajak
Tile	Tile adalah satuan petak berukuran 1x1 dalam <i>map</i>
Fiance	NPC dalam status tunangan (proposed).
Spouse	NPC yang sudah dinikahi. (married). NPC hanya dapat dinikahi jika sudah menjadi fiance.

## Pedoman

- Implementasikan prinsip [SOLID](#).
- Untuk memudahkan fase *development*, **pertimbangkan untuk implementasikan *cheat*** sesuai kebutuhan.
- Lakukan **validasi** untuk setiap input yang akan dimasukkan.
- Gunakan ***build automation tools*** seperti **Gradle** atau **Maven** untuk melakukan manajemen *build* dan *package*.
- Apabila Anda ingin mengimplementasikan ***Graphical User Interface (GUI)***, maka **rencanakan sejak awal**, tidak bisa dikerjakan mendadak di akhir. Pastikan program kalian cukup modular dari awal pengerjaan. Sebagai referensi, Anda bisa menerapkan [MVC](#). Ingat kalian punya **satu bulan** yang tergolong cukup lama.
- Dalam mengerjakan program, **lakukan asesmen untuk meninjau** bagian mana saja yang **perlu konkurensi** dan **tidak perlu konkurensi**. Tidak semua bagian perlu dikonkurensikan. Kalian cukup menerapkan konkurensi pada **aksi yang relevan**
- Penggunaan ***lock*** harus **dilakukan secara hati-hati**, jangan sampai menyebabkan [deadlock](#). Apabila tidak menggunakan *synchronized* maupun *lock*, hati-hati pula terhadap [race condition](#).
- Buat program Anda **mudah diobservasi** sejak awal. Lakukan *logging* secara terstruktur untuk bagian-bagian kode yang dianggap penting. Alih-alih hanya menggunakan `println()`, coba eksplorasi pustaka *logger* seperti **SLF4J**. Anda **dapat memberi nama *thread*** untuk memudahkan proses *debug* melalui *log* yang dihasilkan.
- Penggunaan Generative AI **sangat tidak disarankan** karena tugas besar telah didesain sedemikian rupa agar sulit dikerjakan oleh Generative AI. Jika terdapat penggunaan Generative AI dalam pengerjaan tugas besar ini, seluruh anggota kelompok akan **menerima sanksi berat sesuai dengan aturan akademik**.
- **KERJAKAN SECARA TERSTRUKTUR DAN KOORDINASIKAN PEKERJAAN ANDA**. Program berbasis konkurensi sangat rentan menimbulkan *breaking change* karena banyak *error* yang hanya akan muncul ketika *runtime*.

Terapkan *Git workflow* (misalnya gunakan skema *branching*) untuk memastikan *commit* bermasalah dapat di-*revert* dengan mudah.

- Uji program kalian dengan saksama dan menyeluruh sebelum demo. Hanya karena program kalian bisa di-*compile*, bukan berarti program kalian *foolproof*. Seluruh fitur juga harus diuji bersamaan (*integration test*), sebab fitur dapat saling mempengaruhi fitur lainnya. **GUNAKAN EMPATI DAN BERSIKAP INISIATIF, jangan hanya memedulikan pekerjaan Anda saja.**

## Deskripsi Persoalan

### “That Time I Became a Farming Game Dev to Save the World”



Gambar 1.1 The Labprominos

Setelah kegagalannya yang ke-420 dalam mencoba menguasai wilayah Danville, **Dr. Asep Spakbor** mulai merasakan kelelahan yang mendalam. Tidak ada ledakan besar, tidak ada kekacauan spektakuler—hanya kekalahan yang sunyi. Seperti biasa, **Agen Purry** telah menghentikannya. Tapi kali ini... rasanya berbeda. Bukan kekalahan biasa, tapi semacam... titik balik. Atau mungkin, titik lelah.

Sayangnya, masalah tidak berhenti di sana. Kondisi ekonomi global memburuk. Tarif ekspor-impor melonjak, bahan baku *inator* naik harga, dan investasi Dr. Asep anjlok semua. Saat ia menatap saldo tabungannya yang semakin menipis, dan menyaksikan nilai tukar mata uang Danville merosot tajam, satu kenyataan menghantam:

#### “Dia akan bangkrut”

Bukan hanya gagal menguasai dunia—dunianya sendiri pun hancur berantakan.

Dalam keputusan itu, Dr. Asep memutuskan untuk **me-reset hidupnya** dengan melakukan sesuatu yang belum pernah ia lakukan: **bertani**.

**“Kalau aku tidak bisa menguasai dunia, setidaknya aku bisa menguasai lahan dan tanaman!”**

Kata Dr. Asep Spakbor dengan semangat yang... agak terlalu berapi-api.

Sayangnya, seperti biasa, ide brilian itu berubah jadi bencana. Bukannya panen, tidak ada tanamannya yang berhasil tumbuh. Satu-satunya yang berhasil tumbuh... adalah frustrasi.

Di sisi lain, **O.W.C.A**—Organisasi Warga Cool Abiez—mulai melihat sesuatu yang tidak pernah mereka temui sebelumnya: kesempatan untuk berdamai. Untuk pertama kalinya, Dr. Asep tidak sedang membangun alat penghancur dunia. Ia hanya ingin menanam dan hidup sederhana. Dan O.W.C.A tahu. Jika mereka bisa menjadikan Dr. Asep Spakbor **petani sukses**, mungkin, untuk pertama kalinya dalam sejarah Danville... segalanya akan damai.

Lalu, apa hubungannya dengan kamu? Seorang agen biasa yang kelihatan paling sering menggunakan komputer padahal kamu cuma sering main Minesweeper di jam kerja.. Tapi malam itu, kamu tiba-tiba dipanggil langsung oleh **Purry** dan petinggi-petinggi lainnya melalui *briefing* rahasia.

Misi kamu jelas:

**“Buatkan game bertani khusus untuk Dr. Asep Spakbor.”**

Game yang cukup menarik untuk membuatnya belajar, cukup aman agar tidak ada kerusakan, dan cukup cerdas untuk mengubah mantan ilmuwan jahat menjadi seorang petani handal.

Kamu tidak tahu kenapa kamu yang dipilih. Tapi **agen Purry sudah mengganggu**. Dan kamu tahu, kalau seekor platipus sudah mengganggu, maka dunia sedang serius. Ini bukan cuma soal membuat game. **Ini soal menyelamatkan dunia.**



# Spesifikasi Sistem

## Ketentuan Umum

Anda diminta untuk membuat permainan berbasis **Command Line Interface (CLI)** bernama **Spakbor Hills** menggunakan bahasa pemrograman **Java**. Tuliskan tahapan untuk melakukan kompilasi dan menjalankan program pada *file* README.md yang dikumpulkan bersama dengan *source code*. **Spakbor Hills** dapat dimainkan oleh satu orang pemain. Tujuan dari permainan ini dijelaskan pada bagian [End Game](#). Untuk mewujudkan hal tersebut, ada beberapa aksi yang perlu dilakukan. Penjelasan tentang aksi dapat dibaca [di sini](#). Gambaran besar *flow* permainan dapat dibaca di bagian [ini](#).

## Ketentuan Teknis

Berikut adalah hal-hal yang wajib diimplementasikan di aplikasi yang Anda buat. Perhatikan konsep-konsep OOP serta desain dari aplikasi kalian!

1. *Inheritance*
2. *Abstract Class / Interface*
3. *Polymorphism*
4. *Generics*
5. *Exceptions*
6. *Concurrency*

Penerapan konsep-konsep tersebut dibebaskan kepada Anda. Namun, Anda wajib menerapkan seluruh konsep tersebut pada aplikasi Anda.

Anda juga harus mengimplementasikan **minimal 3 buah Design Pattern** selama proses pembuatan aplikasi. Anda dapat melihat berbagai jenis design pattern, definisi, dan contoh implementasinya melalui pranala berikut [Design Patterns \(refactoring.guru\)](#).

## Entitas Permainan

### Player

Player adalah karakter utama yang dikendalikan pemain. Setiap Player memiliki berbagai data mengenai status Player saat ini seperti jumlah *gold*, isi *inventory*, dan sebagainya.

Seorang Player dapat melakukan berbagai aksi. Aksi dapat membutuhkan energi atau menambah energi. Detail mengenai aksi yang dapat dilakukan Player dapat dilihat di spesifikasi bagian Action. Di awal permainan, energi Player adalah maksimum yaitu 100. Jika energi mencapai 0 saat bekerja, Player memiliki cadangan energi hingga -20 untuk melakukan kegiatan lainnya. Jika energi mencapai -20, *Player* akan langsung melakukan *action* [Sleeping](#) dengan ketentuan yang berlaku. Ingat untuk melakukan validasi energi *Player* sebelum melakukan sebuah *action*. Sebagai contoh, jika energi Player tersisa 5, Player tetap dapat melakukan *action* dengan *energy cost* 10, sehingga tersisa energi sebanyak -5.

Player setidaknya memiliki atribut sebagai berikut.

