BAB 2
LANDASAN TEORI

2.1 *Oral squamous cell carcinoma* (OSCC)

OSCC adalah neoplasma yang berasal dari keratinosit oral yang mengalami mutasi menjadi sel ganas, dan merupakan salah satu kanker mulut yang paling umum dilihat dari angka kesakitan (morbiditas) dan kematian (mortalitas) di seluruh dunia [4, 15]. Berbagai bagian dari rongga mulut dapat terkena tumor ini, seperti bibir, lidah, dan dasar mulut [16].

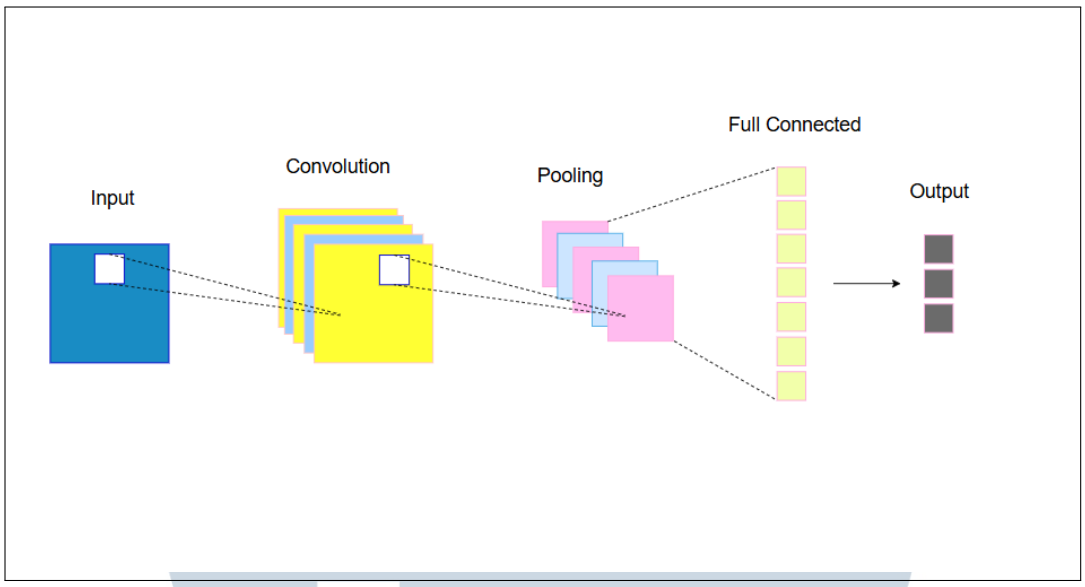
Faktor risiko paling umum yang menyebabkan perkembangan OSCC tampaknya berkaitan dengan penyalahgunaan alkohol dan konsumsi rokok. Secara khusus, zat kimia yang dihasilkan saat merokok tembakau dapat menurunkan respons kekebalan di lingkungan mulut serta merusak DNA sel, yang kemudian mendorong terjadinya karsinogenesis (proses pembentukan kanker) [5].

Akibatnya, pada awal proses karsinogenesis, beberapa lesi (kerusakan jaringan) muncul pada epitel. Lesi ini secara histologis dapat diklasifikasikan berdasarkan perubahan yang diamati pada jaringan yang terkena, seperti modifikasi bentuk keratinosit atau hiperplasia. Perubahan ini menandai tahap kanker sebelum masuk ke proses metastasis [17].

Sayangnya, OSCC seringkali baru terdiagnosis pada tahap yang sangat lanjut [18], sehingga sangat menurunkan kemungkinan bertahan hidup dan mengurangi kualitas hidup pasien. Oleh karena itu, keberhasilan pengobatan OSCC sangat bergantung pada intervensi yang tepat dan cepat pada tahap awal tumor [19].

2.2 Convolutional Neural Network (CNN)

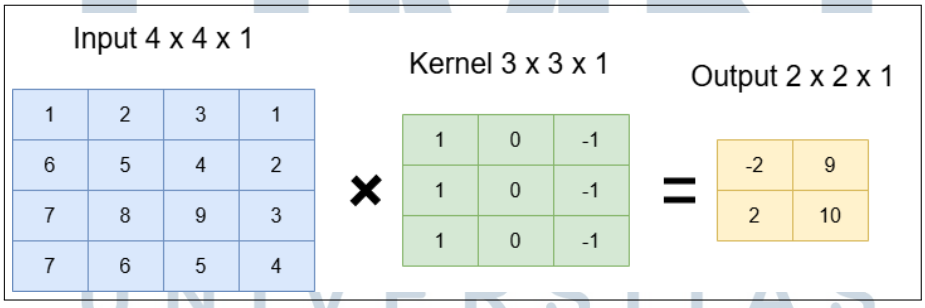
Convolutional Neural Network (CNN) merupakan pengembangan dari arsitektur Multilayer Perceptron (MLP). MLP sendiri termasuk dalam jenis jaringan saraf dalam (deep neural network) yang dirancang untuk menangani data berdimensi dua dengan banyak lapisan. CNN umumnya dimanfaatkan untuk pemrosesan citra digital [20]. Mekanisme kerja CNN mencakup tahap pemberian input ke lapisan konvolusi, dilanjutkan dengan lapisan pooling, dan ditutup oleh lapisan fully-connected sebagai hasil akhir pemrosesan [21]. Ilustrasi alur kerja CNN dapat dilihat pada gambar 2.1.



