

BAB III

FRAME WORK KOLABRASI

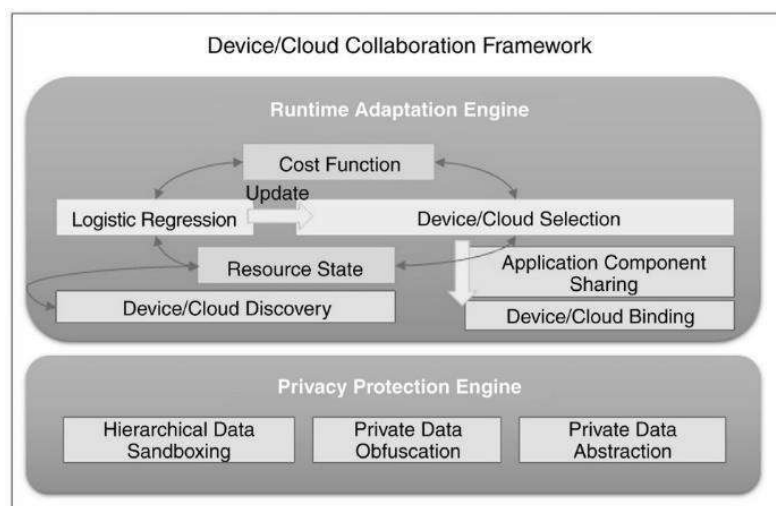
A. Capaian Pembelajaran

Setelah mempelajari materi ini, mahasiswa diharapkan mendapatkan apa yang telah dipelajari sesuai poin-poin dibawah:

1. Framework Kolaborasi Perangkat/Cloud
2. Aplikasi Kolaborasi Perangkat/Cloud

B. Materi

1. Framework Kolaborasi Perangkat/Cloud



Gambar 3. 1 framework kolaborasi

Perhatikan bahwa *Runtime Adaptation Engine* (RAE) berada di inti kerangka kerja, seperti yang

ditunjukkan pada Gambar 3.1. RAE menyimpan daftar perangkat dan cloud yang tersedia untuk memanfaatkan Device/Cloud Discovery, dan memantau status sumber daya yang tersedia. RAE menggunakan algoritme Regresi Logistik untuk mempelajari kebijakan yang paling hemat biaya untuk mendistribusikan tugas di antara perangkat dan cloud, berdasarkan status sumber daya. Di sini, definisi fungsi biaya adalah jumlah bobot status sumber daya (seperti masa pakai baterai), jaringan, dan penggunaan CPU. Kebijakan yang diperoleh dengan menjalankan Regresi Logistik diberlakukan oleh modul Pemilihan Perangkat/Cloud yang memilih sumber daya komputasi paling ekonomis, berdasarkan nilai biaya yang diharapkan untuk tugas tertentu.

Mekanisme RAE sebenarnya adalah agen otonom, yang dapat digunakan di setiap perangkat dan cloud. RAE berkomunikasi satu sama lain untuk berbagi status sumber daya secara transparan guna menentukan cara mendistribusikan beban kerja yang diberikan. RAE sisi cloud juga dapat memodelkan fungsi biayanya sendiri sebagai jumlah tertimbang dari siklus CPU residual dan ruang penyimpanan di seluruh infrastruktur. Jika kapasitas residual turun di bawah ambang batas tertentu, cloud mungkin harus menolak permintaan berbagi sumber daya yang datang dari perangkat yang dipasangkan. Ini karena menjalankan tugas yang diminta akan terlalu mahal. Secara khusus, RAE sisi cloud menyarankan RAE sisi perangkat untuk menjalankan

tugas di dalam perangkat atau cukup menunggu sumber daya komputasi di cloud dibebaskan. Seperti disebutkan sebelumnya, RAE pada perangkat dan cloud membuat keputusan secara mandiri, tanpa sistem perantara pendukung di tengahnya. Namun, RAE sisi perangkat memiliki beban untuk memantau status sumber daya cloud secara berkala. Di sisi lain, RAE sisi cloud tidak harus memantau status sumber daya dari jutaan perangkat yang dipasangkan, karena cloud membuat keputusan yang relatif sederhana, yaitu menolak atau menerima permintaan eksekusi tugas. Secara default, kerangka kerja kami tidak mempertimbangkan untuk memindahkan tugas yang dimulai dari cloud ke perangkat.