Nama Atribut	Keterangan
name	Nama karakter pemain yang dibuat oleh pengguna.
gender	Menyatakan jenis kelamin pemain.
energy	Energi yang dimiliki pemain saat ini. Energi digunakan untuk melakukan aktivitas. Nilai maksimum energi adalah 100. Energi awal sejumlah maksimum energi.
FarmName	Nama Farm yang dimiliki oleh Player
partner	Partner dapat berupa <a href="#">fiance/spouse</a> .
gold	Jumlah uang ( <i>gold</i> ) yang dimiliki oleh pemain.
inventory	Objek inventori yang menyimpan daftar <i>item</i> yang sedang dibawa oleh pemain. Mengacu pada spesifikasi Inventory.
location	Lokasi saat ini dari pemain di dalam <i>map</i>

## Farm

Farm adalah struktur utama yang merepresentasikan suatu dunia permainan. Farm menyimpan seluruh informasi penting sebuah dunia seperti pemain, kondisi dunia, dan sebagainya. Contoh atribut yang bisa dimiliki oleh Farm.

Nama Atribut	Keterangan
name	Nama dari <i>Farm</i> yang dibuat oleh pemain
Player	Objek pemain utama yang merepresentasikan karakter yang dikendalikan oleh pengguna. Mengacu pada spesifikasi Player.
FarmMap	Representasi peta <i>Farm</i> . Mengacu pada spesifikasi Farm Map.
time	Waktu saat ini di hari yang sedang berlangsung dalam Farm.
day	Hari ke-berapa dalam permainan.
season	Musim saat ini dalam permainan.
weather	Cuaca saat ini dalam permainan.

## House

House adalah bangunan tempat tinggal pemain yang secara otomatis tersedia di setiap Farm. House berperan sebagai lokasi utama bagi pemain untuk melakukan aksi yang berkaitan dengan rumah, seperti istirahat, memasak, dan menonton TV.

## NPC

NPC adalah karakter-karakter yang dapat berinteraksi dengan pemain dalam permainan. Pemain dapat berinteraksi dan menikahi **seluruh NPC**. Apabila pemain ingin berinteraksi dengan seorang NPC, maka pemain harus **mengunjungi** rumah NPC tersebut. Asumsikan bahwa NPC selalu ada di rumah miliknya. Pengecualian untuk hal ini adalah **Emily** yang tinggal di **Store**.

Setiap NPC setidaknya memiliki atribut sebagai berikut.

Nama Atribut	Keterangan
name	Nama dari NPC
heartPoints	Seberapa dekat seorang NPC dengan pemain. Poin semakin tinggi, semakin tinggi pemain dengan NPC tersebut. <b>Heart points akan bernilai 0 di awal permainan.</b>  <b>Untuk NPC heartPoints tertinggi adalah 150</b>
lovedItems	Daftar item yang dicintai NPC tersebut
likedItems	Daftar item yang disukai NPC tersebut
hatedItems	Daftar item yang dibenci NPC tersebut
relationshipsStatus	Status hubungan NPC, yaitu <i>single</i> , <i>fiance</i> dan <i>spouse</i> .

CATATAN: Secara *default*, anggap “neutral items” sebagai items yang tidak termasuk ke *loved*, *liked* atau *hated items* kecuali ketika diberikan keterangan khusus (misalnya Mayor Tadi yang tidak memiliki *neutral items*, seluruh *item* lain yang bukan *loved* atau *liked* merupakan item *hated*)

Berikut adalah daftar NPC yang harus ada di dalam permainan beserta nilai atribut mereka masing - masing.

### 1. Mayor Tadi

Mayor Tadi adalah wali kota Spakbor Hills. Dirinya adalah seorang wali kota yang memiliki minat untuk barang-barang yang mewah dan langka. Dirinya membenci apa pun yang dianggap “barang murahan” olehnya.

Atribut	Nilai
name	Mayor Tadi
lovedItems	Legend
likedItems	Angler, Crimsonfish, Glacierfish

hatedItems	Seluruh <i>item</i> yang bukan merupakan <i>lovedItems</i> dan <i>likedItems</i>
------------	--

## 2. Caroline

Caroline adalah seorang tukang kayu lokal di Spakbor Hills. Ia juga memiliki hobi mendaur-ulang bahan-bahan dasar dan mengubahnya menjadi suatu karya seni. Dirinya lebih menyukai makanan yang tawar dan membenci makanan pedas.

Atribut	Nilai
name	Caroline
lovedItems	Firewood, Coal
likedItems	Potato, Wheat
hatedItems	Hot Pepper

## 3. Perry

Perry adalah seorang penulis yang baru saja menerbitkan buku pertamanya. Dia memutuskan untuk pindah ke Spakbor Hills untuk kabur dari lalu-lalang kota. Dia adalah seseorang yang *soft-spoken* dan pemalu. Dia senang ketika kamu menanyakan tentang novel terbarunya. Dirinya menyukai *berries* dan membenci semua item ikan.

Atribut	Nilai
name	Perry
lovedItems	Cranberry, Blueberry
likedItems	Wine
hatedItems	Seluruh item Fish

#### 4. Dasco

Dasco adalah pemilik kasino besar di Spakbor Hills. Dia adalah seorang pria yang menyukai hidangan-hidangan yang mewah. Namun, dia membenci orang-orang yang memberikannya bahan baku mentah, dia merasa bahwa itu merupakan tindakan yang merendahkan dirinya.

Atribut	Nilai
name	Dasco
lovedItems	The Legends of Spakbor, Cooked Pig's Head, Wine, Fugu, Spakbor Salad
likedItems	Fish Sandwich, Fish Stew, Baguette, Fish n' Chips
hatedItems	Legend, Grape, Cauliflower, Wheat, Pufferfish, Salmon

#### 5. Emily

Emily adalah seorang koki yang juga **bekerja dan tinggal di toko lokal Spakbor Hills**. Makanan yang disajikan di toko miliknya dibuat dari bahan-bahan yang ia tumbuhkan di kebun miliknya sendiri. Emily selalu senang ketika seseorang memberikannya bibit yang bisa ditambahkan ke kebun lokal miliknya. Ia juga menyukai ketika seseorang memberikannya bahan baku ikan untuk restoran miliknya. Toko Emily tidak hanya menyajikan hidangan saja tapi juga berbagai hal yang dibutuhkan pemain.

Atribut	Nilai
name	Emily
lovedItems	Seluruh item seeds
likedItems	Catfish, Salmon, Sardine
hatedItems	Coal, Wood

## 6. Abigail

Abigail adalah seorang perempuan yang *outgoing* dan senang melakukan eksplorasi alam. Dia menyukai buah-buahan karena ia membutuhkan energi untuk melakukan ekspedisinya sehari-hari. Walaupun dia menyukai buah-buahan, tetapi dia membenci sayuran.

Atribut	Nilai
name	Abigail
lovedItems	Blueberry, Melon, Pumpkin, Grape, Cranberry
likedItems	Baguette, Pumpkin Pie, Wine
hatedItems	Hot Pepper, Cauliflower, Parsnip, Wheat

## Items

*Items* adalah objek-objek yang dapat digunakan atau dimiliki oleh *Player* dalam permainan. *Item* yang dicantumkan adalah referensi. Silahkan disesuaikan dengan kebutuhan. *Items* dalam **Spakbor Hills** terbagi menjadi 6 kategori utama, yaitu:

### 1. Seeds

*Seeds* adalah biji-bijian yang dapat dibeli dari *store*. *Seeds* dapat ditanam pada *soil* yang telah dibajak (*tilled*) dan akan tumbuh menjadi *Crops* dalam waktu yang telah ditentukan. *Seed* hanya dapat ditanam pada musim yang sesuai. Berikut adalah beberapa contoh *seed* yang dapat digunakan sesuai dengan musimnya. Harga jual *seed* adalah setengah harga belinya. *Seeds* harus disiram setiap 2 hari sekali saat cuaca panas

No	Season	Seed Name	Days To Harvest	Buy Price
1	Spring	Parship Seeds	1	20g
2		Cauliflower Seeds	5	80g
3		Potato Seeds	3	50g
4		Wheat Seeds	1	60g

5	Summer	Blueberry Seeds	7	80g
6		Tomato Seeds	3	50g
7		Hot Pepper Seeds	1	40g
8		Melon Seeds	4	80g
9	Fall	Cranberry Seeds	2	100g
10		Pumpkin Seeds	7	150g
11		Wheat Seeds	1	60g
12		Grape Seeds	3	60g
13	Winter	Tidak ada <i>seed</i> yang dapat tumbuh saat <i>winter</i>		

## 2. Fish

*Fish* adalah ikan yang didapat dari memancing (*fishing*). Terdapat 3 jenis *fish* yang dapat ditangkap:

- *Common fish*, yaitu ikan yang dapat ditangkap pada musim, waktu, dan cuaca apapun, yang terdiri dari:

No	Nama Ikan	Season	Waktu	Weat her	Lokasi
1	Bullhead	<i>Any</i>	<i>Any</i>	<i>Any</i>	Mountain Lake
2	Carp	<i>Any</i>	<i>Any</i>	<i>Any</i>	Mountain Lake, Pond
3	Chub	<i>Any</i>	<i>Any</i>	<i>Any</i>	Forest River, Mountain Lake

