

# Body to Body Network Architecture Design and Interopability Model for Biomedical Data Transmission from Electro Encephalography Sensing Signals

Ida Wahidah<sup>1</sup>, Vinsensius Sigit Widhi Prabowo<sup>1</sup>, Iman Hedi Santoso<sup>1</sup>, Muhammad Rifaldi Septa Pratama<sup>1</sup>, Qonita Nur Ramadhani<sup>1</sup>, Wahid Afiq Anggara<sup>1</sup>

Telkom University,

[wahidah@telkomuniversity.ac.id](mailto:wahidah@telkomuniversity.ac.id),

[vinsensiusvsw@telkomuniversity.ac.id](mailto:vinsensiusvsw@telkomuniversity.ac.id), [qonita986@gmail.com](mailto:qonita986@gmail.com), [faldisepta7@gmail.com](mailto:faldisepta7@gmail.com)

[whdanggara@student.telkomuniversity.ac.id](mailto:whdanggara@student.telkomuniversity.ac.id)

**Abstract**—The brain is one of the most important organs in the human body since it is the center of human equilibrium and the place of human balance. Human brain activity examination is highly suggested as the examination enhances a person's physical and mental state, such as by monitoring brain and mental health, early identification of sickness, and testing cognitive and emotional problems. With the high probability of increasing the number of people who experience decreased brain function, it can be concluded that a system solution that can be proposed for this problem is by designing an EEG (electroencephalogram). EEG (electroencephalogram) is an instrument that measures electrical activity in the human brain. EEG can be used to look for abnormal brain wave patterns that indicate errors and other abnormalities. EEG analysis is one method that aids in the process of assessing brain activity. Besides that, the use of Neurosky as an EEG is also very helpful where the tool has a Neurosky eSense, which can read a person's level of focus to meditation. The results of our research using NeuroSky Mindwave, which is connected to the Arduino UNO microcontroller, can produce brain activity data in the form of "attention" and "meditation" with a range of 0–100% and EEG signals of the beta wave type ( $\beta$ ) with an average value of 50–200 V. With the help of heart rate and neurosky sensors, it can be seen and analyzed by the authorities in medical studies regarding the user's brain activity.

**Keywords**— Brain, NeuroSky, ESP32, EEG, Arduino

## I. INTRODUCTION

The brain is one of the most vital organs in the human body, and it is the epicenter of many processes, particularly balance and human equilibrium. The brain impacts various other organs in the human body, including metabolic processes, blood pressure, coordination of the five senses, and many others. According to the World Health Organization Neurological diseases are the main contributing factor to disability-adjusted life years (DALYs), which result in around 9 million deaths each year. Stroke (42.2%), migraine

(16.3%), dementia (10.4%), meningitis (7.9%), and epilepsy (5%) were the biggest contributors to neurological DALYs in 2016. Parkinson's disease is the fastest-growing neurological condition due to a growing elderly population [1]. In Indonesia itself, based on the data given by the Indonesian Central Statistics Agency, the number of senior citizens (aged 60 and above) in Indonesia is expected to be 22,630,882 people, based on the results of the 2016 Inter-Census Population Survey. The number of this population is predicted to rise to 31,320,066 by 2023 [2]. Indonesia's population is 237.5 million, with a population growth rate of 1.14% (BPS, 2010). Based on that data, it is estimated that there are 2.4 million people with ASD in Indonesia, with an increase of 500 new people per year [3]. With the high number of people suffering from brain health issues, performing brain examinations, particularly on certain targets such as the elderly and autistic patients, can help avoid this by conducting regular checks on a person's psychological and physical status. EEG analysis is one method of assisting in the process of measuring brain activity. EEG can be used to look for unusual brain wave patterns that indicate errors and other abnormalities. As an electroencephalogram tool, the neurosky mindwave tool can assist users in monitoring mental health, undertaking research and scientific studies, improving cognitive function, and developing apps and technology.

## II. METHOD

The project is divided into several parts, which are the multihop topology control process and resource scheduling, data acquisition and reconstruction, and creating an application to display and monitor the data. The schematic will be as follows: by connecting two pulse sensors to each ESP32, which will then carry out the scheduling process, and connecting Neurosky to Arduino, all data from Neurosky and pulse sensors will be sent to Python for the acquisition and

reconstruction process, after which it will be sent to Firebase, where the data will then be displayed by an application created via MIT App Inventor. The system schematic illustrated in Figure 1.

