BAB II

TINJAUAN PUSTAKA

Bagian ini mencantumkan beberapa referensi penelitian sebelumnya pada bidang Federated Learning. Beberapa aspek dari penelitian sebelumnya menunjang penelitian ini. Inti dari setiap jurnal yang menjadi referensi akan dibahas terlebih dahulu, lalu setiap aspek dari masing-masing jurnal yang mempengaruhi penelitian akan dibahas pada bagian akhir. Pembahasan menjadi pengantar ke bagian selanjutnya yang menggambarkan perancangan sistem.

2.1 Penelitian Terdahulu

Terdapat beberapa jurnal terkait Federated Learning yang telah dipublikasikan dan menjadi acuan untuk menerapkan Federated Learning pada bidang Natural Language Processing.

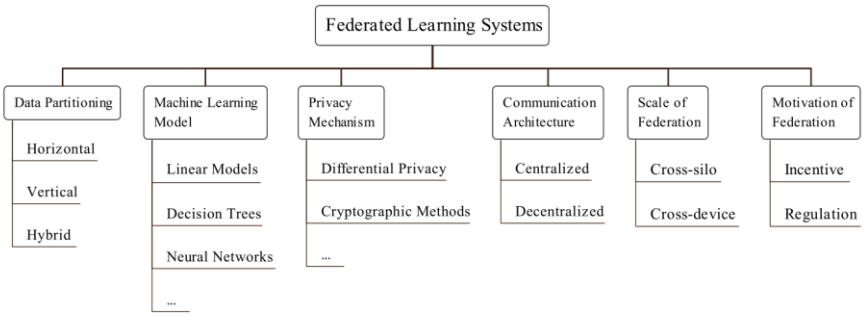
2.1.1 **Federated Learning Versus Classical Machine Learning: Convergence Comparison** [5]

Jurnal yang ditulis oleh Muhammad Asad, Ahmed Moustafa, dan Takayuki Ito memberikan komparasi Classical ML dengan Federated Learning. Classic ML yang didefinisikan pada jurnal mendefinisikan sebuah Machine Learning yang melibatkan pelatihan secara tersentralisasi di mana data dikumpulkan pada satu tempat dan dilatih pada *central server*. Meskipun memberi hasil yang relatif baik, proses pelatihan tersentralisasi memberikan ancaman privasi ke data para *user* karena dikirim dan dikumpulkan pada *cloud server*. Cara yang bisa digunakan untuk melatih ML dan menjaga privasi *user* adalah dengan teknik Federated Learning. Federated Learning memperbolehkan partisipan untuk secara kolaboratif melatih model lokal dengan data lokal tanpa memberikan informasi sensitif mengenai dirinya. Dalam eksekusi algoritma ML tradisional, data yang digunakan untuk pelatihan tersentralisasi biasanya merupakan data yang *independent and identically distributed* (*iid*). Data bersifat *iid* yang biasanya digunakan untuk melatih ML tradisional tidak saling

bergantung namun memiliki properti yang sama. Sementara pada Federated Learning, data bersifat *non-iid* dimana antar *user* memiliki data yang berbeda-beda dan satu *user* memiliki data yang sangat *dependent*. ML tradisional memiliki data yang secara adil terdistribusi, sementara pada Federated Learning data antar *user* bisa memiliki jumlah yang berbeda dan jumlah partisipan bisa memiliki jumlah yang sangat banyak.

Jurnal membandingkan tiga macam pelatihan ML yaitu: Centralized ML, Distributed ML, dan Federated Learning. Pada Centralized ML, partisipan terkoneksi ke server untuk menunggah data dan *server* yang akan melakukan seluruh tugas komputasi untuk melatih model. Cara tersentralisasi memiliki efisiensi yang baik bagi partisipan secara *resource* komputasi. Namun *user data* memiliki ancaman privasi. Pada Distributed ML, model dilatih dengan metodologi yang sama dengan cara tersentralisasi, namun pelatihannya dilakukan secara terpisah ke berbagai partisipan. Cara ini dilakukan untuk melakukan algoritma kompleks pada *dataset* berskala besar. Pada proses pelatihan, partisipan secara *independent* melatih model dan mengirimkan *weight updates* ke *central server*. Pembaharuan *weight* diterima dan *central server* melakukan *averaging* untuk *output*. Mirip dengan Distributed ML, Federated Learning juga melatih model secara independen, namun setiap partisipan memulai pelatihan secara sendiri seperti tidak ada partisipan lain dalam jaringan. Setelah beberapa *epochs* lokal pada suatu partisipan, *updates* akan dikirim ke *central server* dan melakukan agregasi untuk membentuk *global model* selanjutnya. Dari *global model* ini, partisipan mengeksekusi ronde pelatihan selanjutnya menggunakan model tersebut. Hasil eksperimen jurnal menyatakan bahwa Federated Learning merupakan solusi terbaik dalam melatih model ketika dihadapkan dengan *constraint* seperti jumlah partisipan yang terbatas dan juga merupakan cara yang terbaik digunakan untuk melatih model di era yang mementingkan privasi dan keamanan.

