BAB I
PENDAHULUAN

Perkembangan cepat dan pencapaian masif dari kecerdasan buatan dari *deep learning* (DL) dan kecerdasan buatan, atau *artificial intelligence* (AI), berdampak pada semakin maraknya teknologi ini digunakan pada *edge computing* [1]-[4]. Dengan semakin besarnya ukuran model AI pada belakangan ini [5]-[7], ada tantangan dalam mengimplementasikan model tersebut ke perangkat *edge*, yang umumnya memiliki spesifikasi terbatas. Perangkat *edge* pada umumnya dibekali dengan central processing unit (CPU) yang berfokus pada efisiensi energi, dan penyimpanan terbatas [8]-[10]. Terlebih lagi, memori untuk perangkat *edge* biasanya berjenis *unified*, digunakan bersamaan oleh CPU dan graphical processing unit (GPU), dan mempunyai jumlah core GPU yang lebih sedikit dibanding GPU lain yang lebih mahal. Akibatnya, ada beberapa tantangan saat mengimplementasikan model *deep learning* yang rumit dan berukuran besar, seperti beban komputasi dan konsumsi energi yang tinggi, dan kebutuhan penyimpanan yang belum tentu dapat diatasi oleh perangkat *edge*.

Untuk menanggapi tantangan-tantangan ini, teknik kompresi model telah mendapatkan perhatian yang semakin meningkat [11]–[16]. Dengan menggunakan metode seperti kuantisasi dan *pruning*, ukuran model *deep learning* (DL) dapat dikurangi secara signifikan, memungkinkan model tersebut berjalan secara efisien pada perangkat *edge* yang terbatas sumber dayanya. Meskipun kompresi model secara tradisional dilakukan menggunakan GPU yang kuat, ada permintaan yang semakin meningkat untuk kompresi langsung di perangkat *edge* [17]–[20]. Pendekatan ini dapat meningkatkan kemampuan *resource managers* dan *schedulers* untuk mengimplementasikan AI di perangkat *edge*, dengan memungkinkan mereka menciptakan model yang dioptimalkan untuk memenuhi permintaan inferensi AI dan tujuan kinerja yang bervariasi.

Merancang sistem yang mampu melakukan kompresi di perangkat *edge* memerlukan pengetahuan mendalam dan karakterisasi rinci tentang metode kompresi di berbagai perangkat *edge*. Namun, area ini masih belum dieksplorasi secara masif. Memahami overhead dan kebutuhan sumber daya untuk operasi kom-

presi, serta manfaat kinerja yang diperoleh, termasuk peningkatan latensi dan throughput inferensi adalah hal yang amat penting. Sama pentingnya adalah mengidentifikasi dampak kompresi terhadap konsumsi sumber daya dan potensi kerugiannya, seperti penurunan akurasi saat inferensi.

Untuk mengatasi kesenjangan ini, kami melakukan studi karakterisasi untuk memahami peluang dan keterbatasan saat ini dari kompresi model pada perangkat *edge*. Khususnya, kami berfokus pada kuantisasi setelah *training* (*post-training quantization*/PTQ) [12], [21] untuk kompresi model, dengan mempertimbangkan keunggulannya, seperti sifatnya yang ringan dan efisien dalam penggunaan sumber daya, serta penghapusan proses *training* tambahan yang dapat memakan lebih banyak waktu dan sumber daya. Faktor-faktor ini membuat PTQ cocok untuk melakukan kompresi AI langsung dalam perangkat *edge*.

