

## **BAB IV**

### **HASIL PENELITIAN DAN PEMBAHASAN**

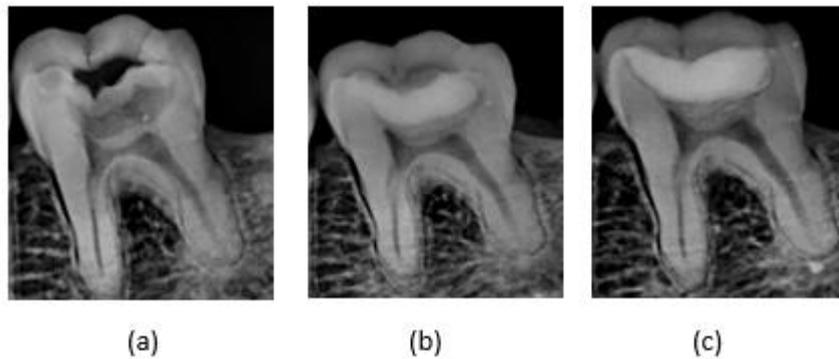
#### **4.1 Prinsip Kerja Program**

Prinsip kerja program yaitu yang pertama dengan melakukan pengambilan citra setelah itu melakukan pemotongan citra dari rontgen pasien dibagian dentin tersier yang sesuai dengan kategori *edge* dan *non edge* yang digunakan sebagai data penelitian. Kumpulan citra tersebut dibagi menjadi citra latih dan citra uji dengan metode *Convolutional Neural Network* dari *deep learning* digunakan untuk melakukan proses latih dan uji sehingga membentuk area klasifikasi. Dalam tahap uji, CNN diimplementasikan dengan lima lapisan agar mencapai akurasi yang lebih tinggi. Pada tahap uji, citra yang sudah di-*replace* dengan kategori *sample edge* dan *non edge* yang sudah dibuat akan tertanda secara otomatis untuk bagian tepinya.

#### **4.2 Hasil Pengambilan Citra**

Data citra yang akan digunakan untuk penelitian ini adalah citra rontgen terhadap pasien yang akan melakukan perawatan kaping pulpa di RSGM Universitas Muhammadiyah Yogyakarta, lalu citra rontgen dalam bentuk *file* nantinya akan diserahkan kepada Fakultas Teknik Universitas Muhammadiyah Yogyakarta guna melakukan penelitian yang akan dilakukan.

Data yang sudah diperoleh lalu dilakukan pemilihan terhadap citra-citra yang akan digunakan sebagai bahan penelitian, karena tidak semua data citra terdapat masing-masing indikasi, K1, dan K2. Total citra yang didapat sebanyak 237 citra yang diambil dari 79 pasien. Dari hasil pengambilan citra pasien yang melakukan perawatan sangat bermacam-macam jenis giginya ataupun bagian karies pada giginya. Oleh karena itu penentuan bagian mana dentin tersier, atau bagian kamar pulpa sangat penting dan diperlukan tenaga ahli yaitu dokter gigi yang menentukan bagian-bagian tersebut. Dapat dilihat pada gambar 4.1 adalah hasil data yang sudah diambil.



**Gambar 4. 1** (a) Citra rontgen indikasi gigi pasien, (b) citra rontgen K1 (kontrol 1), (c) citra rontgen K2 (kontrol 2)

### 4.3 Sampling Klasifikasi

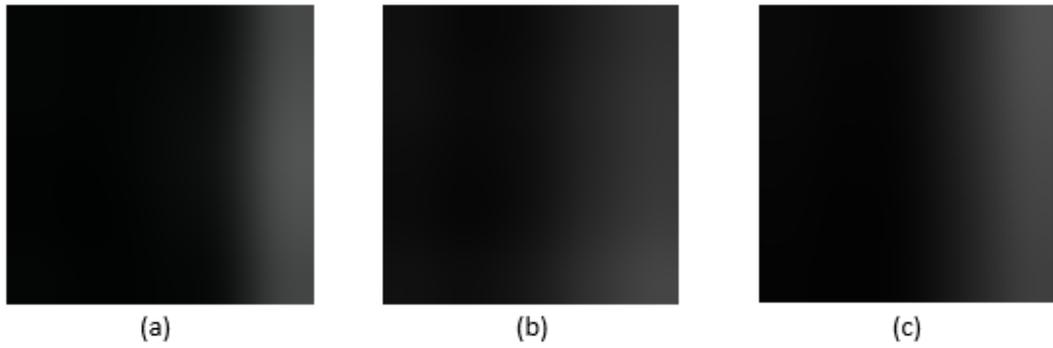
Sampling klasifikasi adalah proses dimana citra yang telah didapat dipotong dibagian dentin tersier sesuai dari bentuk kategori *edge* dan *non edge*. Citra yang dipotong masih dilakukan secara manual karena jika dilakukan oleh sistem hasil dari pengelompokan setiap kategori pasti tidak sesuai. Untuk data penelitian setiap kategori *edge* dan *non edge* akan membutuhkan 20 citra. Setiap jenis rontgen perawatan (indikasi,K1,K2) dilakukan pemotongan citra yang berbeda karena setiap jenis rontgen yang sudah dilakukan akan mengalami perubahan warna. Setiap rontgen perawatan terdapat 11 folder dimana semua itu adalah folder yang sesuai dengan kategori *edge* dan *non edge*. Jadi dalam penelitian terdapat 660 citra yang sudah dipotong sesuai kategori *edge* dan *non edge* untuk masing-masing jenis rontgen perawatan. Setiap citra yang dipotong harus berukuran sama yaitu 5x5 piksel karena menggunakan metode klasifikasi CNN. Pemotongan dan pengelompokkan dari citra ini bertujuan untuk dijadikan data pada saat proses *training* dilakukan.

#### a. Indikasi

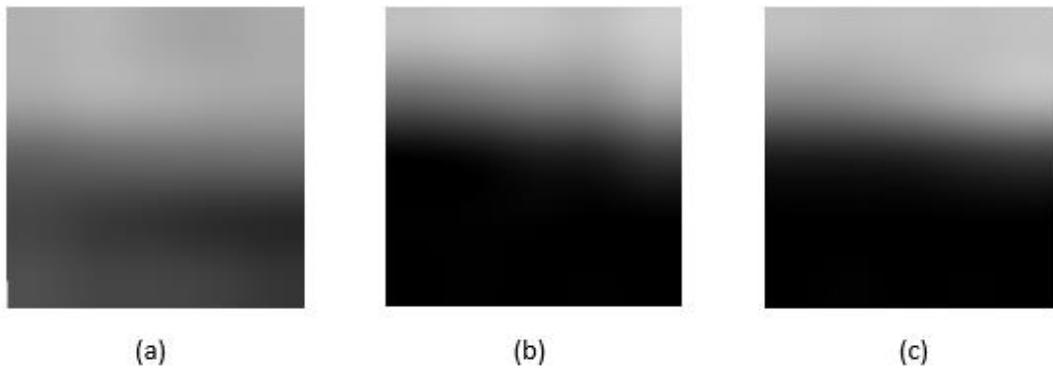
Indikasi adalah rontgen yang diambil kepada pasien tepat sebelum melakukan perawatan kaping pulpa. Pada rontgen indikasi dipotong pada bagian karies gigi untuk melakukan pemotongan kategori *edge* dan *non edge*. *Edge* dan *non edge* ada 11 kategori jadi folder yang dibuat sejumlah 11. Masing-masing folder diisi dengan

20 citra yang didapat dari pemotongan citra berdasarkan kategori *edge* dan *non edge*.