2.1.2 **A Survey on Federated Learning Systems: Vision, Hype, and Reality for Data Privacy and Protection** [8]



Gambar 2. 1 Taksonomi Federated Learning

Jurnal yang ditulis oleh Qinbin Li, Zeyi Wen, Zhaomin Wu, dan peneliti lainnya memberikan gambaran mengenai pengertian *federated learning* dan komponen yang ada pada sistem *federated learning*. Gambar 2.1 menunjukkan enam aspek kategori Federated Learning System yang berbeda: *Data distribution, machine learning model, privacy mechanism, communication architecture, scale of federation, and motivation of federation*. Peneliti jurnal meringkaskan sistem *federated learning* secara sistematis untuk menunjang peneliti lain yang ingin ikut melakukan riset. Jurnal mendefinisikan Federated Learning System sebagai sistem yang mencakup berbagai macam bidang seperti *distributed system*, *machine learning*, dan *privacy*.

2.1.3 **Federated Learning Meets Natural Language Processing: A Survey** [9]

Jurnal yang ditulis oleh Ming Liu, Stella Ho, Mengqi Wang, dan peneliti lainnya membahas mengenai berbagai macam tantangan yang ada dalam menerapkan pelatihan secara *federated* kepada bidang Natural Language Processing (NLP). Tantangan yang disinggung oleh jurnal ini adalah *algorithm challenges, system challenges,* dan *privacy issues*. Jurnal juga membahas tentang metode dan alat untuk melakukan evaluasi hasil pelatihan secara *federated* yang ada. Beberapa *task* dalam NLP yang bisa dilakukan secara *federated* meliputi *language modeling, text classification, speech recognition, sequence tagging, recommendation, health text mining,* dan *task* lainnya.

Menurut jurnal, performa yang dihasilkan dengan melakukan pelatihan secara *federated* menghasilkan performa yang masih lebih rendah dibanding dengan pelatihan yang dilakukan dengan data tersentralisasi.

2.1.4 **Benchmarking PySyft Federated Learning Framework on MIMIC-III Dataset** [10]

Jurnal yang ditulis oleh Andrius Budrionis, Magda Miara, Piotr Miara, dan peneliti lainnya membahas tentang hasil *benchmarking* PySyft. Terdapat tiga macam skenario eksperimen yang dilakukan dengan memainkan ketiga konstan: Data, Nodes, dan Distribution. Metrik yang digunakan untuk mengukur performa adalah ROC AUC dan F1 Score. Eksperimen pertama memiliki jumlah *node* yang konstan berjumlah 32, sementara datanya terus bertambah dan terdistribusi secara seragam. Semakin banyak jumlah data, maka performa model semakin baik dan semakin mirip dengan performa jika dilatih secara *centralized*. Eksperimen kedua membuat jumlah data konstan, namun jumlah *node* terus ditingkatkan, dan data terdistribusi secara seragam. Dari hasil eksperimen kedua, jumlah *node* atau *virtual workers* memiliki sedikit pengaruh ke performa model. Pengaruh jatuhnya performa yang signifikan terlihat ketika meningkatkan jumlah *node* ke maksimal. Hal ini menyatakan bahwa dengan jumlah *node* yang banyak dan setiap *node* memegang jumlah data sedikit bukan cara yang optimal untuk melatih model dengan performa terbaik. Eksperimen ketiga membuat jumlah data dan *node* konstan, namun cara distribusi data dibeda-bedakan. Eksperimen ketiga menyatakan bahwa cara distribusi data tidak mempengaruhi performa model secara signifikan. Durasi *training* dan *inference* dihitung dalam hitungan detik. Dari hasil ketiga eksperimen, ditemukan bahwa performa model Federated tidak dipengaruhi banyak oleh distribusi data yang tidak seimbang. Hal yang perlu diperhatikan pada Federated Learning adalah turunnya efisiensi pelatihan model terjadi karena pengaturan *node*, *privacy preservation*, dan beberapa langkah lain yang tidak ada pada pelatihan secara *centralized*.

