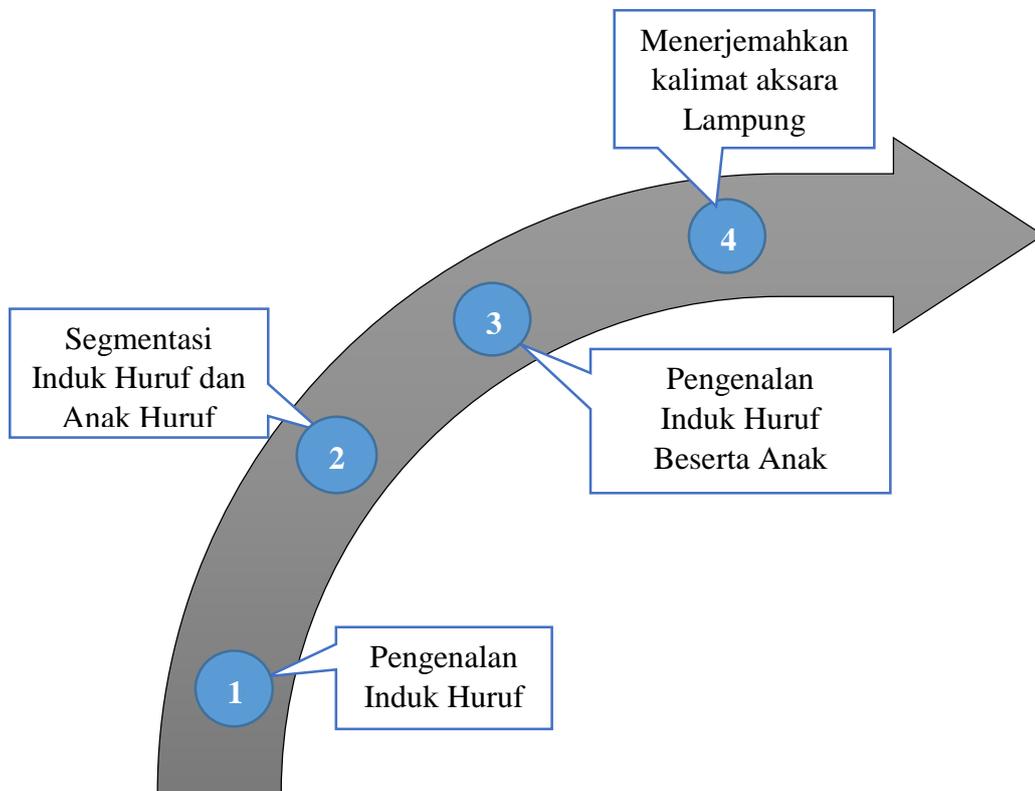


BAB III METODOLOGI PENELITIAN

3.1 *Road Map* Penelitian

Pengenalan aksara Lampung merupakan sebuah penelitian besar yang dapat dilakukan melalui beberapa tahap atau aktivitas, yang dalam penelitian ini digambarkan melalui sebuah *road map* penelitian. *Road map* penelitian menggambarkan berbagai aktivitas yang saling berkaitan dan berkelanjutan, yang dapat dilakukan dalam membangun model pengenalan aksara Lampung. Model pengenalan aksara Lampung yang dibangun dalam penelitian ini berada di tahap atau aktivitas yang pertama, yaitu mengenali induk huruf aksara Lampung. Berdasarkan hasil eksperimen yang telah dilakukan, model yang dibangun dalam penelitian ini, hanya dapat mengenali induk huruf aksara Lampung per karakter. Gambar 3.1 merupakan gambaran dari keseluruhan aktivitas atau tahap yang dapat dilakukan dalam membangun model pengenalan aksara Lampung.



Gambar 3.1. *Road Map* Penelitian.

Aktivitas pertama yang dapat dilakukan dalam membangun model pengenalan aksara Lampung adalah mengenali induk huruf aksara Lampung. Sampel citra induk huruf aksara Lampung yang berjumlah 20 huruf/kelas, digunakan pada proses pelatihan dan pengujian model, sehingga induk huruf aksara Lampung tersebut dapat dikenali oleh model yang dibangun. Setelah aktivitas pertama berhasil dilakukan, dapat memasuki aktivitas selanjutnya, yaitu segmentasi induk huruf dan anak huruf aksara Lampung.