- *Regular fish*, yaitu ikan yang dapat ditangkap pada musim, waktu, dan/atau kondisi tertentu, yang terdiri dari:

No	Nama Ikan	Season	Waktu	Weat her	Lokasi
----	-----------	--------	-------	-------------	--------



1	Largemouth Bass	<i>Any</i>	06.00-18.00	<i>Any</i>	Mountain Lake
2	Rainbow Trout	<i>Summer</i>	06.00-18.00	<i>Sunny</i>	Forest River, Mountain Lake
3	Sturgeon	<i>Summer, Winter</i>	06.00-18.00	<i>Any</i>	Mountain Lake
4	Midnight Carp	<i>Winter, Fall</i>	20.00-02.00	<i>Any</i>	Mountain Lake, Pond
5	Flounder	<i>Spring, Summer</i>	06.00-22.00	<i>Any</i>	Ocean
6	Halibut	<i>Any</i>	06.00-11.00, 19.00-02.00	<i>Any</i>	Ocean
7	Octopus	<i>Summer</i>	06.00-22.00	<i>Any</i>	Ocean
8	Pufferfish	<i>Summer</i>	00.00-16.00	<i>Sunny</i>	Ocean
9	Sardine	<i>Any</i>	06.00-18.00	<i>Any</i>	Ocean
10	Super Cucumber	<i>Summer, Fall, Winter</i>	18.00-02.00	<i>Any</i>	Ocean
11	Catfish	<i>Spring, Summer, Fall</i>	06.00-22.00	<i>Rainy</i>	Forest River, Pond
12	Salmon	<i>Fall</i>	06.00-18.00	<i>Any</i>	Forest River

- *Legendary fish*, yaitu ikan yang dapat ditangkap pada musim, waktu, dan/atau kondisi tertentu, serta memiliki tingkat kesulitan penangkapan yang tinggi, yang terdiri dari:

No	Nama Ikan	Season	Waktu	Weat her	Lokasi
----	-----------	--------	-------	----------	--------

1	Angler	<i>Fall</i>	08.00-20.00	<i>Any</i>	Pond
2	Crimsonfish	<i>Summer</i>	08.00-20.00	<i>Any</i>	Ocean
3	Glacierfish	<i>Winter</i>	08.00-20.00	<i>Any</i>	Forest River
4	Legend	<i>Spring</i>	08.00-20.00	<i>Rainy</i>	Mountain Lake

*Fish* yang baru ditangkap termasuk ke dalam *Edible Items*, sehingga dapat langsung dimakan oleh *Player* dan akan menghasilkan efek berupa penambahan *energy Player* sebesar 1 poin. *Fish* juga dapat dimasak menjadi *Food* dengan *action Cooking*.

Harga jual suatu ikan dihitung berdasarkan lima faktor: jumlah *season*, durasi rentang waktu/jam, jumlah variasi *weather*, jumlah lokasi tempat ikan tersebut bisa diperoleh, serta jenis ikan itu sendiri. Perhitungannya menggunakan formula berikut:

$$\frac{4}{\text{banyak season}} \times \frac{24}{\text{jumlah jam}} \times \frac{2}{\text{jumlah variasi weather}} \times \frac{4}{\text{banyak lokasi}} \times C$$

$C = 10$  untuk *common fish*, 5 untuk *regular fish*, 25 untuk *legendary fish*

Contoh:

Halibut, yang merupakan *regular fish*, dapat ditangkap pada musim apapun (*Any* = 4 musim), pada waktu 06.00-11.00 dan 19.00-02.00 (5 jam + 7 jam = 12 jam), pada cuaca apapun (*Any* = 2 cuaca), dan di Ocean (1 lokasi). Oleh karena itu, harga jual Halibut adalah:

$$\frac{4}{4} \times \frac{24}{12} \times \frac{2}{2} \times \frac{4}{1} \times 5 = 40g$$

### 3. Crops

*Crops* adalah hasil dari *seeds* yang dipanen (*harvest*). Semua *crop* yang telah dipanen tidak akan dapat dipanen lagi (berubah kembali menjadi *soil*). *Crops* dapat dijual, di-*gift*, atau dimasak sesuai dengan keperluan resep masakan. *Crops* dapat dibeli dari [store](#) dengan harga yang telah ditentukan. *Crops* yang telah dipanen dapat dikategorikan sebagai *Edible Items*. **Setiap *crop* yang dimakan akan mengembalikan energi sebanyak 3.**

No	Nama <i>Crop</i>	Harga Beli (per <i>crop</i> )	Harga Jual (per <i>crop</i> )	Jumlah <i>Crop</i> per Panen
1	Parsnip	50g	35g	1
2	Cauliflower	200g	150g	1
3	Potato	-	80g	1
4	Wheat	50g	30g	3
5	Blueberry	150g	40g	3
6	Tomato	90g	60g	1
7	Hot Pepper	-	40g	1
8	Melon	-	250g	1
9	Cranberry	-	25g	10
10	Pumpkin	300g	250g	1
11	Grape	100g	10g	20

#### 4. Food

*Food* adalah sebutan yang digunakan untuk *item-item* yang mampu memulihkan energi karakter. *Food* dapat diperoleh dari memasak atau membeli. *Food* dapat dijual atau di-*gift* kepada NPC yang diinginkan. Semua *Food* termasuk Edible Items.

No	Nama Makanan	Energi	Harga Beli	Harga Jual
1	Fish n' Chips	+50	150g	135g
2	Baguette	+25	100g	80g
3	Sashimi	+70	300g	275g
4	Fugu	+50	-	135g
5	Wine	+20	100g	90g

6	Pumpkin Pie	+35	120g	100g
7	Veggie Soup	+40	140g	120g
8	Fish Stew	+70	280g	260g
9	Spakbor Salad	+70	-	250g
10	Fish Sandwich	+50	200g	180g
11	The Legends of Spakbor	+100	-	2000g
12	Cooked Pig's Head	+100	1000g	0g

5. Equipment

Equipment adalah jenis *item* yang dapat digunakan oleh Player. Kategori ini mencakup *items* seperti Hoe, Watering Can, Pickaxe, dan Fishing Rod.

6. Misc.

Misc. mengandung *item* acak yang tidak tergolong ke dalam 5 kategori di atas, seperti *coal* dan *firewood*. Harga untuk setiap barang dalam kategori misc. dapat ditentukan sendiri dengan ketentuan bahwa harga jual harus lebih murah dari harga beli.

## Maps

### Farm Map

*Farm map* adalah lokasi dimana lahan tanam dan rumah kalian berada. Selain dua lokasi tersebut, terdapat pula *pond* dan *shipping bin* yang dapat dilakukan untuk aktivitas lain.

*Map* ini berbentuk persegi dengan luas 32x32 tiles. Setiap tiles dapat berisikan *deployed object* dan *tile* yang dapat digunakan. Setiap *deployed object* akan dinotasikan dengan sebuah huruf pada tampilan peta. *Deployed object* tersebut adalah:

- House (h) : rumah pemain, berbentuk 6x6
- Pond (o) : tempat untuk memancing, berbentuk 4x3
- Shipping bin (s) : tempat untuk menjual barang, berbentuk 3x2

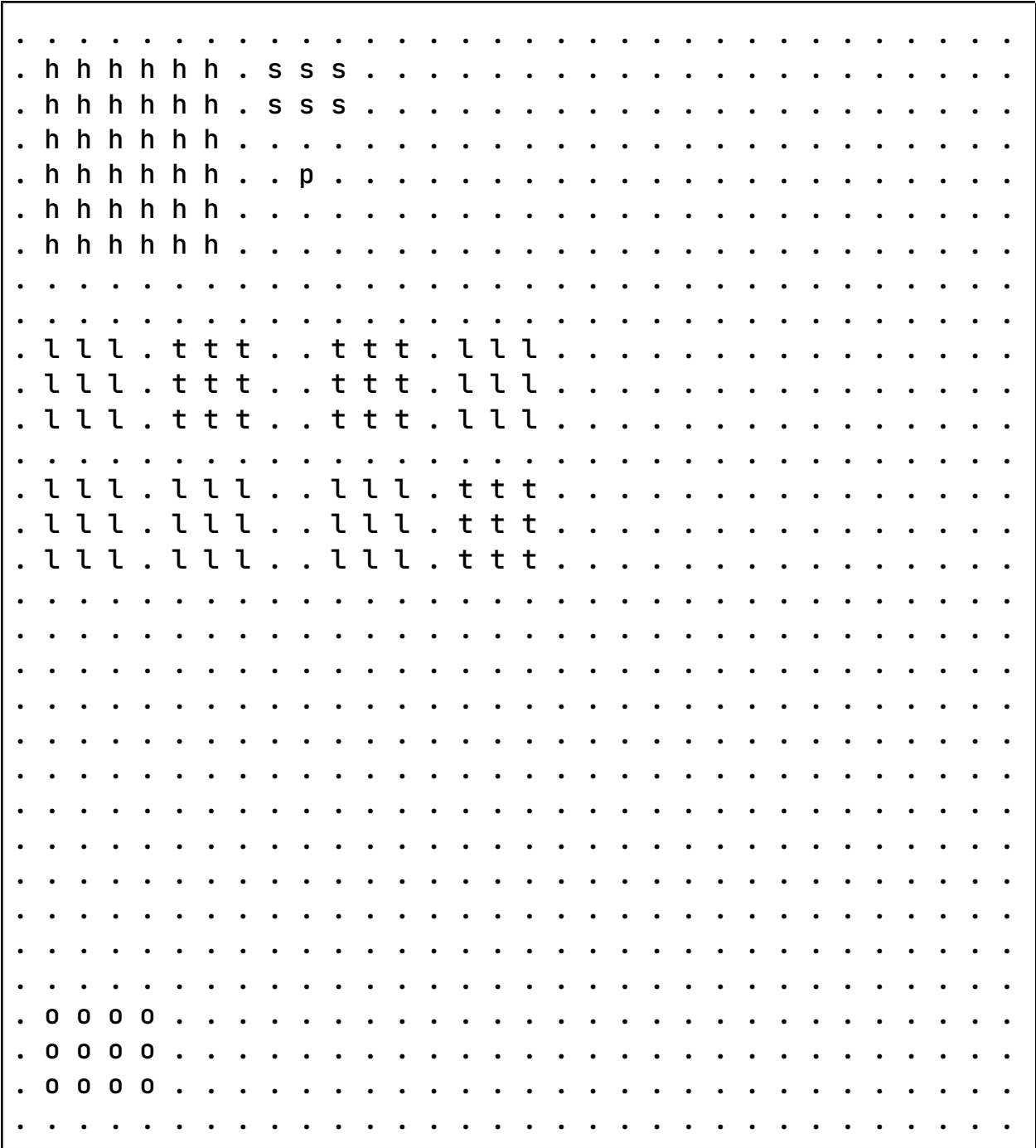
Setiap Farm pasti akan berisi ketiga objek tersebut. Pada awal kreasi *Farm*, penempatan *house* dan *pond* akan di-randomize, sedangkan *shipping bin* akan selalu berjarak 1 petak dari rumah.

