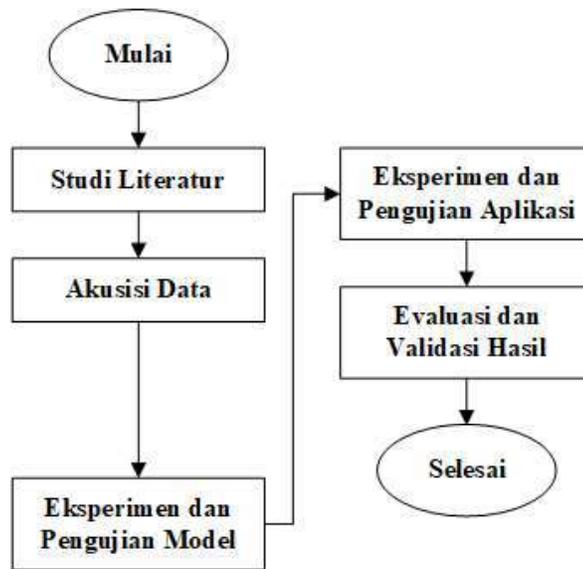


BAB III METODOLOGI PENELITIAN

3.1. Desain Penelitian

Desain penelitian merupakan skema tahapan penelitian yang menjadi acuan dalam melakukan proses penelitian. Pada desain penelitian terdapat tahapan-tahapan yang bertujuan untuk mempermudah penyelesaian masalah yang ada. Pada penelitian ini terdapat tahapan yang akan dilakukan yaitu dapat dilihat pada Gambar 6.



Gambar 3.1. Diagram Alur Penelitian

Diagram alur penelitian pada Gambar 3.1 terdapat tahapan yaitu studi literatur, akuisisi data, praproses data, eksperimen dan pengujian model, eksperimen dan pengujian aplikasi, evaluasi dan validasi hasil. Tahapan studi literatur bertujuan untuk mengumpulkan informasi terkait penelitian yang akan dilakukan. Tahapan akuisisi data merupakan tahapan mengumpulkan data berupa citra teks yaitu cetak huruf dan angka hasil akuisisi menggunakan kamera ponsel pintar Android. Tahapan eksperimen dan pengujian model merupakan tahapan pengolahan data yang sudah dilakukan praproses data. Tujuannya adalah untuk melakukan proses pembuatan model sehingga dapat digunakan dalam penelitian ini. Tahap eksperimen dan pengujian aplikasi merupakan tahapan implementasi dari model

yang telah dihasilkan sebelumnya. Kemudian tahap evaluasi dan validasi hasil yaitu tahapan yang bertujuan untuk mengukur tingkat akurasi hasil yang didapatkan dari tahapan sebelumnya.

3.2. Studi Literatur

Pada tahap studi literatur dilakukan tinjauan pustaka dan tinjauan studi. Tinjauan pustaka dilakukan untuk mengumpulkan teori-teori yang berkaitan dalam mendukung penelitian ini. Tinjauan studi dilakukan untuk komparasi terhadap penelitian-penelitian yang berkaitan dalam penggunaan metode penelitian ini. Penjelasan studi literatur terdapat pada Bab 2.

3.3. Akuisisi Data

Tahap akuisisi data merupakan tahap yang dilakukan untuk mendapatkan data citra yang mengandung teks guna membantu proses pelatihan dan pengujian pada penelitian ini. Data berupa citra teks dibutuhkan dalam penelitian, maka dari itu dilakukannya akuisisi data. Data masukan berupa citra teks cetak huruf dan angka dengan karakter pada teks cetak meliputi huruf cetak A-Z (A B C D E F G H I J K L M N O P Q R S T U V W X Y Z), a-z (a b c d e f g h i j k l m n o p q r s t u v w x y z), dan angka 0-9 (0 1 2 3 4 5 6 7 8 9). Terdapat dua cara dalam melakukan akuisisi, yaitu membangun data citra sintetis dan *capture* data citra natural.