Dalam studi ini, kami menggunakan enam model *deep learning* (DL) yang dikhususkan untuk tugas klasifikasi gambar, dikategorikan berdasarkan ukuran model (dalam MB) menjadi model kecil, sedang, dan besar. Selain itu, kami menggunakan dua perangkat *edge* secara luas dengan spesifikasi perangkat keras yang berbeda: Jetson Xavier (XAVIER) [22] dan Orin Nano (ORIN) [23] dari NVIDIA. Meskipun kedua perangkat memiliki spesifikasi perangkat keras yang serupa untuk CPU dan ukuran memori, ORIN dilengkapi dengan jumlah *core* GPU yang jauh lebih banyak (2,7x lebih banyak), yang diharapkan dapat meningkatkan kinerja pemrosesan AI. Dengan menggunakan perangkat ini, kami bertujuan untuk menilai dampak ukuran GPU pada alur kerja kompresi dan efisiensi pemrosesan model DL yang telah dikuantisasi. Kami melakukan percobaan pada beberapa *precision mode* (jenis tipe data), misalnya, *16-bit floating point* (FP16) dan *8-bit integer* (INT8) dalam PTQ untuk menentukan kelebihan dan kekurangan dari setiap konfigurasi. Bertujuan untuk mengidentifikasi manfaat dan kelemahan yang paling signifikan dari kompresi model dan model yang telah dikuantisasi, khususnya peningkatan latensi dan penurunan akurasi, pada XAVIER dan ORIN.

Berdasarkan pengamatan kami, ada beberapa temuan mengenai proses kuantisasi langsung di perangkat *edge*. Pertama, waktu kompresi (*overhead*) un-

tuk PTQ di perangkat bervariasi tergantung pada ukuran model, spesifikasi perangkat, dan konfigurasi *precision mode*. Sebagai contoh, PTQ dengan mode presisi INT8 mengakibatkan waktu kompresi yang lebih lama karena persyaratan kalibrasi tambahan. Kedua, kompresi berbasis PTQ dapat mengurangi ukuran model sebanyak 58% dari ukuran aslinya, dan model yang dikuantisasi dengan presisi INT8 mendapati ukuran model berkurang 77%. Selain itu, meskipun terjadi pengurangan ukuran model yang signifikan, penurunan akurasi model yang dikuantisasi tetap relatif kecil. Misalnya, model yang dikuantisasi dengan mode presisi FP16 mengalami penurunan akurasi maksimal 4,5%, sementara model yang dikuantisasi berbasis INT8 dapat mengalami penurunan akurasi maksimal 13,9%. Pengurangan ukuran model juga mengakibatkan konsumsi GPU dan memori yang lebih rendah, meskipun penggunaan CPU menjadi lebih tinggi karena CPU mengatur penjadwalan untuk model-model yang lebih kecil (telah dikuantisasi) lebih sering. Terakhir, kami juga memperlihatkan bahwa model-model yang dikuantisasi dapat mencapai peningkatan signifikan dalam latensi inferensi, dengan pengurangan latensi antara 55% hingga 67% dibandingkan dengan model asli.

Studi ini memiliki beberapa kontribusi:

1. Karakterisasi proses PTQ langsung pada sebuah *edge device*: Kami membagi proses kompresi PTQ menjadi dua fase: 1) konversi model dasar ke representasi ONNX (Open Neural Network Exchange) [24] dan 2) konversi TensorRT (kuantisasi). Kami menyediakan analisis tentang *overhead* (waktu yang diperlukan untuk melakukan kuantisasi) dan konsumsi sumber daya untuk setiap langkah dari PTQ.

2. Memeriksa dampak spesifikasi perangkat *edge* terhadap PTQ: Kami menguji perbandingan dampak kapasitas GPU (jumlah core) terhadap efektivitas kompresi model, khususnya mengidentifikasi bagaimana perbedaan jumlah *core* GPU pada perangkat *edge* mempengaruhi alur kerja kompresi PTQ.

3. Melakukan analisis terhadap dampak dan kinerja dari model-model yang telah dikuantisasi: Kami mengevaluasi berbagai aspek dan kinerja dari model-model yang telah dikuantisasi, termasuk pengurangan

ukuran model, pengurangan konsumsi sumber daya setelah kuantisasi, perubahan akurasi dari model yang telah dikuantisasi, serta pengurangan latensi.

Bagian-bagian selanjutnya dari makalah ini disusun sebagai berikut: Bab II menjelaskan latar belakang PTQ pada perangkat *edge*. Bab III menyediakan penjelasan mengenai cara kerja pengukuran kami. Bab IV melaporkan hasil evaluasi dan karakterisasi kami mengenai proses PTQ di perangkat, serta kinerja dan perilaku dari model DL yang telah dikuantisasi. Terakhir, Bagian V menyimpulkan makalah ini.