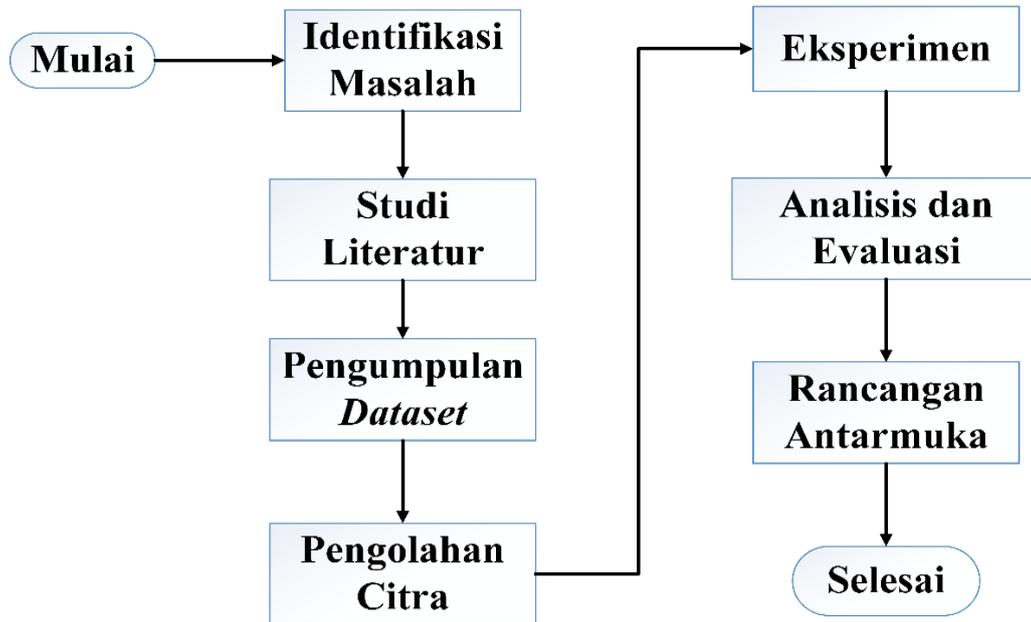
Segmentasi merupakan aktivitas yang dilakukan untuk mempartisi bagian-bagian dari aksara Lampung yang diberikan. Sebagaimana yang telah dijelaskan sebelumnya, aksara Lampung terdiri atas induk huruf dan anak huruf. Induk huruf aksara Lampung dapat dirangkai menjadi sebuah suku kata, kata, atau kalimat, apabila induk huruf tersebut disisipi dengan anak huruf. Agar model dapat mengenali induk huruf beserta anak huruf yang diberikan, perlu dilakukan proses segmentasi seperti dengan melakukan *scan* untuk menemukan blok-blok bagian dari induk huruf maupun anak huruf aksara Lampung.

Setelah model dapat mempartisi bagian-bagian dari aksara Lampung yang diberikan, selanjutnya dapat dilakukan proses pelatihan dan pengujian model kembali untuk mengenali induk huruf aksara Lampung yang telah disisipi anak huruf. Model yang telah berhasil mengenali induk huruf aksara Lampung beserta anak hurufnya, selanjutnya dapat memasuki aktivitas yang terakhir, yaitu pengenalan aksara Lampung dalam bentuk kalimat.

3.2 Diagram Alir Penelitian

Tahap awal yang dilakukan dalam penelitian ini adalah tahap identifikasi masalah, yaitu mencari, menemukan, dan mengumpulkan permasalahan yang berkaitan dengan penelitian. Selanjutnya dilakukan sebuah proses analisis masalah melalui studi literatur untuk mencari solusi dari permasalahan tersebut. Metode ini dilakukan dengan cara mencari referensi terkait penelitian seperti jurnal, buku, skripsi, *proceeding*, dan referensi lainnya. Kemudian dilanjutkan dengan tahap berikutnya seperti pengumpulan *dataset* berupa variasi induk huruf aksara

Lampung, pengolahan citra, eksperimen, analisis dan evaluasi, serta rancangan antarmuka. Diagram alir terkait tahap-tahap yang dilakukan pada penelitian ini dapat dilihat pada Gambar 3.2. berikut:



Gambar 3.2. Diagram Alir Penelitian.

Dalam rangka membangun sistem pengenalan induk huruf aksara Lampung, pada penelitian ini dilakukan 2 buah proses, yaitu proses membangun model dan proses membangun antarmuka. Proses membangun model dilakukan dengan beberapa tahap, antara lain pengumpulan *dataset*, pengolahan citra (*dataset*), eksperimen, dan analisis serta evaluasi hasil eksperimen. Sedangkan untuk membangun antarmuka, dilakukan setelah berhasil mendapatkan model dengan performa terbaik. Antarmuka dibangun dalam bentuk aplikasi web, yang didalamnya telah dilakukan *import* model yang didapat dari proses sebelumnya. Dalam membangun model dilakukan proses *training* dan *testing* dengan menggunakan metode LBP dan *multilayer perceptron*, yang dilakukan pada proses eksperimen. Sedangkan dalam membangun antarmuka, tidak lagi dilakukan proses *training* dan *testing*, karena antarmuka dibangun dengan tujuan hanya untuk melihat bagaimana bentuk implementasi atau pemanfaatan dari model yang telah dibangun.