Gambar 2.1. Convolutional neural network

2.2.1 Convolutional Layer

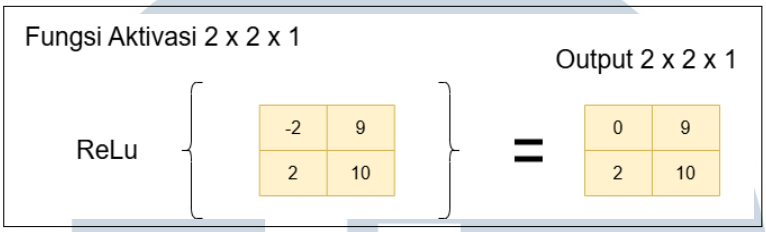
Lapisan konvolusi merupakan bagian dasar dari arsitektur CNN yang bertujuan untuk mengekstraksi fitur penting dari data input citra. Operasi konvolusi dilakukan dengan cara menggeser kernel atau filter melintasi citra input, kemudian menghitung hasil perkalian elemen-elemen antara patch dari citra dan kernel, yang kemudian dijumlahkan menghasilkan nilai pada posisi tertentu dalam output.



Gambar 2.2. Proses konvolusi antara citra input dengan kernel menghasilkan output fitur

Gambar 2.2 menunjukkan sebuah contoh proses konvolusi. Citra input berukuran $4 \times 4 \times 1$ dikenakan kernel berukuran $3 \times 3 \times 1$. Kernel ini digeser melintasi input untuk menghasilkan output berukuran $2 \times 2 \times 1$. Setiap nilai output dihitung berdasarkan jumlah perkalian antara nilai input dan kernel pada patch yang bersesuaian.

Setelah hasil konvolusi diperoleh, biasanya dilanjutkan dengan penerapan fungsi aktivasi non-linear, seperti ReLU (*Rectified Linear Unit*), yang bertugas untuk memperkenalkan non-linearitas dalam model.



Gambar 2.3. Penerapan fungsi aktivasi ReLU pada output konvolusi

Gambar 2.3 memperlihatkan bagaimana fungsi ReLU diterapkan pada output hasil konvolusi. Nilai negatif dikonversi menjadi nol, sedangkan nilai positif dibiarkan tetap. Hal ini meningkatkan kemampuan model dalam mempelajari pola yang kompleks dan mengatasi masalah *vanishing gradient*.

Melalui dua tahapan ini, yaitu operasi konvolusi dan fungsi aktivasi, CNN dapat mengekstrak informasi spasial dari citra secara efisien dan digunakan pada tahap-tahap lanjutan dalam jaringan.

2.2.2 Pooling Layer

Lapisan pooling berperan dalam mengurangi dimensi spasial dari feature maps melalui teknik *downsampling*, seperti *max pooling* atau *average pooling*. Tujuannya adalah untuk menyederhanakan representasi data dan mengurangi kompleksitas komputasi, tanpa kehilangan fitur penting dari hasil ekstraksi [21].

A Global Average Pooling (GAP)

Global Average Pooling (GAP) merupakan teknik pooling yang digunakan dalam arsitektur CNN. Berbeda dengan max pooling yang hanya mengambil nilai maksimum dari suatu area, GAP menghitung rata-rata seluruh nilai dalam satu channel, menghasilkan satu nilai representatif untuk setiap channel. Teknik ini efektif dalam mengurangi jumlah parameter pada jaringan dan membantu mencegah *overfitting* [22, 23].

A.1 Rumus Global Average Pooling

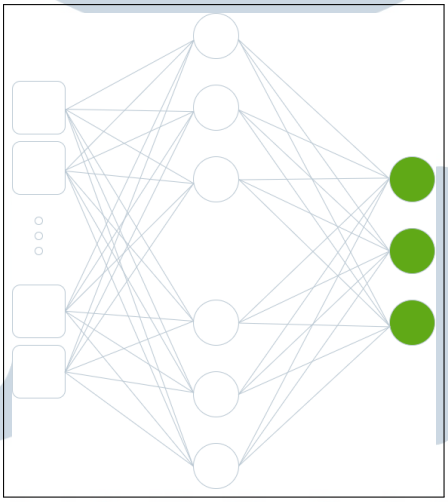
GAP secara matematis dirumuskan sebagai berikut:

$$\text{GAP}(x)_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_{i,j,c} \tag{2.1}$$

Dengan keterangan:

- $x_{i,j,c}$: nilai piksel pada posisi (i, j) di channel ke-c,
- H : tinggi dari *feature map*,
- W : lebar dari *feature map*,
- $\text{GAP}(x)_c$: hasil pooling pada channel ke-c.