Data yang pertama adalah kombinasi dari berbagai gambar alami hasil *capture* yang berisi teks, sedangkan yang kedua citra terdiri dari karakter yang dibuat secara sintetis. Set data sintetis dibuat untuk mengimbangi kurangnya ketersediaan data citra natural. Dengan demikian, kumpulan data sintetis digunakan untuk proses pelatihan model, sedangkan kumpulan data citra natural digunakan untuk pengujian aplikasi. Kemudian set data citra akan dibagi ke dalam dua kelompok yaitu subset data citra *training* dan subset data citra *testing*. Secara berurutan komposisi pembagiannya antarlain jumlah data sebanyak 15000 dataset citra dengan 90% data citra merupakan *training* dan 10% data citra *testing* dengan detail yang telah disebutkan pada subbab sebelumnya yaitu ruang lingkup penelitian.

3.3.1 Data Citra Sintetis

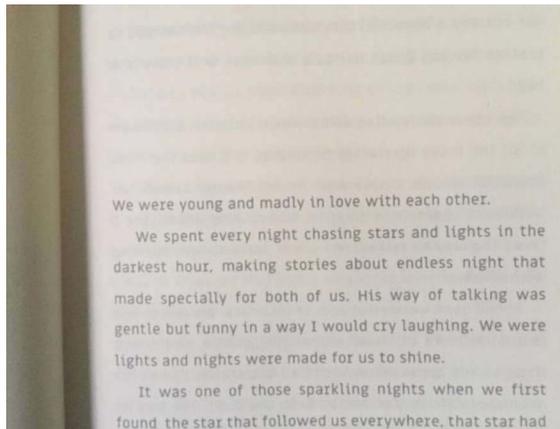
Akuisisi data citra dilakukan dengan sebuah fungsi yang dapat membangun himpunan data untuk digunakan dalam proses pelatihan model yang digunakan. Fungsi ini akan menghasilkan citra teks acak dengan berbeda ketebalan dan *font*. Berikut ini merupakan contoh dari hasil fungsi *generate dataset* pada Gambar 3.2.

1 2 3 4 5 6 7 8 9 0
A B C D E F G H I K L M N O Q R S T U V W X Y Z
a b c d e f g h i k l m n o q r s t u v w x y z

Gambar 3.2. Contoh Hasil *Generate Dataset*.

3.3.2 Data Citra Natural

Akuisisi data citra dilakukan dengan memotret (*capture*) citra teks melalui kamera mobile. Data citra yang menjadi masukan adalah citra *structured text* pada buku atau dokumen fisik yaitu dengan format citra teks penuh tanpa gambar dan berekstensi JPG (*.jpg) yang merupakan *default* format dari kamera mobile. Berikut ini merupakan contoh dari hasil *capture* data citra menggunakan kamera mobile pada Gambar 3.3.



Gambar 3.3. Contoh Akuisisi Citra Natural.

3.4. Eksperimen dan Pengujian Model

3.5.1. Pelatihan Dataset

Pada tahap ini akan dilakukan proses *training* data, data citra dikelompokkan menjadi dua bagian yaitu data *training* dan data *testing*. Tujuan dari proses *training*

ini adalah untuk menemukan ciri (*feature*) dari setiap citra dan dengan menandai *neuron* yang akan diaktifkan ketika citra tersebut diklasifikasikan.

3.5.2. Pengujian Model

Pada tahap pengujian model dilakukan dengan menggunakan data *testing*. Proses ini akan memprediksi kelas dari data *testing*, setiap data *testing* akan dihitung jumlah ketepatan berhasil diprediksi dan jumlah yang tidak tepat diprediksi. Pada Gambar 9 merupakan alur dari struktur jaringan LSTMs yang digunakan untuk mendeteksi setiap data *testing*.