3.3 Pengumpulan *Dataset*

Tahap ini diperlukan untuk mendapatkan *dataset* berupa variasi sampel tulisan tangan induk huruf aksara Lampung yang berguna untuk proses pelatihan dan pengujian dalam membangun model. Pada tahap ini beberapa responden diminta untuk menuliskan sejumlah sampel induk huruf aksara Lampung beserta variasinya menggunakan aplikasi untuk menggambar pada *smartphone*. Responden ditentukan secara acak dan tidak harus mengerti aksara Lampung, karena responden hanya diminta untuk menuliskan induk huruf aksara Lampung sesuai dengan contoh yang diberikan. Sampel data aksara induk yang dibutuhkan dalam penelitian dapat dilihat pada Gambar 3.3. berikut:

Gambar 3.3. Sampel Data Aksara Induk.

3.4 Pengolahan Citra

Dalam membangun model pengenalan induk huruf aksara Lampung, dilakukan sebuah proses pengolahan citra (*dataset*) induk huruf aksara Lampung, yang telah dikumpulkan pada tahap sebelumnya. Proses pengolahan citra (*dataset*) ini adalah sebuah proses yang berkelanjutan dan saling berkaitan, yang meliputi pembagian

dataset dan augmentasi data, *preprocessing* citra, ekstraksi fitur citra, dan klasifikasi citra.

a) Pembagian *Dataset* dan Augmentasi Data

Dataset yang telah dikumpulkan pada tahap sebelumnya, dibagi menjadi 2 bagian, yaitu *dataset* untuk pelatihan, dan *dataset* untuk pengujian. Selanjutnya, pada masing-masing bagian *dataset* tersebut dilakukan proses augmentasi data, yaitu melakukan penambahan atau peningkatan jumlah *dataset* dalam rangka mendapatkan sampel data yang lebih variatif. Proses augmentasi data ini merupakan salah satu alternatif yang dapat dilakukan untuk meningkatkan jumlah *dataset*, tanpa perlu mengambil data kembali dari responden. Salah satu cara yang dapat dilakukan dalam proses augmentasi data adalah dengan melakukan transformasi dari citra asli, seperti rotasi untuk mengubah orientasi dari gambar.

b) *Preprocessing* Citra

Dalam rangka membangun model juga dilakukan proses *preprocessing* citra, yang meliputi *resizing* (menyamakan ukuran citra), konversi ke citra *grayscale*, dan binerisasi citra. *Dataset* yang diakuisisi menggunakan aplikasi untuk menggambar pada *smartphone* dari tahap sebelumnya, terkadang memiliki ukuran yang berbeda antara satu dengan yang lain, untuk itu perlu dilakukan proses *resize* atau menyamakan ukuran seluruh citra (*dataset*). Selain itu, proses *resize* ini juga dilakukan untuk meningkatkan efisiensi pada proses pelatihan dan pengujian dalam membangun model, dengan cara mengubah ukuran dari semua *dataset* menjadi ukuran tertentu yang paling minimal, namun tetap mempertahankan kualitas citra, sehingga citra masih dapat terbaca dengan jelas. *Dataset* yang telah berhasil *dirresize*, kemudian dikonversi ke dalam bentuk citra *grayscale*, agar dapat dengan mudah melakukan proses binerisasi citra. Proses selanjutnya adalah binerisasi citra, yaitu melakukan konversi dari citra *grayscale* ke dalam bentuk citra biner. Hal ini dilakukan untuk membedakan objek dari latar belakangnya. Proses binerisasi citra ini

dilakukan dengan menggunakan metode *thresholding*, yaitu dengan menentukan nilai ambang batas dalam proses binerisasinya.

