	JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA POLITEKNIK NEGERI BANDUNG		
	LEMBAR JAWABAN	EVALUASI AKHIR SEMESTER (EAS)	Halaman 1 dari 6

Nama Lengkap : Aرسال فاديلاه NIM : 201524036 Kelas : 2B Kelompok : 13/2?	Nama Mata Kuliah : Pemodelan Data Kode Mata Kuliah : 16TIN4024 Perkuliahan : PRAKTIKUM Jenjang/TA/Semester : D-IV/2021/2022/Genap
---	--

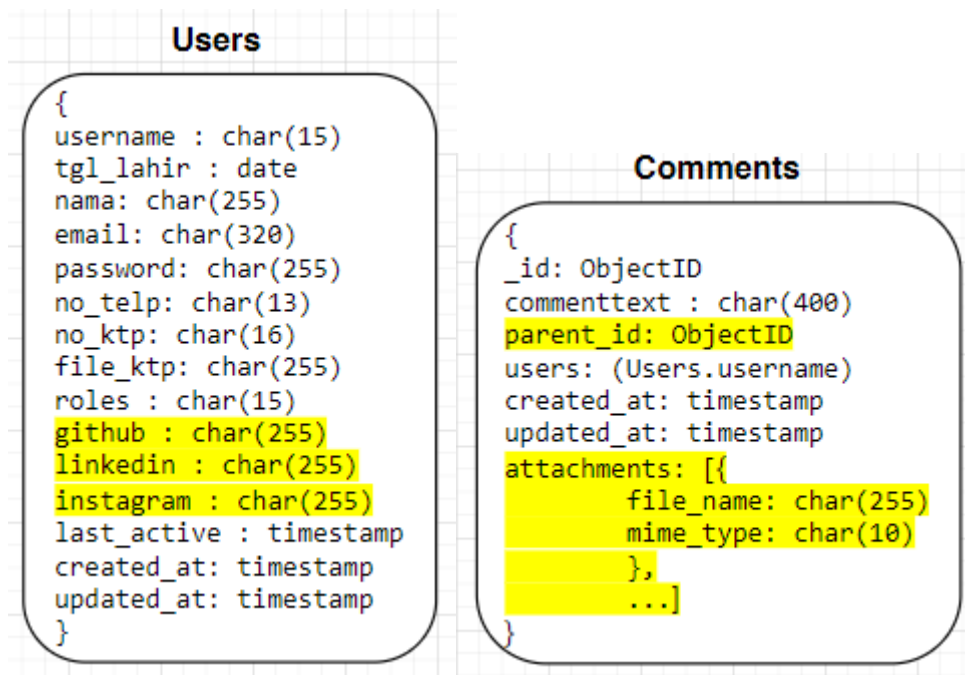
Soal Ujian Terdiri dari 6 nomor

1. Pada rancangan dalam bentuk CRD (*Collection Relationship Diagram*). Bagian *collection/field* manakah yang menurut Anda memerlukan pertimbangan **schemaless**? Jelaskan alasan Anda mengapa memerlukan pertimbangan schemaless pada bagian tersebut! [Bobot:15]
2. Pada rancangan dalam bentuk CRD. Bagian design manakah yang berdampak terhadap **performance**? Jelaskan alasan Anda berdasarkan *performance* seperti apa yang Anda harapkan atau pertimbangkan! [Bobot:15]
3. Buatlah pengembangan **rancangan** CRD dan **implementasi** fitur pada sistem/aplikasi Anda (MongoDB), yaitu: Fitur *user* yang mampu mencatat aktivitas *login*, dengan ketentuan sebagai berikut: [Bobot: 30]
 - a. Menangani aktivitas *login* menggunakan *username/email* dan *password* yang terenkripsi
 - b. Mampu mencatat waktu kapan saja telah melakukan *login*
 - c. Mampu mencatat IP *address* mana saja telah melakukan *login*
 - d. Mampu mencatat waktu penggantian *password* untuk dijadikan acuan setiap setahun sekali akan dilakukan perubahan *password*
 - e. Mampu mencatat percobaan melakukan *login* yang tidak valid (seperti kesalahan pengisian *username/email* atau *password*) untuk membatasi sebanyak tiga kali kesalahan oleh pihak yang tidak berwenang
 - f. Menangani kondisi user menjadi non-aktif
 - g. Menerapkan seluruh *constraint* pada *collection/field* terkait fitur.
 - a. Tuliskan perintah membuat **single index** untuk seluruh bagian pada rancangan Anda! Tambahkan penjelasan mengapa pada bagian tersebut diperlukan *index*!
 - b. Tuliskan perintah membuat **compound index** untuk seluruh bagian pada rancangan Anda! Tambahkan penjelasan mengapa pada bagian tersebut diperlukan *compound index*!
4. Pada rancangan CRD yang telah Anda buat. Jelaskan secara komprehensif, **design pattern** apa saja yang telah diimplementasikan (minimal 3 (tiga) *design pattern*)? Semakin relevan penjelasannya dengan **analisis workload**, semakin optimal nilai yang diperoleh. [Bobot: 20]
5. **Indexing MongoDB** [Bobot: 10]
6. Tuliskan minimal 3 (tiga) perintah membuat **aggregation** untuk seluruh bagian pada rancangan Anda! Tambahkan penjelasan mengapa diperlukan *aggregation* tersebut! [Bobot: 10]

