

Kelas : 02

Nomor Kelompok : 11

Nama Kelompok : KejuCraft

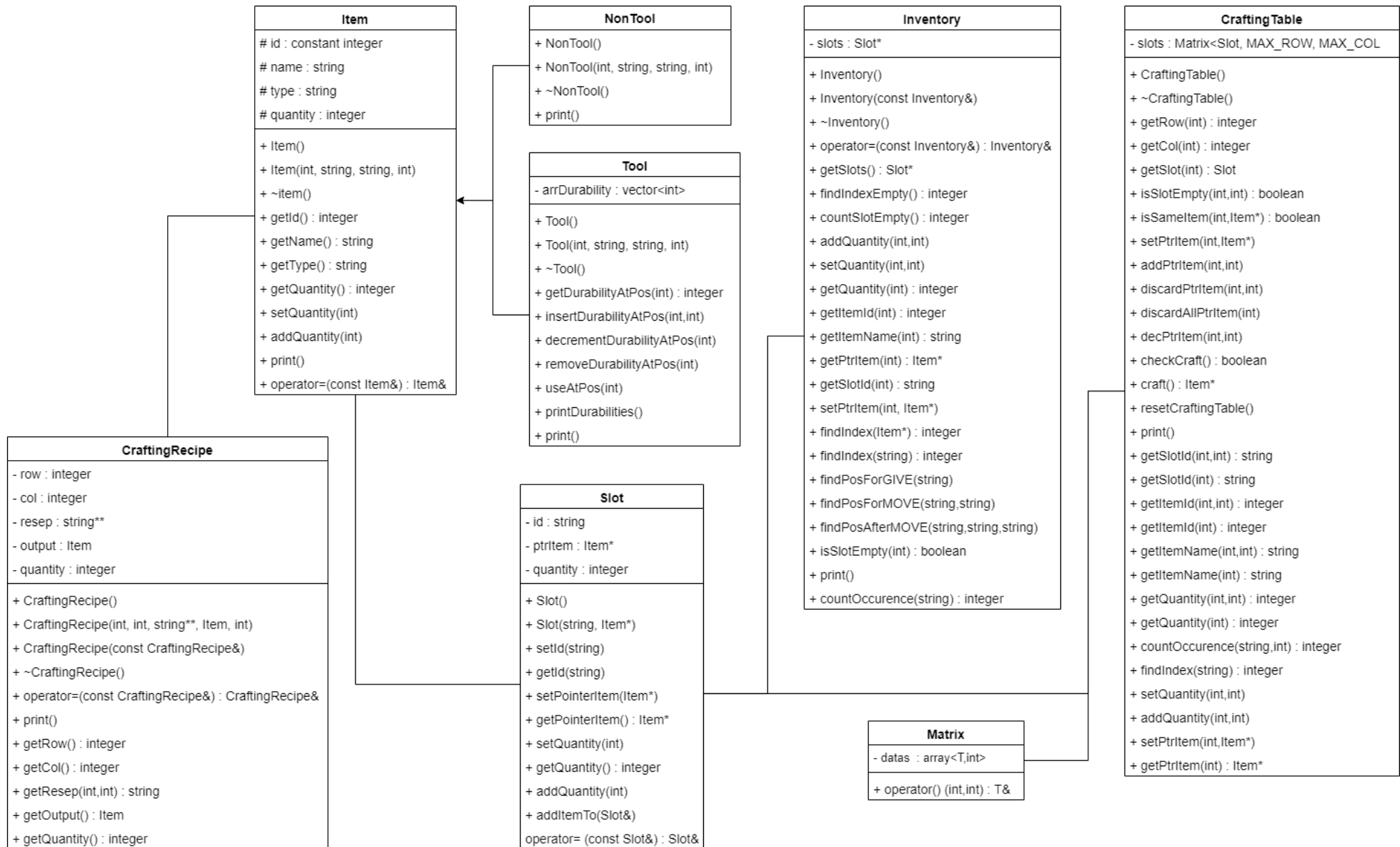
1. 13520017 / Diky Restu Maulana
2. 13520020 / William Manuel Kurniawan
3. 13520035 / Damianus Clairvoyance Diva Putra
4. 13520038 / Shadiq Harwiz
5. 13520158 / Azmi Alfatih Shalahuddin