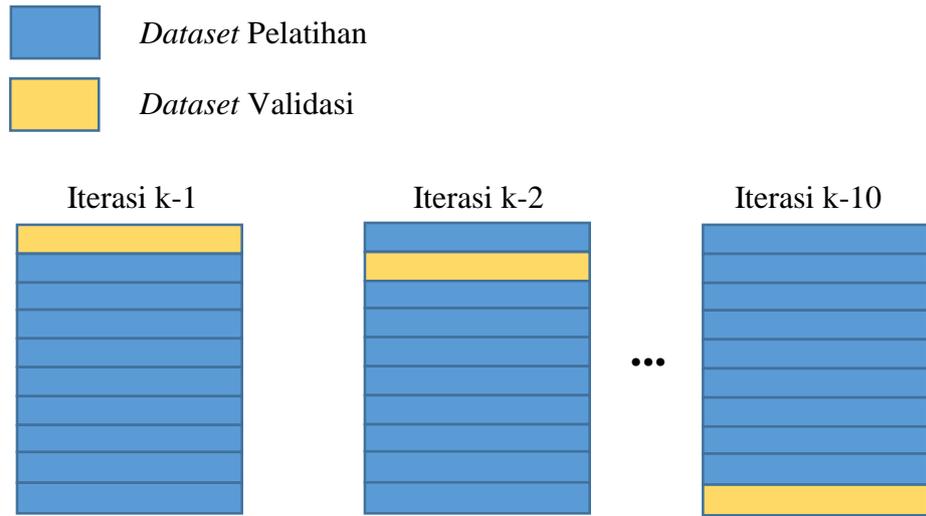
c) Ekstraksi Fitur Citra

Citra biner yang telah didapatkan dari proses sebelumnya, selanjutnya memasuki proses ekstraksi fitur untuk mendapatkan fitur khusus dari tiap-tiap sampel induk huruf aksara Lampung, yang berguna untuk proses pelatihan dan pengujian dalam membangun model MLP pada proses selanjutnya. Ekstraksi fitur ini dilakukan dengan menggunakan metode *Local Binary Python* (LBP), dan menerapkan variasi parameter radius (R) dari metode LBP, yaitu $R = 1$, $R = 2$, dan $R = 3$.

d) Klasifikasi Citra

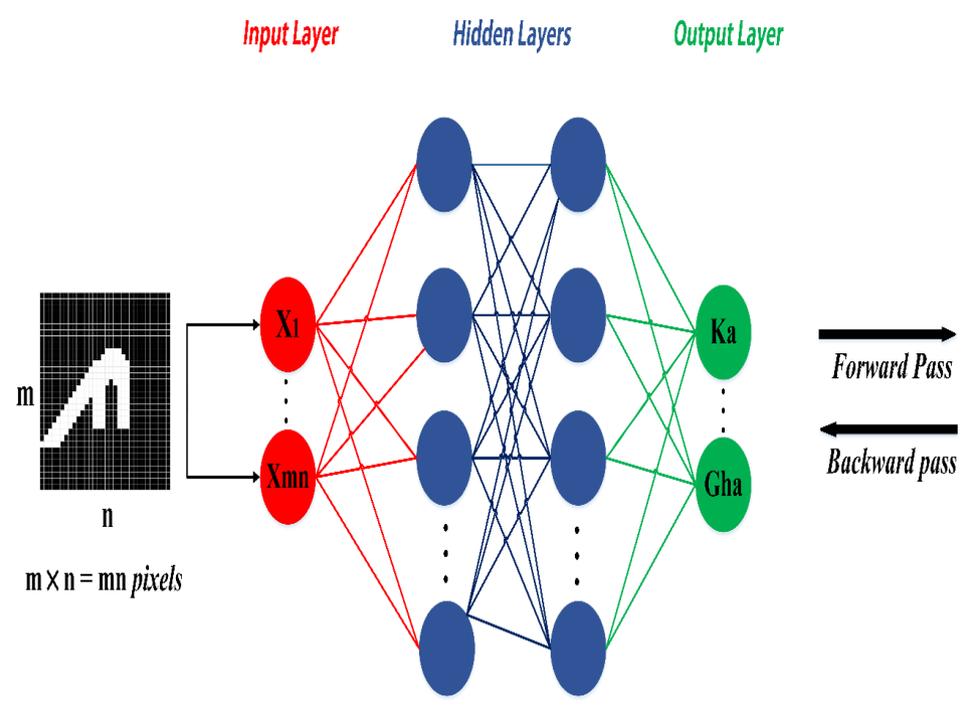
Model yang dibangun dalam penelitian ini, bertujuan untuk dapat mengklasifikasikan fitur dari tiap-tiap citra induk huruf aksara Lampung ke dalam kelas yang tepat. Model ini dibangun menggunakan jaringan MLP, dengan algoritma *backpropagation* sebagai algoritma pembelajarannya. Dalam rangka mendapatkan model dengan performa terbaik, diterapkan sebuah metode yang disebut dengan *k-fold cross validation*, yang digunakan untuk mengetahui adanya *overfitting* pada model, yaitu kondisi saat model yang dibangun memberikan performa yang baik dalam proses pelatihan tetapi buruk dalam mengenali data baru dalam proses pengujian.

Metode ini membagi *dataset* untuk pelatihan menjadi k himpunan bagian. Untuk $k = 10$, data dipartisi dengan perbandingan 9 : 1, yaitu 90% dari total seluruh *dataset* digunakan untuk proses pelatihan, sedangkan *dataset* yang tersisa digunakan untuk proses validasi model. Setiap 10% dari *dataset* yang telah digunakan untuk proses validasi model digunakan kembali untuk proses pelatihan, kemudian dari 90% *dataset* yang telah digunakan dalam proses pelatihan diambil sebanyak 10% data untuk digunakan dalam proses validasi model selanjutnya. Proses tersebut dilakukan berulang sebanyak 10 lipatan. Gambaran penggunaan metode *k-fold cross validation* dapat dilihat pada Gambar 3.4. berikut:



Gambar 3.4. *k-Fold Cross Validation*.