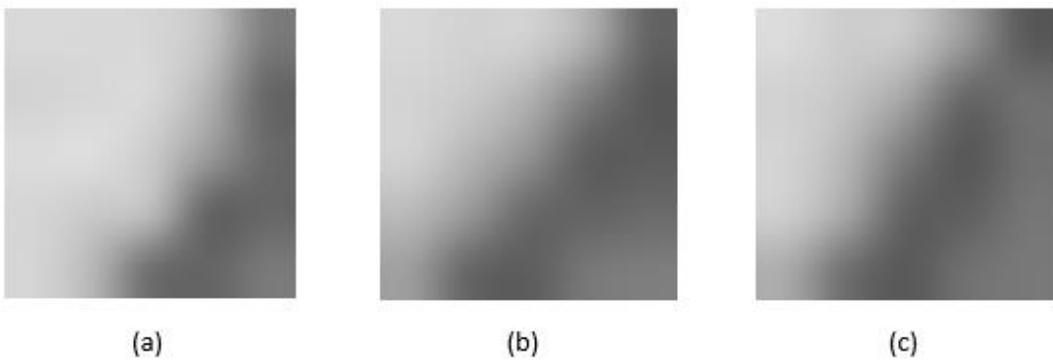
Kategori khusus bagian *edge* memiliki jumlah 8 diantaranya adalah kategori pertama yang memiliki garis lurus horizontal dengan bagian hitam berada di sebelah kiri dan bagian putih berada di kanan, dapat dilihat pada gambar 4.2. Kategori kedua memiliki garis lurus vertikal dengan bagian hitam berada dibawah dan bagian putih diatas, dapat dilihat pada gambar 4.3. Kategori ketiga adalah garis miring yang bagian putih di atas kiri dan bagian hitam di bawah kanan, dapat dilihat pada gambar 4.4. Kategori keempat dengan garis miring bagian putih di atas kanan dan bagian kiri di atas kiri, dapat dilihat pada gambar 4.5. Kategori kelima dengan garis lurus horizontal bagian putih di kiri dan bagian hitam di kanan, dapat dilihat pada gambar 4.6. Kategori keenam yang merupakan garis lurus vertikal dengan bagian putih berada dibawah dan bagian hitam berada diatas, dapat dilihat pada gambar 4.7. Kategori ketujuh merupakan garis miring dengan bagian hitam berada di kiri atas dan bagian putih di kanan bawah, dapat dilihat pada gambar 4.8. Kategori ke delapan dengan garis miring bagian hitam di kanan atas dan bagian putih di kiri bawah, dapat dilihat pada gambar 4.9. Untuk kategori *non edge* adalah kategori kesembilan yang hanya memiliki warna putih, dapat dilihat pada gambar 4.10. Kategori ke sepuluh yang memiliki warna hitam, dapat dilihat pada gambar 4.11. Kategori ke sebelas memiliki warna abu-abu, dapat dilihat pada gambar 4.12.



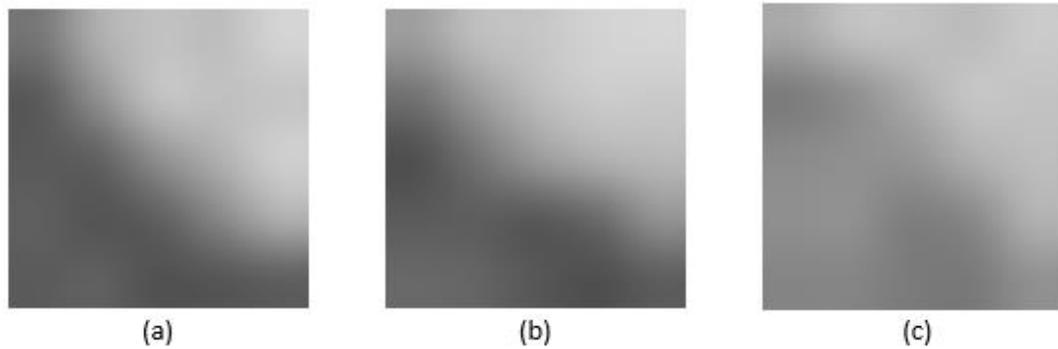
**Gambar 4. 2** (a), (b), (c) citra hasil crop kategori 1 (edge)



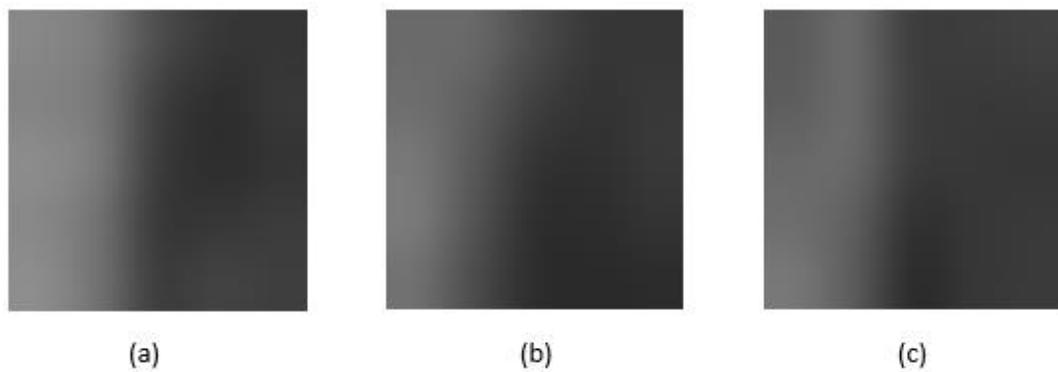
**Gambar 4. 3** (a), (b), (c) citra hasil crop kategori 2 (edge)



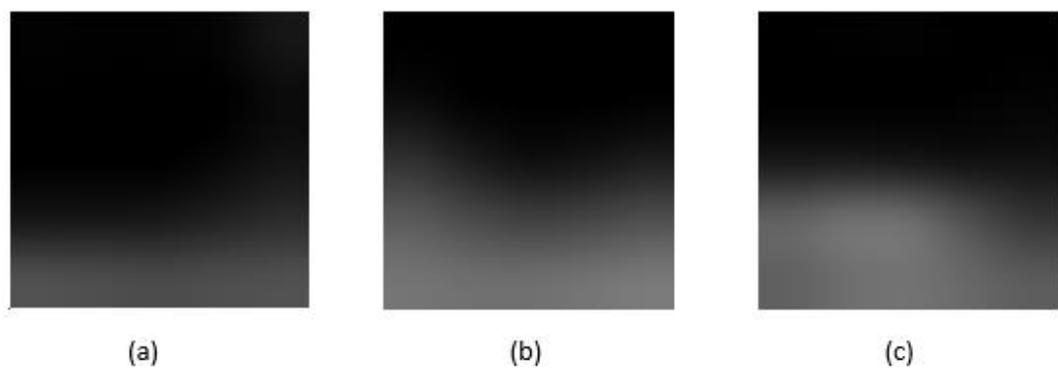
**Gambar 4. 4** (a), (b), (c) citra hasil crop kategori 3 (edge)



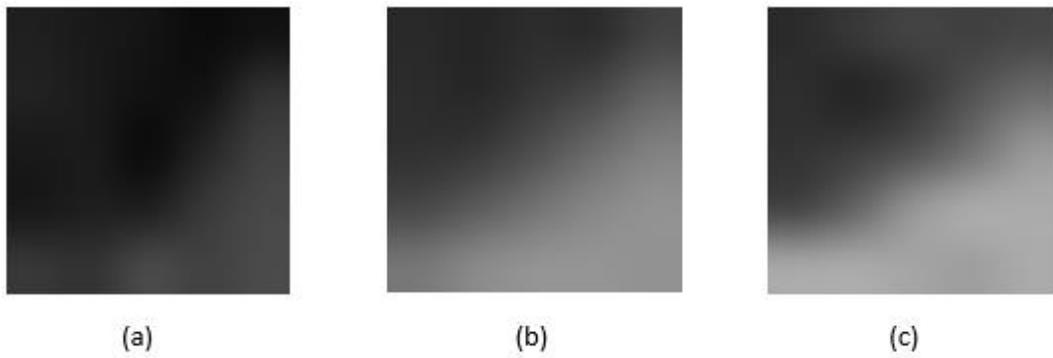
**Gambar 4. 5** (a), (b), (c) citra hasil crop kategori 4 (edge)