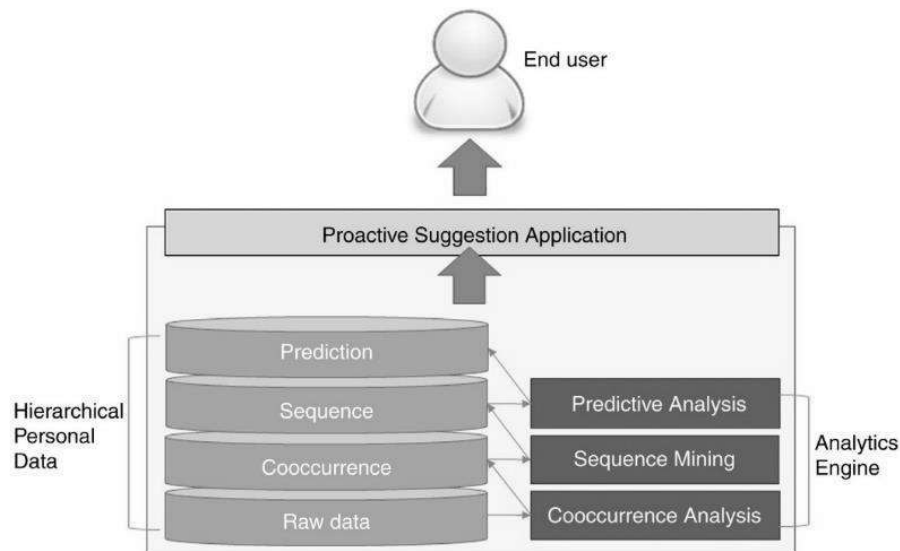
Ketersediaan sumber daya komputasi secara keseluruhan semakin meningkat, dan beban komputasi di cloud dapat dikurangi secara signifikan. Untuk mendukung distribusi tugas di antara perangkat tetangga, kerangka kerja kolaborasi perangkat/cloud mengimplementasikan penemuan perangkat (Device/Cloud Discovery pada Gambar 3.1). Misalkan perangkat (perangkat-A) ingin berkolaborasi dengan perangkat tetangganya (perangkat-B), yang tidak dilengkapi dengan komponen aplikasi yang sesuai untuk memproses tugas yang diberikan. Kemudian, perangkat-A dapat mentransfer komponen aplikasi yang diperlukan ke perangkat-B melalui Berbagi Komponen Aplikasi (ditunjukkan pada Gambar 3.1).

Kerangka kerja kami menggunakan teknologi untuk melindungi privasi dengan memasukkan data aplikasi yang terorganisir secara hierarkis dengan sandbox. Teknologi ini, yang diimplementasikan dalam modul Hierarchical Data Sandboxing (pada Gambar 3.1), mendukung pengguna untuk secara eksplisit menentukan sekelompok data dalam hierarki untuk dibagikan dengan cloud atau tidak. Kelompok data yang diatur untuk disimpan hanya dalam perangkat akan dilindungi oleh sandboxing. Meskipun pendekatan ini mendukung spesifikasi kebijakan perlindungan privasi berbutir halus, pendekatan deklaratif seperti itu akan terlalu rumit bagi banyak pengguna biasa. Dengan demikian, kita dapat mencari solusi alternatif untuk mengaburkan (mengkripsi) data untuk ditransfer ke perangkat tetangga atau ke cloud (Data Obfuscation pada Gambar 3.1). Solusi alami adalah mengenkripsi data saat mentransfer ke cloud. Namun, data terenkripsi dapat diungkapkan melalui pencurian kunci dekripsi dari server yang disusupi di cloud atau dengan spoofing di jaringan yang disadap. Untuk ancaman keamanan ini, biaya penanggulangan, seperti berbagi rahasia (misalnya, kunci dekripsi) di seluruh server yang direplikasi, tidak dapat diabaikan. Sebagai gantinya, kita dapat memiliki pendekatan yang lebih ringan dengan membiarkan cloud menganalisis data yang dikaburkan tanpa menguraikannya, dan membiarkan perangkat mengembalikan bagian yang dikaburkan dari hasil

analisis yang dihasilkan di sisi cloud. Misalnya, pengguna ingin menerima rekomendasi berbasis lokasi berdasarkan log pribadi POI yang dikunjungi. Baik POI itu sendiri maupun waktu kunjungan POI pertama kali dikaburkan di sisi perangkat. Pemetaan antara data asli dan data terenkripsi disimpan di sisi perangkat. Perangkat mengirimkan data terenkripsi ke cloud yang tidak memiliki kunci dekripsi untuk menguraikan data terenkripsi. Di sisi cloud, analisis data seperti penalaran kausal melalui penambangan urutan dilakukan, berdasarkan informasi terenkripsi (misalnya, POI dan waktu kunjungan). Misalnya, pengguna memiliki entri berikut di log lokasi, seperti yang ditunjukkan pada Tabel 3.1. Setiap entri berisi pemetaan antara data asli dan data terenkripsi.