Pada penelitian ini, model dibangun dengan melalui proses pelatihan dan pengujian. Secara umum ilustrasi dari proses pelatihan dalam membangun model pada penelitian ini, dapat dilihat pada Gambar 3.5 berikut:



Gambar 3.5. *Arsitektur MLP dengan Dua Hidden Layers*.

Gambar 3.5. adalah contoh arsitektur MLP untuk proses pelatihan dengan dua *hidden layers*. Proses ini menggunakan *dataset* untuk pelatihan, yang telah dikelompokkan pada proses sebelumnya. Nilai piksel yang merupakan

data fitur dari tiap-tiap *dataset* yang telah didapatkan melalui proses ekstraksi fitur sebelumnya, digunakan sebagai nilai *input* untuk tiap-tiap *neuron* pada *input layer*, yang ditandai dengan *node* berwarna merah pada gambar. Kemudian, nilai-nilai tersebut dikomputasi dari *hidden layers*, menuju *output layers*. Jumlah *hidden layers* beserta *neuron* dalam penelitian ini bervariasi, hal ini dikarenakan dalam membangun model dilakukan berbagai eksperimen untuk mencari performa model terbaik, salah satu eksperimen yang dilakukan adalah dengan uji variasi jumlah *hidden layers* beserta *neuron* nya.

Adapun jumlah *neuron* pada *output layer* dalam penelitian ini berjumlah 20 *neuron*, hal ini sesuai dengan jumlah kelas/jenis induk huruf aksara Lampung, yang ditandai dengan *node* berwarna hijau pada gambar. Nilai prediksi yang dihasilkan oleh *output layers* dibandingkan dengan nilai target/aktual dari tiap-tiap kelas citra induk huruf aksara Lampung. Selisih antara nilai prediksi dengan nilai aktual disebut dengan nilai *error*. Proses yang dimulai dari *input layers* hingga didapatkan nilai *error* tersebut disebut dengan *forward pass*.

Proses selanjutnya disebut dengan *backward pass*, yaitu proses memperbaiki nilai bobot untuk menurunkan nilai *error* yang dihasilkan. Proses ini dimulai dari *output layers* menuju *input layers*. Kedua proses tersebut (*forward* dan *backward pass*) dilakukan berulang hingga didapatkan nilai *error* terendah dalam beberapa kali iterasi/*epoch*, sesuai dengan jumlah iterasi/*epoch* yang ditentukan. Nilai *error* terendah yang dihasilkan, dianggap oleh model sebagai kelas yang tepat dari citra induk huruf aksara Lampung yang diberikan. Hasil dari proses pelatihan ini berupa model jaringan yang dapat digunakan untuk proses identifikasi citra induk huruf aksara Lampung.

Selain proses pelatihan, dalam membangun model dilakukan juga proses pengujian dengan menggunakan *dataset* pengujian, yang sebelumnya telah dikelompokkan. Proses ini dilakukan untuk mengetahui performa dari model dalam mengenali data citra baru yang belum pernah digunakan dalam proses pelatihan sebelumnya. Proses pengujian dilakukan secara langsung

tepat setelah proses pelatihan selesai untuk mengetahui akurasi, *precision* dan *recall* dari model yang telah dilatih. Setelah proses pelatihan dan pengujian selesai, dan telah mendapatkan model pengenalan induk huruf dengan performa terbaik, selanjutnya model tersebut dapat digunakan dengan melakukan *import* model kedalam program aplikasi web, agar aplikasi yang telah dibangun dapat mengidentifikasi induk huruf aksara Lampung yang dituliskan oleh *user* pada *canvas* dari aplikasi web secara langsung.

3.5 Eksperimen

Dalam rangka membangun model pengenalan induk huruf aksara Lampung, dilakukan berbagai eksperimen yang saling berkaitan dan berkelanjutan untuk mencari model dengan performa terbaik. Adapun ringkasan eksperimen yang telah dilakukan, dapat dilihat pada Tabel 3.1. berikut:

Tabel 3.1. Ringkasan Eksperimen.

Eksperimen ke-	Eksperimen yang Dilakukan	Output Eksperimen
1	Membandingkan performa antara model yang dibangun hanya dengan menggunakan <i>library numpy</i> (tanpa menggunakan <i>library/API</i> khusus <i>deep learning</i>), dan model yang dibangun menggunakan <i>API/library deep learning</i> .	Mengetahui cara pengimplementasian dalam membangun model MLP yang memberikan performa model terbaik
2	Melakukan uji variasi jumlah <i>dataset</i> untuk membandingkan performa model berdasarkan perbedaan jumlah data.	Mengetahui jumlah data yang memberikan performa model terbaik.
3	Melakukan uji variasi nilai <i>learning rate</i> , <i>epoch</i> , dan <i>dropout</i> untuk mencari performa model terbaik, dengan menggunakan jumlah <i>dataset</i> yang memberikan performa	Mengetahui nilai dari masing-masing <i>hyperparameter</i> yang memberikan performa model terbaik.

