

Tugas Besar 1

IF3230 - Sistem Paralel dan Terdistribusi

Peer to Peer Collaborative Editing menggunakan Conflict-free Replicated Data Type (CRDT)

Dipersiapkan oleh:

Asisten Laboratorium Sistem Terdistribusi



Start Date: 11 April 2019

End Date: 24 April 2019

A. Persiapan Tugas Besar

Pada tugas besar kali ini, terdapat tiga komputer yang dibutuhkan, masing-masing komputer merupakan sebuah *node* yang terhubung secara *peer-to-peer*. Silahkan membaca pengenalan persoalan di bawah dan artikel [berikut](#) untuk memahami persoalan dengan lebih baik. Konsep yang kental ada di artikel tersebut sehingga diharapkan masing-masing anggota membaca terlebih dahulu artikel tersebut sebelum melakukan pembuatan program.

B. Pengenalan Persoalan

Text Editor

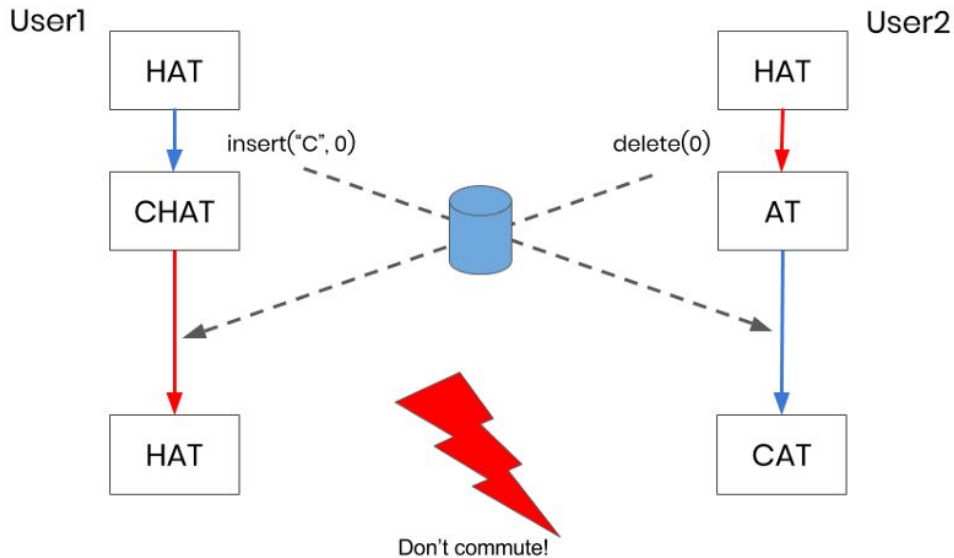
Dalam persoalan ini, text editor didefinisikan sebagai aplikasi untuk menambah (insert) dan menghapus (delete) karakter pada suatu teks. Setiap karakter pada teks memiliki indeks, dengan karakter pertama memiliki indeks 0. Sebagai contoh, misal terdapat teks “saya sedang makan”. Proses penambahan karakter ‘h’ setelah kata ‘saya’ dapat ditulis sebagai Insert(‘h’, 4). Teks hasil dari operasi ini adalah “sayah sedang makan”. Perhatikan bahwa penyisipan karakter ‘h’ setelah kata ‘saya’ menyebabkan semua karakter setelah ‘saya’ mengalami penambahan nilai indeks sebesar satu. Contoh proses penghapusan karakter adalah menghapus karakter pertama dari teks yang dapat ditulis sebagai Delete(0). Teks hasil dari operasi ini adalah “ayah sedang makan”. Pada contoh ini, semua karakter yang tersisa mengalami pengurangan nilai indeks sebesar satu.