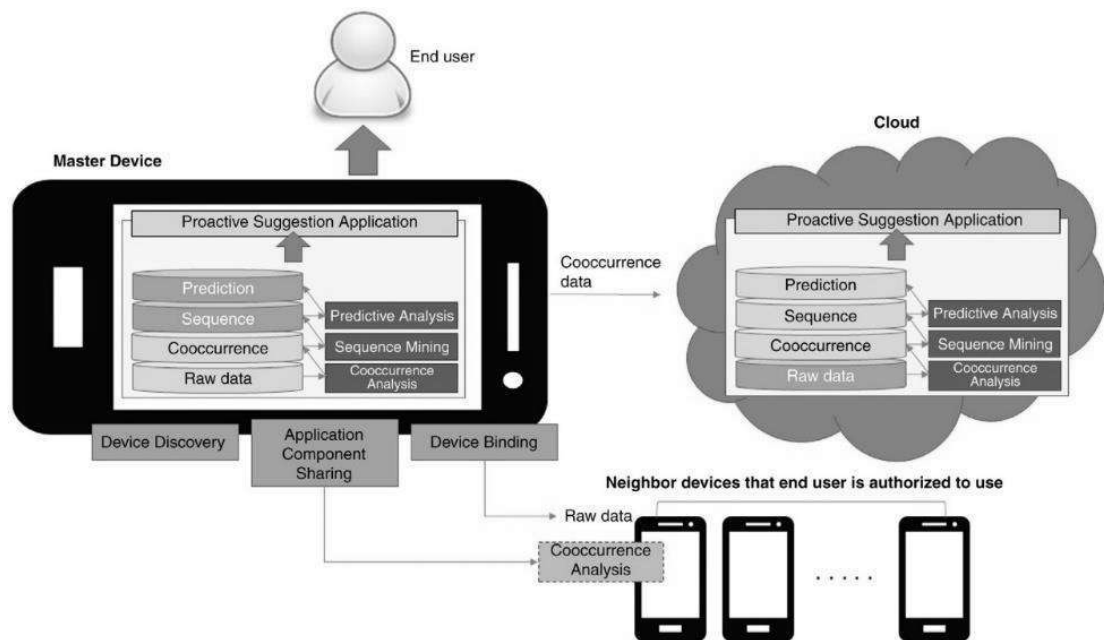
2. Aplikasi Kolaborasi Perangkat/Cloud

Berdasarkan data pribadi yang dikumpulkan di setiap perangkat seluler, kami telah merancang *Proactive Suggestion* (PS), sebuah aplikasi yang membuat rekomendasi kontekstual. Pada Gambar 3.2, masing-masing komponen PS diletakkan.



Gambar 3. 2 komponen PS

Mesin analitik PS menghasilkan data pribadi hierarkis yang saling bergantung satu sama lain. Data mentah seperti koordinat GPS, log panggilan, penggunaan aplikasi, dan kueri pencarian diumpankan ke mesin Analisis mata uang Cooc, yang bertanggung jawab untuk mengidentifikasi aktivitas yang terjadi pada saat yang sama. Misalnya, mesin analisis cooccurrence mungkin mengenali bahwa pengguna mendengarkan musik streaming langsung sambil berjalan di taman. Mengingat data cooccurrence seperti itu, mesin Sequence Mining dapat menyimpulkan hubungan sebab akibat antara aktivitas pribadi yang terjadi dari waktu ke waktu. Pola sekuensial yang dikenali dapat dimasukkan ke dalam mesin Analisis Prediktif untuk menilai kemungkinan aktivitas tertentu yang terjadi dalam konteks tertentu.