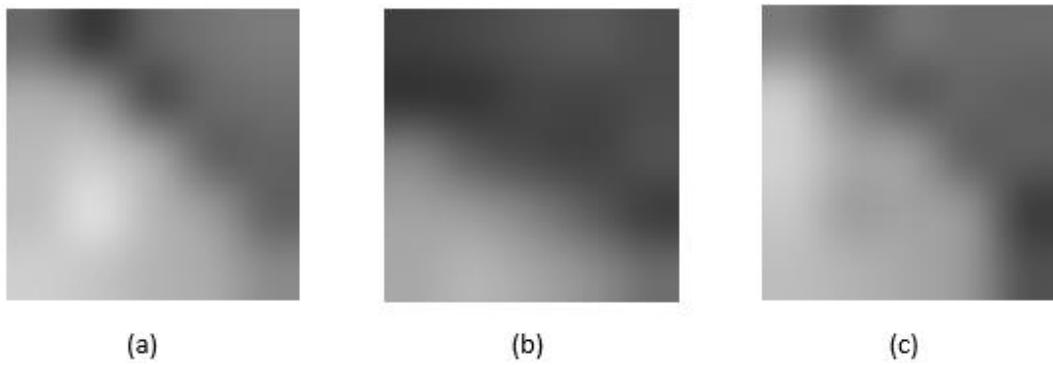
**Gambar 4. 6** (a), (b), (c) citra hasil crop kategori 5 (edge)



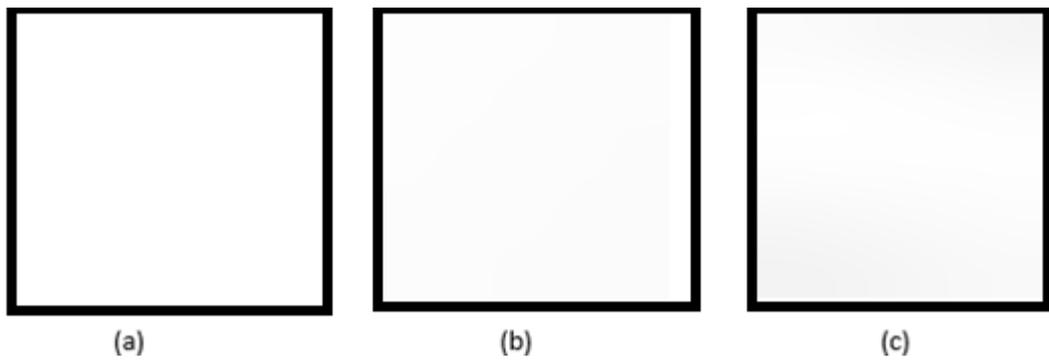
**Gambar 4. 7** (a), (b), (c) citra hasil crop kategori 6 (edge)



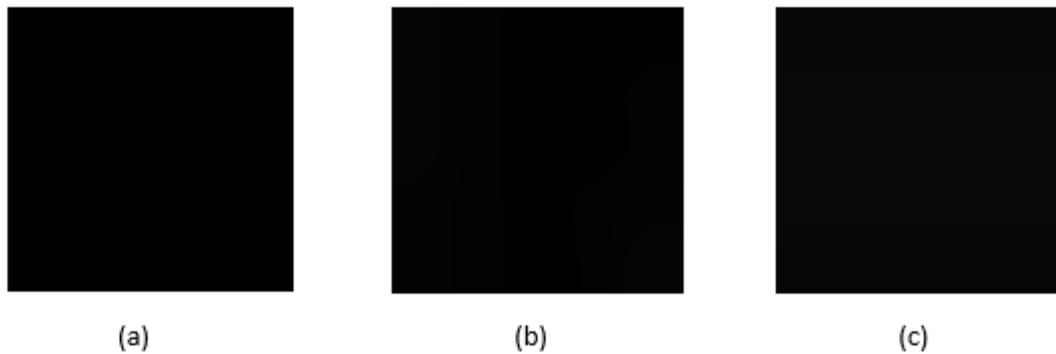
**Gambar 4. 8** (a), (b), (c) citra hasil crop kategori 7 (edge)



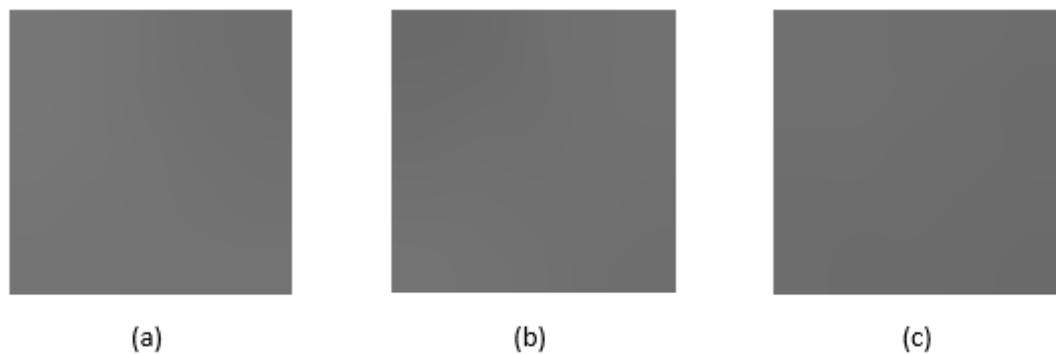
**Gambar 4. 9** (a), (b), (c) citra hasil crop kategori 8 (edge)



**Gambar 4. 10** (a), (b), (c) citra hasil crop kategori 9 (non edge)



**Gambar 4. 11** (a), (b), (c) citra hasil crop kategori 10 (*non edge*)



**Gambar 4. 12** (a), (b), (c) citra hasil crop kategori 11 (*non edge*)

**b. K1 (Kontrol 1)**

K1 adalah rontgen yang diambil kepada pasien yang sudah melakukan perawatan kaping pulpa. Sama seperti indikasi, K1 juga dipotong pada bagian karies gigi untuk melakukan pemotongan kategori *edge* dan *non edge*. *Edge* dan *non edge* ada 11 kategori jadi folder yang dibuat sejumlah 11. Masing-masing folder diisi dengan 20 citra yang didapat dari pemotongan citra berdasarkan kategori *edge* dan *non edge*. Citra hasil pemotongan menyerupai seperti citra hasil pemotongan pada indikasi.

**c. K2 (Kontrol 2)**

K2 adalah rontgen yang diambil kepada pasien yang sudah 1 bulan setelah melakukan perawatan kaping pulpa. Sama seperti indikasi, dan K1, K2 juga dipotong pada bagian karies gigi untuk melakukan pemotongan kategori *edge* dan

*non edge*. *Edge* dan *non edge* ada 11 kategori jadi folder yang dibuat sejumlah 11. Masing-masing folder diisi dengan 20 citra yang didapat dari pemotongan citra berdasarkan kategori *edge* dan *non edge*. Citra hasil pemotongan menyerupai seperti citra hasil pemotongan pada indikasi.

#### 4.4 Training Data

*Training* data dalam penelitian ini dilakukan untuk melatih *deep learning* agar bisa mengklasifikasi citra sample kategori pada saat proses *testing*. Dalam tahap *training*, jumlah *layer* yang digunakan terdiri atas *image input layer* yang digunakan sebagai *input layer*, lapisan tepat setelah *input layer* disebut juga *convolution layer* karena terbentuk dari hasil konvolusi *input layer*, letak layer selanjutnya adalah *maxPooling layer* dimana pada lapisan ini dilakukan penyederhanaan informasi yang merupakan *output* dari *convolutional layer*, selanjutnya ada lapisan *fully connected layer* yang didapat dari masing-masing komponen matriks dari matriks keluaran lapisan konvolusi, *softmax layer* sebagai layer aktivasi, serta *classification output layer* sebagai *output*.

Selain itu dalam penelitian ini menggunakan parameter pengaturan *training* yang sama yaitu nilai maksimum *epoch* adalah 50 yang artinya menandakan siklus *deep learning* dari seluruh data *training* dan ukuran *mini batch* adalah 5. Pada tahapan ini terdiri dari proses *feedforward* dan proses *backpropagation*. Dalam memulai suatu proses *feedforward* diperlukan jumlah dan ukuran layer yang dibentuk, ukuran *max pooling*.

