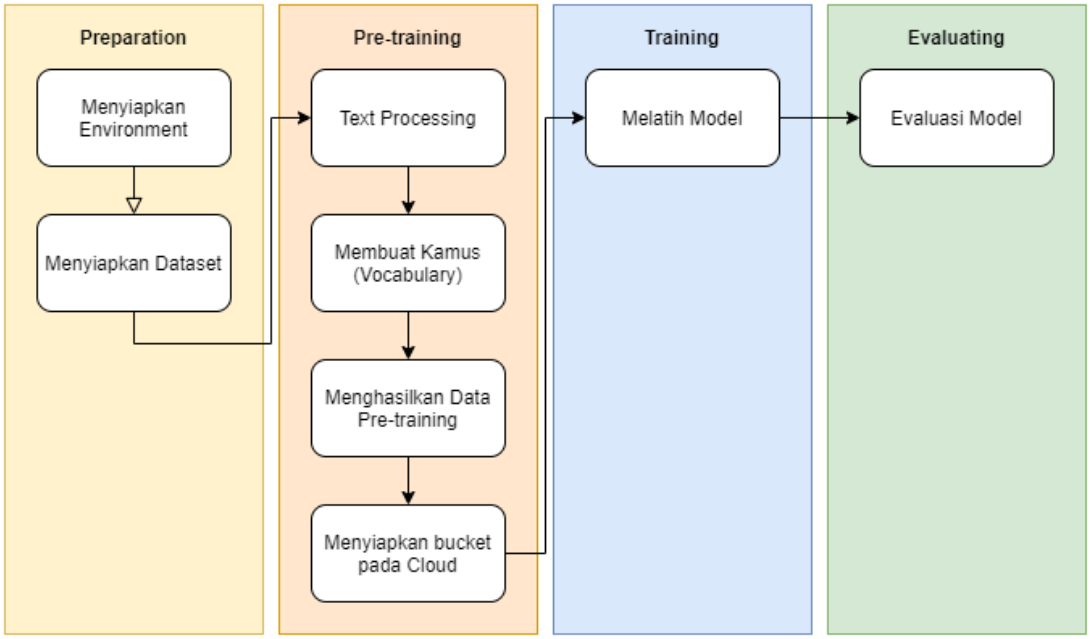
BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1. Metodologi Penelitian

Pada penelitian ini, terdapat sebanyak empat fase yang harus dilakukan dalam pemodelan, yakni; persiapan (*preparation*), pra-latih (*pre-training*), pelatihan model (*training*), dan yang terakhir adalah evaluasi. Terdapat beberapa tahap di setiap fase yang akan dilakukan. Diagram alur dari prosedur pemodelan dapat dilihat pada Gambar 3 (*flowchart*) sebagai gambaran umum proses pembentukkan model.



Gambar 14. Flowchart Tahap Perancangan Model

Untuk pengumpulan data, data yang diperoleh untuk penelitian ini di dapat dari *open corpus* yang tersedia dalam bahasa Indonesia. Ukuran *corpus* yang digunakan

adalah sebesar 12.389.523 baris kalimat. Model akan diprogram dan diimplementasikan sesuai dengan rancangan yang telah disusun pada tahap-tahap yang tertera pada diagram alir. Fungsi-fungsi maupun metode yang akan dibutuhkan dalam pembuatan model akan dikerjakan pada tahap pemrograman dengan menggunakan teknik model Bahasa BERT dan *open-source library* TensorFlow dan *library* lainnya, dengan berbasis bahasa pemrograman Python. Selama proses pemrograman model akan dilakukan *testing* dan *debugging* untuk memastikan perancangan dan pembentukkan model terpenuhi sesuai perencanaan.

3.2. Perancangan Model

Beberapa konfigurasi dari *hyperparameter* disesuaikan dengan kebutuhan *training* model berdasarkan analisis yang didapatkan. *Hyperparameter* tersebut antara lain seperti berikut [23]:

```
{
  "attention_probs_dropout_prob": 0.1,
  "directionality": "bidi",
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "max_position_embeddings": 512,
  "num_attention_heads": 12,
  "num_hidden_layers": 12,
  "pooler_fc_size": 768,
  "pooler_num_attention_heads": 12,
  "pooler_num_fc_layers": 3,
  "pooler_size_per_head": 128,
  "pooler_type": "first_token_transform",
  "type_vocab_size": 2,
  "vocab_size": 32000
```

}

3.2.1. Menyiapkan *environment*

Pada tahap ini peneliti akan menyiapkan *environment* yang dibutuhkan dalam seluruh rangkaian perancangan model. Hal-hal tersebut antara lain adalah; meng-*import libraries* yang akan dibutuhkan seperti OS, ntlk, *logging*, TensorFlow, SentencePiece, BERT, dan lainnya.

