

BAB II. LANDASAN TEORI

2.1 Studi Literatur

Berdasarkan penelitian yang akan dilakukan, acuan dari beberapa penelitian terdahulu menjadi sangat penting dalam melakukan sebuah penelitian dengan tujuan untuk mengetahui hubungan antara penelitian yang akan dilakukan dengan penelitian terdahulu, sehingga dengan menambahkan acuan tersebut dapat menambah referensi, perbedaan, serta keunikan dalam penelitian yang akan dilakukan.

Banyak pengembangan sistem yang menggunakan teknologi *Computer Vision*, seperti: *face detection*, *image recognition* maupun pengenalan dalam pola tertentu. Pengembangan sistem ini menjadi sebuah fungsionalitas yang akan mempermudah pekerjaan dalam berbagai bidang. Pengembangan dari *deep learning* ini sangat tepat dan efektif dalam menyelesaikan permasalahan tersebut. Hal ini tidak lepas dengan adanya riset atau penelitian dalam bidang tersebut. Penelitian terdahulu mengenai *deep learning* dengan menggunakan *Convolutional Neural Network* (CNN) dilakukan oleh para *research* pada berbagai macam *object*.

Adapun penelitian yang dilakukan oleh Ardian Yusuf Wicaksono, (Wicaksono et al., 2017) mengenai Modifikasi Arsitektur Convolutional Neural Network untuk klasifikasi motif gambar batik. Penelitian yang dilakukan oleh Ardian Yusuf Wicaksono, dkk menggunakan metode CNN dengan mengembangkan pada arsitektur dari modelnya dengan mengkombinasi GoogleNet dan Residual Networks yang dinamai IncRes. Penelitian ini menggunakan 11 class dari tipe motif batik dengan jumlah data gambar 7112 yang dibagi kedalam 6401 digunakan untuk data latih (train) dan 711 digunakan untuk data uji (test). Dari hasil penelitian ini memperoleh accuracy sebesar 70.84 % dengan waktu 733 ms (milisecond).

Adapun penelitian yang dilakukan oleh Nadia Azahro Choirunisa, dkk (Azahro Choirunisa et al., 2021) mengenai Deteksi Ras Kucing Menggunakan *Compound Model Scaling Convolutional Neural Network*. Pada penelitian ini Nadia Azahro Choirunisa, dkk melakukan penelitian menggunakan arsitektur

EfficientNet-B0 sebagai base model. Model mendapatkan akurasi yang paling tinggi yaitu sebesar 98% menggunakan optimizer Adam dan RMSprop dengan learning rate 0.001.

Adapun penelitian yang dilakukan oleh Mohammad Arif Rasyidi, dkk (Rasyidi & Bariyah, 2020) mengenai Pengenalan Pola Batik Menggunakan *Convolutional Neural Network*. Pada penelitian ini Mohammad Arif Rasyidi, dkk melakukan penelitian menggunakan algoritma *Convolutional Neural Network* dengan menggunakan arsitektur *DenseNet Network*. Pada penelitian ini menghasilkan akurasi sebesar 94.3% dengan melakukan *image augmentation* pada jenis-jenis batik yang tersedia.

Adapun penelitian yang dilakukan oleh Jonatahan Tristanto, dkk ('Jonathan Tristanto', 2018) mengenai Klasifikasi Motif Batik Menggunakan *Convolutional Neural Network*. Pada penelitian ini Jonatahan Tristanto, dkk mengambil dataset citra batik melalui *Google Images*. Dataset ini berisi 967 citra yang berisikan batik dari Lasem, Yogyakarta, dan Solo dengan 13 motif yang berbeda. 13 motif batik tersebut adalah Aseman, Bledak, Gunung Ringgit, Krecak, Kawung, Latohan, Naga, Parang, Sido Mukti, Sido Luhur, Sekar Jagad, Slobog, dan Tuntum. Ukuran gambar yang digunakan untuk melakukan penelitian ini adalah 160x160. Penelitian ini menghasilkan akurasi sebesar 56%. Model yang digunakan pada penelitian ini bekerja baik saat mengklasifikasikan motif batik Kawung, tetapi gagal dalam mengklasifikasikan motif batik Naga dan batik Sido Mukti secara akurat.

Adapun penelitian yang dilakukan oleh Yufis Azhar, dkk (Azhar et al., 2021) mengenai Ensemble Convolutional Neural Network for Robust Batik Classification. Pada penelitian ini menggunakan arsitektur Convolutional layers, alternating dropout, dan max-pooling layers. Penelitian ini menggunakan 2 tipe dataset. Dataset yang pertama menggunakan dataset Batik300. Dataset ini memiliki 300 batik images dengan memiliki 50 kelas yang berbeda. Setiap kelas memiliki 6 gambar. 6 gambar ini merupakan hasil dari augmentasi dari sebuah gambar dengan dimensi yang besar. Dataset ini memiliki ukuran gambar sebesar 128x128 piksel. Dataset yang kedua adalah dataset Batik41k yang memiliki 41.612 data. Penelitian ini menghasilkan akurasi sebesar 100%

Adapun penelitian yang dilakukan oleh Amin Padmo dan Hamdani (Masa & Hamdani, 2021) mengenai Klasifikasi Motif Citra Batik Menggunakan Convolutional Neural Network Berdasarkan K-means Clustering. Pada penelitian ini menggunakan arsitektur Convolutional Neural Network dengan menggabungkan antara proses segmentasi citra dan *enhancement* dengan *median filter* dan *sharpening*. Penelitian ini menggunakan 3 jenis batik dengan jumlah 300 citra. Proses training menggunakan 200 data citra, sedangkan untuk proses validation menggunakan 100 data citra dan 15 data citra untuk proses testing. Proses klasifikasi citra batik dengan CNN berdasarkan K-means Clustering pada median filter menghasilkan akurasi 100%. Sementara itu, proses klasifikasi citra batik dengan CNN berdasarkan K-means Clustering dari hasil *sharpening* menghasilkan akurasi 80%.

Adapun penelitian yang dilakukan oleh Samuel Febrian, Djoni Haryadi, dkk (Tumewu et al., n.d.) mengenai Klasifikasi Motif Batik menggunakan metode Deep Convolutional Neural Network dengan Data Augmentation. Pada penelitian ini menggunakan metode Convolutional Neural Network dengan arsitektur ResNet untuk mengenali pola pada batik. Selain itu, pada dataset juga diterapkan augmentasi Scale, Random Erase, Rotation dan Flip. Hasil pengujian menunjukkan CNN dengan penggunaan data augmentation pada training dataset memberikan akurasi hingga 84,52% pada Resnet-18 dan 81,90% pada Resnet-50. Adapun pada penggunaan dataset dengan augmentasi rotation memberikan peningkatan akurasi sebesar 4,06%, augmentasi random erase memberikan peningkatan sebesar 9,38%, augmentasi scale sebesar 6,52%, dan pada augmentasi flip sebesar 8,58%.

Adapun penelitian yang dilakukan oleh Lutfi Hakim, dkk (Hakim et al., 2023) mengenai Klasifikasi Motif Batik Banyuwangi Menggunakan Convolutional Neural Network. Pada penelitian ini menggunakan metode Convolutional Neural Network dengan arsitektur yang diberi nama MyCustomModel. Data yang digunakan pada penelitian ini sebanyak 120 citra untuk masing masing motif batik yang dimana terdapat 7 macam motif batik dan hasil prediksi mendapatkan nilai akurasi sebesar 63%.