Syarat Konsistensi

Terdapat dua syarat yang perlu dipenuhi untuk mencapai konsistensi bagi sebuah teks editor, yaitu:

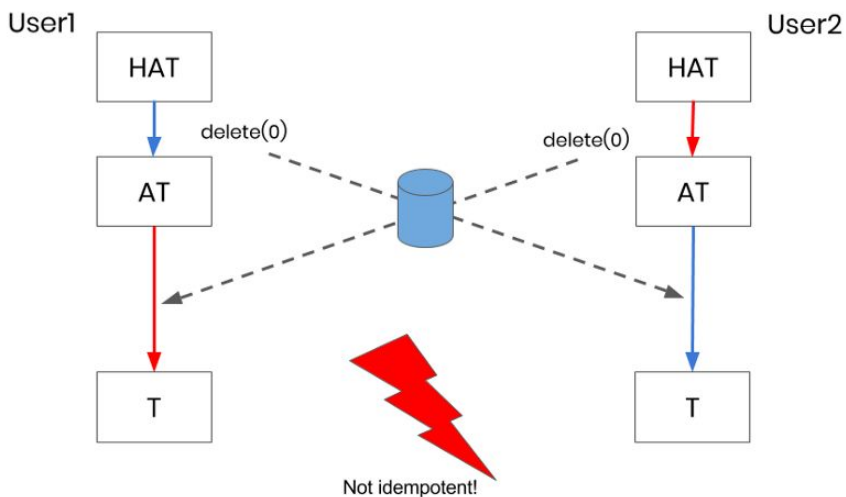
1. Komutatif

Komutatif terjadi saat operasi-operasi menghasilkan hasil yang sama meskipun dilakukan dengan urutan yang berbeda-beda. Misalnya, penjumlahan merupakan hukum komutatif. $a + b = b + a$. Dalam teks editor, hal ini penting. Pada gambar di bawah, digambarkan sebuah skenario dimana sifat komutatif diperlukan dan jika tidak dapat menyebabkan hasil yang tidak konsisten. Operasi insert dan kemudian delete pada teks editor yang tidak komutatif akan menghasilkan hasil yang berbeda dengan operasi delete dahulu baru insert.



2. Idempoten

Idempoten terjadi saat sebuah operasi selalu menghasilkan hasil yang sama terlepas dari berapa kali operasi dilakukan. Contohnya adalah perkalian dengan angka satu. Berapa kalipun sebuah angka dikali dengan satu, hasilnya akan sama. Pada gambar di bawah, digambarkan sebuah skenario dimana sifat ini diperlukan dalam sebuah teks editor. User1 dan User2 menghapus karakter yang sama secara bersamaan. Namun, hasil propagasi yang didapat oleh User1 adalah User2 karakter pertama. Hal tersebut ditangkap dan karakter pertama yang berbeda dihapus oleh User1. Teks editor yang idempoten dapat mengenali setiap karakternya secara unik sehingga penghapusan sebuah karakter yang terdeteksi lebih dari sekali tidak akan merusak hasil.



Collaborative Editing

Collaborative editing adalah proses group-editing yang menghasilkan sebuah pekerjaan akhir melalui kontribusi individual. Collaborative writing adalah penulisan yang dilakukan oleh lebih