1. Pada rancangan dalam bentuk CRD (*Collection Relationship Diagram*). Bagian *collection/field* manakah yang menurut Anda memerlukan pertimbangan schemaless? Jelaskan alasan Anda mengapa memerlukan pertimbangan schemaless pada bagian tersebut! [Bobot:15]

Jawab :

Collection/Field yang menjadi issue schemaless ditandai dengan background color berwarna kuning



Alasan:

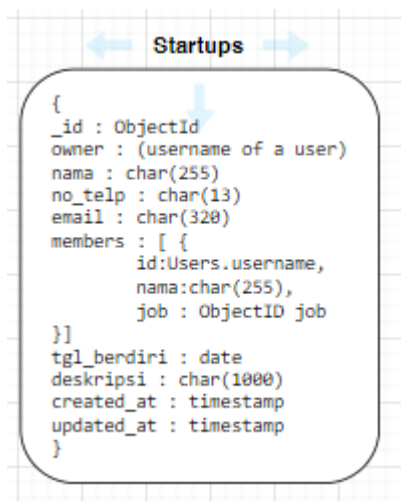
Pada collection Users terdapat tiga field yang menjadi issue schemaless yaitu github, linkedin dan instagram. Kenapa menjadi issue schemaless? Karena pada dasarnya users default (Investor) tidak akan terlalu membutuhkan akun github, linkedin dan instagram tetapi setelah user menjadi member dari salah satu startup maka field tersebut dibutuhkan untuk menarik para investor.

Kemudian untuk pada collection comment terhadap issue schemaless yaitu parent_id dan attachments. Parent_id adalah field untuk mengetahui comment yang diberikan user terhadap salah satu produk itu adalah comment balasan atau tidak. Sehingga jika tidak memiliki parent_id berarti comment tersebut merupakan comment yang berdiri sendiri bukan merupakan reply atau balasan comment lain. Kemudian untuk attachment pada comment dapat ditambahkan atau tidak.

2. Pada rancangan dalam bentuk CRD. Bagian design manakah yang berdampak terhadap performance? Jelaskan alasan Anda berdasarkan performance seperti apa yang Anda harapkan atau pertimbangkan! [Bobot:15]

Jawab :

Issue schemaless menjadi salah satu alasan peningkatan performance dari sisi penyimpanan. Sehingga tidak perlu alokasi field-field yang akan diisi oleh null. Bayangkan jika user memiliki dokumen 1.000.000 dengan hanya 250.000 user yang menjadi member dari startup. Ada sebanyak 750.000 dokumen yang tidak akan memiliki field github, linkedin dan instagram. Begitu pula dengan comment, terdapat 1.000.000 comment yang memiliki attachments file.



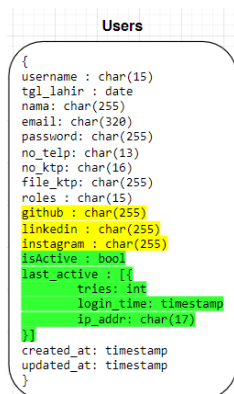
Selain itu, dokumen member yang di embedded kedalam collection start up menjadi salah satu yang akan berdampak terhadap performance. Dengan teknik ini maka tidak perlu lagi operasi join pada collection user karena setiap kali collection Startup dibaca maka collection members akan ikut terbaca.