Tabel 2. 1 Perbandingan Studi Literatur

No	Penulis	Judul	Arsitektur	Akurasi
1.	(Wicaksono et al., 2017)	Modifikasi Arsitektur Convolutional Neural Network untuk Klasifikasi Motif Gambar Batik	<i>Convolutional Neural Network</i> (CNN) dengan kombinasi GoogleNet dan Residual Networks yang dinamai IncRes	70,84%
2.	(Azahro Choirunisa et al., 2021)	Deteksi Ras Kucing Menggunakan <i>Compound Model Scaling Convolutional Neural Network</i>	<i>EfficientNet-B0</i>	98%
3.	(Rasyidi & Bariyah, 2020)	Pengenalan Pola Batik Menggunakan <i>Convolutional Neural Network</i>	<i>DenseNet Network</i>	94,3%
4.	(‘Jonathan Tristanto’, 2018)	Klasifikasi Motif Batik Menggunakan <i>Convolutional Neural Network</i>	<i>Convolutional Neural Network</i> (CNN)	56%
5.	(Azhar et al., 2021)	Ensemble Convolutional Neural Network for Robust Batik Classification	<i>Convolutional Neural Network</i> (CNN)	100%

6.	(Masa & Hamdani, 2021)	Klasifikasi Motif Citra Batik Menggunakan Convolutional Neural Network Berdasarkan K-means Clustering	Convolutional Neural Network	<ul style="list-style-type: none"> • CNN berdasarkan K-means Clustering pada median filter menghasilkan akurasi 100%. • dengan CNN berdasarkan K-means Clustering pada median filter menghasilkan akurasi 100%
7.	(Tumewu et al., n.d.)	Klasifikasi Motif Batik menggunakan metode Deep Convolutional Neural Network dengan Data Augmentation	Convolutional Neural Network dengan arsitektur ResNet	84,52% pada Resnet-18 dan 81,90% pada Resnet-50
8.	(Hakim et al., 2023)	Klasifikasi Motif Batik Banyuwangi Menggunakan Convolutional Neural Network	Convolutional Neural Network dengan arsitektur MyCustomModel	63%

Berdasarkan analisa penelitian terdahulu, dapat diambil kesimpulan bahwa pada penelitian yang dilakukan oleh Yufis Azhar, dkk menggunakan arsitektur

Convolutional layers, alternating dropout, dan max-pooling layers serta pada penelitian yang dilakukan oleh Nadia Azahro, dkk dengan menerapkan *EfficientNet-B0* sebagai base model dan menggunakan *optimizer* RMSprop dan Adam serta menerapkan data augmentasi yang dilakukan oleh Samuel Febrian, Djoni Haryadi, dkk. diharapkan penelitian pada Implementasi Image Classification Pada Jenis-Jenis Batik Menggunakan Arsitektur Convolutional Neural Network akan mendapatkan nilai akurasi yang baik.

2.2 Dasar Teori

2.2.1 Batik

Batik merupakan hal yang tidak asing bagi masyarakat Indonesia saat ini. Batik merupakan salah satu warisan nusantara yang unik. Keunikannya ditunjukkan dengan berbagai macam motif yang memiliki makna tersendiri. Menurut Asti M. dan Ambar B. Arini (2011: 1) berdasarkan etimologi dan terminologinya, batik merupakan rangkaian kata mbat dan tik. Mbat dalam bahasa Jawa dapat diartikan sebagai ngembat atau melempar berkali-kali, sedangkan tik berasal dari kata titik. Jadi, membatik artinya melempar titik berkali-kali pada kain. Adapula yang mengatakan bahwa kata batik berasal dari kata amba yang berarti kain yang lebar dan kata titik. Artinya batik merupakan titik-titik yang digambar pada media kain yang lebar sedemikian sehingga menghasilkan pola-pola yang indah. Dalam Kamus Besar Bahasa Indonesia, batik memiliki arti kain bergambar yang pembuatannya secara khusus dengan menuliskan atau menerakan malam pada kain itu, kemudian pengolahannya diproses dengan cara tertentu.

Perkembangan batik di Indonesia cukup pesat. Tidak hanya secara produksi saja namun juga dengan perkembangan motifnya. Banyak motif-motif baru dan dengan warna-warna yang lebih menarik. Di Jawa terdapat beberapa kelompok motif batik dengan ragam hias geometris, seperti: Ceplok, Kawung, Nitik, Parang, dll. Keanekaragaman motif batik dari seluruh Indonesia ini membuat beberapa orang kesulitan untuk mengenali jenis-jenis batik.

2.2.2 Citra Digital

Citra Digital adalah gambar dua dimensi yang dihasilkan dari gambar analog dua dimensi yang kontinu menjadi gambar diskrit melalui proses sampling.

Proses perubahan citra menjadi citra digital dinamakan dengan digitasi. Digitasi merupakan proses mengubah sebuah gambar, teks, atau suara dari benda yang dapat dilihat ke dalam data elektronik dan dapat disimpan serta diproses untuk keperluan lainnya. Dalam konteks yang lebih luas, pengolahan citra digital lebih mengacu pada pemrosesan setiap dua data dimensi. Pengolahan citra digital adalah sebuah disiplin ilmu yang mempelajari tentang bagaimana teknik pengolahan sebuah citra. Citra yang dimaksud disini adalah sebuah gambar diam (foto) maupun gambar bergerak (Video). Sedangkan digital disini mempunyai maksud penting bahwa pengolahan citra/gambar dilakukan secara digital menggunakan komputer (Sutoyo, 2009). Dalam citra digital terdapat sebuah larik (array) yang berisi nilai-nilai real maupun kompleks yang di representasikan dengan deretan bit tertentu.

Dalam sebuah komputer, citra digital dipetakan menjadi bentuk grid dan elemen piksel berbentuk matriks 2 dimensi. Setiap piksel-piksel tersebut memiliki angka yang mempresentasikan channel warna. Angka pada setiap piksel disimpan secara berurutan oleh sebuah komputer dan sering dikurangi untuk keperluan kompresi maupun pengolahan tertentu. Sebuah citra digital dapat mewakili oleh sebuah matriks yang terdiri dari M kolom N baris, dimana perpotongan antara kolom dan baris disebut piksel. (pixel = picture element), yaitu elemen terkecil dari sebuah citra. Piksel mempunyai dua parameter, yaitu koordinat dan intensitas atau warna. Nilai yang terdapat pada koordinat (x,y) adalah $f(x,y)$, yaitu besar intensitas atau warna dari piksel di titik itu. Oleh karena itu, citra dapat dituliskan kedalam sebuah matriks:

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, M - 1) \\ f(1,0) & f(1,1) & \dots & f(1, M - 1) \\ \vdots & \vdots & \vdots & \vdots \\ f(N - 1, 0) & f(N - 1, 1) & \dots & f(N - 1, M - 1) \end{bmatrix} \quad (2.1)$$

2.2.3 Pengolahan Citra

Pengolahan citra digital (Digital Image Processing) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud disini adalah gambar diam (foto) maupun gambar bergerak (yang berasal dari webcam). Sedangkan digital disini mempunyai maksud bahwa pengolahan

citra/gambar dilakukan secara digital menggunakan komputer (Tompunu, 2011). Tujuan dari pengolahan citra ini adalah untuk memperbaiki kualitas citra sehingga dapat diinterpretasi dengan mudah oleh manusia atau sebuah mesin (komputer).

2.2.4 Machine Learning

Machine Learning (ML) adalah salah satu aplikasi dari Artificial Intelligent (AI) yang fokus kepada pengembangan sebuah sistem yang mampu belajar sendiri tanpa harus diprogram berulang kali. ML membutuhkan sebuah data (data training) sebagai proses learning sebelum menghasilkan sebuah hasil. Jadi, secara sederhana dapat dijelaskan bahwa Machine Learning adalah pemrograman komputer untuk mencapai kriteria/performa tertentu dengan menggunakan sekumpulan data training atau pengalaman di masa lalu (past experience) (Chazar, 2020).

Dalam pembelajaran *machine learning*, terdapat beberapa skenario-skenario, seperti:

1. *Supervised Learning*

Penggunaan skenario *supervised learning*, pembelajaran menggunakan masukan data pembelajaran yang telah diberi label. Setelah itu, membuat prediksi dari data yang telah diberi label.

2. *Unsupervised Learning*

Penggunaan skenario *unsupervised learning*, pembelajaran menggunakan masukan data pembelajaran yang tidak diberi label. Setelah itu, mencoba untuk mengelompokkan data berdasarkan karakteristik-karakteristik yang ditemui.