Struktur jaringan LSTMs digunakan dalam tahapan *testing* yang pada Gambar 13. Jaringan LSTMs terdiri dari blok memori berbeda yang disebut *cell* (persegi panjang yang lihat pada gambar). Ada dua *state* yang akan ditransfer ke *cell* berikutnya yaitu *state cell* dan *hidden state*. Blok memori bertanggung jawab untuk mengingat sesuatu dan manipulasi terhadap memori yang dilakukan melalui tiga mekanisme, yang disebut *gate*. Ketiga mekanisme tersebut adalah :

1. *Forget Gate*, bertanggung jawab untuk menghapus informasi dari *cell state*. Informasi yang tidak lagi diperlukan LSTM atau informasi yang kurang penting dihapus melalui *filter*. *Gate* ini menerima dua masukan yaitu h_{t-1} dan x_t . h_{t-1} adalah *hidden state* dari *cell* sebelumnya atau keluaran dari *cell* sebelumnya dan x_t adalah masukan pada langkah waktu tertentu. *Input* yang diberikan dikalikan dengan matriks bobot dan bias ditambahkan. Kemudian, fungsi sigmoid diterapkan ke nilai ini. Fungsi sigmoid mengeluarkan vektor, dengan nilai mulai dari 0 hingga 1, sesuai dengan setiap angka dalam status sel. Pada dasarnya, fungsi sigmoid bertanggung jawab untuk memutuskan nilai mana yang akan disimpan dan yang akan dibuang. Jika '0' adalah keluaran untuk nilai tertentu dalam status sel, berarti *forget gate* menginginkan status sel untuk sepenuhnya melupakan bagian informasi tersebut. Demikian pula sebaliknya jika '1'. Keluaran vektor dari fungsi sigmoid ini dikalikan dengan *cell state*.
2. *Input Gate* bertanggung jawab atas penambahan informasi ke status sel.
 - Mengatur nilai apa yang perlu ditambahkan ke *cell state* dengan melibatkan fungsi sigmoid. Ini pada dasarnya sangat mirip dengan

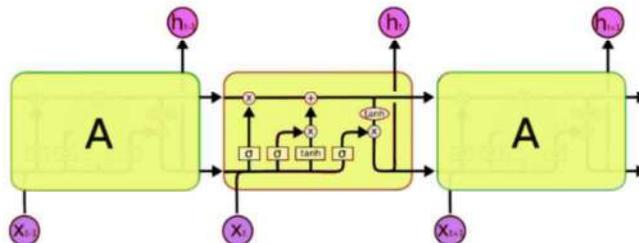
gerbang lupa dan bertindak sebagai filter untuk semua informasi dari h_{t-1} dan x_t .

- Membuat vektor yang berisi semua kemungkinan nilai yang dapat ditambahkan ke *cell state*. Ini dilakukan dengan menggunakan fungsi tanh, yang mengeluarkan nilai dari -1 hingga +1.
- Mengalikan nilai filter (gerbang sigmoid) ke vektor yang dibuat (fungsi tanh) dan kemudian menambahkan informasi yang berguna ini ke *cell state* melalui operasi penjumlahan.

Setelah proses tiga langkah ini selesai, kami memastikan bahwa hanya informasi yang ditambahkan ke *cell state* yang penting dan tidak berlebihan.

3. *Output Gate* berfungsi untuk memilih informasi yang berguna dari *cell state* saat ini dan menampilkannya sebagai keluaran.

- Membuat vektor setelah menerapkan fungsi tanh ke *cell state*, sehingga menskalakan nilai ke rentang -1 hingga +1.
- Membuat filter menggunakan nilai h_{t-1} dan x_t , sehingga dapat mengatur nilai yang perlu di-output dari vektor yang dibuat di atas.
- Mengalikan nilai filter pengaturan ini ke vektor yang dibuat pada langkah sebelumnya, dan mengirimkannya sebagai keluaran dan juga ke *hidden state* berikutnya.