3. Buatlah pengembangan rancangan CRD dan implementasi fitur pada sistem/aplikasi Anda (MongoDB), yaitu: Fitur user yang mampu mencatat aktivitas login, dengan ketentuan sebagai berikut: [Bobot: 30]

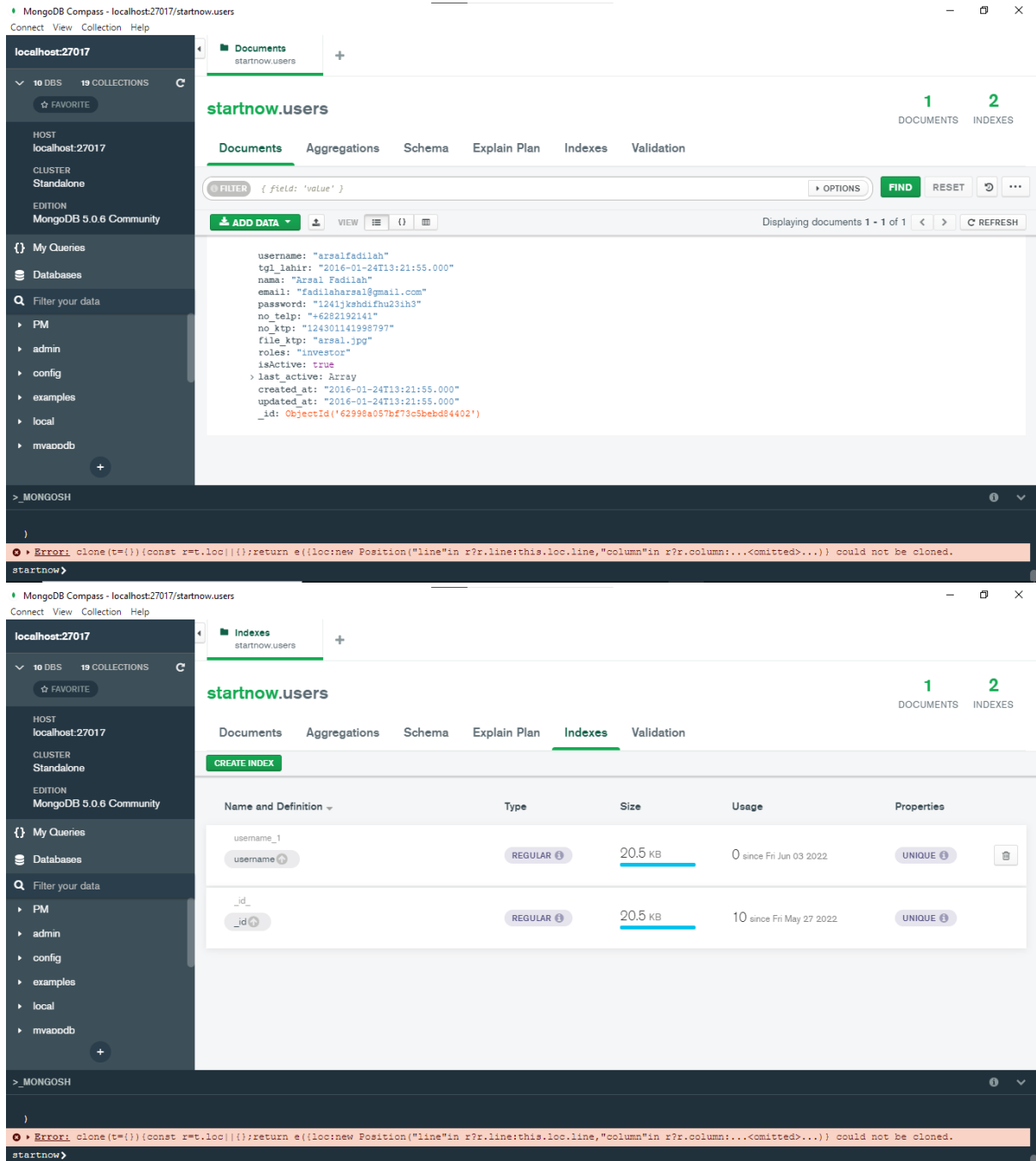
- Menangani aktivitas login menggunakan username/email dan password yang terenkripsi
- Mampu mencatat waktu kapan saja telah melakukan login
- Mampu mencatat IP address mana saja telah melakukan login
- Mampu mencatat waktu penggantian password untuk dijadikan acuan setiap setahun sekali akan dilakukan perubahan password
- Mampu mencatat percobaan melakukan login yang tidak valid (seperti kesalahan pengisian username/email atau password) untuk membatasi sebanyak tiga kali kesalahan oleh pihak yang tidak berwenang
- Menangani kondisi user menjadi non-aktif
- Menerapkan seluruh constraint pada collection/field terkait fitur.