Selain *deployed object*, seluruh *tile* (petak 1x1) yang tidak ditempati dapat digunakan sebagai lahan untuk bercocok tanam. Terdapat 3 jenis tile sebagai berikut:

- Tillable land (.) : tile yang dapat disiapkan untuk tanam
- Tilled land (t) : tile yang dapat ditanamkan *seed*
- Planted land (l) : tile yang sudah ditanamkan *seed*

Terdapat juga entitas Player yang menunjukkan lokasi pemain. Player dinotasikan dengan huruf *p*. Sebuah Player dapat berinteraksi dengan *deployed object* dengan berdiri di sebelah objek tersebut. Tentunya Player juga tidak bisa menembus *deployed object*! Selain itu, untuk berinteraksi dengan sebuah *tile* seorang Player harus berada di atas *tile* tersebut.

Untuk pergi keluar dari *Farm*, misal untuk *visiting*, Player harus bergerak ke salah satu ujung *map*. Hanya pada ujung *map* Player dapat melakukan action *visiting*.



Contoh *Farm Map*

## World Map

*World Map* adalah area di luar *village* yang dapat dikunjungi. World Map terdiri dari tempat tinggal NPC, Forest River, Mountain Lake, Ocean, dan Store. Implementasi visual dari *world map* bersifat opsional. World Map hanya dapat diakses dengan berada di salah satu ujung dari Farm Map. Player akan melakukan *action visiting* agar dapat mengakses World Map.

## Gold

Gold adalah mata uang yang digunakan dalam **Spakbor Hills**. Gold dapat diperoleh dengan melakukan penjualan pada [shipping bin](#), sehingga Gold akan diperoleh setiap hari jika terdapat penjualan yang dilakukan. Jumlah gold yang diterima tergantung pada item yang dijual.

**NB: Buatlah sebuah [file](#) yang mengandung daftar harga jual dan harga beli *item* yang dapat dijual dan dibeli untuk memudahkan pengembangan.**

## Inventory

Inventory merupakan koleksi item yang dimiliki oleh Player. Tidak ada limitasi jumlah item yang dapat disimpan dalam Inventory. Di awal sebuah permainan, Inventory Player telah terisi dengan *item-item* sebagai berikut.

1. Parsnips Seeds x15
2. Hoe
3. Watering Can
4. Pickaxe
5. Fishing Rod

## Shipping Bin

*Shipping Bin* merupakan mekanik yang memungkinkan pemain untuk menjual produknya ke pasar. *Shipping bin* terletak di sebelah kanan rumah pemain. Pemain mampu melakukan penjualan sebanyak satu kali sehari. Di dalam *bin* tersebut terdapat slot-slot yang memungkinkan kalian menjual banyak barang secara langsung. **Item yang sudah masuk ke dalam shipping bin tidak dapat dikembalikan ke dalam *inventory* kembali.**

Penjualan melalui *shipping bin* pada malam hari ketika Player sedang tidur. Uang dari hasil penjualan nantinya akan didapat keesokan harinya. Uang yang didapat dari penjualan masing-masing item mengikuti harga jual yang telah dispesifikasikan.

Maksimal jumlah slot item unik yang dapat dijual ke dalam satu *bin* adalah 16 buah.



## Action

Berikut adalah list dari setiap *action* yang dapat dilakukan oleh Player. Pengurangan waktu bukan mengurangi waktu in-game, tetapi berjalan paralel dengan waktu dalam game. Semua aktivitas adalah aksi aktif sehingga Player tidak bisa melakukan apapun saat aktivitas dijalankan kecuali dirincikan lebih lanjut. Ingat untuk melakukan validasi energi yang perlu dikeluarkan untuk melakukan action tertentu.

No	Nama Aksi	Deskripsi	Efek	Item yang Dibutuhkan
1	Tilling	Mengubah <i>land</i> menjadi <i>soil</i>	<ul style="list-style-type: none"> <li>- 5 energi / tile</li> <li>- 5 menit dalam game / tile</li> </ul>	Hoe
2	Recover Land*	Mengubah <i>soil</i> menjadi <i>land</i>	<ul style="list-style-type: none"> <li>- 5 energi / tile</li> <li>- 5 menit dalam game / tile</li> </ul>	Pickaxe
3	Planting	Menanam <i>seed</i> di <i>soil</i> .	<ul style="list-style-type: none"> <li>- 5 energi / seed</li> <li>- 5 menit dalam game / seed</li> </ul>	Seed yang hendak ditanam
4	Watering	Bergerak ke <i>soil</i> yang telah ditanam dan melakukan <i>watering</i> .	<ul style="list-style-type: none"> <li>- 5 energi / soil</li> <li>- 5 menit dalam game / soil</li> </ul>	Watering Can
5	Harvesting	Berada di area <i>plant</i> . Hanya dapat dilakukan di <i>harvestable plant</i> .	<ul style="list-style-type: none"> <li>- 5 energi / crop</li> <li>- 5 menit dalam game / plant</li> </ul>	-
6	Eating	Recharge energi sesuai dengan deskripsi <i>edible item</i>	<ul style="list-style-type: none"> <li>+ energi menurut <a href="#">Items</a></li> <li>- 5 menit dalam game</li> </ul>	<i>Edible item</i>
7	Sleeping	Melewati waktu sampai pagi. akan mengisi ulang energi penuh. Namun, jika tidur di saat energi sudah menipis,	<ul style="list-style-type: none"> <li>+ MAX_ENERGY</li> <li>+ <math>\frac{1}{2} \times \text{MAX\_ENERGY}</math> jika energi &lt; (10% *</li> </ul>	House / Beds ( <a href="#">bonus</a> )

		maka akan dikenai penalti di mana energi yang terisi ulang hanya setengah penuh. Jika energi habis saat bekerja, lihat di bagian <a href="#">Player</a> . Jika waktu telah mencapai pukul 02.00 dan Player belum tidur pada malam itu, maka akan segera otomatis pergi tidur.	MAX_ENERGY) <ul style="list-style-type: none"> <li>+ 10 jika energi == 0</li> <li>Time skip ke pagi</li> </ul>	
8	Cooking	<b>BIARKAN DIA MEMASAK!</b> Kamu bisa memasak makanan di sini. Untuk bisa memasak, kamu memerlukan bahan bakar (fuel).  Fuel yang bisa digunakan yaitu: <ul style="list-style-type: none"> <li><b>Firewood:</b> Kayu bakar. 1 kayu bisa masak 1 makanan.</li> <li><b>Coal:</b> Arang, 1 arang bisa masak 2 makanan</li> </ul>	Lihat rincian bagian <a href="#">Cooking</a>	<ul style="list-style-type: none"> <li>House, atau Stove jika mengimplimentasi <a href="#">bonus</a>,</li> <li>Recipe</li> </ul>
9	Fishing	Memancing <i>iwak</i> (ikan). <i>Player</i> dapat memancing di 4 lokasi: Forest River, Mountain Lake, Pond, dan Ocean. Setiap ikan dapat ditangkap pada musim, waktu, cuaca, dan lokasi tertentu seperti tertera pada bagian <a href="#">Items</a> nomor 2 (Fish). Mekanik <i>fishing</i> dapat dilihat	<ul style="list-style-type: none"> <li>- 5 energi / <i>attempt</i></li> <li>Lihat bagian <a href="#">fishing</a></li> </ul>	Fishing rod