Proses *feedforward* bekerja dimana citra vektor akan melalui proses konvolusi dan *max pooling* untuk mereduksi ukuran citranya dan memperbanyak neuronnya. Sehingga nantinya akan terbentuk banyak jaringan yang mana menambah varian data untuk dipelajari oleh sistem. Hasil proses ini berupa bobot yang akan digunakan untuk mengevaluasi proses *deep learning*. Proses ini akan menghasilkan beberapa lapisan untuk mengklasifikasi data citra yang mana menggunakan bobot dan bias yang telah diperbarui dari proses *backpropagation*. Proses ini juga akan digunakan kembali pada saat tahap *testing*.

Setelah proses *feedforward*, proses selanjutnya adalah *backpropagation* dimana di proses ini akan di *trace* kesalahan dari lapisan *output* sampai lapisan pertama. Untuk menandai bahwa data tersebut telah di *trace* diperoleh bobot dan bias yang baru. Perhitungan gradien untuk CNN merupakan proses untuk memperoleh nilai bobot dan bias baru yang akan diperlukan saat *training*.

Dalam penelitian ini ada 20 citra dimana 15 citra dijadikan data latih proses *training* dilakukan dengan melatih 15 citra yang sudah di *crop* dan dimasukkan kedalam folder khusus untuk *training*. Dimana folder itu sendiri hanya dibuat untuk melatih data citra. Folder dibuat sesuai dengan jumlah kategori *edge* dan *non edge*. Sehingga ada 11 folder untuk di *training* di setiap jenis rontgennya.

Proses *training* dilakukan menggunakan masing-masing folder dari jenis rontgen perawatan kaping pulpa. Setiap folder jenis rontgen akan melalui proses tahapan 1 kali latihan bersamaan dengan pengujian. Pengklasifikasian yang dilakukan pada proses uji merupakan hasil dari proses pembelajaran pada tahap latih.

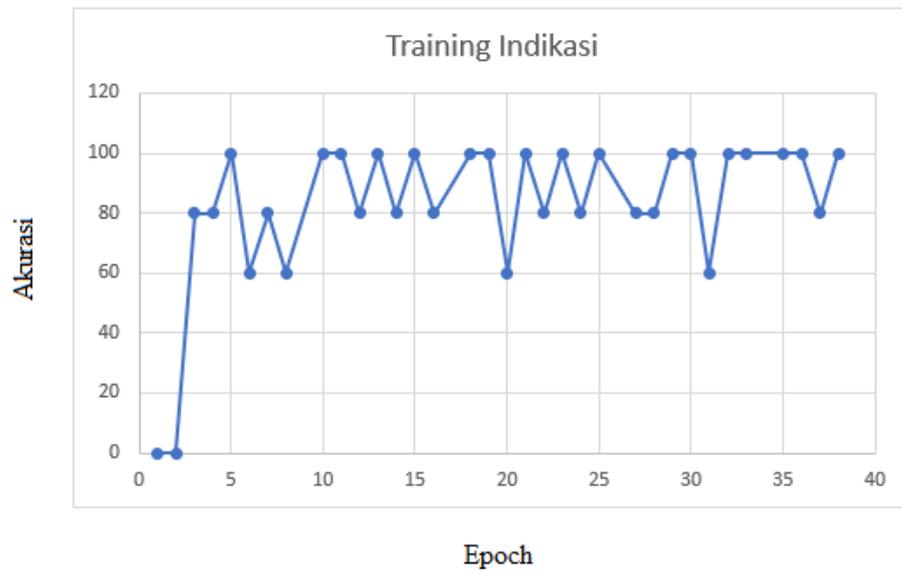
#### a. Hasil *training* Indikasi

Di dalam folder indikasi, disetiap kategori *edge* dan *non edge* terdapat 15 citra yang akan di *training*. Hasil akurasi training dari folder indikasi dapat dilihat pada gambar 4.13.

Epoch	Iteration	Time Elapsed (seconds)	Mini-batch Loss	Mini-batch Accuracy	Base Learning Rate
1	1	17.26	2.5043	0.00%	1.00e-04
2	50	19.34	2.1109	0.00%	1.00e-04
3	100	20.60	1.7138	80.00%	1.00e-04
4	150	21.44	0.5930	80.00%	1.00e-04
5	200	22.30	0.4171	100.00%	1.00e-04
6	250	23.36	1.4871	60.00%	1.00e-04
7	300	24.40	0.7695	80.00%	1.00e-04
8	350	25.35	0.6475	60.00%	1.00e-04
10	400	26.20	0.1538	100.00%	1.00e-04
11	450	27.00	0.2320	100.00%	1.00e-04
12	500	27.82	0.5806	80.00%	1.00e-04
13	550	28.74	0.5466	100.00%	1.00e-04
14	600	29.53	0.4429	80.00%	1.00e-04
15	650	30.31	0.0756	100.00%	1.00e-04
16	700	31.10	0.8092	80.00%	1.00e-04
18	750	31.94	0.3380	100.00%	1.00e-04
19	800	32.87	0.1122	100.00%	1.00e-04
20	850	33.68	3.3788	60.00%	1.00e-04
21	900	34.48	0.2130	100.00%	1.00e-04
22	950	35.28	0.4964	80.00%	1.00e-04
23	1000	36.24	0.1688	100.00%	1.00e-04
24	1050	37.22	0.3552	80.00%	1.00e-04
25	1100	38.21	0.1356	100.00%	1.00e-04
27	1150	39.24	0.3282	80.00%	1.00e-04
28	1200	40.28	1.5369	80.00%	1.00e-04
29	1250	41.53	0.2958	100.00%	1.00e-04
30	1300	42.64	0.1033	100.00%	1.00e-04
31	1350	43.61	0.6240	60.00%	1.00e-04
32	1400	44.78	0.1976	100.00%	1.00e-04
33	1450	45.84	0.2135	100.00%	1.00e-04
35	1500	46.85	0.0577	100.00%	1.00e-04
36	1550	47.97	0.1900	100.00%	1.00e-04
37	1600	48.77	0.3129	80.00%	1.00e-04
38	1650	49.60	0.4625	100.00%	1.00e-04
39	1700	50.38	1.3902	60.00%	1.00e-04
40	1750	51.24	0.1108	100.00%	1.00e-04
41	1800	52.02	0.4395	60.00%	1.00e-04
43	1850	52.80	0.6079	80.00%	1.00e-04
44	1900	53.56	0.1434	100.00%	1.00e-04
45	1950	54.44	0.0348	100.00%	1.00e-04
46	2000	55.23	0.1828	100.00%	1.00e-04
47	2050	56.12	0.1082	100.00%	1.00e-04
48	2100	56.93	0.1047	100.00%	1.00e-04
49	2150	57.75	0.1466	100.00%	1.00e-04
50	2200	58.58	0.0348	100.00%	1.00e-04

**Gambar 4. 13** Hasil proses training indikasi

Dalam tahap *training* ini, untuk *epoch* yang dibuat 50 menghasilkan 2200 iterasi yang dilatih. *Epoch* 50 ini berarti setiap proses *training* akan melalui lima puluh proses *feedforward*, dalam tahapan ini menghasilkan akurasi yang tidak stabil. Akurasi yang rendah didapat karena data *training* tidak mampu mempresentasikan model yang di latih. Akurasi ini adalah akurasi *training* sesuai dengan keakuratan dari *mini batch* pada iterasi yang diberikan. Hasil plot dari *training* indikasi digambarkan seperti pada gambar 4.14.



**Gambar 4. 14** Hasil plot training indikasi

**b. Hasil training K1**

Di dalam folder K1, disetiap kategori *edge* dan *non edge* terdapat 15 citra yang akan di *training*. Hasil akurasi training dari folder K1 dapat dilihat pada gambar 4.16.