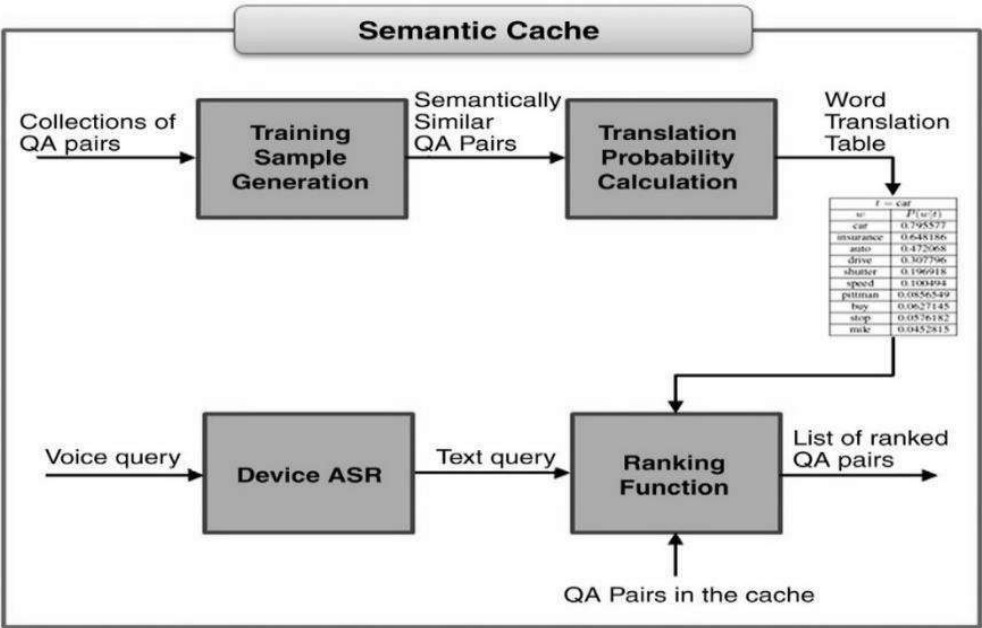


Gambar 3. 3 ilustrasi PS

Gambar 3.3 mengilustrasikan bagaimana PS mengimplementasikan kerangka kerja kolaborasi perangkat/cloud. Perangkat master dapat menemukan perangkat tetangga yang diizinkan untuk digunakan oleh pengguna akhir (*Device Discovery*). Perangkat master dapat mengirim data ke salah satu perangkat tetangga yang memiliki kapasitas komputasi yang cukup (*Device Binding*). Perangkat tetangga dapat mengambil mesin analitik yang sesuai untuk memproses data yang dikirim oleh perangkat master (Berbagi Komponen Aplikasi). Dalam contoh ini, potongan data yang disorot pada perangkat master dibagikan antara cloud dan perangkat tetangga.

Perhatikan bahwa aplikasi PS awalnya memilih Hierarchical Data Sandboxing untuk metode

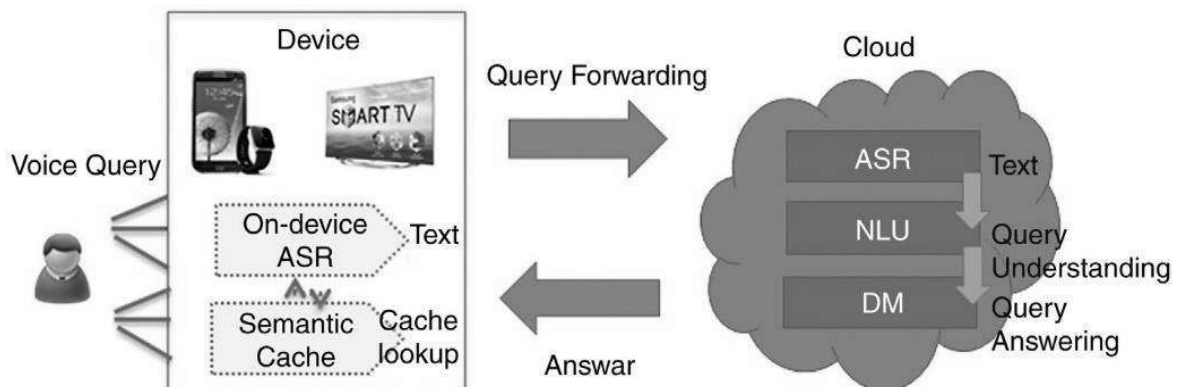
perlindungan privasi eksplisit dan deklaratif. Kami tidak dapat menjalankan metode perlindungan privasi alternatif berdasarkan pengaburan data, karena sumber daya yang terbatas pada perangkat yang sudah terhambat oleh pekerjaan analitik. Namun, ingatlah bahwa kerangka kerja kami cukup fleksibel untuk memungkinkan fungsi biaya yang ditentukan pengguna. Misalnya, jika biaya menjalankan operasi analitik (misalnya, biaya penggunaan masa pakai baterai) berlebihan, maka modul Pemilihan Perangkat/Cloud dalam kerangka kerja dapat memutuskan untuk mentransfer tugas analitik ke cloud atau hanya menunggu baterai naik di atas ambang batas yang dikonfigurasi. Ternyata mentransfer data melalui jaringan menghabiskan energi sebanyak menjalankan operasi analitik di dalam perangkat. Dengan demikian, modul Pemilihan Perangkat/Awan memilih untuk menunggu hingga baterai terisi daya di atas tingkat