B Fully Connected Layer



Gambar 2.4. Struktur fully connected layer

Gambar 2.4 menunjukkan arsitektur *fully connected layer* atau dense layer dalam jaringan saraf tiruan. Pada struktur ini, setiap neuron di lapisan sebelumnya dihubungkan secara penuh ke semua neuron di lapisan berikutnya. Hubungan ini berbentuk transformasi linear dari seluruh input yang diterima oleh setiap neuron.

Secara matematis, proses pada setiap neuron di fully connected layer dapat dirumuskan sebagai:

$$z = (w_1 \times x_1) + (w_2 \times x_2) + (w_3 \times x_3) + \cdots + (w_n \times x_n) + b \tag{2.2}$$

Keterangan:

- $x_1, x_2, \ldots, x_n$ adalah input yang berasal dari neuron-neuron di lapisan sebelumnya.
- $w_1, w_2, \ldots, w_n$ adalah bobot (weight) yang terkait dengan masing-masing input.
- b adalah bias, nilai tambahan yang membantu model menyesuaikan output.
- z adalah hasil dari kombinasi linear sebelum diterapkan fungsi aktivasi.

Nilai z kemudian akan dilewatkan ke fungsi aktivasi seperti ReLU atau *sigmoid* untuk menghasilkan output akhir neuron tersebut. Struktur seperti ini umum digunakan pada bagian akhir arsitektur CNN untuk melakukan klasifikasi berdasarkan fitur yang telah diekstraksi sebelumnya.

2.3 Fungsi Aktivasi (Activation Function)

Fungsi aktivasi merupakan komponen penting dalam jaringan saraf tiruan (Artificial Neural Networks), berperan dalam menentukan output dari sebuah neuron berdasarkan input yang diterima. Tanpa fungsi aktivasi, jaringan saraf hanya akan menjadi model linier dan tidak mampu mempelajari relasi kompleks dalam data non-linier [24].

2.3.1 Rectified Linear Unit (ReLU)

ReLU (Rectified Linear Unit) adalah salah satu fungsi aktivasi paling umum digunakan dalam arsitektur deep learning karena kesederhanaannya dan efisiensi komputasi yang tinggi. Fungsi ini mengubah input negatif menjadi nol dan membiarkan input positif tetap seperti aslinya:

$$\mathrm{ReLU}(x) = \max(0, x) \tag{2.3}$$

ReLU mengatasi masalah vanishing gradient yang umum terjadi pada fungsi aktivasi klasik seperti sigmoid dan tanh. Dengan menjaga gradien tetap saat nilai input positif, ReLU memungkinkan pembelajaran jaringan yang lebih cepat dan stabil. Berdasarkan studi oleh Bai (2022), ReLU menunjukkan performa klasifikasi terbaik dibandingkan fungsi aktivasi lainnya dalam eksperimen pada dataset citra [25].

2.3.2 Sigmoid

Fungsi sigmoid mengubah setiap input menjadi nilai antara 0 dan 1, menjadikannya cocok untuk tugas klasifikasi biner. Fungsi ini didefinisikan sebagai:

$$\text{Sigmoid}(x) = \frac{1}{1+e^{-x}} \tag{2.4}$$

Meskipun sigmoid intuitif dan berguna untuk interpretasi probabilistik, fungsi ini cenderung menyebabkan masalah vanishing gradient karena turunan dari fungsi sigmoid akan sangat kecil jika input berada jauh dari nol. Namun, dalam jaringan kecil atau pada layer output dengan klasifikasi dua kelas, sigmoid tetap efektif dan sering digunakan [26].

2.4 ResNet152 (Residual Network 152)

Residual Network (ResNet) merupakan salah satu jenis arsitektur CNN yang dirancang untuk menekan beban komputasi lebih efisien dibandingkan VGGNet. Arsitektur ini diperkenalkan oleh He, yang mengembangkan model ini berdasarkan permasalahan meningkatnya error saat jaringan deep learning semakin dalam. Untuk mengatasi hal tersebut, digunakan teknik skip connection atau identity mapping sebagai jalur alternatif aliran data dalam jaringan [27]. Kelebihan utama dari struktur ResNet adalah kemampuannya menjaga akurasi model tetap tinggi meskipun jaringan bertambah dalam, serta mampu menghindari isu gradient vanishing yang umum terjadi pada jaringan tradisional [28].