Asisten Pembimbing : Matthew Kevin Amadeus

## 1. Diagram Kelas

Diagram kelas dibuat secara manual menggunakan aplikasi draw.io. Diagram kelas berisi kelas-kelas utama (belum termasuk kelas untuk error handling) yang digunakan dalam file main.cpp. Setiap kelas berisi semua atribut dan method dalam kelas tersebut. Berikut adalah nama-nama kelas yang digunakan dalam main.cpp :

1. Item
2. Tool
3. NonTool
4. Slot
5. Inventory
6. CraftingTable
7. Matrix
8. CraftingRecipe



Class Item memiliki subclass Tool dan subclass Nontool untuk membedakan jenis item yang dimiliki. Dari class Item, dibuat class Slot sebagai representasi slot yang ada di inventory dan crafting table. Class Item dan class slot menyimpan data quantity dimana quantity pada class Item menyimpan data jumlah barang dari seluruh slot inventory dan crafting table sedangkan quantity pada class Slot menyimpan data jumlah barang pada slot tersebut. Perlu diketahui bahwa semua item yang dibaca dari file config hanya diinstansiasi sekali dan disimpan ke dalam sebuah container yang terpisah antara Tool dan NonTool. Semua item akan disimpan, meskipun quantity nya sama dengan nol. Quantity item di dalam container ini akan terus bertambah jika item tersebut dimasukkan ke dalam inventory. Dengan kata lain, quantity suatu item di container menunjukkan jumlah seluruh item tersebut yang hidup, sedangkan quantity di slot adalah jumlah item tersebut di dalam slot. Keuntungan dari penggunaan quantity tersebut adalah lebih mudah untuk memeriksa apakah barang tersebut ada dalam inventory atau tidak dan lebih mudah mengetahui jumlah total barang yang ada tetapi kekurangannya adalah dapat terjadi kebingungan antara quantity yang dipakai serta program menjadi lebih rumit karena setiap kali jumlah item berubah dalam inventory quantity di slot dan item harus diubah. Class Inventory dan berisi pointer menuju Slot dan berfungsi sebagai array of Slot. Array of Slot pada inventory berjumlah 27 dan merepresentasikan inventory pemain yang dihitung mulai dari kiri atas menuju kanan bawah. Class CraftingTable menggunakan template dari class Matrix dan berisi array of Slot seperti pada class Inventory tetapi ditulis seperti matriks dengan bantuan class Matrix. Array of Slot pada class CraftingTable berjumlah 9 dan merepresentasikan crafting table yang digunakan. Class CraftingRecipe berfungsi untuk menyimpan data dari folder recipe sehingga menyimpan jumlah baris resep, jumlah kolom resep, matrix string, Item hasil dari crafting, dan jumlah item yang didapatkan.

## 2. Penerapan Konsep OOP

### 2.1. Inheritance & Polymorphism

```
class Item {
protected:
    const int id;        /* ID of item from item.txt */
    string name;         /* name of item from item.txt */
    string type;         /* type of item from item.txt */
    int quantity;        /* total items owned by player */
public:
    Item();              /* default constructor */
    Item(int id, string name, string type, int quantity); /* user-defined constructor */
    virtual ~Item();     /* destructor */
    int getId();          /* get id attribute of item */
    string getName();     /* get name attribute of item */
    string getType();     /* get type attribute of item */
    int getQuantity();    /* get quantity attribute of item */
    void setQuantity(int quantity);
    void addQuantity(int quantity); /* add quantity of item owned by player */
    virtual void print();
    Item& operator=(const Item& other); /* print info */
};
```

```

class NonTool : public Item {
private:
public:
    NonTool();                /* default constructor */
    NonTool(int id, string name, string type, int quantity); /* user-defined constructor */
    virtual ~NonTool();       /* destructor */
    void print();             /* print info */
};