yang dikonfigurasi.

Gambar 3. 4 Semantic cache

Cache QA semantik adalah aplikasi seluler yang mengambil jawaban atas kueri yang diberikan dari cache yang diisi dengan jawaban atas kueri serupa semantik yang dikeluarkan di masa lalu. Cache QA semantik dapat berguna saat tidak ada konektivitas Internet atau saat pengguna tidak ingin mentransfer kueri pribadi ke cloud. Gambar 3.4 mengilustrasikan bagaimana cache QA semantik dikelola. Cache QA semantik mengembalikan daftar kueri serupa dan jawaban terkait. Cache QA semantik terus memperbarui fungsi peringkat berdasarkan tabel terjemahan kata seperti yang dijelaskan dalam. Fungsi peringkat mengukur kesamaan antara kueri yang baru dikeluarkan dan kueri yang diukur di masa lalu.



Gambar 3. 5 implementasi kerangka kerja

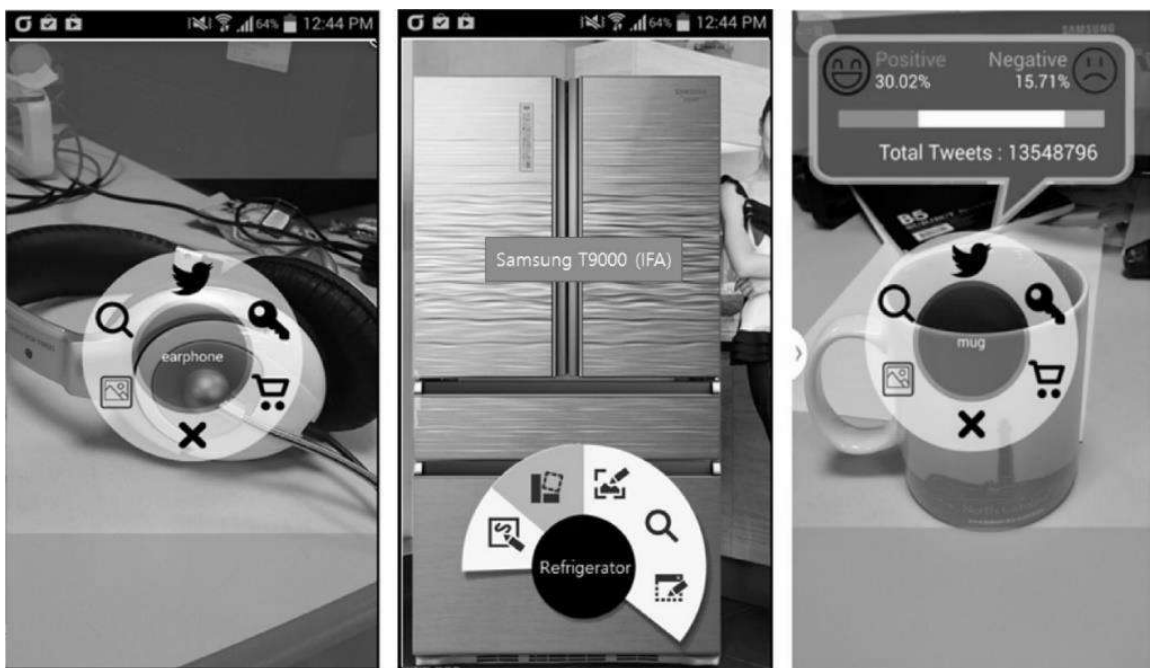
Pada Gambar 3.5, kami telah mendemonstrasikan implementasi kerangka kerja kolaborasi perangkat/cloud oleh cache QA semantik. Secara khusus, kami telah

merancang mesin ASR (*Automatic Speech Recognition*) khusus untuk perangkat seluler dan menggabungkan sistem cloud untuk Samsung S Voice dalam kerangka kerja kolaborasi. Sistem cloud untuk S Voice terdiri dari modul *Natural Language Understanding* (NLU) untuk pemahaman kueri, modul DM (*Dialog Manager*) untuk menjawab kueri, dan mesin ASR yang kuat.