3.2.2. Menyiapkan *dataset*

Data yang dijadikan sebagai *training dataset* akan dicari dan dikumpulkan dari beberapa sumber *corpus* yang *open-source*. *Corpus* tersebut diberi Batasan seberapa besar *corpus* yang diinginkan sebagai *training dataset*.

3.2.3. Memproses Teks

Dataset yang sudah didapat dalam bentuk teks akan dilakukan normalisasi teks. Normalisasi tersebut dapat berupa seperti mengubah seluruh huruf kapital menjadi huruf kecil, menghapus karakter non-UTF, menghapus *punctuation* dengan bantuan *regex*, dan proses lainnya.

3.2.4. Membuat Kamus

Pada tahap ini peneliti menggunakan SentencePiece dalam pembuatan kamus (*vocabulary*). Kamus yang sudah dibuat merupakan pembantu model dalam melakukan tokenisasi sebelum melakukan *training model* atau *fine-tuning pre-trained*

model. Kemudian, akan diberikan control symbol yang diharuskan oleh arsitektur BERT dan disisipkan di bagian awal kamus. Simbol kontrol tersebut antara lain adalah; [PAD], [UNK], [CLS], [SEP], dan [MASK]. Setelah kamus dibuat, kamus tersebut akan diekspor ke dalam sebuah teks *file*.

3.2.5. Menghasilkan Data Pre-*training*

Dalam melakukan *pre-training*, *dataset* yang akan digunakan akan memiliki ukuran yang besar. Untuk mengoptimalisasi dan mencegah adanya kegagalan, *dataset* tersebut dipecah menjadi beberapa bagian *dataset*. Tiap bagian-bagian tersebut akan memanggil skrip `create_pretraining_data.py`. Untuk melakukan *batch running script*, penulis akan menggunakan `xargs` *command*.

3.2.6. Menyiapkan Penyimpanan pada GCP Storage

Agar dalam pelatihan model dapat berjalan secara dinamis, maka penyimpanan hasil *training* tiap *steps*-nya akan disimpan ke dalam Google *Cloud Storage Bucket* dalam bentuk *checkpoints*. Jika terjadi kegagalan *kernel* atau lainnya, *training* model akan tetap bisa dilanjutkan dari *checkpoint* terakhir yang tersimpan di dalam *bucket* Google Cloud Storage.

3.2.7. Melatih Model

Pada tahap ini model dapat mulai dilatih dengan konfigurasi yang sebelumnya telah ditetapkan, konfigurasi fungsi model *builder* dan *input builder* milik

BERT, dan konfigurasi *estimator* TensorFlow. Dalam melatih model akan digunakan TPU *Cloud* yang terdapat pada Google *Cloud*.

3.3. Evaluasi Model

Model akan diuji dengan beberapa *downstream tasks* yang akan diberlakukan juga pada BERT *pre-trained* multilingual dan kemudian dibandingkan. Model dilakukan *fine-tuning* ke pekerjaan yang lebih spesifik yaitu *text classification*. *Fine-tuning* model akan dilakukan dengan menggunakan *dataset* yang memiliki label biner dan berbahasa Indonesia.

3.3.1. Pengumpulan Data Berlabel

Dataset berlabel yang digunakan sebagai bahan pengujian pada *fine-tuning model* Bahasa BERT dan BERT Multilingual, terdapat dua jenis *dataset* berbeda. *Dataset* pertama merupakan *dataset* yang di dapat dari situs researchgate.net, yang berisi sebanyak 553 data *App Review* Playstore. Pada *dataset* ini penulis hanya mengambil data yang berlabel positif dan negatif, dengan total data sebanyak 500 baris kalimat. *Dataset* kedua merupakan *dataset* yang didapat dari perusahaan yang tidak akan disebut Namanya. *Dataset* tersebut merupakan *dataset free text* sebanyak 160 ribu baris kalimat dengan label biner.