3. *Reinforcement Learning*

Pada skenarion *reinforcement learning* fase pembelajaran dan tes saling dicampur. Untuk mengumpulkan informasi pembelajar secara aktif dengan berinteraksi ke lingkungan sehingga untuk mendapatkan balasan untuk setiap aksi dari pembelajar

2.2.5 Deep Learning

Deep Learning adalah salah satu cabang dari *Machine Learning* yang terdiri dari algoritma pemodelan abstraksi tingkat tinggi pada data menggunakan sekumpulan fungsi transformasi non-linear yang ditata berlapis-lapis dan mendalam. *Deep learning* sangat baik untuk diterapkan pada *supervised learning*, *unsupervised learning* dan *semi-supervised learning* maupun untuk *reinforcement*

learning dalam berbagai aplikasi seperti pengenalan citra, suara, klasifikasi teks, dan sebagainya (Cholissodin, I et al, 2020)

Aplikasi konsep jaringan syaraf tiruan yang dalam dan banyak lapisan dapat ditangguhkan pada algoritma *Machine Learning* yang sudah ada sehingga komputer sekarang bisa belajar dengan kecepatan, akurasi dan skala yang besar. Prinsip ini terus berkembang hingga *deep learning* semakin sering digunakan pada komunitas riset dan industri untuk membantu memecahkan banyak masalah data besar seperti *computer vision*, *speech recognition* dan *natural language procesing*. *Feature engineering* adalah salah satu fitur utama dari *deep learning* untuk mengekstrak pola yang berguna dari data yang akan memudahkan model untuk membedakan kelas. *Feature engineering* juga merupakan teknik yang paling penting untuk mencapai hasil yang baik pada tugas prediksi.

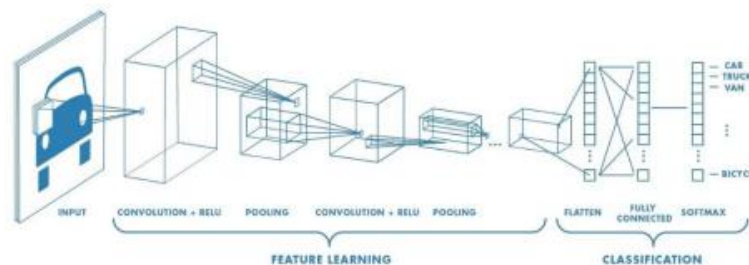
2.2.6 Artificial Neural Network (ANN)

Artificial Neural Network (ANN) adalah sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi, yang merupakan representasi tiruan dari otak manusia, yang berisi berjuta-juta sel syaraf (neuron) dan berfungsi untuk memproses informasi. Neuron mempunyai karakteristik yang sama dalam ANN, terdiri dalam kelompok-kelompok yang disebut layer. Neuron-neuron dalam satu layer terhubung dalam layer-layer lainnya yang berdekatan. Kekuatan hubungan antar neuron yang berdekatan dipresentasikan dalam kekuatan hubungan atau bobot (Dharma et al., 2011).

Sebuah *Artificial Neural Network* (ANN) umumnya terdiri dari tiga layer, yaitu: *input layer*, *hidden layer* dan *output layer*. Layer input (*input layer*) terdiri dari neuron-neuron yang menerima sebuah input dari lingkungan luar. Input yang dimasukkan merupakan penggambaran dari suatu masalah. Layer tersembunyi (*hidden layer*) terdiri dari neuron-neuron yang menerima masukan dari input layer dan kemudian membawa output ke layer berikutnya. Lapisan *output* disebut unit-unit *output* yang terdiri dari neuron-neuron yang menerima output dari *hidden layer* dan mengirimkannya kepada pemakai.

2.2.7 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan pengembangan dari *multilayer perceptron* (MLP) yang didesain untuk mengolah data dua dimensi dalam bentuk citra. *Convolutional Neural Network* (CNN) ini termasuk kedalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Pada dasarnya klasifikasi citra dapat digunakan dengan *multilayer perceptron* (MLP), akan tetapi dengan metode *multilayer perceptron* (MLP) kurang sesuai untuk digunakan karena tidak menyimpan informasi spasial dari data citra dan menganggap setiap piksel adalah fitur yang independen sehingga menghasilkan hasil yang kurang baik. Penelitian awal yang mendasari penemuan *Convolutional Neural Network* (CNN) ini pertama kali dilakukan oleh Hubel dan Wiesel (Hubel & Wiesel, T, 1968) mengenai *virtual cortex* pada indera penglihatan kucing. Secara teknis, *Convolutional Neural Network* (CNN) adalah sebuah arsitektur yang dapat dilatih dan terdiri dari beberapa tahap. Masukan (*input*) dan keluaran (*output*) dari setiap tahap adalah terdiri dari beberapa *array* yang biasa disebut *feature map*. Setiap tahap terdiri dari tiga layer yaitu konvolusi, fungsi aktivasi layer dan pooling layer. Berikut adalah jaringan arsitektur *Convolutional Neural Network*:



Gambar 2. 1 *Convolutional Neural Network* (CNN)

Berdasarkan gambar diatas, Tahap pertama pada arsitektur *Convolutional Neural Network* (CNN) adalah tahap konvolusi. Tahap ini dilakukan dengan menggunakan sebuah kernel dengan ukuran tertentu. Perhitungan jumlah kernel yang dipakai tergantung dari jumlah fitur yang dihasilkan. Kemudian dilanjutkan menuju fungsi aktivasi, biasanya menggunakan fungsi aktivasi ReLU (*Rectifier Linear Unit*), Selanjutnya setelah keluar dari proses fungsi aktivasi kemudian melalui proses *pooling*. Proses ini diulang beberapa kali sampai didapatkan peta

fitur yang cukup untuk dilanjutkan ke *fully connected neural network*, dan dari *fully connected network* adalah *output class*.

2.2.7.1 Convolution Layer

Convolutional layer merupakan bagian dari tahap pada arsitektur *Convolutional Neural Network* (CNN). Tahap ini melakukan operasi konvolusi pada *output* dari layer sebelumnya. Layer tersebut adalah proses utama yang mendasari jaringan arsitektur *Convolutional Neural Network* (CNN). Konvolusi merupakan istilah matematis yang dimana pengaplikasian sebuah fungsi pada *output* fungsi lain secara berulang. Operasi konvolusi merupakan operasi pada dua fungsi argumen bernilai nyata. Operasi ini menerapkan fungsi *output* sebagai *Feature Map* dari input citra. *Input* dan *output* ini dapat dilihat sebagai dua argumen bernilai riil. Operasi konvolusi dapat dituliskan sebagai berikut:

$$s(t) = (x * t)(t) = \sum_{a=-\infty}^{\infty} x(a) * w(t - a) \quad (2.2)$$

Keterangan:

$s(t)$ = fungsi hasil operasi konvolusi

x = input

w = bobot (kernel)

Fungsi $s(t)$ memberikan *output* tunggal berupa *feature map*. Argumen pertama adalah *input* yang merupakan x dan argumen kedua w sebagai kernel atau filter. Apabila dilihat *input* sebagai citra dua dimensi, maka bisa dikatakan t sebagai piksel dan menggantinya dengan i dan j . Maka dari itu, operasi untuk konvolusi ke *input* dengan lebih dari satu dimensi dapat menulis sebagai berikut:

$$s(i, j) = (K * I)(i, j) = \sum_{\infty} \sum_{\infty} I(i - m, j - n)K(m, n)$$

$$s(i, j) = (K * I)(i, j) = \sum_{\infty} \sum_{\infty} I(i + m, j + n)K(m, n)$$

Berdasarkan kedua persamaan diatas merupakan perhitungan dasar dalam operasi konvolusi, dengan i dan j adalah sebuah piksel dari citra. Perhitungan tersebut bersifat komulatif dan muncul saat K sebagai kernel, kemudian I sebagai *input* kernel yang dapat dibalik relatif terhadap *input*. Sebagai alternatif operasi konvolusi dapat dilihat sebagai perkalian matriks antara citra masukan dan kernel

dimana keluarannya dihitung dengan *dot product*. Selain itu, penentuan volume *output* juga dapat ditentukan dari masing-masing lapisan dengan *hyperparameters*. *Hyperparameters* yang digunakan pada persamaan di bawah ini digunakan untuk menghitung banyaknya neuron aktivasi dalam sekali *output*. Perhatikan persamaan berikut

$$(W - F + 2P)/(S + 1)$$

Keterangan:

W = Ukuran volume gambar

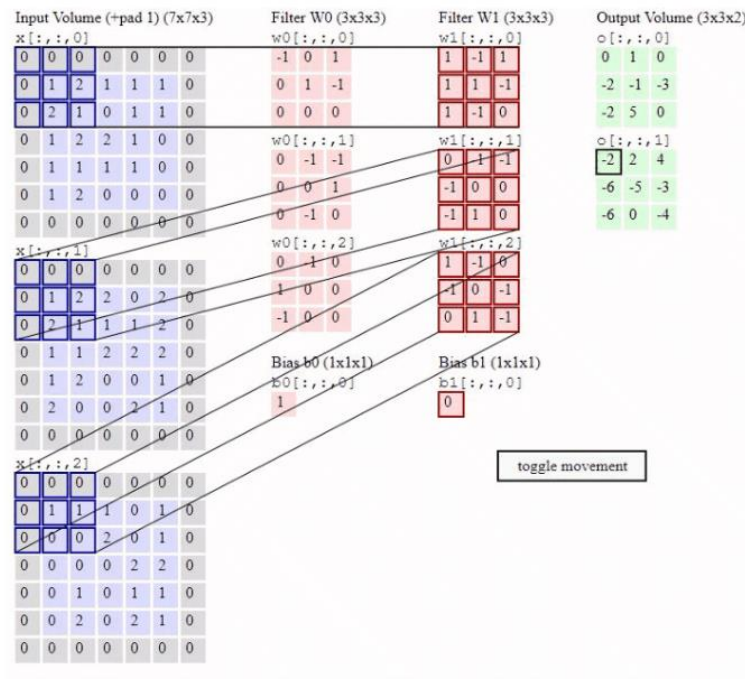
F= Ukuran filter

P= Nilai *padding* yang digunakan

S= Ukuran pergeseran (*Stride*)

Berdasarkan persamaan di atas, dapat dihitung ukuran spasial dari volume output dimana *hyperparameter* yang dipakai adalah ukuran volume (W), *filter* (F), *Stride* yang diterapkan (S) dan jumlah *padding* nol yang digunakan (P). *Stride* merupakan nilai yang digunakan untuk menggeser *filter* melalui *input* citra dan *zero padding* adalah nilai untuk mendapatkan angka nol di sekitar *border* citra.

Convolutional layer terdiri dari *neuron* yang tersusun sedemikian rupa sehingga membentuk sebuah filter dengan panjang dan tinggi (*pixels*). Sebagai contoh, layer pertama pada *feature extraction layer* biasanya adalah *convolutional layers* dengan ukuran 5x5x3. Panjang 5 *pixels*, tinggi 5 *pixels* dan tebal sejumlah 3 buah sesuai dengan *channel* dari *image* tersebut. Ketiga filter ini akan digeser keseluruhan bagian dari gambar. Setiap pergeseran akan dilakukan operasi “dot” antara input dan nilai dari filter tersebut sehingga menghasilkan sebuah output atau biasa disebut sebagai *activation map* atau *feature map*. Perhatikan ilustrasi berikut:



Gambar 2. 2 Convolution Layer

2.2.7.2 Operasi Pooling

Pooling merupakan pengurangan ukuran matriks dengan menggunakan operasi *pooling*. *Pooling layer* biasanya berada setelah *convolutional layer*. Pada dasarnya *pooling layer* terdiri dari sebuah filter dengan ukuran dan *stride* tertentu yang akan secara bergantian bergeser pada seluruh area *feature map*. Dalam *pooling layer* terdapat dua macam *pooling* yang biasa digunakan yaitu *average pooling* dan *max-pooling*. Nilai yang diambil pada *average pooling* adalah nilai rata-rata, sedangkan pada *max-pooling* adalah nilai maksimal. Lapisan *pooling* yang dimasukkan diantara lapisan konvolusi secara berturut-turut dalam arsitektur model *Convolutional Neural Network* (CNN) dapat secara progsrif meurangi ukuran pada *feature map*, sehingga mengurangi jumlah parameter dan perhitungan di jaringan untuk mengendalikan *overfitting*. Lapisan *pooling* bekerja di setiap tumpukan *feature map* dan melakukan pengurangan pada ukurannya. Bentuk lapisan *pooling* umumnya dengan menggunakan filter dengan ukuran 2x2 yang diaplikasikan dengan langkah sebanyak dua dan beroperasi pada setiap irisan dari inputnya. Berikut ini adalah contoh gambar operasi *max-pooling*:

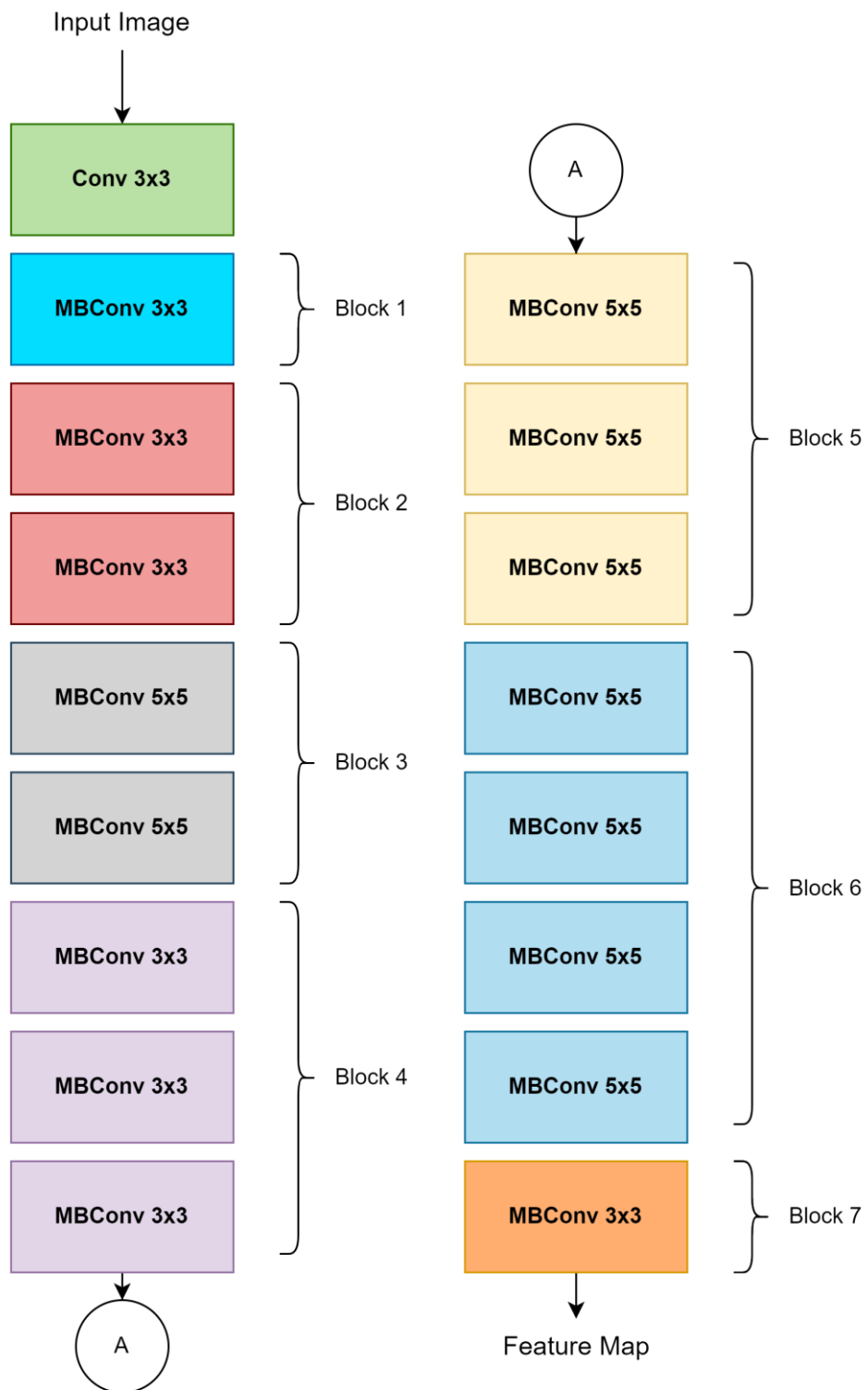
Tan dan Quoc V. Le. Mereka menggunakan model *scaling*, yaitu melakukan *scaling* pada model yang telah ada dalam hal *model depth*, *model width*, dan resolusi gambar untuk memperbaiki performa model. Melakukan *scaling* pada ketiganya secara bersamaan ini disebut dengan metode *compound scaling* (Tan & Le, n.d.). *EfficientNet-B0* merupakan versi terkecil dari *EfficientNet*. Berikut merupakan gambar arsitektur B0:

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels \hat{C}_i	#Layers \hat{L}_i
1	Conv3x3	224×224	32	1
2	MBCConv1, k3x3	112×112	16	1
3	MBCConv6, k3x3	112×112	24	2
4	MBCConv6, k5x5	56×56	40	2
5	MBCConv6, k3x3	28×28	80	3
6	MBCConv6, k5x5	28×28	112	3
7	MBCConv6, k5x5	14×14	192	4
8	MBCConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