Perhatikan bahwa kami telah mengadaptasi kerangka kerja untuk menghitung probabilitas cache QA semantik pada perangkat untuk menjawab kueri yang diberikan dengan benar. Jika probabilitasnya cukup tinggi, modul Pemilihan Perangkat/Cloud akan mengambil risiko mencari jawaban dari cache QA semantik. Jika cache tidak mengembalikan jawaban yang benar dan memaksa pengguna untuk bertanya lagi ke cloud, maka kerangka kerja kami akan menyesuaikan probabilitasnya.

Kami mengevaluasi manfaat kinerja menggunakan kerangka kerja kolaborasi perangkat/cloud untuk cache QA semantik. Dari log aplikasi QA berbasis suara kami, kami memperoleh 50 pertanyaan teratas yang sering dikeluarkan tentang cuaca, restoran, orang, dan perintah khusus perangkat. Kami memilih kueri acak dari himpunan menurut distribusi acak seragam (Metode 1) dan distribusi Zipf (Metode 2). Latensi mendapatkan respons untuk kueri diuji pada mode khusus cloud dan mode kolaborasi perangkat/cloud. Dalam mode khusus cloud, Google Voice API versi server digunakan untuk

ASR, dan DBpedia serta Freebase digunakan untuk menjawab kueri. Dalam mode kolaborasi perangkat/cloud, mesin ASR yang dibuat khusus dan cache QA semantik digunakan, bersama dengan layanan QA berbasis cloud. Memanfaatkan kolaborasi perangkat/cloud meningkatkan kinerja untuk kedua jenis beban kerja kueri. Latensi berkurang 56,7 dan 69,5% untuk Metode 1 dan Metode 2, masing-masing.



Gambar 3. 6 snapshot Watch&Go

Mengenali gambar dan ucapan secara otomatis dapat sangat meningkatkan pengalaman pengguna dalam menggunakan aplikasi. Misalnya, dengan pengenalan gambar otomatis, foto yang diambil oleh pengguna dapat secara otomatis ditandai dengan metadata dan dikatalogkan dengan lebih mudah. Mirip dengan Amazon Firefly, kami telah mengembangkan

aplikasi bernama Watch&Go, yang memungkinkan pengguna memperoleh informasi mendetail tentang suatu produk saat mengambil foto. Gbr. 3.6 menunjukkan snapshot Watch&Go yang memandu pengguna untuk fokus dengan benar pada beberapa produk elektronik, dan secara otomatis mengambil informasi seperti jenis, vendor, nama model, dan hasil sentimen sosial tentang produk tersebut.

Kepraktisan aplikasi pengenalan ini telah sangat meningkat, berkat kemajuan *Deep Learning* (DL) baru-baru ini. DL mengikuti pendekatan mempelajari korelasi antara parameter di berbagai lapisan perceptron. Namun, metode pelatihan model DL biasanya mengalami kurva pembelajaran yang lambat dibandingkan dengan metode pembelajaran mesin konvensional lainnya. Meskipun secara umum diyakini bahwa model DL yang lebih besar meningkatkan akurasi pengenalan melalui serangkaian data pelatihan yang disempurnakan dengan baik, memperoleh parameter yang memadai saat kami melatih beberapa lapisan pada saat yang sama merupakan tantangan. Kemunculan metode *Restricted Boltzmann Machine* (RBM) baru-baru ini, yang memungkinkan pelatihan berlapis dan tanpa pengawasan, dapat mengurangi batasan yang disebutkan di atas hingga tingkat tertentu. Namun, overhead komputasi keseluruhan masih tangguh, bahkan untuk cloud dengan sumber daya komputasi yang melimpah. Masalah kinerja

ini telah memotivasi kami untuk menggunakan kerangka kerja kolaborasi perangkat/cloud kami sebagai berikut.