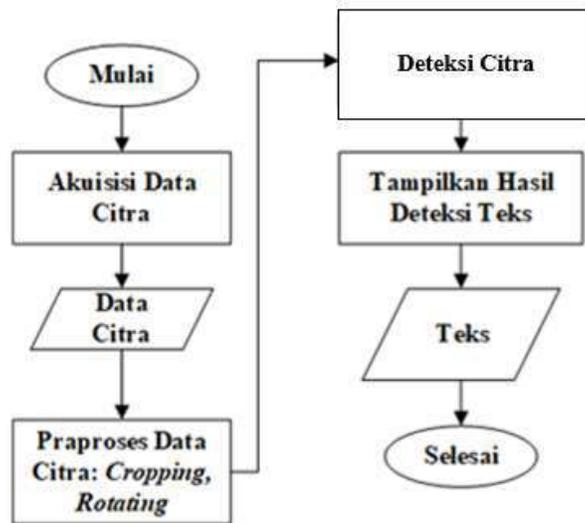
Gambar 3.4. Model LSTM - Alur Struktur Jaringan[12].

3.5. Eksperimen dan Pengujian Aplikasi

3.6.1. Rancangan Aplikasi

Pada penelitian ini dibuat sebuah aplikasi yang mampu mengenali karakter dari citra teks cetak hasil *capture* oleh kamera pada perangkat Android dan hasil pengenalan teks tersebut kemudian dijadikan sebagai dokumen teks digital.

Secara keseluruhan diagram rancangan alur proses sistem secara keseluruhan yang akan dibuat pada penelitian ini dapat dilihat pada Gambar 3.5.



Gambar 3.5. Rancangan alur proses sistem secara keseluruhan.

Proses pengenalan karakter tulisan membutuhkan beberapa tahapan pengolahan citra. Secara umum dapat diperhatikan melalui Gambar 3.5, proses pengenalan karakter (OCR) pada citra teks meliputi akuisisi citra, pra-pengolahan citra, *feature extraction* dan pengenalan karakter dengan LSTM.

1. Akuisisi data citra, kertas yang berisi kalimat teks tulisan cetak yang ditangkap citraannya oleh kamera pada perangkat Android.
2. Tahap pra proses data citra dengan melakukan *cropping, rotating* jika diperlukan setelah melihat hasil akuisisi data citra. Tahap ini dilakukan untuk menangani dan mengondisikan data citra yang akan dideteksi.
3. Selanjutnya tahap deteksi teks dengan algoritma LSTM. Penerapan model dan pemrosesan data citra dilakukan dalam sistem Android secara langsung. Proses *feature extraction* dan pengenalan karakter. Untuk dapat mengenali karakter pada citra tulisan, karakter tulisan pada citra harus diekstraksi terlebih dahulu berdasarkan karakteristik setiap pola huruf tertentu dengan metode ekstraksi ciri kemudian ciri yang dihasilkan dapat menjadi pola acuan dalam pengenalan karakter (klasifikasi) proses.

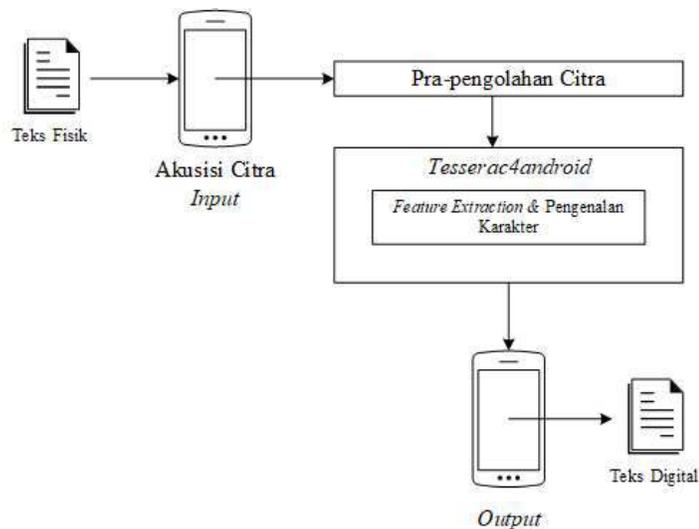
4. Hasil keluaran dari aplikasi ini adalah teks dalam bentuk digital.