Pada arsitektur ResNet152, struktur jaringan terdiri atas blok residual dengan komposisi convolution layer 3x3, batch normalization 1x1, dan ReLU activation 1x1. Di tahap awal, lapisan conv1 menggunakan convolution 7x7 dengan output berdimensi 112×112. Kemudian, lapisan conv2 melibatkan

proses maxpooling 3x3 dan terdiri dari tiga blok dengan keluaran berukuran 56×56. Tahapan berikutnya, lapisan conv3 memiliki delapan blok dengan output 28×28, diikuti oleh conv4 yang terdiri atas 36 blok dan menghasilkan keluaran 14×14. Selanjutnya, conv5 memproses tiga blok dengan keluaran berdimensi 7×7, dilanjutkan dengan average pooling, fully connected layer, dan pada bagian akhir terdapat lapisan klasifikasi menggunakan softmax activation dengan hasil keluaran 1×1 [29].

2.5 ResNet101 (Residual Network 101)

ResNet101 adalah arsitektur jaringan saraf dalam (*Deep Neural Network*) dengan 101 lapisan yang diperkenalkan oleh Kaiming He et al. Arsitektur ini menggunakan pendekatan *residual learning* yang mengimplementasikan *skip connection* agar model dapat mempelajari fungsi identitas, sehingga mampu menghindari masalah degradasi akurasi pada jaringan yang sangat dalam.

Dalam penelitian oleh Heo et al. (2022), ResNet101 digunakan untuk mendiagnosis kanker lidah menggunakan citra endoskopi dan menunjukkan performa klasifikasi yang tinggi [30]. Arsitektur ini juga banyak digunakan dalam tugas-tugas seperti deteksi objek dan pengenalan wajah.

2.6 K-Fold Cross Validation

K-Fold Cross Validation (CV) merupakan metode di mana dataset dibagi menjadi K bagian atau *fold*. Setiap *fold* akan secara bergantian digunakan sebagai data uji, sementara *fold* yang lain digunakan untuk pelatihan model. Sebagai contoh, dalam skenario validasi silang dengan 5 *fold* (K = 5), data dibagi menjadi lima bagian. Pada iterasi pertama, bagian pertama digunakan sebagai data uji, dan empat bagian lainnya digunakan untuk melatih model. Pada iterasi berikutnya, bagian kedua menjadi data uji, sedangkan bagian lainnya digunakan untuk pelatihan. Proses ini terus berlanjut hingga setiap *fold* digunakan satu kali sebagai data uji[31].

2.7 Optimazer AdamW

Sebagai penyempurnaan dari optimizer Adam, AdamW memperkenalkan cara yang lebih efektif dalam menangani Weight Decay [32]. Algoritma ini dimodifikasi agar proses regularisasi Weight Decay tidak lagi menyatu dengan

pembaruan gradien, melainkan diterapkan secara terpisah sehingga mengatasi keterbatasan yang ada pada Adam.

2.8 Evaluasi Model

Evaluasi merupakan tahap yang sangat penting dalam menentukan kualitas prediksi model klasifikasi, khususnya dalam pengenalan kualitas kacang hijau. Model yang telah dilatih akan diuji menggunakan data pelatihan dan data pengujian, lalu dievaluasi menggunakan sejumlah metrik. Beberapa metrik evaluasi yang digunakan antara lain *accuracy*, *precision*, *recall*, dan *F1-score*, yang semuanya dihitung berdasarkan *confusion matrix*[33].

Tabel 2.1. Tabel *confusion matrix*

Actual / Predicted	**Positive**	**Negative**
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

Precision (P) mengukur proporsi prediksi positif yang benar-benar relevan, atau dapat didefinisikan sebagai rasio antara prediksi positif yang benar terhadap seluruh prediksi positif. Formula untuk precision diberikan dalam Persamaan 2.5[33].

$$P = \frac{TP}{TP+FP} \times 100\% \tag{2.5}$$

Recall (R) merepresentasikan seberapa banyak dari total sampel positif yang berhasil diidentifikasi dengan benar oleh model. Persamaan untuk recall ditunjukkan dalam Persamaan 2.6[33].

$$R = \frac{TP}{TP+FN} \times 100\% \tag{2.6}$$

F1-score merupakan rata-rata harmonik dari precision dan recall yang digunakan untuk menilai keseimbangan antara keduanya. Rumus F1-score disajikan dalam Persamaan 2.7[33].

$$F1 = \frac{2 \cdot P \cdot R}{P+R} \tag{2.7}$$

Accuracy menunjukkan rasio prediksi yang benar terhadap keseluruhan prediksi yang dilakukan oleh model. Ini adalah metrik yang paling umum dan intuitif dalam evaluasi klasifikasi, sebagaimana dinyatakan dalam Persamaan 2.8[33].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{2.8}$$