Epoch	Iteration	Time Elapsed (seconds)	Mini-batch Loss	Mini-batch Accuracy	Base Learning Rate
1	1	0.06	2.3294	40.00%	1.00e-04
1	50	0.72	2.0709	20.00%	1.00e-04
2	100	1.39	1.5555	50.00%	1.00e-04
3	150	2.07	1.5982	50.00%	1.00e-04
4	200	2.74	2.7274	20.00%	1.00e-04
5	250	3.43	3.2635	0.00%	1.00e-04
6	300	4.11	0.8210	50.00%	1.00e-04
7	350	4.80	1.5123	50.00%	1.00e-04
8	400	5.47	1.2085	60.00%	1.00e-04
9	450	6.15	0.8349	50.00%	1.00e-04
10	500	6.83	1.4991	40.00%	1.00e-04
11	550	7.50	1.1699	50.00%	1.00e-04
12	600	8.18	1.2462	50.00%	1.00e-04
13	650	8.85	2.1917	20.00%	1.00e-04
14	700	9.53	1.4560	60.00%	1.00e-04
15	750	10.20	0.4237	100.00%	1.00e-04
16	800	10.88	1.4729	60.00%	1.00e-04
17	850	11.57	0.3406	100.00%	1.00e-04
18	900	12.26	1.2098	50.00%	1.00e-04
19	950	12.95	0.7397	50.00%	1.00e-04
20	1000	13.64	1.6940	50.00%	1.00e-04
21	1050	14.33	0.8368	100.00%	1.00e-04
22	1100	15.00	2.5465	60.00%	1.00e-04
23	1150	15.76	1.9585	50.00%	1.00e-04
24	1200	16.44	0.3322	100.00%	1.00e-04
25	1250	17.13	1.8159	60.00%	1.00e-04
26	1300	17.80	1.1316	80.00%	1.00e-04
27	1350	18.48	0.5677	100.00%	1.00e-04
28	1400	19.16	0.8649	100.00%	1.00e-04
29	1450	19.85	0.6882	80.00%	1.00e-04
30	1500	20.54	2.8806	60.00%	1.00e-04
31	1550	21.22	3.2810	20.00%	1.00e-04
32	1600	21.91	0.5889	50.00%	1.00e-04
33	1650	22.60	1.1091	50.00%	1.00e-04
34	1700	23.28	0.8489	60.00%	1.00e-04
35	1750	23.98	0.6653	50.00%	1.00e-04
36	1800	24.71	1.3903	40.00%	1.00e-04
37	1850	25.40	0.9683	50.00%	1.00e-04
38	1900	26.08	1.0575	50.00%	1.00e-04
39	1950	26.80	1.9817	20.00%	1.00e-04
40	2000	27.50	1.3347	60.00%	1.00e-04
41	2050	28.19	0.2518	100.00%	1.00e-04
42	2100	28.90	1.2057	60.00%	1.00e-04
43	2150	29.61	0.1962	100.00%	1.00e-04
44	2200	30.30	1.0348	50.00%	1.00e-04
45	2250	31.00	0.5973	50.00%	1.00e-04
46	2300	31.69	1.3959	50.00%	1.00e-04
47	2350	32.38	0.6630	100.00%	1.00e-04
48	2400	33.07	2.1259	60.00%	1.00e-04
49	2450	33.77	1.5550	50.00%	1.00e-04
50	2500	34.46	0.2824	100.00%	1.00e-04
50	2550	35.15	1.5715	60.00%	1.00e-04
50	2600	35.84	0.9914	50.00%	1.00e-04

**Gambar 4. 15** Hasil proses training K1

Dalam tahap *training* ini, untuk *epoch* yang dibuat 50 menghasilkan 2600 iterasi yang dilatih. *Epoch* 50 ini berarti setiap proses *training* akan melalui lima puluh proses *feedforward*, dalam tahapan ini menghasilkan akurasi yang tidak stabil. Akurasi yang rendah didapat karena data *training* tidak mampu mempresentasikan model yang di latih. Akurasi ini adalah akurasi *training* sesuai

dengan keakuratan dari *mini batch* pada iterasi yang diberikan. Hasil plot dari *training* K1 digambarkan seperti pada gambar 4.16.



**Gambar 4. 16** Hasil plot training K1

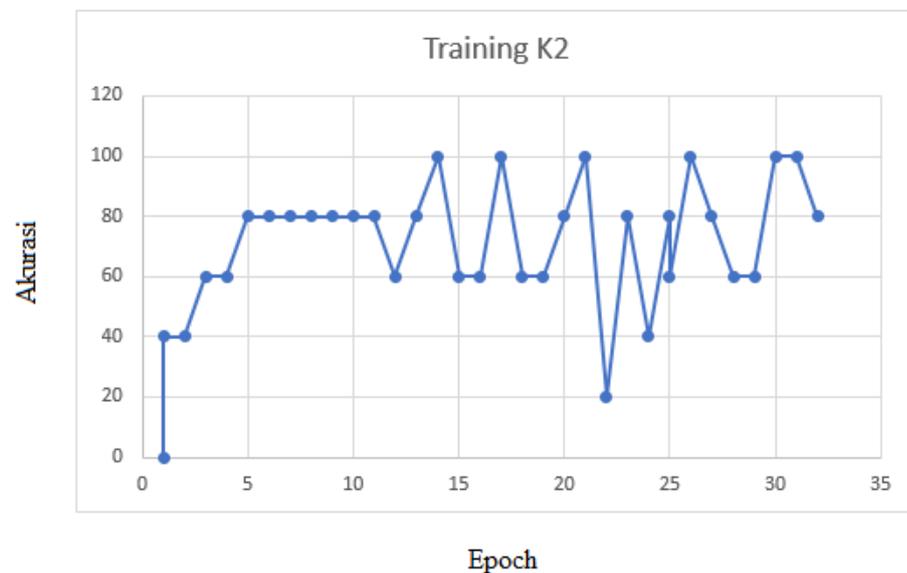
### c. Hasil training K2

Di dalam folder K2, disetiap kategori *edge* dan *non edge* terdapat 15 citra yang akan di *training*. Hasil akurasi training dari folder K2 dapat dilihat pada gambar 4.17.

Epoch	Iteration	Time Elapsed (seconds)	Mini-batch Loss	Mini-batch Accuracy	Base Learning Rate
1	1	0.04	2.3936	0.00%	1.00e-04
1	50	0.70	1.5118	40.00%	1.00e-04
2	100	1.38	1.8281	40.00%	1.00e-04
3	150	2.09	1.6759	60.00%	1.00e-04
4	200	2.77	1.4254	60.00%	1.00e-04
5	250	3.46	1.2989	80.00%	1.00e-04
6	300	4.14	1.1923	80.00%	1.00e-04
7	350	4.83	0.5302	80.00%	1.00e-04
8	400	5.51	0.7677	80.00%	1.00e-04
9	450	6.22	0.9108	80.00%	1.00e-04
10	500	6.91	0.8508	80.00%	1.00e-04
11	550	7.60	0.4484	80.00%	1.00e-04
12	600	8.29	1.0606	60.00%	1.00e-04
13	650	8.99	0.6746	80.00%	1.00e-04
14	700	9.68	0.4625	100.00%	1.00e-04
15	750	10.37	1.5999	60.00%	1.00e-04
16	800	11.07	1.5980	60.00%	1.00e-04
17	850	11.77	0.5206	100.00%	1.00e-04
18	900	12.47	1.5829	60.00%	1.00e-04
19	950	13.17	1.7925	60.00%	1.00e-04
20	1000	13.88	1.2243	80.00%	1.00e-04
21	1050	14.57	0.4501	100.00%	1.00e-04
22	1100	15.28	2.4359	20.00%	1.00e-04
23	1150	15.97	0.9990	80.00%	1.00e-04
24	1200	16.67	2.4197	40.00%	1.00e-04
25	1250	17.36	0.9463	80.00%	1.00e-04
25	1300	18.05	1.3175	60.00%	1.00e-04
26	1350	18.75	0.2039	100.00%	1.00e-04
27	1400	19.44	0.8023	80.00%	1.00e-04
28	1450	20.15	1.0439	60.00%	1.00e-04
29	1500	20.84	1.1200	60.00%	1.00e-04
30	1550	21.54	0.6130	100.00%	1.00e-04
31	1600	22.24	0.3573	100.00%	1.00e-04
32	1650	22.95	0.3171	80.00%	1.00e-04
33	1700	23.65	0.4879	80.00%	1.00e-04
34	1750	24.35	0.6185	80.00%	1.00e-04
35	1800	25.05	0.6089	80.00%	1.00e-04
36	1850	25.76	0.3314	80.00%	1.00e-04
37	1900	26.45	0.6099	80.00%	1.00e-04
38	1950	27.19	0.4291	80.00%	1.00e-04
39	2000	27.89	0.2034	100.00%	1.00e-04
40	2050	28.61	1.2916	60.00%	1.00e-04
41	2100	29.31	1.0546	60.00%	1.00e-04
42	2150	30.03	0.3543	100.00%	1.00e-04
43	2200	30.73	1.3649	60.00%	1.00e-04
44	2250	31.50	1.4227	60.00%	1.00e-04
45	2300	32.20	0.9854	80.00%	1.00e-04
46	2350	32.91	0.3537	100.00%	1.00e-04
47	2400	33.61	2.0631	20.00%	1.00e-04
48	2450	34.33	0.6221	80.00%	1.00e-04
49	2500	35.04	1.7853	40.00%	1.00e-04
50	2550	35.75	0.7832	100.00%	1.00e-04
50	2600	36.45	1.1303	60.00%	1.00e-04