		pada bagian <a href="#">Fishing</a> .		
10	Proposing	Melamar NPC untuk menjadi <i>fiance</i> . Agar sebuah lamaran dapat diterima, NPC yang akan dilamar harus sudah memiliki atribut heartPoints dengan nilai maksimal, yaitu sebesar 150 poin. Proposal Ring tidak hilang setelah digunakan ( <i>reusable</i> ).	<ul style="list-style-type: none"> <li>- 10 energi apabila lamaran diterima</li> <li>- 20 energi apabila lamaran ditolak</li> <li>- 1 jam <i>in game</i></li> </ul>	Proposal Ring
11	Marry	Menikahi NPC yang telah menjadi <i>fiance</i> . Marry dapat dilakukan paling cepat satu hari setelah NPC tersebut menjadi <i>fiance</i> . Pada hari pernikahan tersebut, <i>Player</i> akan menghabiskan waktunya seharian bersama <i>spouse</i> atau pasangan hidupnya. Permainan akan langsung <i>time skip</i> ke pukul 22.00 dan <i>Player</i> dikembalikan ke rumah. Proposal Ring tidak hilang setelah digunakan.	<ul style="list-style-type: none"> <li>- 80 energi</li> <li>- <i>Time skip</i> ke 22.00</li> </ul>	Proposal Ring
12	Watching	Menonton televisi, hanya dapat dilakukan di rumah <i>Player</i> .	<ul style="list-style-type: none"> <li>- 15 menit dalam game</li> <li>- 5 energi</li> </ul>	House / TV ( <a href="#">bonus</a> )
13	Visiting	Mengunjungi area di luar Farm	<ul style="list-style-type: none"> <li>- 15 menit dalam game</li> <li>- 10 energi</li> </ul>	-

13	Chatting	Berbicara dengan NPC, hanya dapat dilakukan di rumah NPC tersebut.	<ul style="list-style-type: none"> <li>- 10 menit dalam game atau buat percakapan sendiri</li> <li>• 10 heartPoints untuk NPC tersebut</li> <li>• - 10 energi</li> </ul>	-
14	Gifting	<p>Memberikan <i>item</i> kepada seorang NPC. Item yang diberikan akan hilang dari <i>inventory</i>.</p> <p>Efek untuk aksi ini bergantung kepada jenis <i>item</i> yang diberikan.</p> <ul style="list-style-type: none"> <li>• lovedItems: +25 heartPoints</li> <li>• likedItems: +20 heartPoints</li> <li>• hatedItems: -25 heartPoints</li> </ul> <p>Apabila item yang diberikan tidak ada di loved, liked, atau hated items maka items tersebut adalah items <b>neutral</b> dan tidak memberikan poin apapun.</p> <p>Aksi ini hanya bisa dilakukan di rumah NPC tersebut.</p>	<ul style="list-style-type: none"> <li>- 10 menit dalam game atau buat percakapan sendiri</li> <li>• Menambah atau mengurangi heartPoints NPC tersebut tergantung dengan jumlah item.</li> <li>• - 5 energi</li> </ul>	Gift
15	Moving	Berpindah dari location A ke location B	Memindahkan posisi Player dalam Farm Map	-

16	Open Inventory	Memperlihatkan isi inventory	-	-
17	Show Time	Memperlihatkan waktu (musim, hari, waktu)	-	-
18	Show Location	Memperlihatkan lokasi pengguna	-	-
19	Selling	Menaruh item ke dalam <i>shipping bin</i> untuk dijual.	Menghentikan waktu selama penjualan dan menghabiskan waktu 15 menit dalam game setelah selesai Selling.	Shipping Bin

## Fishing

*Fishing* merupakan *basic skill* yang harus dimiliki oleh semua orang di Spakbor Hills. Oleh karena itu, Dr. Asep Spakbor juga harus bisa (atau mau belajar) memancing. *Fishing* dapat dilakukan di 4 lokasi, yakni Pond, Mountain Lake, Forest River, dan Ocean.

Berhubung Pond terdapat pada [Farm Map](#), maka untuk memancing di Pond, *Player* harus berada di Farm-nya terlebih dahulu dan bergerak (melakukan *action Moving*) hingga berjarak 1 *tile* dari Pond (bukan di atas Pond). Sedangkan untuk 3 lokasi lainnya, *Player* harus mengunjungi (melakukan *action Visiting*) ke Mountain Lake, Forest River, atau Ocean terlebih dahulu baru melakukan aktivitas memancing.

Pada saat action Fishing dimulai, *world time* akan dihentikan, ditambah 15 menit, dan baru berlanjut kembali setelah action Fishing selesai. Mekanik dari action Fishing ini adalah dengan menggunakan RNG (*random number generator*),

- Tipe *common*: tebak angka 1-10 (maks. 10 percobaan)
- Tipe *regular*: tebak angka 1-100 (maks. 10 percobaan)
- Tipe *legendary*: tebak angka 1-500 (maks. 7 percobaan)

Penjelasan skenario:

1. *Player* melakukan action Fishing
2. Hentikan *world time*, kurangi energi *Player* sebanyak 5 poin, tambahkan 15 menit ke *world time*
3. Permainan akan melakukan *randomizing* terhadap ikan yang akan ditangkap, misalnya hasil *random*-nya adalah ikan Halibut yang merupakan *regular fish*
4. Permainan akan melakukan *randoming* terhadap angka 1-100, misalnya hasil *random*-nya adalah 77
5. *Player* diberikan 10 kesempatan untuk menebak angka tersebut
6. Apabila *Player* berhasil menebak dengan benar, maka ikan tersebut akan ditambahkan ke *inventory*
7. Lanjutkan *world time*
8. *Action* Fishing berakhir

## Cooking

*Cooking* adalah bakat tersembunyi dari Dr. Asep Spakbor. Walaupun ia telah dimasak oleh Agen Purry, ia masih bisa memasak hal yang lain seperti makanan. Berikut merupakan spesifikasi aksi *Cooking*.

1. *Cooking* hanya dapat dilakukan di dalam rumah.
2. Jika kalian mengimplementasi bonus furnitur Stove, maka aktivitas memasak hanya dapat dilakukan melalui interaksi dengan Stove.
3. Dalam memasak, terdapat beberapa parameter yang dapat kalian isi, yaitu:
  - a. Resep yang akan dimasak. Kalian hanya dapat memasak satu resep untuk satu aksi
  - b. Bahan-bahan yang diperlukan untuk resep tersebut
  - c. Bahan bakar untuk memasak.
4. Jika ketiga parameter tersebut telah terpenuhi, maka aktivitas memasak dapat dilakukan.
5. Setiap aksi memasak akan menghabiskan *fuel*/bahan bakar. Jumlah makanan yang bisa dimasak per *fuel* dapat dilihat pada deskripsi singkat di bagian **Action**.
6. Kalian dibebaskan menuliskan alur program yang mungkin dapat memenuhi spesifikasi dari nomor 1 - 5.

7. Durasi memasak semua makanan adalah **1 jam** dan aksi ini akan bersifat pasif, yang membuat Player dapat melakukan hal lain selama proses memasak dilaksanakan.
8. Walaupun sebagai aksi pasif, memasak tetap memerlukan energi pada saat inisiasinya, yaitu - 10 energi untuk tiap percobaan memasak.
9. Setelah memasak selesai, makanan akan langsung masuk ke inventory.

Berikut merupakan penjelasan dari resep-resep yang dapat dimasak.

Item ID	Nama	Bahan Baku	Cara mendapatkan/ <i>unlock</i>
recipe_1	Fish n' Chips	<ul style="list-style-type: none"> <li>- Any Fish x2</li> <li>- Wheat x1</li> <li>- Potato X1</li> </ul>	Beli di store
recipe_2	Baguette	<ul style="list-style-type: none"> <li>- Wheat x3</li> </ul>	<i>Default</i> /Bawaan
recipe_3	Sashimi	<ul style="list-style-type: none"> <li>- Salmon x3</li> </ul>	Setelah memancing 10 ikan (jumlah akumulasi ikan yang dipancing, bukan 10 jenis ikan berbeda)
recipe_4	Fugu	<ul style="list-style-type: none"> <li>- Pufferfish x1</li> </ul>	Memancing pufferfish
recipe_5	Wine	<ul style="list-style-type: none"> <li>- Grape x2</li> </ul>	<i>Default</i> /Bawaan
recipe_6	Pumpkin Pie	<ul style="list-style-type: none"> <li>- Egg x1</li> <li>- Wheat x1</li> <li>- Pumpkin x1</li> </ul>	<i>Default</i> /Bawaan
recipe_7	Veggie Soup	<ul style="list-style-type: none"> <li>- Cauliflower x1</li> <li>- Parsnip x1</li> <li>- Potato x1</li> <li>- Tomato x1</li> </ul>	Memanen untuk pertama kalinya
recipe_8	Fish Stew	<ul style="list-style-type: none"> <li>- Any fish 2x</li> </ul>	Dapatkan "Hot