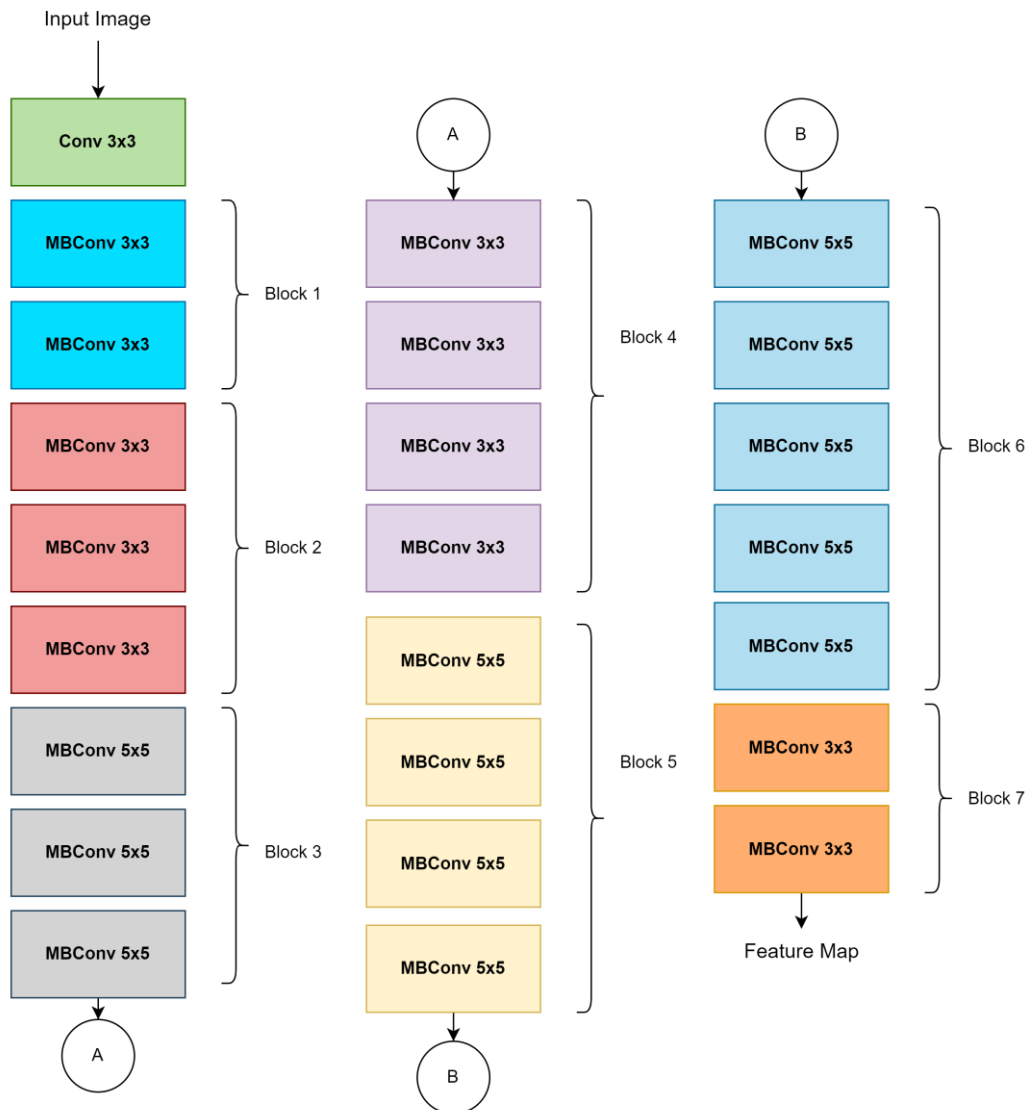
Gambar 2. 4 Baseline jaringan EfficientNet-B0

Blok MBCConv merupakan blok *inverted residuals structure* yang terdapat pada arsitektur *MobileNetV2*, dengan pengaturan yang berbeda. Dari *baseline* ini, mereka mengaplikasikan metode *compound scaling* dan mendapatkan arsitektur *EfficientNet-B0* sampai B7.

Dari ketujuh arsitektur *EfficientNet*, setiap *EfficientNet* memiliki jumlah layer pada setiap blok yang berbeda. Berikut merupakan perbedaan layer dari setiap *EfficientNet*.



Gambar 2. 5 Struktur Layer *EfficientNetB0*



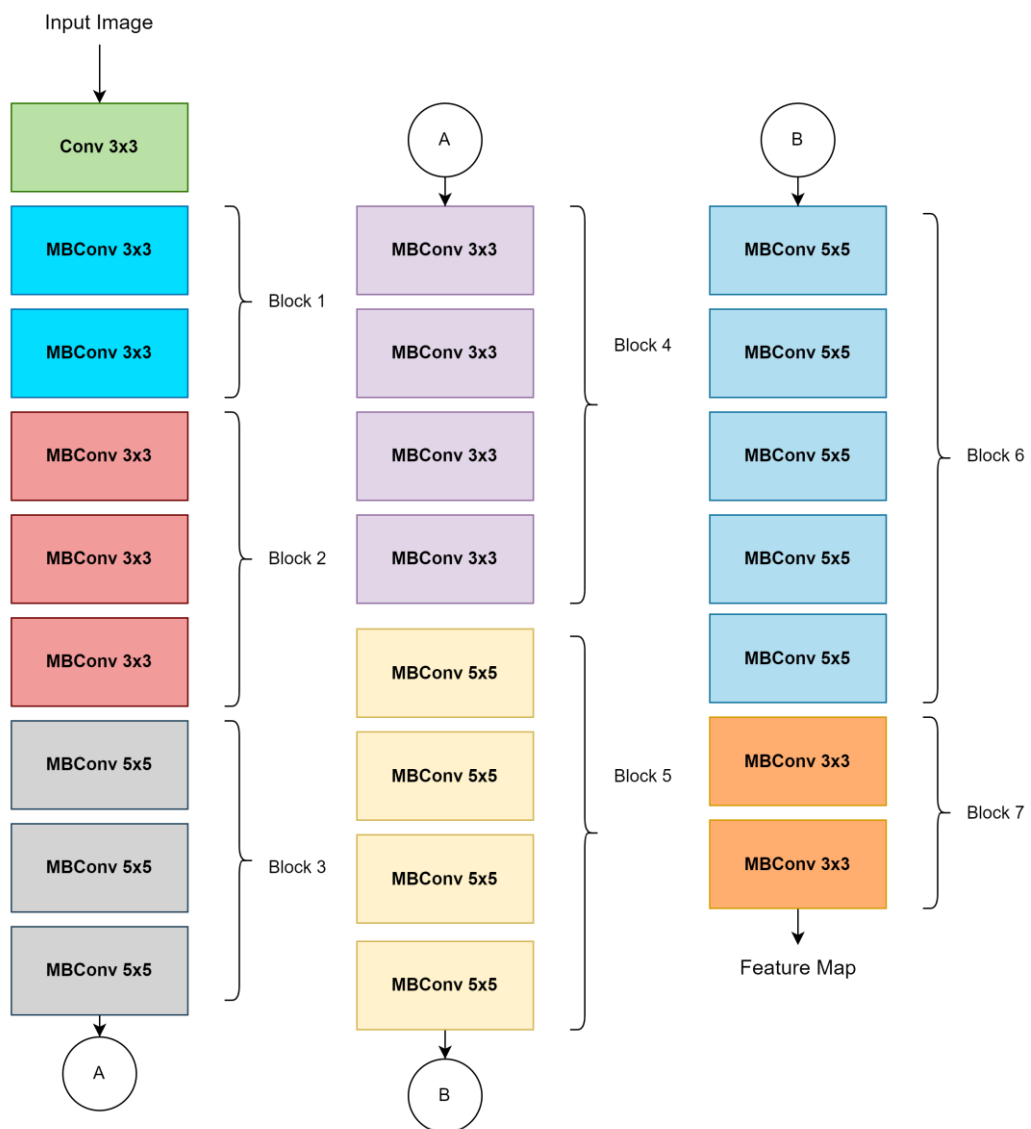
Gambar 2. 6 Struktur Layer *EfficientNetB1*