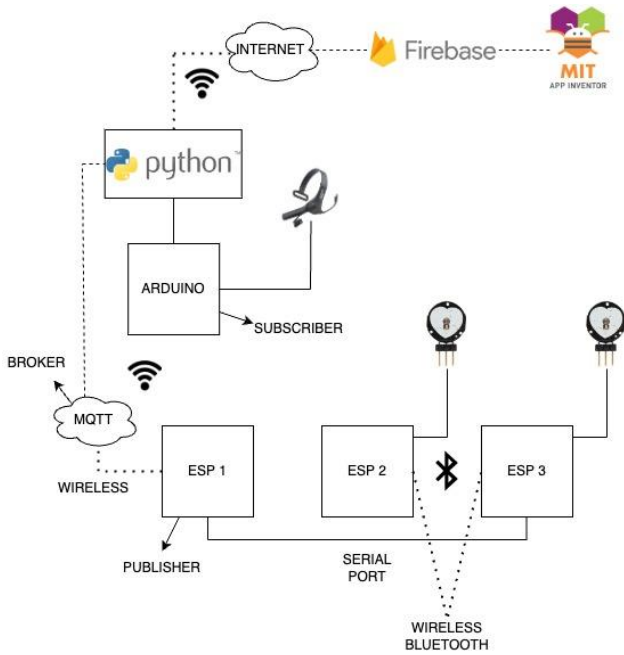


Figure 1 System Schematic

### A. Sensor Pulse & Scheduling

a) A pulse sensor is a sensor that can calculate heart rate. This sensor uses infrared and photodiode. The next step is for us to try the Pulse Sensor which is connected to the ESP 32.



1. First connect the Pulse Sensor to ESP 32 where the jumper cable is at:
  - 1) Ground to GND
  - 2) 3.5 Volts to VCC
  - 3) Pin 4 to VP
2. Second, connect the first ESP 32 with the Pulse Sensor installed to the second ESP 32:
  - 1) Rx 16 to Tx17
  - 2) Tx 17 to Rx 16
  - 3) Ground to GND

