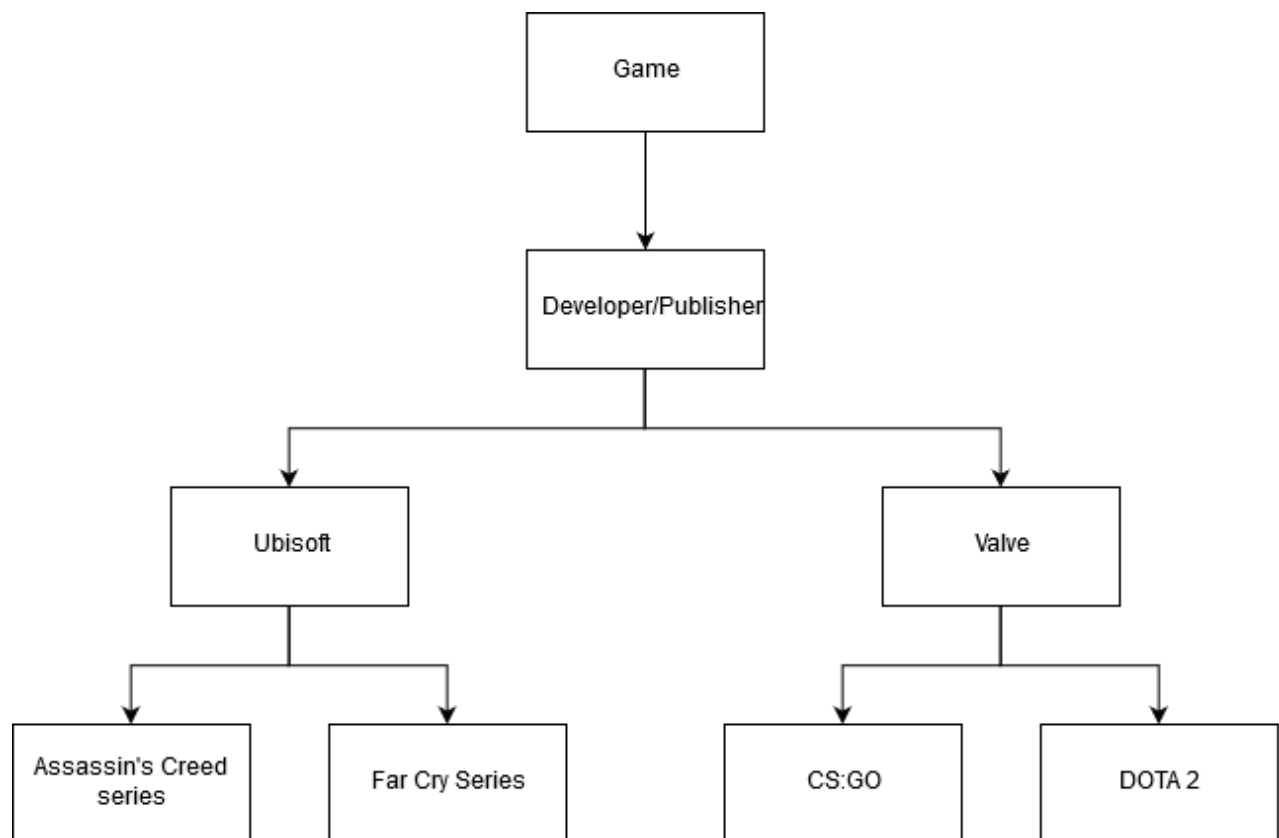


NAMA : TONY BASKORO
NIM : 17050623016
KELAS : MI 2017



➤ Model Tree Structures with Parent References

```
Command Prompt - mongo
> show databases
admin            0.000GB
config           0.000GB
kelas            0.000GB
local            0.000GB
perpustakaan     0.000GB
> use game_parent
switched to db game_parent
> db.game_parent.insert( { _id: "Assassin's Creed Series", parent: "Ubisoft" } )
WriteResult({ "nInserted" : 1 })
> db.game_parent.insert( { _id: "Far Cry Series", parent: "Ubisoft" } )
WriteResult({ "nInserted" : 1 })
> db.game_parent.insert( { _id: "CS:GO", parent: "Valve" } )
WriteResult({ "nInserted" : 1 })
> db.game_parent.insert( { _id: "DOTA 2", parent: "Valve" } )
WriteResult({ "nInserted" : 1 })
> db.game_parent.insert( { _id: "Ubisoft", parent: "Developer/Publisher" } )
WriteResult({ "nInserted" : 1 })
> db.game_parent.insert( { _id: "Valve", parent: "Developer/Publisher" } )
WriteResult({ "nInserted" : 1 })
> db.game_parent.insert( { _id: "Developer/Publisher", parent: "Game" } )
WriteResult({ "nInserted" : 1 })
> db.game_parent.insert( { _id: "Game", parent: "null" } )
WriteResult({ "nInserted" : 1 })
>
```

```
Command Prompt - mongo
> db.game_parent.find().pretty()
{ "_id" : "Assassin's Creed Series", "parent" : "Ubisoft" }
{ "_id" : "Far Cry Series", "parent" : "Ubisoft" }
{ "_id" : "CS:GO", "parent" : "Valve" }
{ "_id" : "DOTA 2", "parent" : "Valve" }
{ "_id" : "Ubisoft", "parent" : "Developer/Publisher" }
{ "_id" : "Valve", "parent" : "Developer/Publisher" }
{ "_id" : "Developer/Publisher", "parent" : "Game" }
{ "_id" : "Game", "parent" : "null" }
>
```

```
Command Prompt - mongo
> db.game_parent.findOne( { _id: "CS:GO" } ).parent
Valve