Di dalam setiap blok MBConv, model *EfficientNetB1* memiliki beberapa jumlah layer yang berbeda, berikut merupakan penjelasan dari setiap blok MBConv pada model *EfficientNetB1*

Tabel 2. 2 Baseline Jaringan *EfficientNetB1*

Stage	Operator	Resolution	Channels	Layers
1	Conv3x3	112 x 112	32	1
2	MBConv1 3x3	112 x 112	32	2

3	MBCConv6 3x3	112 x 112	96	3
4	MBCConv6 5x5	56 x 56	144	3
5	MBCConv6 3x3	28 x 28	240	4
6	MBCConv6 5x5	14 x 14	480	4
7	MBCConv6 5x5	14 x 14	672	5
8	MBCConv6 3x3	7 x 7	1152	2
9	Conv1x1 & Pooling & FC	7 x 7	1280	1



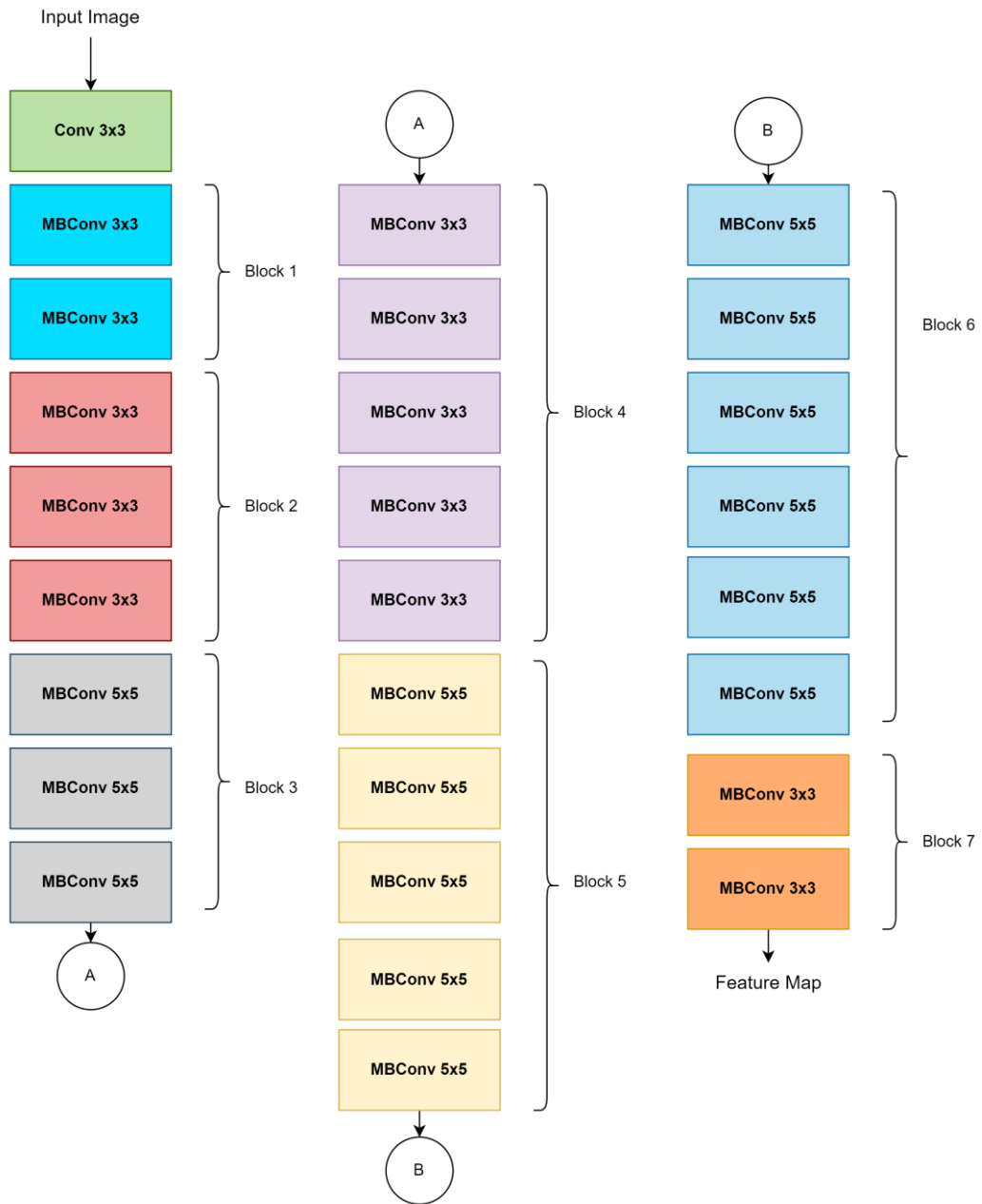
Gambar 2. 7 Struktur Layer *EfficientNetB2*

Di dalam setiap blok MBConv, model *EfficientNetB2* memiliki beberapa jumlah layer yang berbeda, berikut merupakan penjelasan dari setiap blok MBConv pada model *EfficientNetB2*.

Tabel 2. 3 Baseline Jaringan *EfficientNetB2*

Stage	Operator	Resolution	Channels	Layers
1	Conv3x3	112 x 112	32	1
2	MBConv1 3x3	112 x 112	32	2

3	MBCConv6 3x3	112 x 112	96	3
4	MBCConv6 5x5	56 x 56	144	3
5	MBCConv6 3x3	28 x 28	288	4
6	MBCConv6 5x5	14 x 14	528	4
7	MBCConv6 5x5	14 x 14	720	5
8	MBCConv6 3x3	7 x 7	1248	2
9	Conv1x1 & Pooling & FC	7 x 7	1408	1



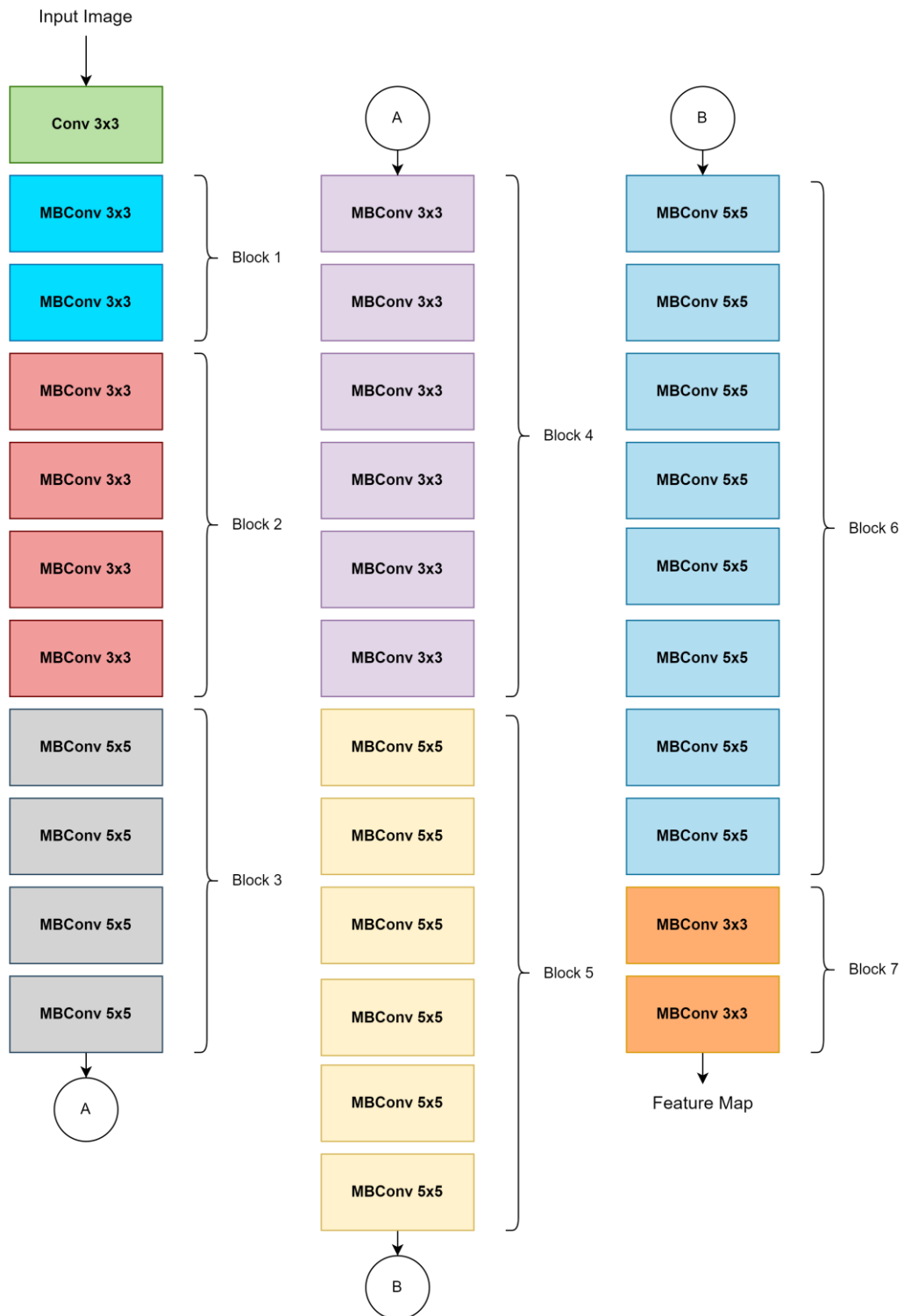
Gambar 2. 8 Struktur Layer *EfficientNetB3*

