

Nama : Dhea Safira

Kelas : TK 2A

Prodi : D3 Teknik Komputer

Mata Kuliah : Pemrograman Microservice

Ringkasan mengenai materi VIDIO 11 Service Choreography:

Kita akan bahas tentang fans service koreografi Hai Masterpiece koreografi ini berbeda-beda ampir hampir berbeda kebalikannya yang hampir kebalikannya dibanding service or questions Nah kalau dalam service offices and itu kan kalau kita mau berkomunikasi dengan micro service lain itu kita akan melakukan daerah call menggunakan remote prosedur invocation Natus Apa bedanya dengan servis koreografi kalau service choreography komunikasi antara graduation service dengan Microsoft yang lainnya dia tidak ngelakuin direct call lagi tapi dia akan lakuin menggunakan messaging nah saya udah bahas juga tentang messaging kalau temen-temen belum sempet nonton teman-teman bisa nonton tentang episode messaging di belajar microservice Hai nan dalam service or questions jadi kalau kita pilih pola service station aggregation microservice adalah servis yang sangat kompleks dan mengerti semua alur bisnisnya jadi Yahudi orkes nya itu kan bisa selanjutnya dia disimpan by aggregation Maker service jadi kalau kita bisa kita pengen tahu semua bisnis logic itu adanya di action episode jadi seakan-akan si persistent microservice itu ia kasarnya service yang sedikit bego gitu dia nggak ngerti bisnis ojek decu bisa ngelakuin Insert update delete sebagainya Jadi kalau misalnya di Yang sebelumnya Yang di sini ya ini itu dianggapnya servicenya pinter banget karena dia ngerti semua bisnis ojeknya tapi kalau yang di sininya kita anggap ini agak sedikit bodoh Kenapa karena dia tidak ngerti bisnis logic apapun dia cuma tahunya diperintahkan sama si agression untuk melakukan sesuatu Jadi mungkin disini Epic yang dibikin cuma Insert update delete silek dan lain-lain untuk mengisi selanjutnya semuanya di hendel sama sih transaction Services ini kalau kita pilih service or questions nah bagaimana kalau kita pilih tentang dengan servis koreografi Hai Nah kalau misalnya kita memilih service koreografi semua Michael service dituntut untuk menjadi pintar jadi kalau sebelumnya cuma aggregation microservice yang pintar Sekarang semua makroskopis harus pintar maksudnya Apa

maksudnya sekarang itu yang sebelumnya kan si persisten Eko situs diperintahkan sama si aggregation Nah kalau sekarang bisa nggak Yusuf nggak ada lagi perintah yang dilakukan si aggression Michael service daerah ke servicenya Kenapa Karena sekarang komunikasinya udah via Messenger dengan kalau misalnya via messaging artinya siaga Khan Cuma ngirim event nanti servis-servis lain persistent service k atau status service lah itu dia harus bisa bereaksi terhadap event yang dikirim oleh si aggregation microservice jadi bisnis logikanya lebih bekerja ceker ya di semua microservices jadi enggak terpusat lagi di satu microservice Hai nah ini contohnya filmnya Hai macho ntohnya adalah Hai kalau misalnya Jadi flownya itu kita nggak bisa lihat dari satu flower Contohnya kayak PTN section kita langsung bayar itu kalau pedasnya jadi kita cuma ngaber interaksi sama transaction Services kalau Sebelumnya kan teteh service kita berinteraksi ke berbagai macam ekor service kalau sekarang udah nggak kayak gitu lagi kalau sekarang ini beneran agak susah untuk melihat full flow bisnisnya Kenapa karena semua udah bercecer di semua microservice artinya Apa artinya semua masker service akan lebih pintar dibanding yang service or questions nah Disini yang tadinya kita paint section habis itu transaksinya akan ngekol mesin service balance dan juga member Sekarang udah enggak seperti itu lagi karena semuanya Sekarang kita akan berkomunikasi menggunakan messaging alias ada perantaranya meses broker maka kita akan komunikasinya berbasiskan menggunakan Hai Jadi yang tadi sebelumnya Kalau saya bayar untuk ngambil data mercon saya harus ngambil dari mesin service kalau sekarang isu kebalik Jadi kalau misalnya ada perubahan zat diservis mercon service merchant ini akan mengirim data perubahannya Kemensos broker nah siapapun yang butuh data perubahan itu dia boleh ngambil data perubahannya dari simetris broker yg contohnya transaction item kan dia butuh data marchen ya maka dia akan menyimpan data mercon di dalam transaction service artinya nanti tax service ini akan punya database duplikasi dari data marginnya itu termasuk ketika ada member ada member baru ya terus kalau misalnya ada member baru atau perubahan member Makasih member service akan mempublish data perubahannya ke dalam message broker siapapun yang butuh Silahkan ambil datanya dari simetris prokernya nah contohnya Sutra servers itu butuh data member gitu Karena kan untuk transaksi kita butuh data member dan juga tamachen maka secara otomatis si member event ini akan dikonsumsi oleh si transaction service dan akan disimpan di database transaction Services ini Hai jadi kebalik ya sekarang ya kalau sebelumnya kantong service dia manggil mesin service di sekarang kebalik sudah saya service yang ngelis ton atau nge konsumen data