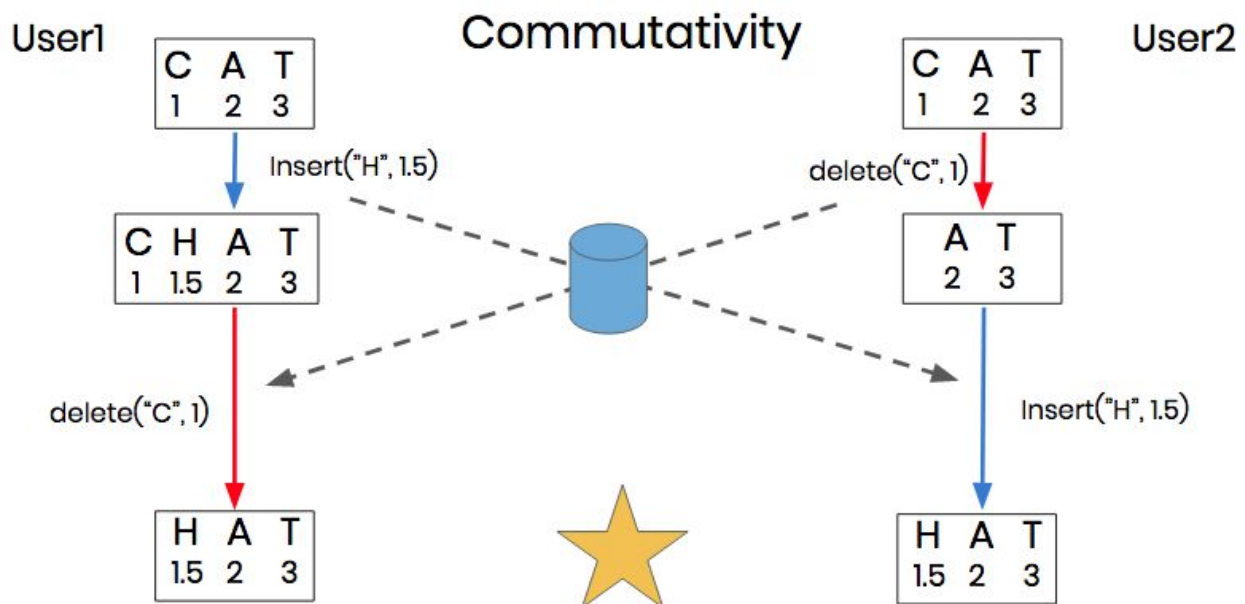
dari satu orang. Pembagian penulisan dapat dilakukan dengan banyak cara, misalkan setiap orang bertanggung jawab untuk menulis di bawah heading tertentu, baik secara berurutan maupun secara bersamaan.

Seringkali, collaborative editing diterapkan untuk menulis dokumen (contohnya Google Docs, Wiki, dsb) ataupun source code dari sebuah program (contohnya Version Control System). Kontribusi asynchronous (non-simultaneous) sangat time-efficient, karena anggota kelompok tidak harus mengerjakan pekerjaannya pada satu mesin. System collaborative editing juga dapat menyimpan label pada setiap bagian yang menandakan siapa kontributor dari bagian tersebut.[1]

Conflict-free Replicated Data Type (CRDT)

Pada pemrograman terdistribusi, CRDT adalah tipe data yang dapat direplikasi pada beberapa komputer dalam sebuah jaringan. Replika tersebut dapat diperbaharui secara independen dan konkuren tanpa koordinasi antar replika, di mana selalu ada cara matematis untuk menyelesaikan *inconsistencies* yang dapat terjadi. Contoh penerapan CRDT adalah pada sistem online chatting dan SoundCloud. Distributed database NoSQL seperti Redis dan Riak mendukung tipe data CRDT.

Contoh operasi yang dilakukan pada CRDT dan dieksekusi secara komutatif adalah sebagai berikut:



Pada gambar tersebut, awalnya pada node User1 tertulis "CAT", User1 ingin melakukan operasi insert "H" di antara C dan A. Berdasarkan CRDT posisi "H" merupakan posisi fraksional antara C dan A yaitu 1.5, pada program akan di tulis [1,5]. User1 melakukan broadcast ke User2, namun secara konkuren User2 juga melakukan perubahan terhadap teks awal yaitu "CAT". Perubahan yang dilakukan User2 adalah menghapus karakter C pada posisi ke 1. Pesan operasi ini juga akan dibroadcast ke node User1.

State sekarang pesan pada User2 adalah “AT”, sedangkan pesan pada User1 adalah “CHAT”. Ketika broadcast pesan User1 sampai ke User2, User2 memeriksa apakah operasi tersebut konflik atau tidak, dan ternyata operasi tersebut tidak konflik sehingga User2 memiliki tulisan “HAT”. Ketika pesan broadcast User2 sampai ke User1, User1 memeriksa operasi tersebut apakah konflik atau tidak, dan ternyata tidak konflik sehingga User1 menerapkan operasi dari User2, yaitu **delete** karakter C pada posisi 1. User1 memiliki state teks berupa “HAT”, sedangkan User2 juga, oleh karena itu operasi yang dilakukan oleh kedua user ini komutatif sehingga operasi memenuhi syarat pertama konsistensi.