		<ul style="list-style-type: none"> <li>- Hot pepper x1</li> <li>- Cauliflower x2</li> </ul>	Pepper” terlebih dahulu agar bisa membuka resepnya
recipe_9	Spakbor Salad	<ul style="list-style-type: none"> <li>- Melon x1</li> <li>- Cranberry x1</li> <li>- Blueberry x1</li> <li>- Tomato x1</li> </ul>	<i>Default/Bawaan</i>
recipe_10	Fish Sandwich	<ul style="list-style-type: none"> <li>- Any fish x1</li> <li>- Wheat 2x</li> <li>- Tomato 1x</li> <li>- Hot Pepper 1x</li> </ul>	Beli di store
recipe_11	The Legends of Spakbor	<ul style="list-style-type: none"> <li>- Legend fish 1x</li> <li>- Potato 2x</li> <li>- Parsnip 1x</li> <li>- Tomato 1x</li> <li>- Eggplant 1x</li> </ul>	Memancing “Legend”



## Time, Season, and Weather

### Time

Berikut merupakan rincian spesifikasi terkait waktu di dalam permainan:

1. Waktu berjalan di dalam permainan yaitu: “1 detik di dunia nyata = 5 menit di dunia permainan”.
2. Waktu dibagi menjadi 2 fase, yaitu siang (06.00 - 17.59) dan malam (18.00 - 05.59).
3. Waktu akan terus berjalan apapun kondisinya, baik saat pemain *idle* maupun saat pemain melakukan aksi
4. Aksi **tidur** mampu membuat pemain melakukan *time-skipping* atau melompati waktu.
5. Semua aksi/*state* yang bergantung pada waktu harus mengikuti hukum waktu yang telah ditentukan.

### Season

Hal lain yang perlu diperhatikan juga adalah musim (*Season*). 1 Season berdurasi 10 hari. Terdapat hal-hal yang perlu diperhatikan setiap *season*, yaitu:

1. Jenis ikan yang hanya muncul di *season* tertentu
2. *Crops* yang mati saat perubahan *season*

### Weather

Terdapat 2 jenis *weather* pada permainan Spakbor Hills, yaitu Rainy dan Sunny. Seperti namanya, Rainy merupakan kondisi ketika hujan turun pada hari tersebut, dan Sunny merupakan kondisi cuaca cerah pada hari tersebut. Untuk dapat mengetahui *weather* pada hari tersebut, Player dapat melakukan *action Watching* dalam rumah atau furnitur TV jika mengimplementasikan furnitur.

Efek hujan:

1. Dalam satu season, Rainy Day minimal terjadi 2 kali.
2. Saat Rainy, seluruh *tile* yang tidak ditempati (*tillable land*, *tilled land*, dan *planted land*) akan menjadi “basah”, sehingga *Player* tidak perlu melakukan *action Watering* pada *tile* manapun pada hari tersebut.
3. Beberapa *fish* juga hanya dapat ditangkap pada *weather* tertentu.

NB: Buatlah *cheat* untuk *manually set season* dan *weather* untuk memudahkan proses *development*.

## End Game

Tidak ada *end game* dalam permainan Spakbor Hills sehingga permainan akan terus berjalan hingga kapan pun (*infinite gameplay*). Namun, terdapat *milestone* berupa:

1. *Player* sudah memiliki gold sebesar 17.209g
2. *Player* sudah menikah

Apabila salah satu dari *milestone* tersebut sudah tercapai, maka permainan akan menampilkan semacam *end game statistics*. Seluruh statistik yang dimunculkan harus divalidasi terlebih dahulu. *End game statistics* minimal menampilkan:

- *Total income*, yaitu total *gold* yang didapatkan oleh *Player*.
- *Total expenditure*, yaitu total *gold* yang dikeluarkan oleh *Player*.
- *Average season income*, yaitu rata-rata *gold* yang didapatkan oleh *Player* selama satu musim.
- *Average season expenditure*, yaitu rata-rata *gold* yang dikeluarkan oleh *Player* selama satu musim.
- *Total days played*, yaitu jumlah hari yang telah dilalui oleh *Player* dalam permainan.
- *NPCs status*, yang terdiri dari:
  - *Relationship status*, status hubungan *Player* dengan semua NPC dalam permainan.
  - *Chatting Frequency*, frekuensi *chatting* dengan semua NPC
  - *Gifting Frequency*, frekuensi *gifting* dengan semua NPC
  - *Visiting Frequency*, frekuensi *visiting* NPC
- *Crops harvested*, yaitu jumlah tanaman yang telah dipanen oleh *Player*.
- *Fish caught*, yaitu jumlah ikan yang telah ditangkap oleh *Player*, termasuk rincian jumlah tiap jenis ikan (*common*, *regular*, *legendary*).

## Menu Game

Pada awal membuka permainan, menu utama akan ditampilkan. Menu ini berisi pilihan aksi yang dapat dilakukan oleh pemain. Menu setidaknya memiliki pilihan aksi berikut.

Pilihan	Keterangan
New Game	Memulai permainan baru. Mekanisme memulai game. Mengikuti bagian <a href="#">Flow Permainan</a> .
Load Game (Bonus)	<a href="#">Save and Load</a>
Help	Memberikan deskripsi dari permainan dan juga arahan mengenai cara bermain.
View Player Info	Menu ini hanya bisa digunakan ketika dalam permainan. Menu ini digunakan untuk melihat atribut pengguna, yaitu: <ol style="list-style-type: none"><li>1. Nama</li><li>2. Gender</li><li>3. Energy</li><li>4. Partner</li><li>5. Favorite Item</li><li>6. Gold</li></ol>
List Object (Bonus)	Melihat objek apa saja di dalam rumah ( <a href="#">bonus furnitur</a> )
Statistics	Melihat statistik pemain (seperti di <a href="#">End Game</a> ).
Actions	Melakukan aksi pada objek
Credits	Menampilkan informasi pembuat permainan.
Exit	Keluar dari permainan.

## Flow Permainan

1. Inisiasi Environment beserta atribut yang diperlukan. **(sesuaikan dengan kebutuhan)**
  - a. Musim (Spring, Summer, Fall, Winter)
  - b. [Weather](#) (Sunny, Rainy)
  - c. [Time](#) / GameCalendar
2. Inisiasi World Map berdasarkan ketentuan pada [World Map](#).
3. Inisiasi Farm beserta atribut yang diperlukan. **(sesuaikan dengan kebutuhan):**
  - a. Land (tilled, planted, default)
  - b. Pond
  - c. Shipping Bin
  - d. House **(sesuaikan atribut dengan kebutuhan)**
4. Inisiasi inventory untuk menyimpan seluruh *item* yang akan dimiliki Player.
5. Inisiasi Player sesuai dengan bagian [Player](#)
6. Setelah seluruh objek utama telah diinisiasi, game sudah dapat dimainkan. *Player* bebas melakukan apapun untuk mencapai *milestone* yang telah ditentukan pada [End Game](#).
7. Untuk melakukan aksi, pemain dapat memasukkan *input* sesuai dengan aksi yang akan dilakukan pada *command line*. Untuk menggunakan sebuah objek, pemain perlu memindahkan Player ke objek tersebut terlebih dahulu dan melakukan aksi jika sudah berada di dekat objek tersebut.
8. Selamat bermain! 🎮🎉

## Bonus

### Furnitures (+1 per furnitur)

Buatlah *map* di dalam House seperti contoh pada Farm Map dengan ukuran 24 x 24. Kalian dapat membuat furnitur agar rumah Asep Spakbor makin cantik. Furnitur yang dapat dibuat adalah sebagai berikut:

Item ID	Nama	Deskripsi	Batasan dan Efek
bed_1	Single Bed	Kasur ukuran <i>single</i> yang mampu ditempati maks 1 orang	<ul style="list-style-type: none"><li>• Maks orang: 1</li><li>• Ukuran: 2 x 4</li></ul>
bed_2	Queen Bed	Kasur ukuran <i>queen</i> yang mampu ditempati maks 2 orang	<ul style="list-style-type: none"><li>• Maks orang: 2</li><li>• Ukuran: 4 x 6</li></ul>
bed_3	King Bed	Kasur ukuran <i>king</i> yang mampu ditempati maks 2 orang	<ul style="list-style-type: none"><li>• Maks orang: 2</li><li>• Ukuran: 6 x 6</li></ul>
stove	Stove	Tungku untuk memasak makanan	<ul style="list-style-type: none"><li>• Maks makanan yang bisa dimasak hanya satu.</li><li>• Perlu <i>fuel</i> untuk menyalakannya.</li><li>• Ukuran: 1 x 1</li></ul>
tv	Television (TV)	Televisi untuk melihat cuaca pada hari tersebut	<ul style="list-style-type: none"><li>• Ukuran: 1 x 1</li></ul>

Selain item tersebut, kalian bebas membuat furnitur lain, yang terpenting furnitur tersebut jelas (seperti perabotan untuk dekorasi, perabotan yang memiliki fungsi tertentu seperti menyimpan item, dan lain-lain)

## GUI (+3 ~ +10)

Kalian disarankan untuk mengimplementasi GUI pada tugas besar ini. GUI membuat game kalian memiliki UI dan visual yang membuat game menjadi lebih hidup dan mudah berinteraksinya. Dalam melakukan implementasi GUI, kalian dibebaskan untuk menggunakan *package* apapun, contoh *package* GUI yang dapat digunakan yaitu Java Swing (sudah bawaan dari Java sehingga kalian tidak perlu meng-*install* sendiri), atau JavaFX (*package third-party* sehingga kalian perlu meng-*install*-nya). Contoh implementasi program menggunakan GUI dapat dilihat referensi kodenya pada tautan [ini](#). Perlu diingat bahwa kalian diberi kebebasan dalam implementasi dan tidak harus mengikuti alternatif pada referensi. Untuk mempermudah proses pengembangan dan kolaborasi, **sangat disarankan** menggunakan MVC (Model, View, Controller).