yang dikirim oleh mesin event dan juga member event jadi kebalik mulai kebalik hatinya sekarang menservis juga harus punya kepintaran Ketika ada perubahan data mercon di harus mengirim data eventnya ini juga sama ya Ini juga sekarang logikanya bukan lagi manggil service lain tapi ngelis tensi perubahan dari meses broker nah terus kalau misalnya saya mau ngelakuin transaksi Gimana kalau saya melakukan transaksi cukup mengambil data merconnya didatabase lokalnya karena sudah saya copy dari meses broker dan untuk ngecek data membernya cukup dari database juga database lokal saya sendiri instructions that service ini sendiri karena sudah mengcopy data yang dari si member event ini lalu tinggal dibikin transaction eventnya saya itu dikirim ke message broker Hai jadi sekian selesai ini dia nggak perlu lagi connect ke email service atau kembalian service Nah sekarang apa yang terjadi sekarang siapapun yang butuh data transaction event silahkan konsumsi data transaksi eventnya contohnya adalah Jalan service udah pasti dia butuh data transaction event Kenapa karena Berlin service perlu memotong Beyonce member dan juga menaikkan Beyonce mercon kalau dan saksikan jadi setelah situasinya di Pengabdian Setan section solita service Nova bisa sini pen Makasih bln service akan ke konsumen data eventnya lalu akan ngelakuin perubahan didatabase Siberian sini yaitu menurunkan Belen si usernya atau si marcy member dan menaikkan Beyonce merconnya nah Selain itu transaction event tidak hanya dikonsumsi oleh Siberian service dia akan dikonsumsi juga sama si email service untuk apa untuk mengirim email notifikasi adegan demikian logic logikanya sekarang akan tersebar disemua microservice jadi semua servicenya sekarang nggak ada lagi servis yang Dam atau yang bodoh dan service yang pintar semua service akan menjadi pintar dengan demikian distribusi pekerjaannya tidak terpusat di tim yang gede Flop aggregation microservice jadi semua tim sekarang akan harus ngelakuin development karena ia bisnis ojeknya akan terpecah di semua microservice Hai nah ini adalah konsep dari service koreografi Jadi hampir kembalikan dari service fashion pertanyaannya Plus minusnya sekarang Apa nih sembunyikan kita udah bahas fans-nya Nah sekarang di sini ini Plus minusnya apa Hai Oke kita bahas dari keuntungan service choreography dulu nah Age She makes service tidak tergantung dengan micro service lain ini udah pasti Hai Coba lihat di sini enggak ada transaction service itu dia tidak connect ke server service yang manapun lihat dia cuma terkoneksi ke mesin broker dia tidak langsung ngocor ml0 mebel Nobel Sanyo call macan dengan demikian kalau di service-service yang lain itu ada masalah ataupun tiba-tiba error atau tiba-tiba lemot service sih aggregation dia tetap jalan dengan aman ke contohnya kalau email service

nya bermasalah atau mati gitu ya ya tasnya sukses karena kan si recent Event sudah sorry transaction service sudah berhasil ngirim event Cuma mungkin email-nya agak sedikit delay agak sedikit ya nunggu sih servicenya nyala dulu misalnya simpelnya tapi nanti setelah dia nyala data Dimas ebooknya akan otomatis dikonsumsi sama si Imel service jadi tidak ada data yang hilang kalau diservis orkestrasi on itu ketika transaction gagangnya Hai data Melan susah ya gagal call balanced itu tahun bisa jadi balesnya nggak rubah tapi tasnya sukses jadi sedikit berbahaya kalau di koreografi itu harusnya datanya aman karena semua disimpan di message broker jadi walaupun ada yang mati di sini servicenya itu tetap aman karena semua sudah tersimpan di MSS broker Hai selanjutnya aggregation microservice akan lebih cepat karena tidak perlu komunikasi dengan mikosis lain Jadi kalau yang sebelumnya diservis orkes fashion itu Age service anggap aja butuh waktu yang lambat karena dia harus nol semua service Kalau sekarang nggak perlu itu harusnya lebih cepat harusnya jatuhnya cepat banget ya karena dia cuma ngirim data Kemenkes bocor habis itu selesai jadi kalau kita pakai konsep service koreografi harusnya si aggression microservice akan sangat cepat respon Time I Hai jika ada mengko service baru aggregation MB saves tidak perlu melakukan perubahan lagi nah ini yang paling penting kalau tiba-tiba Oh saya mau ngirim pakai SMS itu udah enggak pula giada perubahan di translation service contoh di sini ada SMS service cukup bikin SMS service lalu dia konsumen data dari sih message broker jadi nggak perlu ada lagi perubahan di transaction service AC Hai naju seperti ini ya Jadi kalau misalnya tiba-tiba saya mau bikin mengirim notifikasi via SMS cukup saya bikin SMS service Hai tinggal dia konsumen dari transaction event jadi transaction server sendiri dia tidak ada perubahan jadi Zero changers jadi ini lebih enak buat tim yang develops ITS section service karena siapapun yang fitur yang baru itu di develop diservis yang baru Tinggal di Angel konsumen dari meses yang sudah ada jadi ini lebih cepat harusnya adalah dari sisi development karena cukup di development sama tim si SMS service owner Hai Nah sekarang kita bahas minusnya Hai Kelihatannya sih kayak dikit ya minusnya tapi semuanya ini manusia lumayan bahaya juga contohnya yang paling pertama adalah sulit dijemak ketika terjadi masalah kalau misalnya yang koreografi sore yang orkestrasi on itu gampang single track errornya kita tinggal lihat di transaction servis-servis tinggal Cek dia Error ketika ngocol bareng service artinya errornya disitu Tapi kalau ini ini susah banget ngecek errornya dimana karena logika sudah tersebar dan semua komunikasinya udah asynchronus jadi enggak sinkronus lagi udah asynchronus jadi kita aku misalnya dari transaction service dan kirim fashion event tiba-