```

```

class Tool : public Item {
private:
    vector<int> arrDurability; /* durability of item */
public:
    Tool();                /* default constructor */
    Tool(int id, string name, string type, int quantity); /* user-defined constructor */
    virtual ~Tool();       /* destructor */
    int getDurabilityAtPos(int pos); /* get durability attribute of tool */
    void insertDurabilityAtPos(int pos, int durability);
    void decrementDurabilityAtPos(int pos);
    void removeDurabilityAtPos(int pos);
    void useAtPos(int pos);
    void printDurabilities(); /* use nontool (decrease durability) */
    void print();             /* print info */
};

```

Inheritance digunakan untuk class Item agar dapat menyimpan bagian Item yang dimiliki alat dan Item yang bukan merupakan alat. Pada dasarnya semua barang memiliki id, nama, dan tipe tetapi alat memiliki durability dan hanya bisa ada 1 alat pada setiap slot sedangkan item biasa dapat ditumpuk sebanyak 64. Untuk menangani kasus tersebut, dibuat class Item untuk menyimpan id, nama, dan tipe lalu dibuat subclass Tool untuk menyimpan durability alat dan subclass Nontool. Keuntungan penggunaan inheritance ini adalah tidak perlu dibuat banyak variabel untuk item alat dan bukan alat.

```
class Slot {
protected:
    string id;      /* I0, I1, ... */
    Item* ptrItem; /* pointer to Item */
    int quantity; /* quantity of item in slot */
public:
    /* main */
    Slot();
    Slot(string id, Item* item, int quantity);
    ~Slot();
    Slot& operator=(const Slot& other);
    void setId(string id);
    string getId();
    void setPointerItem(Item* ptr);
    Item* getPointerItem();
    void setQuantity(int quantity);
    int getQuantity();
    void addQuantity(int quantity);
    void addItemTo(Slot& other);
};
```

Polymorphism diterapkan dalam kelas Slot. Kelas ini memiliki atribut berupa pointer Item. Atribut ini dapat diisi oleh kelas turunan dari Item, yaitu Tool dan NonTool. Keuntungan dari penerapan ini adalah kita tidak perlu membuat kelas berbeda untuk menerima objek dari kelas Tool dan NonTool. Kita juga bisa menganggap kedua jenis Item itu adalah benda yang sama, yaitu Item sebagai kelas dasarnya. Dampaknya, kita sebagai pemrogram juga tidak terlalu pusing untuk memikirkan begitu banyak kelas.

## 2.2. Method/Operator Overloading

```
template<class T, int row, int col>
class Matrix
{
    array<T, row * col> datas;
public:
    T& operator()(int y, int x){
        return datas[y * col + x];
    }

    // more methods go here
};

#endif
```

Crafting table sejatinya merupakan array tetapi ingin digunakan seperti matrix. Oleh karena itu, dibuatlah sebuah operator overloading berupa tanda kurung supaya array tersebut seolah-olah menjadi sebuah matriks yang bisa diakses melalui baris dan kolomnya. Keuntungan menggunakan operator ini adalah penggunaan memori yang lebih sedikit karena sebenarnya kelas Matrix adalah sebuah array.

```

class CraftingTable {
private:
    Matrix<Slot, MAX_ROW, MAX_COL> slots; /* matrix
    // Slot slots[MAX_ROW][MAX_COL];
public:
    CraftingTable();
    ~CraftingTable();
    int getRow(int id);
    int getCol(int id);
    Slot getSlot(int index);
    bool isSlotEmpty(int index);
    bool isSameItem(int index, Item* ptr);
    Item* getPtrItem(int index);
    void setPtrItem(int index, Item* ptr);
    void addPtrItem(int index, int quantity);
    void discardPtrItem(int index, int quantity);
    void discardAllPtrItem(int index);
    void decPtrItem(int index, int quantity);
    bool checkCraft();
    void resetCraftingTable();
    string getSlotId(int row, int col);
    int getItemId(int row, int col);
    int countOccurence(string itemName, int index);
    int findIndex(string slotIdDest);
    int getQuantity(int index);
    int getQuantity(int row, int col);
    void setQuantity(int index, int quantity);
    void addQuantity(int index, int quantity);
    string getItemName(int index);
    string getItemName(int row, int col);
    int getItemId(int index);
    string getSlotId(int index);
    void print();
};

