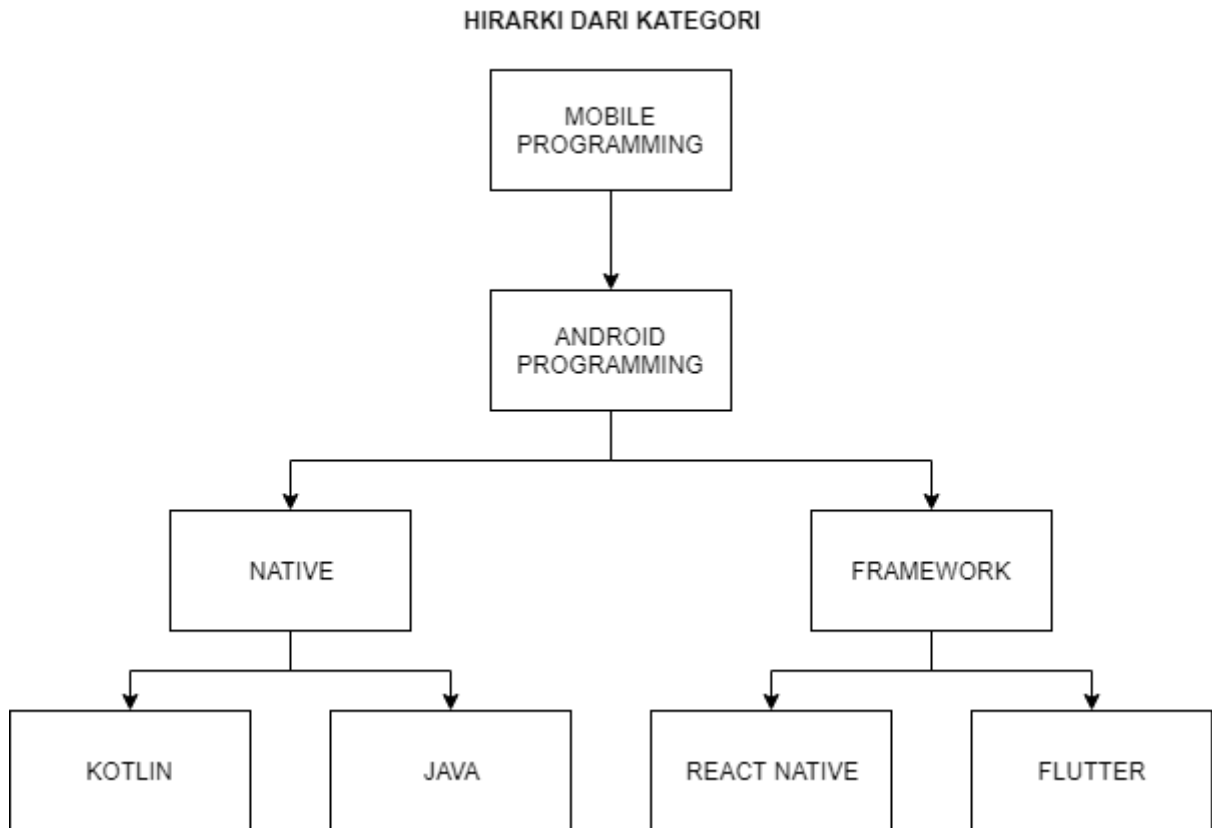


TUGAS BASIS DATA LANJUT

MODEL TREE STRUKTUR



Parent References :

Create db untuk Parent References:

```
> db
belajar_tree2
```

Insert db untuk Parent References:

```
> db.kategori.insert({_id:"ReactNative", parent:"Framework"})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({_id:"Flutter", parent:"Framework"})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({_id:"Framework", parent:"AndroidProgramming"})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({_id:"Kotlin", parent:"Native"})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({_id:"Java", parent:"Native"})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({_id:"Native", parent:"AndroidProgramming"})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({_id:"AndroidProgramming", parent:"MobileProgramming"})
WriteResult({ "nInserted" : 1 })
```

Query untuk pencarian Parent dari suatu Child :

```
> db.kategori.findOne({_id:"Flutter"}).parent
Framework
```