**Gambar 4. 17** Hasil proses training K2

Dalam tahap *training* ini, untuk *epoch* yang dibuat 50 menghasilkan 2600 iterasi yang dilatih. *Epoch* 50 ini berarti setiap proses *training* akan melalui lima puluh proses *feedforward*, dalam tahapan ini menghasilkan akurasi yang tidak stabil. Akurasi yang rendah didapat karena data *training* tidak mampu mempresentasikan model yang di latih. Akurasi ini adalah akurasi *training* sesuai dengan keakuratan dari *mini batch* pada iterasi yang diberikan. Hasil plot dari *training* K1 digambarkan seperti pada gambar 4.18.



**Gambar 4. 18** Hasil plot training K2

#### 4.5 Testing Data

Pada tahap *testing* data merupakan tahap dimana klasifikasi menggunakan bobot dan bias dari hasil tahap *training*. Tahap ini tidak jauh berbeda dengan tahap *training* namun yang membedakannya hanya tidak terdapat proses *backpropagation* setelah proses *feedforward*. Sehingga hasil akhir dari proses ini menghasilkan akurasi dari klasifikasi yang dilakukan.

##### a. Hasil *testing* Indikasi

Hasil *testing* dapat dilihat pada tabel 4.1 dimana ada dua kesalahan pada saat klasifikasi yang menimbulkan akurasi tidak maksimal.

**Tabel 4. 1** Hasil testing indikasi - part 1

Gambar edge	Kategori edge	Kategori klasifikasi <i>deep learning</i>	Keterangan
	1	1	Benar
	1	1	Benar
	1	1	Benar
	1	1	Benar
	1	1	Benar
	2	2	Benar
	2	2	Benar
	2	2	Benar
	2	2	Benar
	2	2	Benar
	3	3	Benar
	3	3	Benar
	3	3	Benar
	3	3	Benar
	3	3	Benar
	4	4	Benar
	4	4	Benar
	4	4	Benar
	4	4	Benar
	4	4	Benar
	5	5	Benar
	5	5	Benar
	5	5	Benar
	5	5	Benar
	5	5	Benar
	6	6	Benar

**Tabel 4. 2 Hasil testing indikasi - part 2**

	6	6	Benar
	6	6	Benar
	6	6	Benar
	6	6	Benar
	7	7	Benar
	7	7	Benar
	7	7	Benar
	7	7	Benar
	7	1	Salah
	8	8	Benar
	8	8	Benar
	8	8	Benar
	8	8	Benar
	8	8	Benar
	8	8	Benar
	10	10	Benar
	10	10	Benar
	10	10	Benar
	10	10	Benar
	10	10	Benar
	11	11	Benar
	11	11	Benar
	11	11	Benar
	11	11	Benar
	11	10	Salah

Dapat diketahui dari tabel 4.1 menunjukkan dari 55 citra *training* terdapat 2 kesalahan yang terjadi pada saat pengklasifikasian data *training*. Kesalahan yang

pertama ada di kategori 7 yang seharusnya pada saat mengkategorikan klasifikasi *deep learning* juga menunjukkan kategori 7 tetapi di atas diklasifikasikan sebagai kategori 1. Kesalahan kedua ditunjukkan pada saat pengklasifikasian kategori 11 yang dimana seharusnya itu kategori 11, akan tetapi pada saat proses klasifikasi itu menjadi kategori 10. Dari hasil tersebut didapat jumlah akurasi sebesar 96% untuk *testing* indikasi.

#### b. Hasil *testing* K1

Hasil *testing* K1 terlihat pada tabel 4.2 dimana ada tujuh kesalahan pada saat klasifikasi yang menimbulkan akurasi tidak maksimal.

**Tabel 4. 3 Hasil *testing* K1 – part 1**

Gambar <i>edge</i>	Kategori <i>edge</i>	Kategori klasifikasi <i>deep learning</i>	Keterangan
	1	1	Benar
	1	1	Benar
	1	1	Benar
	1	1	Benar
	1	11	Salah
	2	2	Benar
	2	2	Benar
	2	2	Benar
	2	2	Benar
	2	2	Benar
	3	3	Benar
	3	3	Benar

**Tabel 4. 4** Hasil testing KI – part 2

	3	3	Benar
	3	3	Benar
	3	3	Benar
	4	4	Benar
	4	4	Benar
	4	4	Benar
	4	4	Benar
	4	4	Benar
	5	5	Benar
	5	5	Benar
	5	5	Benar
	5	5	Benar
	5	3	Salah
	6	6	Benar
	6	6	Benar
	6	6	Benar
	6	6	Benar
	6	6	Benar
	7	7	Benar
	7	7	Benar
	7	7	Benar
	7	7	Benar
	7	10	Salah
	8	8	Benar

**Tabel 4.5 Hasil testing K1 – part 3**

	8	8	Benar
	8	8	Benar
	8	8	Benar
	8	6	Salah
	10	10	Benar
	10	10	Benar
	10	10	Benar
	10	10	Benar
	10	10	Benar
	11	11	Benar
	11	11	Benar
	11	2	Salah
	11	10	Salah
	11	10	Salah

Dapat diketahui dari tabel 4.2 menunjukkan dari 55 citra *training* terdapat 7 kesalahan yang terjadi pada saat pengklasifikasian data *training*. Kesalahan yang pertama ada di kategori 1 yang seharusnya pada saat mengkategorikan klasifikasi *deep learning* juga menunjukkan kategori 1 tetapi di atas diklasifikasikan sebagai kategori 11. Kesalahan kedua ditunjukkan pada saat pengklasifikasian kategori 5 yang dimana seharusnya itu kategori 5, akan tetapi pada saat proses klasifikasi itu menjadi kategori 3. Kesalahan ketiga dapat dilihat pada saat pengklasifikasian kategori 7 yang seharusnya tetap menjadi kategori 7, namun diklasifikasikan menjadi kategori 10. Kesalahan ke empat ada di kategori 8 yang mestinya tetap kategori 8, tapi diklasifikasi ke kategori 6. Kesalahan ke lima, enam, dan tujuh berada pada saat pengklasifikasian di kategori 11 yang semestinya terisi dengan kategori 11, namun malah terbaca 1 kategori 2 dan 2 kategori 10. Dari hasil tersebut didapat jumlah akurasi sebesar 85% untuk *testing* K1.

**c. Hasil testing K2**

Dalam tahap *testing* didapatkan hasil yang terlihat pada tabel 4.3 dimana terdapat 5 kesalahan pada saat klasifikasi yang menimbulkan akurasi tidak maksimal.