We need to know that here we are connecting ESP 32 with Rx Tx in a cross manner.
3. After the Pulse Sensor has been successfully connected to the ESP 32 a display like the one below will appear:

```

46 | // Serial.println() -> startHere: New Connection, model = 'nrf', model:
47 | }
48 |
49 | void changeConnectionCallback() {
50 | // Serial.println("Change connection");
51 | }
52 |
53 | void modelAdjustmentCallback(uint32_t offset) {
54 | // Serial.println("Adjusted time by: offset = " + offset + "ms", mesh-getNodeLine1,offset);
55 | }
56 |
57 | void setup() {
58 | Serial.begin(115200);
59 | SerialPort.begin(ESP8266, SERIAL_8N1, 16, 17);
60 |
61 |
62 | //mesh_setMeshType( ERROR | MESH_STATUS | CONNECTION | SYNC | COMMUNICATION |
63 | mesh_setMeshType( ERROR | STARTUP ); // set before start() so that you can
64 |
65 | mesh.setMeshPrefix( MESH_PREFIX, MESH_PASSWORD, userScheduler, MESH_PORT );
66 | mesh.onMeshEventReceivedCallback();
67 | mesh.onMeshEventSentCallback();
68 | }
69 |
70 |
71 |
72 |
73 |
74 |
75 |
76 |
77 |
78 |
79 |
80 |
81 |
82 |
83 |
84 |
85 |
86 |
87 |
88 |
89 |
90 |
91 |
92 |
93 |
94 |
95 |
96 |
97 |
98 |
99 |
100 |
101 |
102 |
103 |
104 |
105 |
106 |
107 |
108 |
109 |
110 |
111 |
112 |
113 |
114 |
115 |
116 |
117 |
118 |
119 |
120 |
121 |
122 |
123 |
124 |
125 |
126 |
127 |
128 |
129 |
130 |
131 |
132 |
133 |
134 |
135 |
136 |
137 |
138 |
139 |
140 |
141 |
142 |
143 |
144 |
145 |
146 |
147 |
148 |
149 |
150 |
151 |
152 |
153 |
154 |
155 |
156 |
157 |
158 |
159 |
160 |
161 |
162 |
163 |
164 |
165 |
166 |
167 |
168 |
169 |
170 |
171 |
172 |
173 |
174 |
175 |
176 |
177 |
178 |
179 |
180 |
181 |
182 |
183 |
184 |
185 |
186 |
187 |
188 |
189 |
190 |
191 |
192 |
193 |
194 |
195 |
196 |
197 |
198 |
199 |
200 |
201 |
202 |
203 |
204 |
205 |
206 |
207 |
208 |
209 |
210 |
211 |
212 |
213 |
214 |
215 |
216 |
217 |
218 |
219 |
220 |
221 |
222 |
223 |
224 |
225 |
226 |
227 |
228 |
229 |
230 |
231 |
232 |
233 |
234 |
235 |
236 |
237 |
238 |
239 |
240 |
241 |
242 |
243 |
244 |
245 |
246 |
247 |
248 |
249 |
250 |
251 |
252 |
253 |
254 |
255 |
256 |
257 |
258 |
259 |
260 |
261 |
262 |
263 |
264 |
265 |
266 |
267 |
268 |
269 |
270 |
271 |
272 |
273 |
274 |
275 |
276 |
277 |
278 |
279 |
280 |
281 |
282 |
283 |
284 |
285 |
286 |
287 |
288 |
289 |
290 |
291 |
292 |
293 |
294 |
295 |
296 |
297 |
298 |
299 |
300 |
301 |
302 |
303 |
304 |
305 |
306 |
307 |
308 |
309 |
310 |
311 |
312 |
313 |
314 |
315 |
316 |
317 |
318 |
319 |
320 |
321 |
322 |
323 |
324 |
325 |
326 |
327 |
328 |
329 |
330 |
331 |
332 |
333 |
334 |
335 |
336 |
337 |
338 |
339 |
340 |
341 |
342 |
343 |
344 |
345 |
346 |
347 |
348 |
349 |
350 |
351 |
352 |
353 |
354 |
355 |
356 |
357 |
358 |
359 |
360 |
361 |
362 |
363 |
364 |
365 |
366 |
367 |
368 |
369 |
370 |
371 |
372 |
373 |
374 |
375 |
376 |
377 |
378 |
379 |
380 |
381 |
382 |
383 |
384 |
385 |
386 |
387 |
388 |
389 |
390 |
391 |
392 |
393 |
394 |
395 |
396 |
397 |
398 |
399 |
400 |
401 |
402 |
403 |
404 |
405 |
406 |
407 |
408 |
409 |
410 |
411 |
412 |
413 |
414 |
415 |
416 |
417 |
418 |
419 |
420 |
421 |
422 |
423 |
424 |
425 |
426 |
427 |
428 |
429 |
430 |
431 |
432 |
433 |
434 |
435 |
436 |
437 |
438 |
439 |
440 |
441 |
442 |
443 |
444 |
445 |
446 |
447 |
448 |
449 |
450 |
451 |
452 |
453 |
454 |
455 |
456 |
457 |
458 |
459 |
460 |
461 |
462 |
463 |
464 |
465 |
466 |
467 |
468 |
469 |
470 |
471 |
472 |
473 |
474 |
475 |
476 |
477 |
478 |
479 |
480 |
481 |
482 |
483 |
484 |
485 |
486 |
487 |
488 |
489 |
490 |
491 |
492 |
493 |
494 |
495 |
496 |
497 |
498 |
499 |
500 |
501 |
502 |
503 |
504 |
505 |
506 |
507 |
508 |
509 |
510 |
511 |
512 |
513 |
514 |
515 |
516 |
517 |
518 |
519 |
520 |
521 |
522 |
523 |
524 |
525 |
526 |
527 |
528 |
529 |
530 |
531 |
532 |
533 |
534 |
535 |
536 |
537 |
538 |
539 |
540 |
541 |
542 |
543 |
544 |
545 |
546 |
547 |
548 |
549 |
550 |
551 |
552 |
553 |
554 |
555 |
556 |
557 |
558 |
559 |
560 |
561 |
562 |
563 |
564 |
565 |
566 |
567 |
568 |
569 |
570 |
571 |
572 |
573 |
574 |
575 |
576 |
577 |
578 |
579 |
580 |
581 |
582 |
583 |
584 |
585 |
586 |
587 |
588 |
589 |
590 |
591 |
592 |
593 |
594 |
595 |
596 |
597 |
598 |
599 |
600 |
601 |
602 |
603 |
604 |
605 |
606 |
607 |
608 |
609 |
610 |
611 |
612 |
613 |
614 |
615 |
616 |
617 |
618 |
619 |
620 |
621 |
622 |
623 |
624 |
625 |
626 |
627 |
628 |
629 |
630 |
631 |
632 |
633 |
634 |
635 |
636 |
637 |
638 |
639 |
640 |
641 |
642 |
643 |
644 |
645 |
646 |
647 |
648 |
649 |
650 |
651 |
652 |
653 |
654 |
655 |
656 |
657 |
658 |
659 |
660 |
661 |
662 |
663 |
664 |
665 |
666 |
667 |
668 |
669 |
670 |
671 |
672 |
673 |
674 |
675 |
676 |
677 |
678 |
679 |
680 |
681 |
682 |
683 |
684 |
685 |
686 |
687 |
688 |
689 |
690 |
691 |
692 |
693 |
694 |
695 |
696 |
697 |
698 |
699 |
700 |
701 |
702 |
703 |
704 |
705 |
706 |
707 |
708 |
709 |
710 |
711 |
712 |
713 |
714 |
715 |
716 |
717 |
718 |
719 |
720 |
721 |
722 |
723 |
724 |
725 |
726 |
727 |
728 |
729 |
730 |
731 |
732 |
733 |
734 |
735 |
736 |
737 |
738 |
739 |
740 |
741 |
742 |
743 |
744 |
745 |
746 |
747 |
748 |
749 |
750 |
751 |
752 |
753 |
754 |
755 |
756 |
757 |
758 |
759 |
760 |
761 |
762 |
763 |
764 |
765 |
766 |
767 |
768 |
769 |
770 |
771 |
772 |
773 |
774 |
775 |
776 |
777 |
778 |
779 |
780 |
781 |
782 |
783 |
784 |
785 |
786 |
787 |
788 |
789 |
790 |
791 |
792 |
793 |
794 |
795 |
796 |
797 |
798 |
799 |
800 |
801 |
802 |
803 |
804 |
805 |
806 |
807 |
808 |
809 |
810 |
811 |
812 |
813 |
814 |
815 |
816 |
817 |
818 |
819 |
820 |
821 |
822 |
823 |
824 |
825 |
826 |
827 |
828 |
829 |
830 |
831 |
832 |
833 |
834 |
835 |
836 |
837 |
838 |
839 |
840 |
841 |
842 |
843 |
844 |
845 |
846 |
847 |
848 |
849 |
850 |
851 |
852 |
853 |
854 |
855 |
856 |
857 |
858 |
859 |
860 |
861 |
862 |
863 |
864 |
865 |
866 |
867 |
868 |
869 |
870 |
871 |
872 |
873 |
874 |
875 |
876 |
877 |
878 |
879 |
880 |
881 |
882 |
883 |
884 |
885 |
886 |
887 |
888 |
889 |
890 |
891 |
892 |
893 |
894 |
895 |
896 |
897 |
898 |
899 |
900 |
901 |
902 |
903 |
904 |
905 |
906 |
907 |
908 |
909 |
910 |
911 |
912 |
913 |
914 |
915 |
916 |
917 |
918 |
919 |
920 |
921 |
922 |
923 |
924 |
925 |
926 |
927 |
928 |
929 |
930 |
931 |
932 |
933 |
934 |
935 |
936 |
937 |
938 |
939 |
940 |
941 |
942 |
943 |
944 |
945 |
946 |
947 |
948 |
949 |
950 |
951 |
952 |
953 |
954 |
955 |
956 |
957 |
958 |
959 |
960 |
961 |
962 |
963 |
964 |
965 |
966 |
967 |
968 |
969 |
970 |
971 |
972 |
973 |
974 |
975 |
976 |
977 |
978 |
979 |
980 |
981 |
982 |
983 |
984 |
985 |
986 |
987 |
988 |
989 |
990 |
991 |
992 |
993 |
994 |
995 |
996 |
997 |
998 |
999 |
1000 |