3.6.2. Komponen Aplikasi

Komponen aplikasi merupakan kebutuhan-kebutuhan baik *software* maupun *hardware* dalam menjalankan aplikasi pada penelitian ini. Adapun komponen yang terkait antara lain :

1. Ponsel pintar android, merupakan perangkat sebagai media dalam memanfaatkan dan menggunakan aplikasi pada penelitian ini.
2. Tesseract4android, merupakan pustaka (*library*) perangkat lunak yang daapt digunakan melakukan pengenalan karakter .

Berikut Gambar 3.6 merupakan gambaran umum dalam pemanfaatan komponen pada aplikasi, menurut penjelasan sebelumnya.



Gambar 3.6. Komponen Aplikasi

3.6.3. Proses Konversi *Image to Text* Menggunakan Aplikasi

Pada tahap ini dilakukan proses konversi *image to text* menggunakan aplikasi sebagai media implementasi dan untuk menguji aplikasi yang sudah dihasilkan dari tahap perancangan aplikasi. Proses konversi dilakukan terhadap semua kelas yang ada, selama proses berlangsung setiap aspek akan dicatat seperti waktu yang dibutuhkan seberapa lama mulai dari pendeteksian citra hingga dengan hasil. Data dari hasil tersebut yang kemudian akan digunakan pada tahap evaluasi dan validasi hasil.

3.6.4. Skema Pengujian Fungsionalitas Aplikasi

Black-Box Testing adalah serangkaian pengujian pada aplikasi atau perangkat lunak di mana merupakan tes pada fungsionalitas dari aplikasi tersebut dengan tidak mengacu pada struktur internal. Aktor penguji tidak membutuhkan pengetahuan khusus terkait kode program pada aplikasi tersebut berikut dengan pengetahuan pemrograman[22]. Sehingga dapat disimpulkan bahwa *Black-Box Testing* merupakan pengujian yang berorientasi pada fungsionalitas yaitu perilaku dari perangkat lunak atas input yang diberikan pengguna sehingga mendapatkan/menghasilkan output yang diinginkan tanpa melihat proses internal atau kode program yang dieksekusi oleh perangkat lunak. Berikut tabel merupakan rancangan pertanyaan dalam menguji kondisi aplikasi.

Tabel 3.1. Daftar Pertanyaan Pengujian Aplikasi

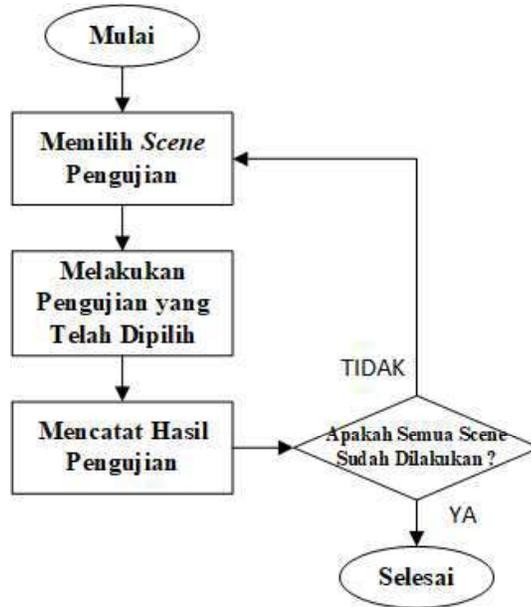
NO	Keterangan	Hasil	
		Sukses	Tidak
1	<i>Install</i> aplikasi pada ponsel pintar Android.		
2	Akuisisi data citra.		
Pra proses data citra			
3	<i>Cropping</i> data citra.		
4	<i>Rotating</i> data citra.		
5	Deteksi teks pada data citra.		
6	Menampilkan hasil deteksi teks.		