**Tabel 4.6 Hasil testing K2 – part 1**

Gambar edge	Kategori edge	Kategori klasifikasi <i>deep learning</i>	Keterangan
	1	1	Benar
	1	1	Benar
	1	1	Benar
	1	1	Benar
	1	7	Salah
	2	2	Benar
	2	2	Benar
	2	2	Benar
	2	2	Benar
	2	2	Benar
	3	3	Benar
	3	3	Benar

**Tabel 4. 7 Hasil testing K2 – part 2**

	3	3	Benar
	3	3	Benar
	3	3	Benar
	4	4	Benar
	4	4	Benar
	4	4	Benar
	4	4	Benar
	4	2	Salah
	5	5	Benar
	5	5	Benar
	5	5	Benar
	5	5	Benar
	5	5	Benar
	6	6	Benar
	6	6	Benar
	6	6	Benar
	6	6	Benar
	6	11	Salah
	7	7	Benar
	7	7	Benar
	7	7	Benar
	7	7	Benar
	7	7	Benar
	8	8	Benar
	8	8	Benar
	8	8	Benar

**Tabel 4. 8 Hasil testing K2 – part 3**

	8	8	Benar
	8	8	Benar
	10	10	Benar
	10	10	Benar
	10	10	Benar
	10	10	Benar
	10	10	Benar
	11	11	Benar
	11	11	Benar
	11	11	Benar
	11	1	Salah
	11	10	Salah

Dapat diketahui dari tabel 4.3 menunjukkan dari 55 citra *training* terdapat 5 kesalahan yang terjadi pada saat pengklasifikasian data *training*. Kesalahan yang pertama ada di kategori 1 yang seharusnya pada saat mengkategorikan klasifikasi *deep learning* juga menunjukkan kategori 1 tetapi di atas diklasifikasikan sebagai kategori 7. Kesalahan kedua ditunjukkan pada saat pengklasifikasian kategori 4 yang dimana seharusnya itu kategori 4, akan tetapi pada saat proses klasifikasi itu menjadi kategori 2. Kesalahan ketiga dapat dilihat pada saat pengklasifikasian kategori 6 yang seharusnya tetap menjadi kategori 6, namun diklasifikasikan menjadi kategori 11. Kesalahan ke empat dan lima berada pada saat pengklasifikasian di kategori 11 yang semestinya terisi dengan kategori 11, namun malah terbaca kategori 1 dan kategori 10. Dari hasil tersebut didapat jumlah akurasi sebesar 90% untuk *testing K2*.

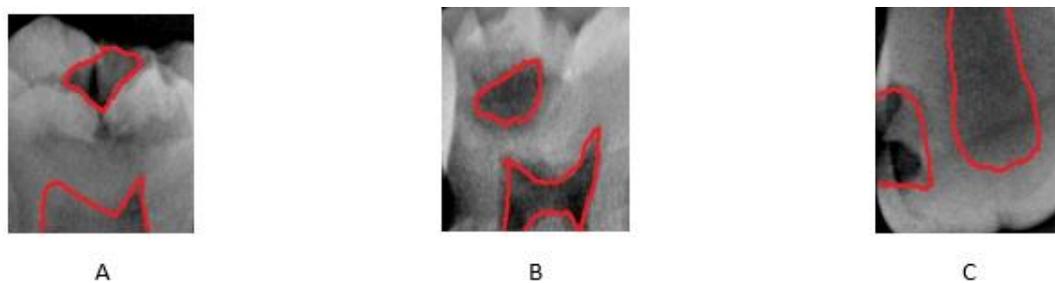
#### 4.6 Penentuan *Edge*

Tahap penentuan *edge* ialah dimana sistem akan mendeteksi secara otomatis pada bagian tepi dari dentin tersier. Untuk citra yang ingin di deteksi kemudian dibaca per setiap 5 piksel untuk mengklasifikasi bahwa potongan citra itu termasuk

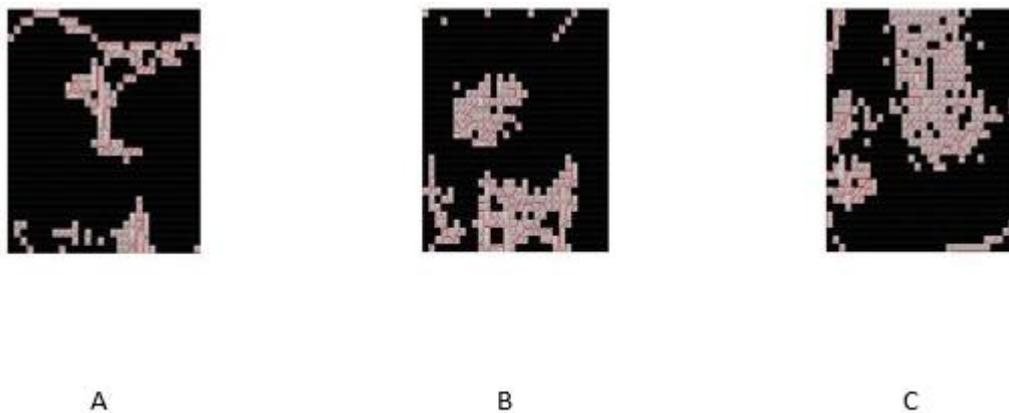
bagian kategori *edge* yang mana lalu akan di *replace* oleh citra sample kategori hingga mendapat citra yang sudah terdeteksi bagian tepinya.

**a. Hasil penentuan *edge* indikasi**

Hasil yang didapat dari penentuan *edge* indikasi ini dapat dilihat pada gambar 4.20. Umumnya kesalahan pada penandaan tepi dikarenakan faktor input citra gigi yang sulit untuk dideteksi bagian *edge* nya karena warna dari citra gigi cenderung sama dimana hanya ada warna abu-abu dan hitam.



**Gambar 4. 19** A, B, dan C sample citra indikasi sebelum melalui proses *edge*



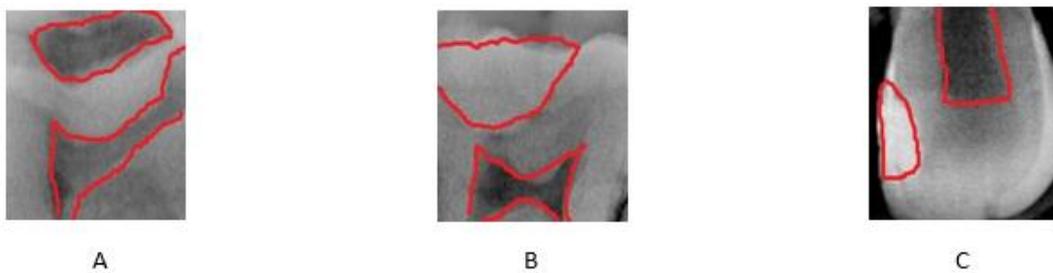
**Gambar 4. 20** A, B, dan C adalah citra hasil proses *edge* indikasi

Dapat dilihat dari gambar 4.20 diatas hasil indikasi yang sudah di proses dapat mendeteksi bagian yang dinilai memiliki perbedaan warna sehingga dapat membentuk tepi pada garis yang menonjol. Untuk bagian gigi yang memiliki warna yang berbeda secara mencolok akan terdeteksi secara sempurna dibagian tepinya. Karena untuk deteksi tepi pada tahap penentuan *edge* akan diklasifikasi sesuai

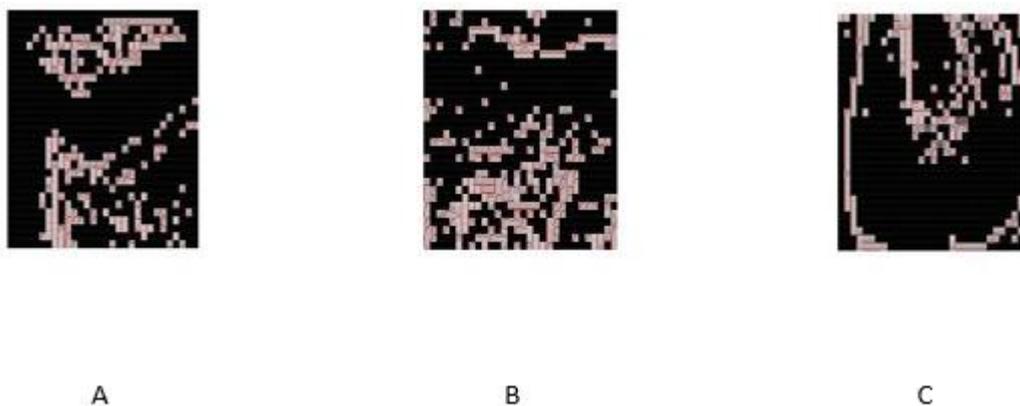
sample kategori yang dilihat memiliki 2 warna yang berbeda untuk menandai bagian tepi dari citra tersebut.