> db.game_parent.createIndex( { parent: 1 } )
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
> db.categories.find( { parent: "Ubisoft" } )
> db.game_parent.find( { parent: "Ubisoft" } )
{ "_id" : "Assassin's Creed Series", "parent" : "Ubisoft" }
{ "_id" : "Far Cry Series", "parent" : "Ubisoft" }
>
```

➤ Model Tree Structures with Child References

```
Command Prompt - mongo
> show databases
admin            0.000GB
config           0.000GB
game_parent      0.000GB
kelas            0.000GB
local            0.000GB
perpustakaan     0.000GB
> use game_child
switched to db game_child
> db.game_list.insert( { _id: "Assassin's Creed Series", children: "[]" } )
WriteResult({ "nInserted" : 1 })
> db.game_list.insert( { _id: "Far Cry Series", children: "[]" } )
WriteResult({ "nInserted" : 1 })
> db.game_list.insert( { _id: "Ubisoft", children: ["Assassin's Creed Series", "Far Cry Series"] } )
WriteResult({ "nInserted" : 1 })
> db.game_list.insert( { _id: "CS:GO", children: [] } )
WriteResult({ "nInserted" : 1 })
> db.game_list.insert( { _id: "DOTA 2", children: [] } )
WriteResult({ "nInserted" : 1 })
> db.game_list.insert( { _id: "Valve", children: ["CS:GO", "DOTA 2"] } )
WriteResult({ "nInserted" : 1 })
> db.game_list.insert( { _id: "Developer/Publisher", children: ["Ubisoft", "Valve"] } )
WriteResult({ "nInserted" : 1 })
> db.game_list.insert( { _id: "Game", children: ["Developer/Publisher"] } )
WriteResult({ "nInserted" : 1 })
>
```

```
Command Prompt - mongo
> db.game_list.findOne( { _id: "Ubisoft" } ).children
[ "Assassin's Creed Series", "Far Cry Series" ]
> db.game_child.createIndex( { children: 1 } )
{
  "createdCollectionAutomatically" : true,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
>
```