## Keyboard and Mouse Input (+1 ~ +3)

Dengan mengimplementasikan bonus ini, kalian bisa menggunakan *keyboard movement* (WASD atau arrows) untuk bergerak dari suatu titik ke titik lainnya dan melakukan aksi. Aksi dapat dilakukan dengan *keyboard input* atau *mouse click* pada *object* yang akan digunakan. Implementasi ini dianjurkan jika akan menerapkan GUI.

## Free Market (+3)

Pasti kalian pernah mendengar konsep *supply and demand*. Ketika sebuah barang banyak dijual di pasar maka harganya akan menurun karena banyak alternatif yang dapat dipilih dan juga sebaliknya. Sayangnya, desa Spakbor Hills tidak bisa kabur dari prinsip ekonomi ini. Setiap *crop* yang dijual oleh Anda, warga desa, bahkan desa lain dapat menurunkan nilai ekonomi sebuah *crop*. Dengan jumlah warga desa yang tidak bertambah dalam kurun waktu 5 tahun terakhir, *supply crop* akan terus bertambah tanpa adanya *demand*.

Tugas anda adalah untuk mensimulasikan proses *supply and demand* ini! Setiap *crop* yang akan dijual tidak akan memiliki nilai yang konstan, tetapi dinamis. Proses perhitungan nilai jual *crop* dapat menggunakan formula berikut.

$$\frac{4}{4 - |season_{now} - season_{grow}|} \times \min\left(\frac{20}{jumlah\ crop\ yang\ sudah\ terjual}, 1\right) \times N \times D$$

dengan  $N = \text{Harga awal crop}$ ;  $D = \text{Multiplier demand crop}$

Sesuai rumus, setiap *crop* yang sudah dijual akan mengurangi nilai jual dari *crop* tersebut. Namun, perhitungan harga hanya akan dilakukan setelah sudah *crop* tersebut terjual. Artinya, jika seorang petani langsung menjual 200 pada waktu yang bersamaan, seluruh harga *crop* akan mengikuti harga awal dengan multiplier supply bernilai 1. Namun, jika ia akan menjual *crop* yang sama, maka ia akan terkena *multiplier* 20/200 yaitu 0,1.

Setiap 4 hari, akan terbit koran edisi baru yang dapat di beli pada *store*. Isi koran akan berisi 2 *headline* yang mendeskripsikan *demand* sebuah *crop* pada musim sekarang. *Demand* akan berupa *multiplier* yang dapat bernilai dari 0,5 kali lipat sampai 3 kali lipat.

Pada setiap pergantian *season*, pasar akan mengalami *market correction* yang akan mengembalikan seluruh nilai *crop* yang terjual menjadi 0.

## Let's Go Gambling! (+2)

Salah satu warga desa (yang tidak perlu disebut namanya) ingin mencari cara untuk mendapatkan uang secara cepat. Ketika sedang mengunjungi desa Kamboja, warga tersebut menemukan solusinya, *casino*! Setelah pulang kembali ke Spakbor Hills, warga tersebut ~~meminjam uang~~ meminta investasi ke Mayor Tadi untuk mendirikan **Dasco's Gambling Den!**

Casino akan memiliki 4 buah alat perjudian berbeda, yaitu sebagai berikut:

1. Slot

Sesuai dengan namanya, permainan ini akan memutar 3 buah *reel* yang masing-masing akan memberikan simbol khusus. Terdapat simbol buah, lonceng, hati dan angka tujuh. Pemain akan menaruh (*bet*) sejumlah

uangnya untuk bermain dan akan mendapatkan hadiah sejumlah *multiplier* dari *bet* ketika mendapatkan kombinasi berikut.

- Mendapatkan 2 buah : 2x *initial bet*
- Mendapatkan 3 buah : 3x *initial bet*
- Mendapatkan 3 lonceng atau 3 hati : 5x *initial bet*
- Mendapatkan 3 angka tujuh : 10x *initial bet*

Karena Dasco merasa kasihan bila warga desa Spakbor Hills jatuh miskin akibat permainannya, seluruh simbol memiliki kemungkinan muncul yang sama.

## 2. Blackjack

Blackjack adalah permainan kartu dimana pemain akan menebak apakah kartu yang dimilikinya lebih baik dibandingkan kartu *dealer*. Sebelum bermain, menaruh *bet* yang akan dimainkan. Pada awal permainan, pemain dan lawan akan mendapatkan dua buah kartu. Seluruh kartu pemain akan ditunjukkan sedangkan kartu *dealer* hanya 1 yang dapat terlihat.

Pada setiap giliran, pemain dapat memilih untuk *hit* (menambah kartu) atau *stand* (tahan di kartu tersebut). Jika memilih *stand*, maka *dealer* akan memulai bermain. *Dealer* akan membuka kartunya dan wajib melakukan *hit* sampai setidaknya kartu bernilai 17. Jika kartu *dealer* melebihi 21 atau kurang atau sama dengan kartu pemain maka pemain akan menggandakan uangnya. Namun, jika saat melakukan *hit* pemain melebihi 21, maka pemain akan kehilangan seluruh uangnya.

Jika masih bingung, kalian dapat membaca tutorial [Blackjack berikut](#). Perlu diketahui, bagian *special gameplay techniques* tidak harus diimplementasi.

## 3. Ride the Bus

Ride the Bus adalah permainan tebak-tebakan dengan kartu. Sebelum memulai, pemain akan menaruh *bet* yang akan dimainkan. Permainan ini terdiri dari 4 tahap, yaitu:

- Red or black?

Pada *stage* ini pemain akan menebak apakah kartu yang keluar akan berwarna merah atau hitam. Jika berhasil menebak, maka *bet* akan dikalikan 2.



- Higher or lower?

Pada *stage* ini pemain akan menebak apakah kartu yang keluar akan bernilai lebih tinggi atau kecil dibanding kartu pada *stage* sebelumnya. Jika berhasil menebak, maka *bet* akan dikalikan 3.

- Inside or outside?

Pada *stage* ini pemain akan menebak apakah kartu yang keluar akan berada di antara dua kartu pada *stage* sebelumnya atau di luar. Jika berhasil menebak, maka *bet* akan dikalikan 5.

- Guess the suit

Pada *stage* ini pemain akan menebak simbol yang ada pada kartu yang akan keluar. Jika berhasil menebak, maka *bet* akan dikalikan 10.

Pemain dapat memilih keluar pada *stage* manapun, maka total hadiah yang didapatkan adalah *multiplier* pada *stage* terakhir yang berhasil ditebak pemain. Namun, jika pemain kalah pada *stage* manapun maka seluruh uang akan hilang.

Jika masih bingung, kalian dapat membaca tutorial [Ride the Bus berikut](#) khusus bagian *dealing cards*.

## Save dan Load (+3)

Fitur *save and load* adalah fitur yang memungkinkan pengguna untuk menyimpan data atau *progress* yang telah dibuat pada dunia **Spakbor Hills** dan memuatnya kembali di waktu yang akan datang. Tipe penyimpanan dibebaskan, disarankan berupa suatu *file* JSON.

## NPC Easter Egg (+1 per NPC)

*Apakah kalian merasa NPC dalam permainannya kurang bervariasi?* Kalau begitu di bonus ini kalian bisa **menambahkan NPC kalian sendiri!** Silahkan berkreasi dan membuat NPC yang lucu! :D semakin lucu dan kreatif maka poin kalian semakin tinggi.

## Kelompok


Kelompok tugas besar akan diacak dengan masing-masing kelompok beranggota 4-5 mahasiswa. Pembagian kelompok akan dilakukan pada **Rabu, 16 April 2025** pukul 20.10 secara daring dengan menggunakan Google Meet pada [link https://meet.google.com/snc-bjyo-bbn](https://meet.google.com/snc-bjyo-bbn).

Pembagian kelompok dapat dilihat pada *link* berikut.

 [Pembagian Kelompok Tubes](#)

## QnA

Jika ada spesifikasi yang belum jelas, dapat ditanyakan pada Form QnA

 [QnA Tugas Besar IF2010 Pemrograman Berbasis Objek 2024/2025](#)

## Asistensi

Asistensi dilakukan dengan ketentuan sebagai berikut.