C. Spesifikasi Tugas

Flow program secara umum beserta contoh yang diminta untuk dikerjakan adalah sebagai berikut (3 Node):

1. Inisialisasi:
 - a. Setiap node memiliki replikanya masing-masing
 - b. Setiap node melakukan sinkronisasi antar replikasi menggunakan CRDT
 - c. Setiap node berinteraksi dengan menggunakan pertukaran pesan ke node-node lainnya (asumsi hubungan hanya terhadap 3 node tersebut)
2. Node 1 melakukan perubahan pada text editor untuk menulis “CAT”
3. Node 1 melakukan perubahan terhadap CRDT yang dimilikinya untuk menyesuaikan perubahan yang ada pada *text editor*
4. Node 1 mengubah masing-masing operasi ke sebuah objek karakter (**site-id**: sumber operasi, **value**: karakter, **type**: tipe operasi, **position**: posisi karakter yang ingin dilakukan operasi) :
 - a. OP(1, 'Insert', 'C', 0) //insert karakter C pada posisi ke 0
 - b. OP(1, 'Insert', 'A', 1) //insert karakter A pada posisi ke 1
 - c. OP(1, 'Insert', 'T', 2) //insert karakter T pada posisi ke 2
5. Node 1 melakukan broadcast ke node-node lainnya (Node 2 dan Node 3) menggunakan sebuah kelas *messenger*
6. Node 2 menerima broadcast *message* operasi tersebut menggunakan kelas *messenger* miliknya
7. Node 2 melakukan verifikasi terhadap message operasi tersebut menggunakan kelas CRDT dan kelas Version Vector sebelum menerapkan ke *text editor* miliknya
8. Node 2 berhasil melakukan verifikasi terhadap operasi tersebut, dan melakukan pembaharuan terhadap CRDT dan Version Vector miliknya
9. Node 2 menerapkan operasi-operasi tersebut ke replika miliknya sendiri kemudian menampilkannya di text editor

Perlu diperhatikan contoh di atas hanyalah contoh yang disederhanakan, karena Node 1 dan Node 2 mungkin melakukan perubahan secara konkuren/berbarengan, dan mungkin saja terdapat latency antara Node 1 dan Node 2 sehingga masing-masing operasi mungkin akan sampai secara tidak sesuai dengan keinginan.

Pada tugas ini, Anda diminta untuk mengimplementasikan sebuah program untuk melakukan collaborative editing yang dapat dicapai dengan melakukan implementasi beberapa kelas, yaitu:

1. Controller
2. CRDT
3. Version Vector
4. Messenger

Masing-masing kelas tersebut dapat dijelaskan sebagai berikut:

1. Controller: kelas ini memiliki fungsi untuk melakukan
 - a. Melakukan pembaharuan CRDT
 - b. melakukan pembaharuan Version Vector
 - c. Menerima *notify* operasi-operasi yang diterapkan ke *text editor*
 - d. Mengubah operasi-operasi ke dalam objek operasi yang siap dikirim oleh messenger
 - e. Meminta *messenger* untuk mengirim operasi-operasi tersebut
 - f. Menerima *notify* operasi dari *messenger*
 - g. Melakukan verifikasi operasi yang diterima terhadap CRDT dan Version Vector miliknya
 - h. Menerapkan perubahan ke *text editor*
 - i. Menyimpan **Deletion Buffer** yang digunakan untuk mempertahankan *causality* jika terjadi kasus di mana operasi delete seharusnya dilakukan belakangan belakangan dibanding insert terhadap suatu karakter di posisi tertentu, namun karena *latency* operasi delete masuk terlebih dahulu.
 - j. Menerapkan operasi delete pada **Deletion Buffer** jika sudah memenuhi syarat
2. CRDT: kelas ini memiliki fungsi untuk melakukan operasi terhadap CRDT baik itu **insert** maupun **delete** dan melakukan **update** terhadap struktur datanya
3. Version Vector: kelas ini memiliki fungsi untuk menyimpan version vector yang dimiliki oleh sebuah node (**id-node** dan **counter** operasi)
4. Messenger: kelas ini memiliki fungsi untuk melakukan *broadcast* objek operasi dan menerima *broadcast* objek operasi

Selain itu, Anda diwajibkan untuk mengimplementasi *text editor*, text editor dapat berupa *command line interface* (CLI) ataupun GUI (akan mendapatkan bonus).