2.1.5 **Federated Learning from Small Datasets** [7]

Jurnal yang ditulis oleh Michael Kamp, Jonas Fischer, dan Jilles Vreeken membahas tentang bagaimana meningkatkan akurasi model yang dilatih secara Federated Learning menggunakan ukuran *dataset* yang kecil. Terdapat tiga macam cara Federated Learning yang telah dilakukan sebelum jurnal ini ditulis yaitu Federated Learning biasa yang menggunakan algoritma FedAvg untuk memperbaharui global model [6], FedProx yang mengubah algoritma averaging milik FedAvg [11], dan Gossip Federated Learning [12] yang tidak melakukan agregasi secara centralized sehingga disebut jurnal sebagai Decentralized Federated Learning. Jurnal ini melakukan penyederhanaan metode Federated Learning milik [7] dan menamai metodenya simple daisy-chaining. Teknik daisy¬-chaining juga diterapkan oleh jurnal ke metode Federated Learning yang menggunakan averaging dan menamainya FedDC. FedDC melakukan agregasi model seperti Federated Learning biasa, namun lanjut melakukan perputaran model antar worker layaknya cara daisy-chaining. Pada jurnal ini, peneliti jurnal membandingkan performa model berdasarkan jenis *dataset* yang terdiri dari: CIFAR10, MRI, Pneumonia dan berdasarkan jenis model yang terdiri dari: FedDC, Daisy-chaining, FedAvg dengan aggregation period 10, FedAvg dengan aggregation period 1, dan FedProx. Hasil yang didapatkan dapat dilihat pada Gambar 2.2.

dataset	FedDC	Daisy-Chaining	FedAvg(b=10)	FedAvg(b=1)	FedProx
CIFAR10	**62.8**	59.2	51.0	56.3	54.5
MRI	**78.4**	57.7	75.6	74.1	76.5
Pneumonia	**82.5**	78.8	79.0	78.8	79.7.0

Gambar 2. 2 Gambar Tabel Perbandingan Akurasi

Sumber: Federated Learning From Small Datasets [7, p. 8]

Dari hasil percobaan peneliti, FedDC milik peneliti yang melakukan daisy-chaining memiliki akurasi paling tinggi dibanding model lainnya. Metode daisy-chaining biasa juga menghasilkan akurasi yang sebanding dengan model lain.

2.2 Tinjauan Teori

Berikut tinjauan beberapa teori yang digunakan dalam mengerjakan Tugas Akhir, sebagai berikut.

2.2.1 Horizontal Data Partitioning

Data pada Federated Learning dapat dibagi menjadi tiga berdasarkan bagaimana data terdistribusi dilihat dari *sample* dan *feature space*: Horizontal, Vertical, dan Hybrid. Data horizontal memiliki *feature space* yang serupa, namun sedikit memiliki *sample space* yang mirip [8, p. 7]. Sebagai contoh, beberapa kalimat memiliki jumlah kata yang hampir sama, namun makna dari kalimat tersebut belum tentu sama. Partisi data secara horizontal adalah tipe yang sering digunakan pada penelitian Federated Learning. Beberapa pihak dapat melakukan pelatihan secara lokal dengan arsitektur model yang sama. *Global model* hanya melakukan *averaging* dari seluruh model lokal. Pengumpulan data tersebut dilakukan dengan *framework* FedAvg. Penelitian yang akan dilakukan mengambil data dengan tipe partisi horizontal dan menerapkan algoritma Federated Averaging. Federated Averaging adalah algoritma yang efisien dalam melakukan komunikasi antar *client* dengan jumlah yang banyak. *Client* tetap dapat menyimpan data lokal secara aman, sementara sebuah *central server* akan mendistribusi parameter ke seluruh *client* dan mendapatkan kembali parameter yang telah diperbaharui dari *client*.