Pengujian dilakukan secara sederhana dan hanya dikerjakan oleh peneliti yang artinya merupakan pengujian mandiri. Adapun tahapan proses yang dilalui yaitu :

1. Melihat dan memilih kondisi pengujian pada daftar pertanyaan yang telah dirancang pada Tabel 3.1.
2. Setelah memilih maka dilakukannya pengujian aplikasi pada kondisi yang telah dipilih sebelumnya.
3. Selanjutnya memperhatikan hasil dari proses sebelumnya dan mencatat hasilnya.
4. Melihat daftar pertanyaan pada tabel jika sudah terjawab semua maka pengujian dapat dihentikan, jika masih terdapat pertanyaan yang belum

terjawab maka dapat mengulang dari proses 1 dan dilanjutkan hingga semua pertanyaan terjawab semua.

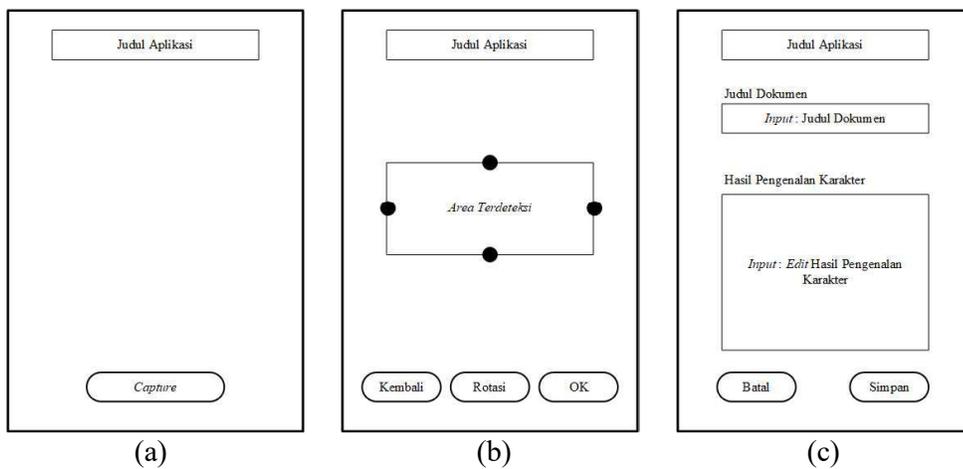
Berikut merupakan *flowchart* dari tahapan proses pengujian aplikasi dapat dilihat pada Gambar 3.7.



Gambar 3.7. *Flowchart* Tahapan Pengujian Aplikasi.

3.6.5. User Interface Design

Berikut adalah rancangan *user interface* dari aplikasi pada penelitian ini :



Gambar 3.8. Rancangan *User Interface Design*.
(a) camera module (b) sesuaikan citra (c) hasil pengenalan karakter.

Pada Gambar 3.8 bagian (a) yaitu tampilan dari *camera module* terdapat *button* untuk melakukan *capture* citra yang dideteksi. Setelah mengambil citra, aplikasi akan berpindah tampilan seperti pada bagian (b) dan merupakan tahap penyesuaian citra seperti mengubah area (*cropping*) yang akan dideteksi, mengubah rotasi citra sebanyak 90^0 untuk sekali tekan pada *button* Rotasi, *button* Kembali untuk mengulang *capture* citra (*discard*), dan *button* OK untuk menggunakan citra tersebut dan lanjut ke tahap selanjutnya. Pada bagian (c) merupakan tampilan hasil pendeteksian karakter pada citra, terdapat dua *input* yaitu Judul untuk dokumen dan *Edit* untuk melakukan perubahan pada hasil pendeteksian.

3.6. Evaluasi dan Validasi Hasil

Pengujian ini dilakukan untuk melakukan evaluasi terhadap sistem yang menggunakan model yang dihasilkan oleh LSTMs. Adapun metode pengukuran akurasi menggunakan perhitungan persentase dari prediksi huruf/kata yang benar. Hasil perhitungan pengukuran akurasi digunakan untuk mempertimbangkan evaluasi model, yaitu jumlah ketepatan prediksi yang dihasilkan kemudian nilai dari akurasi dinyatakan dengan persentase (%).