tiba balesnya nggak berubah Nah kita masih bingung nih gamenya Kenapa bingung yaitu errornya di balanced dia gagal konsumen atau errornya di transaction service dia gagal ngepublish itu harus dicek di beberapa tempat akhirnya atau contohnya saya transaksi tapi kok selalu error kalo transaksi di member Agit Uya ini di cek lagi apakah errornya ketika not consume data membernya atau gagal ngirim membernya dan ini harus dicek beneran di semua service yang kira-kira eh eh yang apa ya berelasi sama sih error yang terjadi itu jadi Gedebage ini agar bisa banget dibanding kalau kita pakai sih service or questions di Indonesia apalagi kalau misalnya koreografinya udah ngelewatin banyak service ya itunya komunikasinya itu bakal lebih susah lagi untuk ditebak ketika terjadi masalah Nah karena bisnis logikanya ini akan terdistribusi di semua microservice sehingga akan menyulitkan untuk dimengerti secara keseluruhan Jadi sekarang itu kalau misalnya ada misalnya ada karyawan baru gitu ya terus dia bilang eh pengen tahu dong flownya atas itu seperti apa nah dia nggak bisa tuh buka langsung satu service ini karena diservis ini enggak ada flow fullnya flownya dia cuma ngelakuin public event transaction Ivan udah selesai Dedi harus ngetrek sebelum transaksi event ini dipublish di harus ngetrek si transaction of service Ini dia ke konsumen MSS mana aja kalau dia bisa apa Bisa sini pen Siapa yang konsumennya itu harus di trek di semua Michael service jadi akan lebih susah untuk dilihat bisnis logic nya jadi walaupun ini kelihatannya lebih lebih cepet dia Enggak dia servicenya lebih independen karena tidak ketergantungan antar microservice tapi dia akan susah banget untuk di ngerti nah Saran saya kalau selain pribadi saya kalau misalnya teman-teman timnya misalnya masih sedikit gitu ya atau juga timnya anggap aja belum banyak yang senior itu kalau bisa pakai yang cara yang paling gampang dulu yaitu service or questions kalau teman-teman timnya misalnya udah banyak dan nggak mau ada ketergantungan microservice bahkan saya udah banyak yang senior yang udah ngerti cara nge-bug nah sebagainya itu boleh teman-teman pindah dari konsep service organization ke service koreografi Kalau saya sendiri secara ideal selalu merekomendasikan untuk menggunakan service koreografi karena ini lebih independency Maker servicenya dan dia lebih scale dalam sisi development dan juga skill dalam sisi skalabilitas si aplikasinya nah tapi konsekuensinya adalah yang tadi ya lebih susah untuk ditebak dan juga bisa selanjutnya berceceran dimana-mana sehingga susah untuk dimengerti Jadi butuh orang-orang yang udah senior atau pengalaman untuk mendevlop arsitektur dengan konsep service koreografi nah Selain itu ada yang kelewat sepertinya kekurangannya kekurangannya adalah dia satu lagi adalah dia

tergantung banget sama si meses lokernya Jadi kalau teman-teman perhatikan Sekarang semua komunikasinya lewat message broker Jadi kalau sebabnya mati gitu ya tiba-tiba bermasalah semua aplikasinya Udah akan jalan lagi karena semua tergantung Kemensos brokernya jadi teman-teman juga hati-hati dalam memilih masih brokernya jangan sampai mengirim SMS brokernya yang tidak scalable terlihat misalnya sering matilah atau sering bermasalah itu jangan sampai milih hasil Dokternya yang salah jadi harus pilih meses brokernya yang beneran bagus dan bisa skillnya bagus juga Khan mungkin Tentang pembahasan service koreografi jadi saya harapan teman-teman sekarang ngerti tentang service offices in dan koreografi dan kapan harus memilih salah satunya dan kapan apa aja plus dan minus dari kedua Konsep ini