Di dalam setiap blok MBConv, model *EfficientNetB3* memiliki beberapa jumlah layer yang berbeda, berikut merupakan penjelasan dari setiap blok MBConv pada model *EfficientNetB3*.

Tabel 2. 4 Baseline Jaringan *EfficientNetB3*

Stage	Operator	Resolution	Channels	Layers
1	Conv3x3	112 x 112	40	1

2	MBCConv1 3x3	112 x 112	40	2
3	MBCConv6 3x3	112 x 112	14	3
4	MBCConv6 5x5	56 x 56	192	3
5	MBCConv6 3x3	28 x 28	288	5
6	MBCConv6 5x5	14 x 14	576	5
7	MBCConv6 5x5	14 x 14	816	6
8	MBCConv6 3x3	7 x 7	1392	2
9	Conv1x1 & Pooling & FC	7 x 7	1536	1

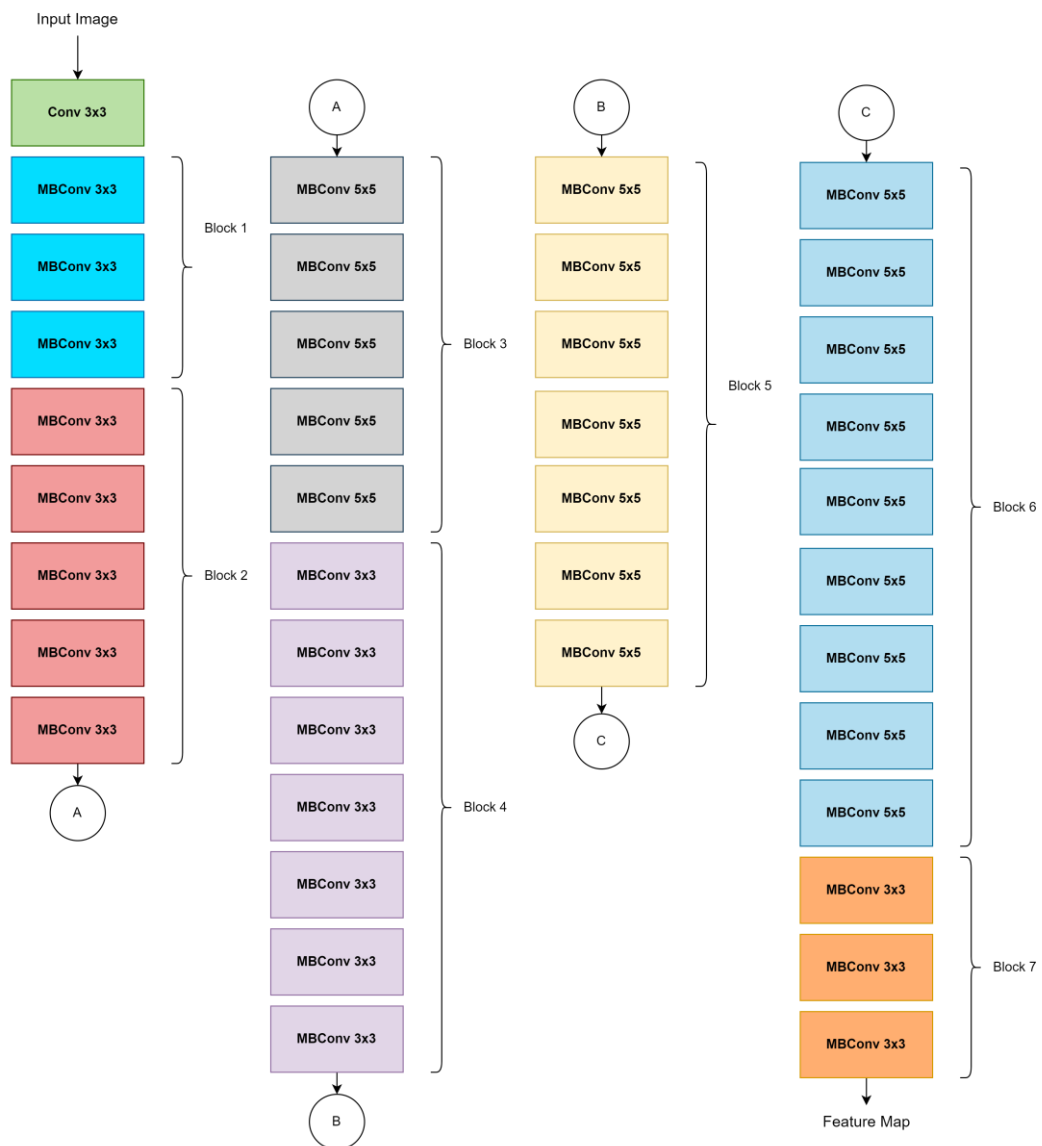


Gambar 2. 9 Struktur Layer *EfficientNetB4*

Di dalam setiap blok MBConv, model *EfficientNetB4* memiliki beberapa jumlah layer yang berbeda, berikut merupakan penjelasan dari setiap blok MBConv pada model *EfficientNetB4*.

Tabel 2. 5 Baseline Jaringan *EfficientNetB4*

Stage	Operator	Resolution	Channels	Layers
1	Conv3x3	112 x 112	48	1
2	MBCConv1 3x3	112 x 112	48	2
3	MBCConv6 3x3	112 x 112	14	4
4	MBCConv6 5x5	56 x 56	192	4
5	MBCConv6 3x3	28 x 28	336	6
6	MBCConv6 5x5	14 x 14	672	6
7	MBCConv6 5x5	14 x 14	960	8
8	MBCConv6 3x3	7 x 7	1632	2
9	Conv1x1 & Pooling & FC	7 x 7	1792	1