Jawab :

Pada soal nomor satu telah diberikan gambar mengenai collection. Untuk dapat mengatasi fitur fitur diatas maka design collection akan menjadi seperti berikut ini :



Implementasi pada mongoDb untuk penerapan constraint sebagai berikut:

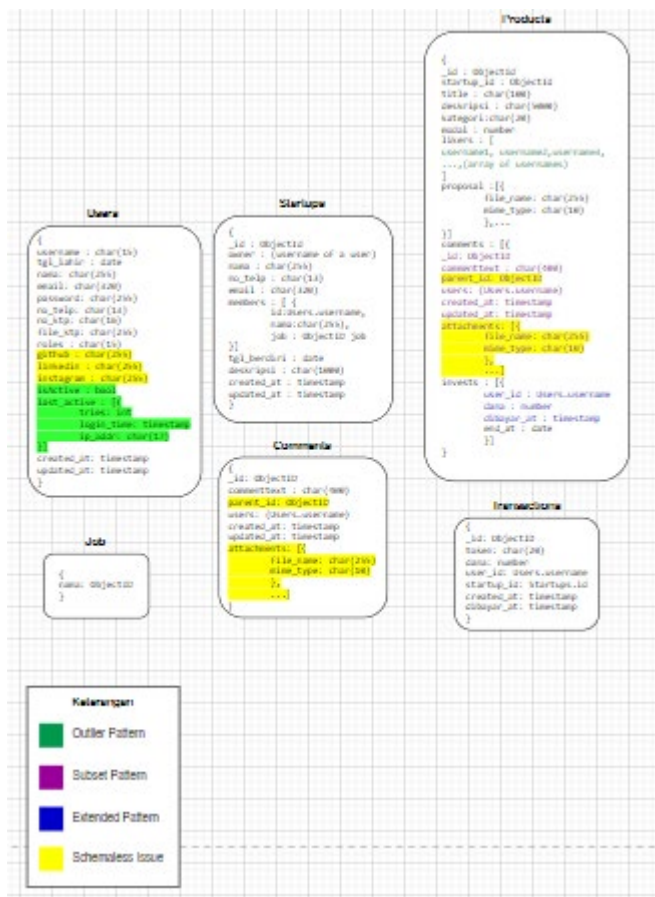


The screenshot displays the MongoDB Compass interface for the 'startnow.users' collection. The left sidebar shows the database structure, including the 'startnow' database and the 'users' collection. The main panel shows the 'Documents' tab with a single document displayed. The document contains fields for 'username', 'tgl_lahir', 'nama', 'email', 'password', 'no_telp', 'no_ktp', 'file_ktp', 'roles', 'isActive', 'last_active', 'created_at', 'updated_at', and '_id'. The bottom panel shows the 'Indexes' tab with a table of indexes.

Name and Definition	Type	Size	Usage	Properties
username_1 username	REGULAR	20.5 KB	0 since Fri Jun 03 2022	UNIQUE
id _id	REGULAR	20.5 KB	10 since Fri May 27 2022	UNIQUE

4.

Jawab :



- Subset Pettern

Pettern ini digunakan pada saat user membuka komentar dari start up, jika ada 200, jika seluruh komentar ditampilkan pada waktu yang sama, maka ini akan membutuhkan memori yang banyak. Untuk itu, demi meningkatkan efisiensi, kami menggunakan Subset Pettern, agar komentar tidak ditampilkan semua nya secara langsung. Komentar yang ditampilkan diurutkan dari komentar yang sering direply dan banyak interaksi didalamnya.

- Extended Pettern

Pettern ini digunakan Ketika user ingin berinvestasi di suatu produk dalam hal transaksi. Daripada menduplikasi semua hal yang ada di collection transaksi, dengan pettern ini cukup mereference field yang diperlukan saja.

- Outlier Pettern

Pettern ini digunakan untuk merekam siapa user menyukai suatu produk. Hal ini bertujuan agar kita bisa merekomendasikan kepada user apa produk yang mirip atau sejenis.

5.

Jawab :

Single index pada aplikasi kami dapat diaplikasikan pada collection transactios untuk data invests. Hal ini bertujuan untuk mengurutkan jumlah dana yang diinvestasikan oleh investor pada suatu produk. Dengan begitu, Start Up mengetahui siapa yang berinvestasi lebih banyak dan siapa juga yang lebih sedikit.

Syntax:

	JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA POLITEKNIK NEGERI BANDUNG		
	LEMBAR JAWABAN	EVALUASI AKHIR SEMESTER (EAS)	Halaman 6 dari 6

db.transactions.createIndex({ "dana": 1 })

6.

Jawab:

```
//aggreration
db.invest.aggregate( [
  {
    $match: { karya_id: 10 }
  },
  {
    $group: { _id: "$karya_id", totalDana: { $sum: "$dana" } }
  }
] )
```

Hasil :

```
{ "_id" : 10, "totalDana" : 1100000 }
```

```
db.invest.aggregate( [
  {
    $match: { investor_id: 2 }
  },
  {
    $group: { _id: "$karya_id", dana: { $sum: "$dana" } }
  }
] )
```