Anda diharapkan untuk menyelesaikan seluruh permasalahan menggunakan struktur data yang sudah dijelaskan, tidak diperkenankan menggunakan konsep *locking*. Operasi-operasi dasar yang wajib diimplementasikan pada struktur data ini adalah **insert** dan **delete**. Operasi-operasi lainnya merupakan pengembangan dari kedua operasi tersebut.

Beberapa kasus yang perlu ditangani adalah:

- Kasus konkuren *editing*

- Kasus yang merupakan syarat konsistensi (komutatif dan idempoten)
- Kasus yang membutuhkan **Deletion Buffer** dan **Version Vector**

Selain itu terdapat beberapa hal yang perlu diperhatikan, yaitu:

- Beberapa bahasa pemrograman yang diijinkan untuk digunakan adalah C/C++ dan Java.
- Tidak diperkenankan untuk menggunakan *library* apapun selain *library* default dari masing-masing bahasa (jika membutuhkan sebuah *library* dipersilahkan menghubungi salah satu asisten beserta alasan justifikasi *library* tersebut diperbolehkan)
- Hubungan antar node wajib menggunakan WebSocket (TCP/UDP).
- Implementasi text editor (*front end*) dan controller (*back end*) diperkenankan untuk menggunakan *thread* maupun *polling*

D. Pengumpulan dan Deliverables

Tugas dikerjakan dalam kelompok sebanyak maksimal 3 orang (anggota kelompok tidak boleh dari kelas yang berbeda). Fork spesifikasi tugas ini serta contoh *source code* dari *repository* [link](#). Kerjakan persoalan yang diberikan pada deskripsi di atas dan lakukan merge request paling lambat 24 April 2019 pukul 23.59 WIB.

Merge request dilakukan dengan judul TugasBesar1_K0[1|2|3]_<NIM1>_<NIM2>. Perhatikan bahwa keterlambatan pengumpulan dapat mengakibatkan nilai 0 (nol).

Beberapa file yang harus ada dalam repository tersebut diantaranya:

- Direktori **src** yang berisi source code yang anda buat.
- File README.md yang berisi:
 - Petunjuk penggunaan program.
 - Pembagian tugas yang mencantumkan NIM, persentase pengerjaan, serta deskripsi **singkat**. Harap diisi sejujur mungkin karena akan berdampak ke penilaian. Asisten juga melihat commit kalian untuk melakukan justifikasi terhadap apa yang ditulis.
13500999 (50%): fungsi YYYY, ZZZZ, dan YYZZ.
13500998 (50%): fungsi AAAA dan laporan.
13500997 (0%): tidak mengerjakan.
- Laporan yang berisi beberapa poin berikut:
 - Jelaskan cara kerja program kalian beserta desain program (arsitektur program). Berikan ilustrasi jika perlu.
 - Jelaskan fungsi **CRDT**, **Version Vector** dan **Deletion Buffer** di program kalian beserta desain struktur data tersebut di program kalian
 - Berikan analisis terhadap program yang kalian buat, apakah ada solusi yang lebih baik (baik dari segi arsitektur maupun yang lainnya)?
 - Beberapa kasus uji yang kalian gunakan untuk menguji kebenaran program kalian (minimal empat kasus uji)
 - Screenshot / video demo program kalian yang bekerja

E. Survey Asisten

<https://forms.gle/SUBjntdBn9tZe1dj7>

Segala bentuk kecurangan yang terjadi akan ditindaklanjuti oleh asisten. Mohon jangan melakukan plagiarisme karena akan menyebabkan nilai 0. Have fun.

F. Referensi

[1] https://en.wikipedia.org/wiki/Collaborative_editing

[2] https://en.wikipedia.org/wiki/Conflict-free_replicated_data_type

[3]
<https://hackernoon.com/building-conclave-a-decentralized-real-time-collaborative-text-editor-a6ab438fe79f>