```

Kelas CraftingTable menyimpan banyak sekali method overloading. Kelas ini menyimpan atribut berupa kelas Matrix yang diimplementasikan sebagai STL array, tetapi memiliki operator overloading berupa tanda kurung untuk pengaksesan baris dan kolom. Akibat dari bentuk aktualnya



yang merupakan sebuah array, kelas Matrix ini pun bisa diakses menggunakan indeks. Oleh karena itu, dibuatlah beberapa fungsi bernama sama, tetapi parameter masukannya berbeda. Ada yang menerima masukan berupa indeks, ada juga yang menerima baris dan kolom. Penggunaan keduanya disesuaikan dengan kebutuhan. Keuntungan penggunaan operator overloading ini adalah memudahkan pengaksesan kelas CraftingTable. Misalnya, ketika ingin dilakukan perulangan bersarang untuk mengiterasi baris dan kolom, maka lebih mudah menggunakan operator yang menerima masukan baris dan kolom daripada indeks.

### 2.3. Template & Generic Classes

```
template<class T, int row, int col>
class Matrix
{
    array<T, row * col> datas;
public:
    T& operator()(int y, int x){
        return datas[y * col + x];
    }

    // more methods go here
};

#endif
```

Template digunakan pada class Matrix yang akan digunakan untuk class CraftingTable. Template digunakan karena class Matrix dapat menerima class lain yang akan diproses menjadi matrix. Penggunaan template & generic classes tersebut memiliki keuntungan dapat membuat beberapa matriks dari class lain dengan lebih mudah.

## 2.4. Exception

Kami tidak menerapkan exception di dalam program ini karena kami sudah menangani hampir semua kemungkinan error dalam percabangan if-else. Jika dimungkinkan akan terjadi error, program akan mengeluarkan pesan kesalahan dan menggagalkan command terakhir.

## 2.5. C++ Standard Template Library

```
class Tool : public Item {
private:
    vector<int> arrDurability;    /* durability of item */
public:
    Tool();
    Tool(int id, string name, string type, int quantity);
    virtual ~Tool();
    int getDurabilityAtPos(int pos);
    void insertDurabilityAtPos(int pos, int durability);
    void decrementDurabilityAtPos(int pos);
    void removeDurabilityAtPos(int pos);
    void useAtPos(int pos);
    void printDurabilities();
    void print();
};
```

STL Vector digunakan untuk menyimpan array durability yang merupakan atribut dari kelas Tool. Setiap objek yang termasuk ke dalam kelas Tool memiliki durability yang berbeda, sehingga perlu disimpan sebagai array dinamis berbentuk vector yang diurutkan sesuai urutan kemunculan objek tersebut di Inventory. Perlu diketahui bahwa kami hanya menginstansiasi setiap objek satu kali dan semuanya disimpan di dalam sebuah container yang berbentuk Map (dijelaskan lebih lanjut di bawah). Penggunaan vector memiliki banyak keuntungan, antara lain mudah untuk menambah dan menghapus elemen di dalamnya sehingga tidak perlu repot untuk menggeser elemen setelah penghapusan.

```
map<string, Tool> toolContainer;
map<string, NonTool> nonToolContainer;
```

```
// Memisahkan item tool dan nontool dan meletakkan dalam container masing-masing
while (getline(itemConfigFile, intext)) {
    stringstream ss(intext);
    string name, type, isTool;
    int id;
    ss >> id >> name >> type >> isTool;
    if (isTool == "TOOL") {
        Tool item(id, name, type, 0); // Durability awal = 10
        toolContainer.insert(pair<string, Tool> (item.getName(), item));
    } else {
        NonTool item(id, name, type, 0);
        nonToolContainer.insert(pair<string, NonTool> (item.getName(), item));
    }
}
```

STL map digunakan untuk menyimpan item dan tool yang diberikan dalam item.txt. Penggunaan STL map untuk menyimpan item dan tool memiliki fungsi sebagai array dinamis sehingga tidak perlu mendeklarasikan ukuran array untuk menyimpan data terlebih dahulu. STL map juga mengurutkan bagian string pada container sehingga item dan tool yang disimpan terurut berdasarkan abjad

### 3. Bonus Yang dikerjakan

### 3.1. Bonus yang diusulkan oleh spek

### 3.1.1. Multiple Crafting

[illegible]