Melalui kerangka kerja kami, bagian intensif komputasi dari DL (yaitu, pelatihan) ditugaskan ke cloud. Setelah pembelajaran selesai, kerangka kerja kami mem-port model pengenalan ke perangkat untuk pelaksanaan tugas pengenalan yang sebenarnya. Secara khusus, kami menggunakan model ImageNet-1000 yang dibangun berdasarkan metode *Convolutional Neural Networking* (CNN) [23,24]. Dengan model ini, klasifikasi hingga 1000 objek yang berbeda dimungkinkan. Namun, pengklasifikasi gambar sumber terbuka (OpenCV) pada perangkat Android membutuhkan waktu lebih dari 20 detik untuk mengklasifikasikan objek dengan ImageNet-1000. Hal ini disebabkan karena perkalian matriks pada perangkat tidak efisien. Kami telah mengatasi masalah ini dengan memparalelkan perkalian matriks berdasarkan OpenCL, menghasilkan latensi klasifikasi turun menjadi rata-rata 400 ms per objek. Dengan memanfaatkan pengklasifikasi gambar pada perangkat dengan latensi rendah pada jutaan perangkat seluler yang digunakan sebelumnya, kami dapat mengurangi beban komputasi di cloud secara signifikan.

Kami telah mencapai peningkatan kinerja serupa untuk aplikasi pengenalan ucapan dengan DL melalui kerangka kerja kolaborasi perangkat/cloud kami. Secara khusus, kami pertama-tama mengekstraksi data pidato senilai 400 jam dari *Fisher Corpus*. Berlawanan dengan

masalah pengenalan gambar, kami telah menggunakan model *Deep Neural Network* (DNN), yang terbukti efektif dalam membangun model akustik yang akurat. Mirip dengan pengenalan gambar, kami telah menetapkan tugas konstruksi model akustik dan tugas klasifikasi masing-masing ke cloud dan perangkat seluler. Secara khusus, kami mem-porting Kaldi ke perangkat Android untuk memproses permintaan pengenalan ucapan berdasarkan yang dibangun model akustik. Pemisahan tugas melalui kerangka kerja kolaborasi perangkat/cloud dan akselerasi tambahan melalui OpenCL membantu kami memperoleh hasil pengenalan dalam 0,9 RT (*RealTime*), yang merupakan penundaan yang dapat ditoleransi bagi pengguna akhir.

Kami dapat meringankan beban komputasi di sisi cloud lebih jauh dengan membagi bagian pembelajaran. Model yang ringan dapat dibuat dalam perangkat seluler. Namun, akurasi klasifikasi dapat dikompromikan ketika model ini digunakan. Kami telah mengamati bahwa tingkat akurasi toleran bervariasi antara pengguna akhir yang berbeda. Oleh karena itu, kerangka kerja kami dapat disesuaikan untuk mempelajari tingkat toleransi pribadi dan menentukan model mana yang akan dibangun sesuai dengan itu.

3. Kesimpulan

Kami telah mempresentasikan manfaat menggunakan kerangka kerja komputasi kolaboratif antara perangkat dan cloud melalui studi kasus aplikasi intelijen dunia nyata terpilih yang dirancang di Samsung Electronics. Aplikasi yang mengimplementasikan kerangka kerja kolaborasi perangkat/cloud dapat menghasilkan kinerja tinggi, seperti pengurangan latensi dalam memproses permintaan pengguna. Selain itu, biaya pengelolaan cloud dapat dikurangi ketika sumber daya komputasi pada jutaan perangkat seluler pintar digunakan. Selain manfaat dalam hal biaya dan kinerja, kerangka kerja membantu aplikasi melindungi privasi pengguna akhir dengan memproses data pribadi di dalam perangkat atau menganalisis versi data pribadi yang dikaburkan di cloud.

C. Latihan

1. Apa yang dimaksud dengan mekanisme RAE ?
2. Apa yang di maksud dengan definisi fungsi biaya ?
3. Apa itu Cache QA Semantik ?
4. Berdasarkan data pribadi yang dikumpulkan di setiap perangkat seluler, kami telah merancang Proactive Suggestion (PS). Apa itu *Proactive Suggestion* ?

D. Refrensi

Buyya, Rajkuma. Dastjerdi, Vahid, Amir. 2016. Internet Of Thing “*Principles and Paragdigms*”. Cambridge: Todd Green.