Eksperimen ke-	Eksperimen yang Dilakukan	Output Eksperimen
	model terbaik berdasarkan eksperimen ke-2.	
4	Melakukan uji variasi jumlah <i>neuron</i> pada <i>hidden layers</i> terhadap performa model dengan menggunakan nilai <i>hyperparameter</i> terbaik, berdasarkan hasil yang telah didapat dari eksperimen ke-3.	Mengetahui jumlah <i>neuron</i> pada <i>hidden layers</i> , yang memberikan performa terbaik pada model.
5	Melakukan uji variasi jumlah <i>hidden layers</i> beserta <i>neuron</i> terhadap performa model, dengan menggunakan konfigurasi jumlah <i>neuron</i> terbaik yang didapatkan dari eksperimen ke-4.	Mengetahui jumlah <i>hidden layers</i> beserta <i>neuron</i> yang memberikan performa terbaik pada model.
6	Melakukan uji variasi parameter radius pada metode LBP terhadap performa model, dengan konfigurasi jumlah <i>hidden layers</i> beserta <i>neuron</i> yang terbaik, berdasarkan hasil eksperimen ke-5.	Mengetahui nilai radius dari metode LBP, yang memberikan performa terbaik pada model.
7	Eksperimen terakhir yang dilakukan dengan menggunakan konfigurasi arsitektur model terbaik berdasarkan keseluruhan eksperimen yang telah dilakukan, dan menggunakan tambahan <i>dataset</i> .	Mendapatkan model dengan performa terbaik dari keseluruhan eksperimen yang telah dilakukan.

Pada eksperimen pertama dilakukan perbandingan antara 2 buah model yang dibangun dengan pengimplementasian yang berbeda. Model pertama dibangun dengan hanya menggunakan *library numpy* (tanpa menggunakan *library/API* khusus *deep learning*). Sedangkan untuk model kedua, dibangun dengan menggunakan *library/API* khusus *deep learning*. Dalam eksperimen pertama ini tidak menggunakan *10-fold cross validation*, karena tujuan dari eksperimen ini

hanya ingin mengetahui perbedaan performa dan tingkat efisiensi komputasi dari kedua cara pengimplementasian model tersebut. Cara pengimplementasian model yang memberikan performa dan tingkat efisiensi komputasi tertinggi, akan diterapkan pada semua eksperimen berikutnya.

Pada eksperimen kedua dilakukan uji variasi jumlah data terhadap performa model, dengan menggunakan *10-fold cross validation*. Pada eksperimen ini membandingkan performa 2 buah model dengan jumlah data pelatihan yang berbeda, yang bertujuan untuk mengetahui apakah jumlah data memengaruhi performa dari model yang dibangun. Jumlah data yang memberikan performa terbaik, akan digunakan kembali untuk eksperimen ketiga, hingga eksperimen keenam. Adapun pada eksperimen ketiga, dilakukan uji variasi nilai *learning rate*, *epoch*, dan *dropout*, serta menerapkan juga *10-fold cross validation* untuk mengetahui nilai yang memberikan performa model terbaik.

Pada eksperimen keempat, dilakukan uji variasi jumlah *neuron* pada 1 *hidden layer*, dengan menerapkan nilai *learning rate*, *epoch*, dan *dropout* terbaik, berdasarkan eksperimen ketiga. Eksperimen ini juga menerapkan *10-fold cross validation* dalam proses pencarian model terbaik. Kemudian pada eksperimen kelima, dilakukan uji variasi jumlah *hidden layers* beserta *neuron* terhadap performa model, dengan menerapkan jumlah *neuron* terbaik pada tiap *hidden layer* pertama, berdasarkan hasil eksperimen keempat. Eksperimen ini juga menggunakan *10-fold cross validation* dalam rangka pencarian model terbaik.

Selanjutnya, pada eksperimen keenam dilakukan uji variasi parameter radius pada metode LBP terhadap performa model, dengan jumlah *hidden layers* beserta *neuron* terbaik, berdasarkan hasil eksperimen kelima. Eksperimen ini juga menggunakan *10-fold cross validation* dalam rangka pencarian model terbaik. Untuk eksperimen terakhir, yaitu eksperimen ketujuh, dilakukan eksperimen untuk memaksimalkan performa dari model yang dibangun, dengan menggunakan arsitektur model MLP yang terbaik berdasarkan keseluruhan eksperimen yang telah dilakukan. Pada eksperimen ini juga dilakukan penambahan *dataset*, serta menerapkan *10-fold cross validation* dalam rangka pencarian model terbaik.