MOVE I2 1 C7																										
SHOW																										
			[C0	0	0]	[C1	0	0]	[C2	0	0]															
			[C3	0	0]	[C4	0	0]	[C5	0	0]															
			[C6	0	0]	[C7	10	1]	[C8	0	0]															
[I0	15	10]	[I1	17	10]	[I2	10	1]	[I3	0	0]	[I4	0	0]	[I5	0	0]	[I6	0	0]	[I7	0	0]	[I8	0	0]
[I9	0	0]	[I10	0	0]	[I11	0	0]	[I12	0	0]	[I13	0	0]	[I14	0	0]	[I15	0	0]	[I16	0	0]	[I17	0	0]
[I18	0	0]	[I19	0	0]	[I20	0	0]	[I21	0	0]	[I22	0	0]	[I23	0	0]	[I24	0	0]	[I25	0	0]	[I26	0	0]
MOVE I2 1 C7																										
SHOW																										
			[C0	0	0]	[C1	0	0]	[C2	0	0]															
			[C3	0	0]	[C4	0	0]	[C5	0	0]															
			[C6	0	0]	[C7	10	2]	[C8	0	0]															
[I0	15	10]	[I1	17	10]	[I2	0	0]	[I3	0	0]	[I4	0	0]	[I5	0	0]	[I6	0	0]	[I7	0	0]	[I8	0	0]
[I9	0	0]	[I10	0	0]	[I11	0	0]	[I12	0	0]	[I13	0	0]	[I14	0	0]	[I15	0	0]	[I16	0	0]	[I17	0	0]
[I18	0	0]	[I19	0	0]	[I20	0	0]	[I21	0	0]	[I22	0	0]	[I23	0	0]	[I24	0	0]	[I25	0	0]	[I26	0	0]

Crafting table akan menerima barang menggunakan command “move” secara satu per satu (angka 1 di tengah tidak boleh diubah) dan dapat menerima lebih dari 1 item seperti slot inventory.

```

[C0 10 1] [C1 10 1] [C2 10 1]
[C3 0 0] [C4 7 2] [C5 0 0]
[C6 0 0] [C7 7 2] [C8 0 0]
[I0 0 0] [I1 0 0] [I2 0 0] [I3 0 0] [I4 0 0] [I5 0 0] [I6 0 0] [I7 0 0] [I8 0 0]
[I9 0 0] [I10 0 0] [I11 0 0] [I12 0 0] [I13 0 0] [I14 0 0] [I15 0 0] [I16 0 0] [I17 0 0]
[I18 0 0] [I19 0 0] [I20 0 0] [I21 0 0] [I22 0 0] [I23 0 0] [I24 0 0] [I25 0 0] [I26 0 0]
CRAFT
item crafted
SHOW
[C0 0 0] [C1 0 0] [C2 0 0]
[C3 0 0] [C4 7 1] [C5 0 0]
[C6 0 0] [C7 7 1] [C8 0 0]
[I0 15 10] [I1 0 0] [I2 0 0] [I3 0 0] [I4 0 0] [I5 0 0] [I6 0 0] [I7 0 0] [I8 0 0]
[I9 0 0] [I10 0 0] [I11 0 0] [I12 0 0] [I13 0 0] [I14 0 0] [I15 0 0] [I16 0 0] [I17 0 0]
[I18 0 0] [I19 0 0] [I20 0 0] [I21 0 0] [I22 0 0] [I23 0 0] [I24 0 0] [I25 0 0] [I26 0 0]

```

Ketika memberi command “CRAFT”, program akan mengurangi jumlah item yang ada di slot sebesar 1 dan mengeluarkan item. Sisa item yang ada di slot crafting table dapat digunakan untuk melakukan craft item lain.