2.2.2 Gated Recurrent Unit Machine Learning Model

Tidak semua model *machine learning* dapat di-train secara *federated*. Model *machine learning* paling populer yang dapat menghasilkan hasil *state-of-the-art* adalah Neural Network (NN). Cara paling populer untuk mengoptimasi NN secara *federated* adalah Stochastic Gradient Descent (SGD) [8, p. 8]. Model *machine learning* yang akan digunakan adalah Gated Recurrent Unit (GRU). GRU merupakan penyederhanaan dari model Long Short-Term Memory (LSTM) yang merupakan NN dengan tipe Recurrent Neural Network (RNN). GRU memiliki parameter yang lebih sedikit dibanding LSTM sehingga lebih ringan dan lebih cepat dibanding LSTM. Dikarenakan GRU lebih ringan dari LSTM, model ini lebih cepat waktu pelatihannya jika dilatih di perangkat

keras semacam *smartphone*. GRU memiliki dua *vector* yang menentukan apakah informasi diteruskan atau dihapus yaitu Update Gate and Reset Gate. Update Gate membuat model dapat menentukan seberapa banyak informasi sebelumnya yang akan diteruskan. Update Gate bisa meneruskan seluruh informasi sebelumnya sehingga meminimalisir terjadinya permasalahan *vanishing gradient*. Reset Gate menentukan seberapa banyak informasi sebelumnya yang akan dilupakan atau dihapus. [13]

2.2.3 Centralized Communication Architecture

Terdapat dua macam cara untuk berkomunikasi pada sistem Federated Learning yaitu Centralized Design dan Decentralized Design. Cara komunikasi yang akan dilakukan adalah Centralized Design. Pada cara ini, sebuah *central server* bertindak sebagai *manager* yang akan mengumpulkan informasi dari berbagai pihak *client*. Pembaharuan parameter *global model* akan selalu dilakukan pada *manager*. Komunikasi *manager* dengan *client* dapat dilakukan secara *synchronous* atau *asynchronous*. Algoritma Federated Averaging yang akan dilakukan untuk melakukan agregasi merupakan tipe komunikasi yang akan melakukan pengumpulan parameter dari *client* secara *synchronous* dan berusaha meminimalisir jumlah *communication rounds*. Setiap *communication round* memakan waktu secepat kecepatan maksimum perangkat terlambat yang berpartisipasi. Proses pelatihan melakukan beberapa ronde pelatihan secara lokal. Setiap ronde lokal telah dilakukan untuk jumlah yang telah ditetapkan, lalu parameter dikirim dan digunakan untuk memperbaharui *global model* [8, p. 9].

2.2.4 Scale of Federation

Penerapan sistem Federated Learning yang akan dilakukan pada penelitian ini adalah Cross-Device FL dimana terdapat dua *client* yang memiliki jumlah data yang sama. Jumlah data dibagi rata untuk dua *client* dikarenakan jumlah data yang terbatas. Kedua *client* tersebut berbentuk sebuah *virtual worker* yang bekerja pada satu mesin yang sama, namun secara logika seperti mesin yang terpisah. Pembuatan *virtual worker*, pembagian *dataset*, dan simulasi pelatihan secara Federated disediakan oleh *framework* PySyft.

2.2.5 **Summary**

Dari jurnal yang telah dibahas kesimpulan yang dapat dibuat adalah:

1) Jurnal [5] memberi pengetahuan bahwa Federated Learning memiliki performa yang bisa mengalahkan cara Centralized ML dan Distributed ML. Federated Learning juga cocok digunakan untuk melatih AI di era yang mementingkan privasi.
2) Jurnal [8] memberikan gambaran mengenai apa saja komponen yang ada dalam Federated Learning dan apa saja yang perlu diperhatikan ketika merancang sistem Federated Learning. Komponen yang dibahas adalah: bagaimana jenis distribusi data, jenis model yang dilatih, mekanisme privasi, komunikasi arsitektur, skala federasi, dan motivasi federasi.
3) Jurnal [9] memberikan gambaran mengenai bagaimana menerapkan NLP pada Federated Learning beserta tantangan yang ada. Jurnal mengatakan bahwa *text classification* dapat dilakukan secara Federated Learning. *Tools* dan *framework* yang disarankan ditulis dalam jurnal. Penelitian ini menggunakan PySyft. Model GRU juga disebut oleh jurnal cocok pada tata cara Federated Learning.
4) Jurnal [10] memberikan pengetahuan bahwa dari ketiga komponen berupa jumlah data, node, dan jenis distribusi data, jumlah data merupakan *variable* yang paling berpengaruh ke performa model. Semakin banyak jumlah data maka semakin bagus performa modelnya. Performa model diukur dengan ROC AUC dan F1 Score. Penelitian ini menerapkan ROC AUC untuk mengukur performa model pengklasifikasi.
5) Jurnal [7] memberikan hasil perbandingan akurasi dari model-model Federated Learning yang umum dipakai. Dari jurnal tersebut, penulis ingin mencoba melatih model secara Federated Learning dengan agregasi *averaging* dan Federated Learning tanpa agregasi *averaging* yang memakai teknik *daisy-chaining*.