3.6 Analisis dan Evaluasi

Tahap analisis dan evaluasi dilakukan dengan menggunakan hasil perhitungan akurasi yang didapatkan melalui metode *confusion matrix*. *Confusion matrix* adalah metode yang dapat digunakan untuk membandingkan hasil klasifikasi yang dilakukan oleh model dengan hasil klasifikasi yang seharusnya. Berikut adalah tabel yang digunakan pada metode *confusion matrix*:

Tabel 3.2. *Confusion Matrix*.

		<i>True Values</i>	
		<i>True</i>	<i>False</i>
<i>Prediction</i>	<i>True</i>	TP <i>Correct result</i>	FP <i>Unexpected result</i>
	<i>False</i>	TN <i>Missing result</i>	FN <i>Correct absence of result</i>

Tabel 3.2. merupakan tabel acuan yang digunakan untuk menganalisis hasil dari pengujian model sebelumnya, dengan metode *confusion matrix*. Untuk melakukan perhitungan akurasi dengan metode *confusion matrix*, dapat menggunakan Persamaan 3.1 berikut:

$$\text{Akurasi} = \frac{(TP+TN)}{(TP+TN+FP+FN)} * 100\% \quad (3.1)$$

Selain berdasarkan akurasi, nilai sensitifitas/*recall* dan *precision* juga dapat dipertimbangkan untuk mengetahui performa dari model yang telah dibangun. Untuk melakukan perhitungan nilai sensitifitas/*recall* dengan metode *confusion matrix*, dapat menggunakan Persamaan 3.2 berikut:

$$\text{Recall} = \frac{(TP)}{(TP+FN)} * 100\% \quad (3.2)$$

Sedangkan untuk melakukan perhitungan nilai *precision* dengan metode *confusion matrix*, dapat menggunakan Persamaan 3.3 berikut:

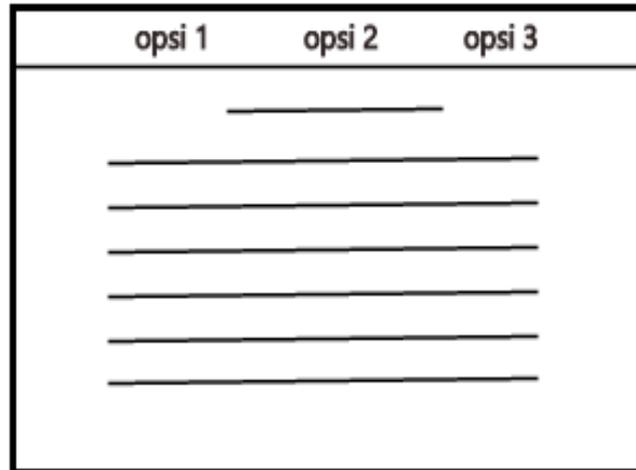
$$Precision = \frac{(TP)}{(TP+FP)} * 100\% \quad (3.3)$$

Nilai *True Positive* (TP) adalah jumlah data aksara yang terdeteksi dengan benar. Nilai ini didapat pada saat model mengenali dengan baik citra induk huruf yang diberikan oleh *user*. Kemudian, nilai *False Positive* (FP) adalah nilai data yang salah namun terdeteksi sebagai data yang benar. Nilai ini didapat saat model salah mengenali kelas, yang dianggap sebagai kelas sebenarnya. Adapun nilai *False Negative* (FN) adalah nilai data yang benar namun terdeteksi sebagai data yang salah. Nilai ini didapat saat model tidak berhasil mengenali citra induk huruf aksara Lampung yang diberikan, sehingga hasil *output* tidak sesuai dengan *input* yang diberikan.

Faktor-faktor yang dianalisis dalam penelitian ini meliputi semua proses eksperimen yang telah dilakukan. Analisis dilakukan dengan cara membandingkan hasil dari tiap-tiap eksperimen. Selanjutnya, dilakukan evaluasi terhadap keseluruhan hasil dari analisis yang telah didapatkan. Evaluasi tersebut dilakukan untuk menilai proses eksperimen yang memberikan performa model terbaik.