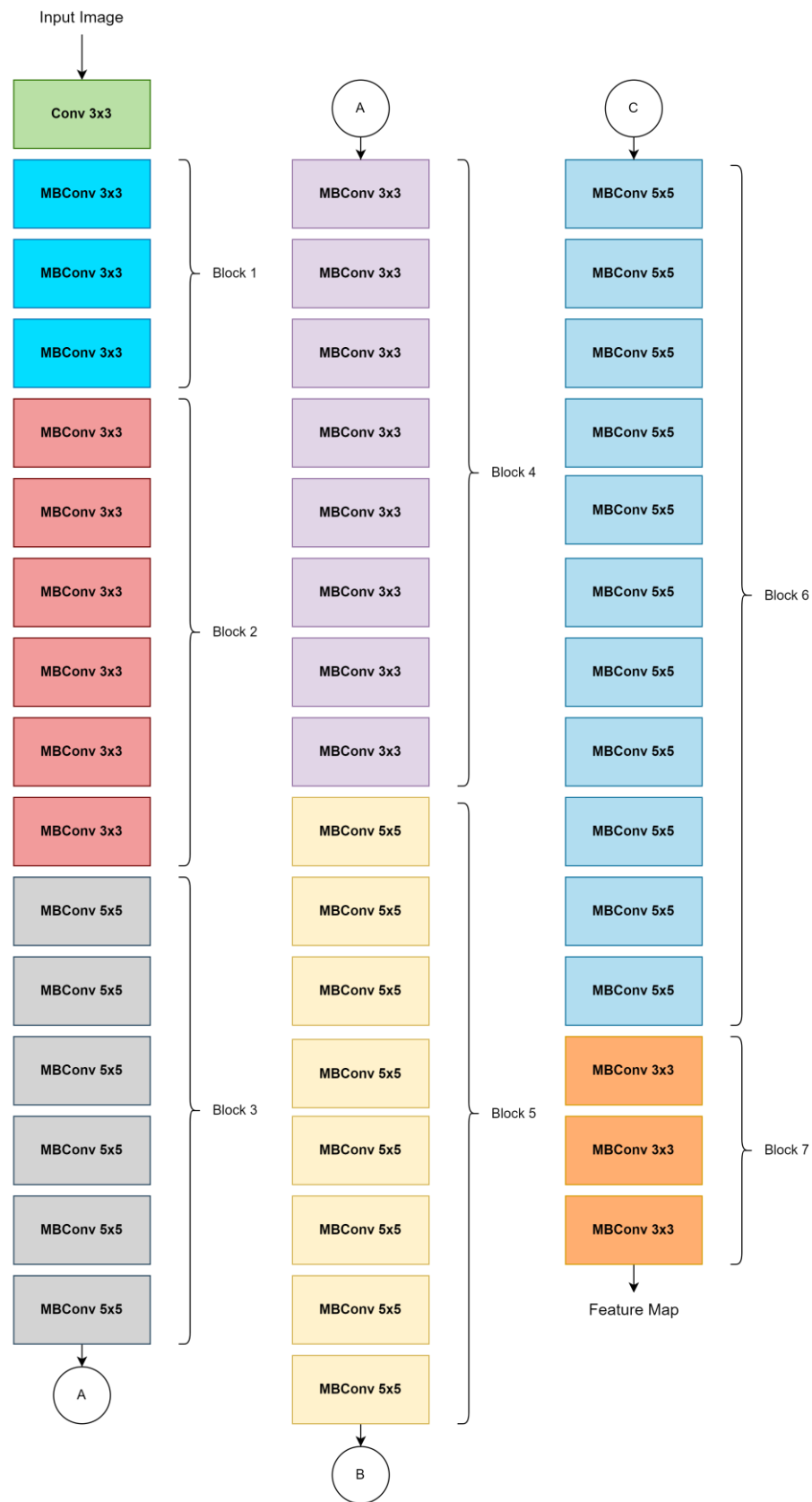


Gambar 2. 10 Struktur Layer *EfficientNetB5*

Di dalam setiap blok MBConv, model *EfficientNetB5* memiliki beberapa jumlah layer yang berbeda, berikut merupakan penjelasan dari setiap blok MBConv pada model *EfficientNetB5*.

Tabel 2. 6 Baseline Jaringan *EfficientNetB5*

Stage	Operator	Resolution	Channels	Layers
1	Conv3x3	112 x 112	48	1
2	MBCConv1 3x3	112 x 112	48	3
3	MBCConv6 3x3	112 x 112	14	5
4	MBCConv6 5x5	56 x 56	240	5
5	MBCConv6 3x3	28 x 28	384	7
6	MBCConv6 5x5	14 x 14	768	7
7	MBCConv6 5x5	14 x 14	1056	9
8	MBCConv6 3x3	7 x 7	1824	3
9	Conv1x1 & Pooling & FC	7 x 7	2048	1

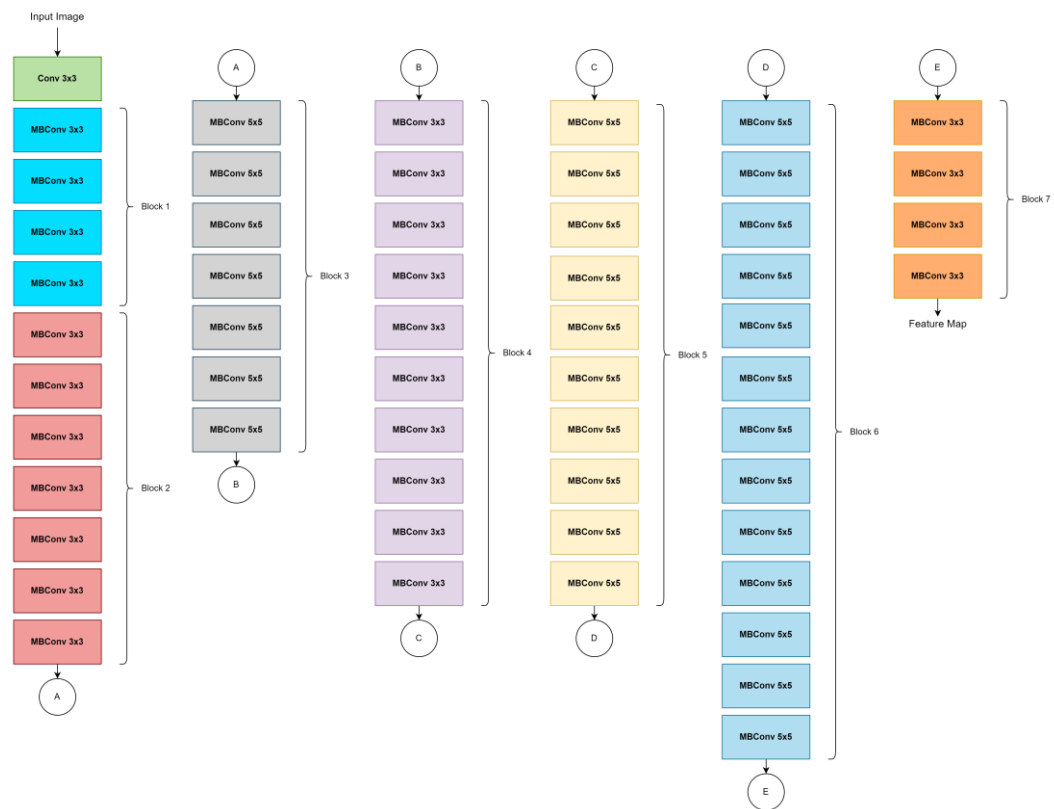


Gambar 2. 11 Struktur Layer *EfficientNetB6*

Di dalam setiap blok MBConv, model *EfficientNetB6* memiliki beberapa jumlah layer yang berbeda, berikut merupakan penjelasan dari setiap blok MBConv pada model *EfficientNetB6*.

Tabel 2. 7 Baseline Jaringan *EfficientNetB6*

Stage	Operator	Resolution	Channels	Layers
1	Conv3x3	112 x 112	56	1
2	MBConv1 3x3	112 x 112	56	3
3	MBConv6 3x3	112 x 112	19	6
4	MBConv6 5x5	56 x 56	240	6
5	MBConv6 3x3	28 x 28	432	8
6	MBConv6 5x5	14 x 14	864	8
7	MBConv6 5x5	14 x 14	1200	11
8	MBConv6 3x3	7 x 7	2064	3
9	Conv1x1 & Pooling & FC	7 x 7	2304	1



Gambar 2. 12 Struktur Layer *EfficientNetB7*

Di dalam setiap blok MBConv, model *EfficientNetB7* memiliki beberapa jumlah layer yang berbeda, berikut merupakan penjelasan dari setiap blok MBConv pada model *EfficientNetB7*.

Tabel 2. 8 Baseline Jaringan *EfficientNetB7*

Stage	Operator	Resolution	Channels	Layers
1	Conv3x3	112 x 112	64	1
2	MBConv1 3x3	112 x 112	64	4
3	MBConv6 3x3	112 x 112	19	7
4	MBConv6 5x5	56 x 56	288	7
5	MBConv6 3x3	28 x 28	480	10

6	MBCConv6 5x5	14 x 14	960	10
7	MBCConv6 5x5	14 x 14	1344	13
8	MBCConv6 3x3	7 x 7	2304	4
9	Conv1x1 & Pooling & FC	7 x 7	2560	1

2.2.9 Phyton

Python adalah bahasa pemrograman tingkat tinggi. Python dibuat oleh Guido van Rossum di Centrum Wiskunde & Informatica (CWI), Belanda dan pertama kali dirilis pada Tahun 1991. Python dapat dipergunakan untuk proyek skala kecil ataupun besar. Python saat ini sudah mencapai versi 3x dan dapat digunakan untuk berbagai kebutuhan seperti web development, GUI development, scientific, software development, dan system administration (Pandika Pinata et al., 2020).