Query untuk membuat index pada Parent untuk mengaktifkan pencarian cepat oleh simpul parent:

```
> db.kategori.createIndex({parent:1})
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```

Query Pencarian Child melalui Parent :

```
> db.kategori.find({parent:"Framework"})
{ "_id" : "ReactNative", "parent" : "Framework" }
{ "_id" : "Flutter", "parent" : "Framework" }
```

Child Reference

Create db untuk Child Reference :

```
C:\WINDOWS\system32\cmd.exe - mongo.exe RAM: 66 % CPU: 27 %
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.
To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMo
nitoring()
---
> use belajar_tree
switched to db belajar_tree
> db
belajar_tree
```

Insert db untuk Child Reference:

```
> db.kategori.insert({_id:"ReactNative", children:[]})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({_id:"Flutter", children:[]})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({_id:"Framework", children:["ReactNative","Flutter"]})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({_id:"Kotlin", children:[]})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({_id:"Java", children:[]})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({_id:"Native", children:["Kotlin","Java"]})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({_id:"AndroidProgramming", children:["Native","Framework"]})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({_id:"MobileProgramming", children:["AndroidProgramming"]})
WriteResult({ "nInserted" : 1 })
```

Query untuk pencarian Child dari suatu Parent :

```
> db.kategori.findOne({_id:"Framework"}).children
[ "ReactNative", "Flutter" ]
```

Query untuk membuat index pada Child untuk mengaktifkan pencarian cepat oleh simpul Child:

```
> db.kategori.createIndex({children:1})
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```

Query Pencarian Parent melalui Child :

```
> db.kategori.find({children:"ReactNative"})
{ "_id" : "Framework", "children" : [ "ReactNative", "Flutter" ] }
```

Array of Ancestors

Create db untuk Array of Ancestors :

```
C:\WINDOWS\system32\cmd.exe - mongo.exe RAM: 63% CPU:13%
> use belajar_tree3
switched to db belajar_tree3
> db
belajar_tree3
```

Insert db untuk Array of Ancestors :

```
> db.kategori.insert({_id:"ReactNative", ancestors:["MobileProgramming","AndroidProgramming","Framework"], parent:"Framework"})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({_id:"Flutter", ancestors:["MobileProgramming","AndroidProgramming","Framework"], parent:"Framework"})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({_id:"Framework", ancestors:["MobileProgramming","AndroidProgramming"], parent:"AndroidProgramming"})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({_id:"Java", ancestors:["MobileProgramming","AndroidProgramming","Native"], parent:"Native"})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({_id:"Kotlin", ancestors:["MobileProgramming","AndroidProgramming","Native"], parent:"Native"})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({_id:"Native", ancestors:["MobileProgramming","AndroidProgramming"], parent:"AndroidProgramming"})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({_id:"AndroidProgramming", ancestors:["MobileProgramming"], parent:"MobileProgramming"})
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({_id:"MobileProgramming", ancestors:[ ], parent: null})
WriteResult({ "nInserted" : 1 })
```

Query untuk pencarian Child berdasarkan Ancestorsnya :

```
> db.kategori.findOne({_id:"Flutter"}).ancestors
[ "MobileProgramming", "AndroidProgramming", "Framework" ]
```

Query untuk membuat index pada Ancestors untuk mengaktifkan pencarian cepat oleh simpul ancestors:

```
> db.kategori.createIndex({ancestors:1})
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```

Query Pencarian Ancestors dari “AndroidProgramming” :

```
> db.kategori.find({ancestors:"AndroidProgramming"})
{ "_id" : "ReactNative", "ancestors" : [ "MobileProgramming", "AndroidProgramming", "Framework" ], "parent" : "Framework" }
{ "_id" : "Flutter", "ancestors" : [ "MobileProgramming", "AndroidProgramming", "Framework" ], "parent" : "Framework" }
{ "_id" : "Framework", "ancestors" : [ "MobileProgramming", "AndroidProgramming" ], "parent" : "AndroidProgramming" }
{ "_id" : "Java", "ancestors" : [ "MobileProgramming", "AndroidProgramming", "Native" ], "parent" : "Native" }
{ "_id" : "Kotlin", "ancestors" : [ "MobileProgramming", "AndroidProgramming", "Native" ], "parent" : "Native" }
{ "_id" : "Native", "ancestors" : [ "MobileProgramming", "AndroidProgramming" ], "parent" : "AndroidProgramming" }
```

Materialized Paths

Create db untuk Materialized Paths :