➤ Model Tree Structures with An Array Of Ancestor

```
Command Prompt - mongo
> show databases
admin                0.000GB
config               0.000GB
game_child           0.000GB
game_parent          0.000GB
kelas                0.000GB
local                0.000GB
perpustakaan         0.000GB
> use game_array
switched to db game_array
> db.game_list.insert( { _id: "Far Cry Series", ancestors: [ "Game", "Developer/Publisher", "Ubisoft" ], parent: "Ubisoft" } )
WriteResult({ "nInserted" : 1 })
> db.game_list.insert( { _id: "Assassin's Creed Series", ancestors: [ "Game", "Developer/Publisher", "Ubisoft" ], parent: "Ubisoft" } )
WriteResult({ "nInserted" : 1 })
> db.game_list.insert( { _id: "Ubisoft", ancestors: [ "Game", "Developer/Publisher" ], parent: "Developer/Publisher" } )
WriteResult({ "nInserted" : 1 })
> db.game_list.insert( { _id: "CS:GO", ancestors: [ "Game", "Developer/Publisher", "Valve" ], parent: "Valve" } )
WriteResult({ "nInserted" : 1 })
> db.game_list.insert( { _id: "DOTA 2", ancestors: [ "Game", "Developer/Publisher", "Valve" ], parent: "Valve" } )
WriteResult({ "nInserted" : 1 })
> db.game_list.insert( { _id: "Valve", ancestors: [ "Game", "Developer/Publisher" ], parent: "Developer/Publisher" } )
WriteResult({ "nInserted" : 1 })
> db.game_list.insert( { _id: "Developer/Publisher", ancestors: [ "Game" ], parent: "Game" } )
WriteResult({ "nInserted" : 1 })
> db.game_list.insert( { _id: "Game", ancestors: [], parent: null } )
WriteResult({ "nInserted" : 1 })
>
```

```
Command Prompt - mongo
> db.game_list.findOne( { _id: "Far Cry Series" } ).ancestors
[ "Game", "Developer/Publisher", "Ubisoft" ]
> db.game_list.createIndex( { ancestors: 1 } )
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
> db.game_list.find( { ancestors: "Developer/Publisher" } )
{ "_id" : "Far Cry Series", "ancestors" : [ "Game", "Developer/Publisher", "Ubisoft" ], "parent" : "Ubisoft" }
{ "_id" : "Assassin's Creed Series", "ancestors" : [ "Game", "Developer/Publisher", "Ubisoft" ], "parent" : "Ubisoft" }
{ "_id" : "Ubisoft", "ancestors" : [ "Game", "Developer/Publisher" ], "parent" : "Developer/Publisher" }
{ "_id" : "CS:GO", "ancestors" : [ "Game", "Developer/Publisher", "Valve" ], "parent" : "Valve" }
{ "_id" : "DOTA 2", "ancestors" : [ "Game", "Developer/Publisher", "Valve" ], "parent" : "Valve" }
{ "_id" : "Valve", "ancestors" : [ "Game", "Developer/Publisher" ], "parent" : "Developer/Publisher" }
```

➤ Model Tree Structures with Materialized Paths

```
Command Prompt - mongo
> show databases
admin          0.000GB
config         0.000GB
game_array     0.000GB
game_child     0.000GB
game_parent    0.000GB
kelas          0.000GB
local          0.000GB
perpustakaan  0.000GB
> use game_MP
switched to db game_MP
> db.game_list.insert( { _id: "Game", path: null } )
WriteResult({ "nInserted" : 1 })
> db.game_list.insert( { _id: "Developer/Publisher", path: ",Game," } )
WriteResult({ "nInserted" : 1 })
> db.game_list.insert( { _id: "Ubisoft", path: ",Game,Developer/Publisher," } )
WriteResult({ "nInserted" : 1 })
> db.game_list.insert( { _id: "Valve", path: ",Game,Developer/Publisher," } )
WriteResult({ "nInserted" : 1 })
> db.game_list.insert( { _id: "CS:GO", path: ",Game,Developer/Publisher,Valve," } )
WriteResult({ "nInserted" : 1 })
> db.game_list.insert( { _id: "DOTA 2", path: ",Game,Developer/Publisher,Valve," } )
WriteResult({ "nInserted" : 1 })
> db.game_list.insert( { _id: "Assassin's Creed Series", path: ",Game,Developer/Publisher,Ubisoft," } )
WriteResult({ "nInserted" : 1 })
> db.game_list.insert( { _id: "Far Cry Series", path: ",Game,Developer/Publisher,Ubisoft," } )
WriteResult({ "nInserted" : 1 })
>
```

```
Command Prompt - mongo
> db.game_list.find().sort( { path: 1 } )
{ "_id" : "Game", "path" : null }
{ "_id" : "Developer/Publisher", "path" : ",Game," }
{ "_id" : "Ubisoft", "path" : ",Game,Developer/Publisher," }
{ "_id" : "Valve", "path" : ",Game,Developer/Publisher," }
{ "_id" : "Assassin's Creed Series", "path" : ",Game,Developer/Publisher,Ubisoft," }
{ "_id" : "Far Cry Series", "path" : ",Game,Developer/Publisher,Ubisoft," }
{ "_id" : "CS:GO", "path" : ",Game,Developer/Publisher,Valve," }
{ "_id" : "DOTA 2", "path" : ",Game,Developer/Publisher,Valve," }
> db.game_list.createIndex( { path: 1 } )
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```