1. Setiap kelompok akan mendapatkan satu orang Asisten Pembimbing.
2. Asistensi wajib dilakukan sebanyak minimal dua kali, satu kali sebelum *milestone* 1 dan satu kali sebelum *milestone* 2. Berikut adalah ketentuan asistensi wajib.
  - a. Asistensi 1 dilakukan paling lambat tanggal **27 April 2025** pukul 00.01.
  - b. Asistensi 2 dilakukan paling lambat tanggal **29 Mei 2025 23.59**.
3. Waktu dan tempat asistensi ditentukan berdasarkan perjanjian asisten dan kelompok.
4. Perjanjian waktu asistensi harus dilakukan paling lambat H-3 asistensi.
5. Asistensi **bukan sesi untuk membahas ulang spesifikasi**, tetapi untuk bertanya bagian yang tidak dimengerti. Asisten **tidak akan membahas kembali spesifikasi saat asistensi**.
6. Paling lambat 1 jam setelah asistensi selesai, masing-masing anggota kelompok yang hadir pada asistensi wajib mengisi form asistensi secara daring melalui Google Form.
  - a. Form catatan asistensi: <https://forms.gle/wMNjZFp2yZuG7rch7>  
Sertakan juga bukti kehadiran asistensi berupa *screenshot* atau foto dengan asisten.

- b. Tidak diperkenankan untuk menyalin catatan kemajuan milik orang lain.

## Pengumpulan

### Milestone 1 - 27/04/2025 20:10

Pada *Milestone 1*, anda perlu mengumpulkan *link repository* GitHub dan 2 dokumen, yaitu pembagian tugas setiap anggota kelompok dan struktur dari objek-objek yang terdapat pada game (disarankan menggunakan *class diagram*). Harap mengundang asisten yang bersangkutan pada *repository* Github. Akun Github asisten akan dituliskan pada sheets pembagian kelompok. Tidak ada aturan penamaan khusus *repository* Github. Format dari kedua dokumen adalah PDF dan wajib dikumpulkan pada *link TBA*.

### Milestone 2 - 30/04/2025 - 01/06/2025 01:02

Pada *Milestone 2*, anda perlu mengumpulkan *Source Code*, Buklet, dan *Log Activity*. Pengumpulan akan dilakukan dengan membuat **release** pada Repository Github kelompok masing-masing. Release harus sudah mencakup *Source Code*, Buklet dan *Log Activity*. Waktu Release tidak boleh melewati *deadline* pengumpulan. Berikut adalah spesifikasi *source code* dan buklet.

Dengan pengunduran *deadline* tugas besar ini, terdapat aturan yang harus dipatuhi oleh seluruh peserta kuliah IF2010 yaitu **TIDAK ADA rescheduling demo dan debugging** ketika demo sedang berlangsung. Program yang disubmit pada waktu *deadline* adalah program yang digunakan untuk penilaian akhir.

## Source Code

Teknik yang digunakan dalam pembuatan aplikasi permainan **Spakbor Hills** dibebaskan, namun terdapat beberapa konsep OOP yang wajib diimplementasikan di aplikasi yang Anda buat, yaitu:

1. *Inheritance*
2. *Abstract Class / Interface*
3. *Polymorphism*
4. *Generics*
5. *Exceptions*
6. *Concurrency*

Program harus diimplementasikan menggunakan bahasa **Java**. Tuliskan tahapan untuk melakukan kompilasi dan menjalankan program pada file README.md. Kumpulkan berkas ZIP *source code* beserta README.md.

## Buklet

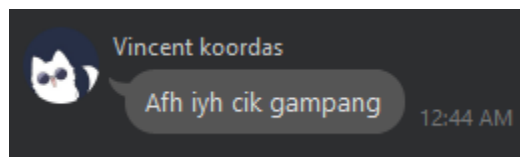
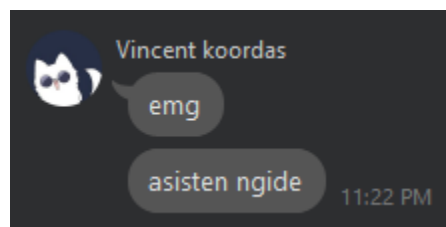
Pada Tugas Besar kali ini, format laporan berupa buklet. Buklet yang dibuat memiliki ketentuan sebagai berikut.

1. Berukuran A5
2. Tidak lebih dari 15 halaman
3. Format PDF
4. Berisi setidaknya:
  - a. Halaman *cover*, yang berisi nama dan nomor kelompok, serta nama dan NIM anggota kelompok.
  - b. *User manual* dan deskripsi jalannya permainan (*gameplay*).
  - c. Nama, peran, dan pembagian tugas anggota kelompok.
  - d. Struktur final dari objek-objek yang terdapat pada game (disarankan menggunakan *class diagram*)
  - e. Cerita tentang proses pengembangan
    - i. Pada bagian ini kalian menceritakan proses pengembangan Tugas Besar ini dari awal sampai akhir. Kalian boleh mencantumkan *screenshot* rapat, hasil ideasi, corat-coret Miro/Jamboard, atau dokumentasi apapun yang mendukung jalan cerita kalian.
    - ii. Cantumkan juga rancangan awal dan apa saja perubahan yang dilakukan (apabila implementasi beda dari rancangan) dan kenapa perlu dilakukan perubahan tersebut.
5. Dibuat kreatif dan semenarik mungkin. Desain, tipografi, pewarnaan, dan tata letak dibebaskan kepada kalian.
6. Bahasa yang digunakan tidak harus baku, tetapi tetap serius dan terkesan *friendly*.

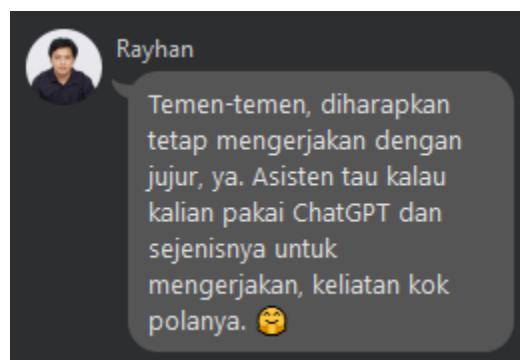
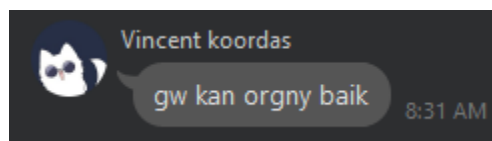
## Demo

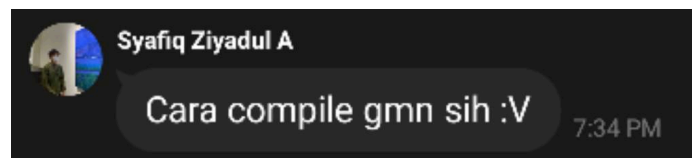
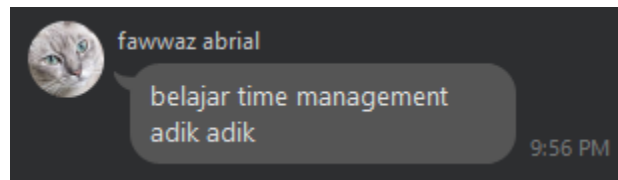
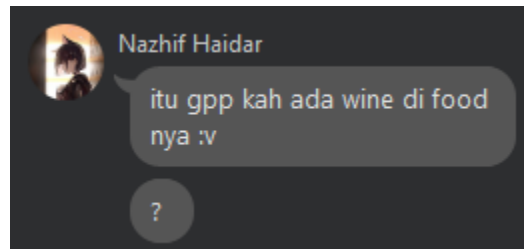
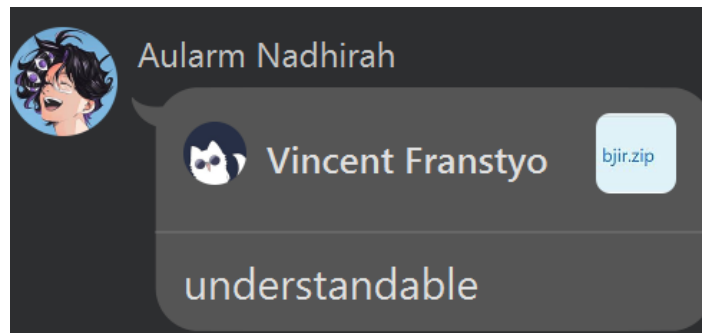
1. Demo akan dilakukan baik secara luring ataupun daring dengan menggunakan Google Meet, sesuai kesepakatan dengan asisten penguji.
2. Mahasiswa menghubungi asisten demo untuk melakukan perjanjian jadwal demo.
3. Mahasiswa mempersiapkan program dan menjalankan eksekusi program sesuai dengan instruksi dari asisten penguji.
4. Penilaian demo dilakukan dengan membandingkan hasil eksekusi program dengan hasil yang diharapkan.
5. Demo dapat dilakukan pada waktu yang akan ditentukan kemudian dengan pembagian asisten yang berbeda. Pembagian asisten dapat dilihat pada *Sheets* Pembagian Kelompok.

## Extras



^ minimal berkonteks  
Lucunya di situ (no konteks)





Semangat gang ngerjain tubesnya!  
- Aularm