```
> use belajar_tree4
switched to db belajar_tree4
> db
belajar_tree4
```

<<< PENTING >>>
KERJAKAN LAPORAN PI !!

<done>
• JARKOM LANJUT
• SISTEM INFORMASI

Insert db untuk Materialized Paths :

```
> db.kategori.insert({ _id : "MobileProgramming", path : null })
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({ _id : "AndroidProgramming", path : ",MobileProgramming," })
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({ _id : "Framework", path : ",MobileProgramming,AndroidProgramming," })
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({ _id : "Native", path : ",MobileProgramming,AndroidProgramming,Framework," })
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({ _id : "ReactNative", path : ",MobileProgramming,AndroidProgramming,Framework,ReactNative," })
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({ _id : "Flutter", path : ",MobileProgramming,AndroidProgramming,Framework,Flutter," })
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({ _id : "Kotlin", path : ",MobileProgramming,AndroidProgramming,Native," })
WriteResult({ "nInserted" : 1 })
> db.kategori.insert({ _id : "Java", path : ",MobileProgramming,AndroidProgramming,Native,Java," })
WriteResult({ "nInserted" : 1 })
```

Query Pencarian seluruh data dengan mengurutkan berdasar jalur bidangnya:

```
> db.kategori.find().sort({path:1})
{ "_id" : "MobileProgramming", "path" : null }
{ "_id" : "AndroidProgramming", "path" : ",MobileProgramming," }
{ "_id" : "Framework", "path" : ",MobileProgramming,AndroidProgramming," }
{ "_id" : "Native", "path" : ",MobileProgramming,AndroidProgramming,Framework," }
{ "_id" : "ReactNative", "path" : ",MobileProgramming,AndroidProgramming,Framework,ReactNative," }
{ "_id" : "Flutter", "path" : ",MobileProgramming,AndroidProgramming,Framework,Flutter," }
{ "_id" : "Kotlin", "path" : ",MobileProgramming,AndroidProgramming,Native," }
{ "_id" : "Java", "path" : ",MobileProgramming,AndroidProgramming,Native,Java," }
```

Query Pencarian Ancestors dari "AndroidProgramming" :

```
> db.kategori.find({path:/,AndroidProgramming,/})
{ "_id" : "Framework", "path" : ",MobileProgramming,AndroidProgramming," }
{ "_id" : "Native", "path" : ",MobileProgramming,AndroidProgramming,Framework," }
{ "_id" : "ReactNative", "path" : ",MobileProgramming,AndroidProgramming,Framework,ReactNative," }
{ "_id" : "Flutter", "path" : ",MobileProgramming,AndroidProgramming,Framework,Flutter," }
{ "_id" : "Kotlin", "path" : ",MobileProgramming,AndroidProgramming,Native," }
{ "_id" : "Java", "path" : ",MobileProgramming,AndroidProgramming,Native,Java," }
```

Query Pencarian Ancestors dari "MobileProgramming" :

```
> db.kategori.find({path:/,MobileProgramming,/})
{ "_id" : "AndroidProgramming", "path" : ",MobileProgramming," }
{ "_id" : "Framework", "path" : ",MobileProgramming,AndroidProgramming," }
{ "_id" : "Native", "path" : ",MobileProgramming,AndroidProgramming,Framework," }
{ "_id" : "ReactNative", "path" : ",MobileProgramming,AndroidProgramming,Framework,ReactNative," }
{ "_id" : "Flutter", "path" : ",MobileProgramming,AndroidProgramming,Framework,Flutter," }
{ "_id" : "Kotlin", "path" : ",MobileProgramming,AndroidProgramming,Native," }
{ "_id" : "Java", "path" : ",MobileProgramming,AndroidProgramming,Native,Java," }
```

Query untuk membuat index pada Path untuk mengaktifkan pencarian cepat oleh simpul ancestors:

```
> db.kategori.createIndex({path:1})
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```