3.7 Rancangan Antarmuka

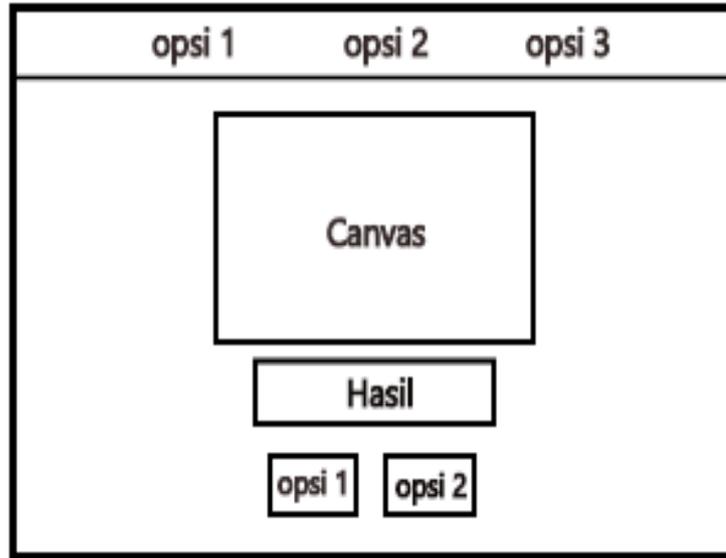
Setelah berhasil membangun model dengan performa terbaik dari keseluruhan eksperimen yang dilakukan, selanjutnya dibangun antarmuka dari model pengenalan induk huruf aksara Lampung dalam bentuk aplikasi web. Antarmuka ini dibangun dengan tujuan untuk mengetahui pemanfaatan dari model MLP yang telah dibangun. Aplikasi web yang dibangun dalam penelitian ini memiliki 3 menu di dalamnya, yaitu menu tulis aksara, menu tentang web, dan menu yang berisi informasi terkait kebudayaan Lampung. Menu tentang web berisi keterangan mengenai aplikasi web pengenalan induk huruf aksara Lampung yang telah dibuat, meliputi tujuan dibuatnya aplikasi, menu yang ada pada aplikasi, serta batasan pada aplikasi. Gambar 3.6. merupakan rancangan tampilan dari menu tentang aplikasi.



Gambar 3.6. Menu Tentang.

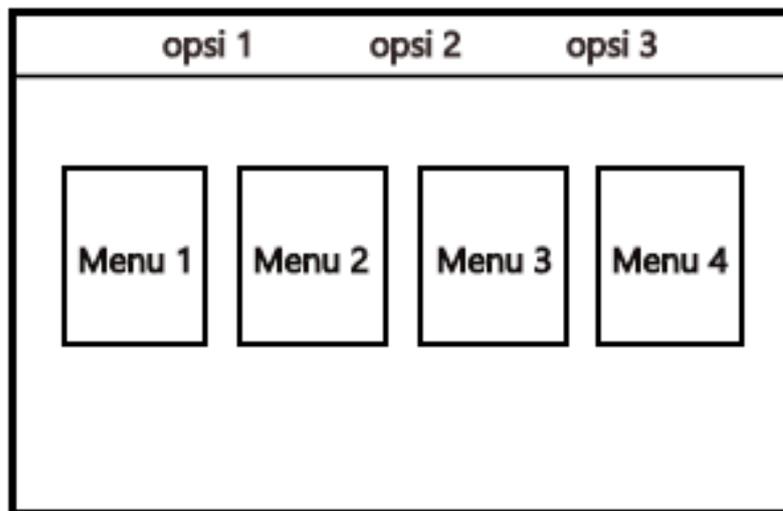
Selanjutnya adalah menu tulis aksara yang dapat digunakan oleh *user* untuk menuliskan induk huruf aksara Lampung yang ingin diidentifikasi. *User* dapat menuliskan induk huruf pada fitur *canvas*. Setelah *user* menuliskan induk huruf aksara Lampung tersebut, selanjutnya *user* memiliki dua opsi, opsi pertama dapat digunakan oleh *user* untuk menghapus tulisan, dan opsi kedua dapat digunakan untuk memprediksi gambar yang telah dituliskan. Jika *user* memilih opsi kedua, maka sistem akan mengolah citra tulisan tangan tersebut, serta melakukan ekstraksi fitur, seperti pada pengolahan citra dan ekstraksi fitur yang dilakukan dalam proses membangun model sebelumnya.

Pada proses membangun model, citra (*dataset*) *dirsize* dengan tujuan untuk menyamakan ukuran citra, dan meningkatkan efisiensi komputasi, sedangkan dalam implementasi antarmuka, citra *dirsize* dengan tujuan hanya untuk meningkatkan efisiensi komputasi, karena pada implementasi antarmuka ini, citra yang diakuisisi dari fitur *canvas* memiliki ukuran yang sama. Selanjutnya, hasil dari pengolahan citra dan ekstraksi fitur tersebut diklasifikasikan menggunakan model yang telah dibangun sebelumnya. Setelah model dapat mengklasifikasikan citra tulisan tangan yang ditulis oleh pengguna, hasil dari prediksi tersebut ditampilkan oleh sistem ke layar. Adapun rancangan tampilan dari menu tersebut ditunjukkan pada Gambar 3.7. berikut:



Gambar 3.7. Menu Tulis Aksara.

Selain itu, pada aplikasi web ini juga memiliki menu yang berisi informasi terkait kebudayaan Lampung. Tujuannya adalah untuk memperkenalkan atau mengingatkan kembali pada pengguna tentang beberapa kebudayaan masyarakat Lampung. Adapun rancangan tampilan dari menu tersebut ditunjukkan pada Gambar 3.8. berikut:



Gambar 3.8. Menu Edukasi.