### 3.1.2. Item dan Tool Baru

Kami berusaha untuk mengumpulkan beberapa resep baru dari referensi game asli Minecraft. Item yang kami punya berjumlah 60, terdiri atas 17 NonTool dan 43 Tool. Rinciannya adalah sebagai berikut

id	Nama	Tipe	Jenis
1	OAK_LOG	LOG	NONTOOL
2	SPRUCE_LOG	LOG	NONTOOL
3	BIRCH_LOG	LOG	NONTOOL

4	OAK_PLANK	PLANK	NONTOOL
5	SPRUCE_PLANK	PLANK	NONTOOL
6	BIRCH_PLANK	PLANK	NONTOOL
7	STICK	-	NONTOOL
8	COBBLESTONE	STONE	NONTOOL
9	BLACKSTONE	STONE	NONTOOL
10	IRON_INGOT	-	NONTOOL
11	IRON_NUGGET	-	NONTOOL
12	DIAMOND	-	NONTOOL
13	WOODEN_PICKAXE	-	TOOL
14	STONE_PICKAXE	-	TOOL
15	IRON_PICKAXE	-	TOOL
16	DIAMOND_PICKAXE	-	TOOL
17	WOODEN_AXE	-	TOOL
18	STONE_AXE	-	TOOL
19	IRON_AXE	-	TOOL
20	DIAMOND_AXE	-	TOOL
21	WOODEN_SWORD	-	TOOL

22	STONE_SWORD	-	TOOL
23	IRON_SWORD	-	TOOL
24	DIAMOND_SWORD	-	TOOL
25	GOLD_INGOT	-	NONTOOL
26	GOLD_NUGGET	-	NONTOOL
27	STRING	-	NONTOOL
28	RABBIT_HIDE	-	NONTOOL
29	LEATHER	-	NONTOOL
30	SHIELD	-	TOOL
31	BOW	-	TOOL
32	FISHING_ROD	-	TOOL
33	LEATHER_BOOTS	-	TOOL
34	IRON_BOOTS	-	TOOL
35	GOLDEN_BOOTS	-	TOOL
36	DIAMOND_BOOTS	-	TOOL
37	LEATHER_PANTS	-	TOOL
38	IRON_LEGGINGS	-	TOOL
39	GOLDEN_LEGGINGS	-	TOOL

40	DIAMOND_LEGGINGS	-	TOOL
41	LEATHER_CAP	-	TOOL
42	IRON_HELMET	-	TOOL
43	GOLDEN_HELMET	-	TOOL
44	DIAMOND_HELMET	-	TOOL
45	LEATHER_TUNIC	-	TOOL
46	IRON_CHESTPLATE	-	TOOL
47	GOLDEN_CHESTPLATE	-	TOOL
48	DIAMOND_CHESTPLATE	-	TOOL
49	WOODEN_SHOVEL	-	TOOL
50	IRON_SHOVEL	-	TOOL
51	GOLDEN_SHOVEL	-	TOOL
52	DIAMOND_SHOVEL	-	TOOL
53	WOODEN_HOE	-	TOOL
54	STONE_HOE	-	TOOL
55	IRON_HOE	-	TOOL
56	GOLDEN_HOE	-	TOOL
57	DIAMOND_HOE	-	TOOL



58	GOLDEN_PICKAXE	-	TOOL
59	GOLDEN_AXE	-	TOOL
60	GOLDEN_SWORD	-	TOOL

### 3.2. Bonus Kreasi Mandiri

Kami membuat sebuah banner yang ditampilkan saat permainan dimulai. Banner bertuliskan nama kelompok kami, yaitu KejuCraft dilengkapi dengan warna yang bernuansa MineCraft. Contoh tampilan banner dapat dilihat di bawah ini.



### 4. Pembagian Tugas

Modul (dalam poin spek)	Implementer	Tester
Kelas Item, Tool, dan NonTool	13520017	13520017, 13520020
Kelas Slot	13520017, 13520020, 13520035	13520017, 13520020
Kelas Inventory	13520017, 13520158	13520017, 13520020, 13520158

Kelas CraftingTable	13520020, 13520035	13520020, 13520035
Kelas CraftingRecipe	13520020	13520020
Command Show	13520017, 13520035	13520017, 13520035
Command Give	13520017, 13520038	13520017, 13520038
Command Discard	13520017	13520017, 13520035
Command Move	13520017	13520017, 13520020
Command Use	13520017, 13520158	13520017, 13520158
Command Export	13520017	13520017, 13520035
Program Utama	13520017, 13520020, 13520035	13520017, 13520020, 13520035