**b. Hasil penentuan *edge* K1**

Hasil yang didapat dari penentuan *edge* indikasi ini dapat dilihat pada gambar 4.22. Umumnya kesalahan pada penandaan tepi dikarenakan faktor input citra gigi yang sulit untuk dideteksi bagian *edge* nya karena warna dari citra gigi cenderung sama dimana hanya ada warna abu-abu dan hitam.



**Gambar 4. 21** A, B, dan C sample citra K1 sebelum melalui proses *edge*



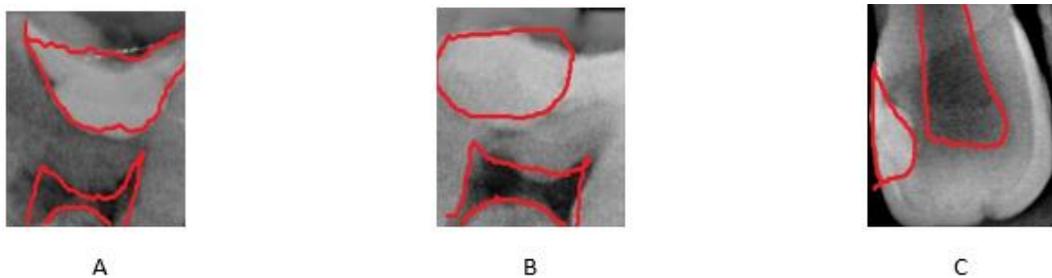
**Gambar 4. 22** A, B, dan C adalah citra hasil proses *edge* K1

Dapat dilihat dari gambar 4.22 diatas hasil indikasi yang sudah di proses dapat mendeteksi bagian yang dinilai memiliki perbedaan warna sehingga dapat membentuk tepi pada garis yang menonjol. Untuk bagian gigi yang memiliki warna yang berbeda secara mencolok akan terdeteksi secara sempurna dibagian tepi nya. Karena untuk deteksi tepi pada tahap penentuan *edge* akan diklasifikasi sesuai

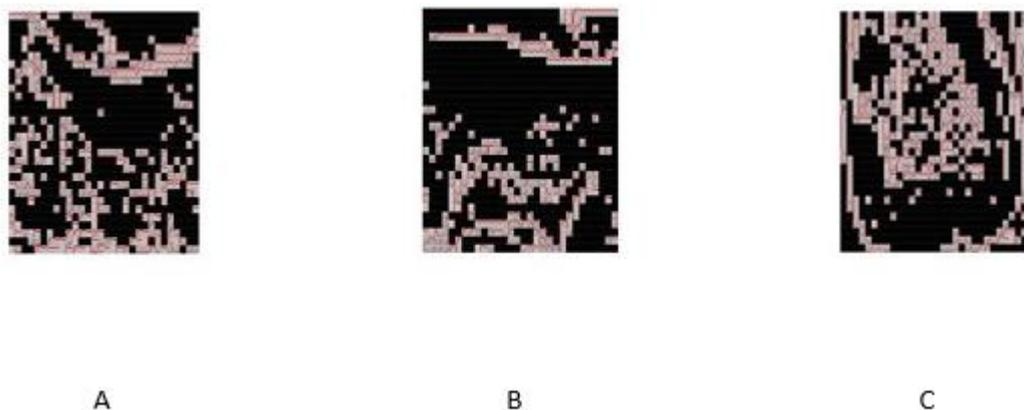
sample kategori yang dilihat memiliki 2 warna yang berbeda untuk menandai bagian tepi dari citra tersebut.

**c. Hasil penentuan *edge* K2**

Hasil yang didapat dari penentuan *edge* indikasi ini dapat dilihat pada gambar 4.24. Umumnya kesalahan pada penandaan tepi dikarenakan faktor input citra gigi yang sulit untuk dideteksi bagian *edge* nya karena warna dari citra gigi cenderung sama dimana hanya ada warna abu-abu dan hitam.



**Gambar 4. 23** *A, B, dan C sample citra K2 sebelum melalui proses edge*



**Gambar 4. 24** *A, B, dan C adalah citra hasil proses edge K2*

Dapat dilihat dari gambar 4.24 diatas hasil indikasi yang sudah di proses dapat mendeteksi bagian yang dinilai memiliki perbedaan warna sehingga dapat membentuk tepi pada garis yang menonjol. Untuk bagian gigi yang memiliki warna yang berbeda secara mencolok akan terdeteksi secara sempurna dibagian tepi nya. Karena untuk deteksi tepi pada tahap penentuan *edge* akan diklasifikasi sesuai

sample kategori yang dilihat memiliki 2 warna yang berbeda untuk menandai bagian tepi dari citra tersebut.

#### 4.7 Validasi

Tahap terakhir yaitu tahap validasi, dimana tujuannya adalah untuk menilai keakuratan sebuah model yang dibangun berdasarkan set tertentu. Pada tahap ini dilakukan secara manual yaitu dengan membandingkan hasil dari *edge* menggunakan *deep learning* dengan metode *edge Sobel*, *edge Canny*, dan *edge Prewitt* maupun *edge* buatan *expert*. Hasil dari validasi dapat dilihat pada gambar 4.25.



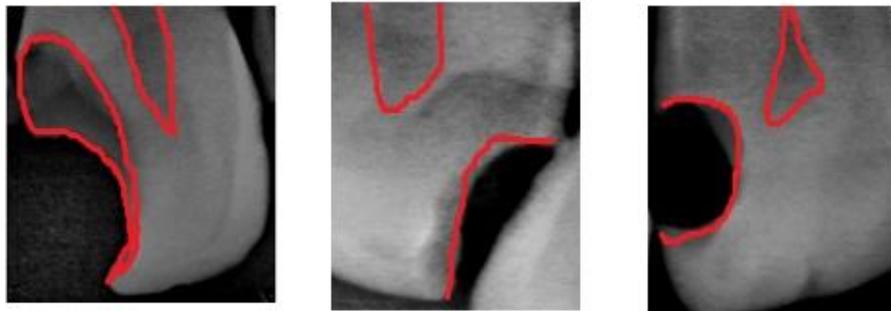
A. Metode *deep learning*



B. Metode Prewitt



C. Metode *sobel*

D. Metode *Canny*E. Hasil *expert***Gambar 4. 25** Perbandingan antara beberapa metode

Pada gambar 4.25 bisa diketahui dimana A adalah citra *edge* hasil pembelajaran *deep learning*. Pada B adalah citra yang menggunakan metode *Prewitt*. Gambar C adalah citra *edge* hasil dari metode *Sobel*. D ialah hasil dari *edge* yang menggunakan metode *Canny*. Dan pada gambar E adalah citra *edge* yang dibuat oleh *expert*. Dari hasil perbandingan ini dapat dikatakan bahwa metode *deep learning* yang hampir menyerupai dengan hasil *expert*, karena metode ini membuat sistem belajar dimana bagian tepi yang diklasifikasikan akan sesuai dengan bentuk aslinya. Dari metode *deep learning* juga dapat diketahui dibagian mana yang menjadi akar gigi dan bagian mana gigi yang berlubang sehingga para ahli dapat menentukan area dentin tersiernya. Karena dari ketiga metode yang telah diolah dari Matlab dengan citra yang sama menghasilkan citra yang tidak jelas semua maka disarankan untuk menggunakan metode *deep learning* yang bisa menyerupai dengan bentuk gigi aslinya, karena pada metode ini dapat diketahui dimana garis tepi berada.