```

b) Data scheduling is data scheduling that is carried out so that the queuing process can be carried out correctly and efficiently. for example, if the bound process is bound with I/O bound, if scheduling is not carried out then the I/O process may wait too long for the bound process. We do this scheduling using the priority scheduling method where we take data from the two Pulse Sensors 5 times. After the data is obtained, we average the data.

1. First connect each Pulse Sensor to the ESP 32
2. After the Pulse Sensor has been successfully connected to each ESP 32 then the data is scheduled using coding.
3. After we enter the coding as in the image above, the data display will appear like this:

```

Output Serial Monitor x
Message (Enter to send message to 'ESP32 Wrover Module' on '/dev/cu. New Line 115200 baud
0x00000000 0E (245) psram: PSRAM ID read error: 0xffffffff
Bluetooth Device is Ready to Pair
BPM: 28
BPM: 53
BPM: 121
BPM: 124
Received: 56
Received: 64
Received: 57
Received: 98

Output Serial Monitor x
Message (Enter to send message to 'ESP32 Wrover Module' on '/d New Line 115200 baud
0x00000000 0E (245) psram: PSRAM ID read error: 0xffffffff
Bluetooth Device is in master mode. Connecting to the host...
Connected Successfully!
Deteksi
BPM: 38
Deteksi
BPM: 39
Deteksi

```

### B. data acquisition and data reconstruction

a) Based on what the previous Step did. The next step is to acquire data carried out by a program based on output data from the Arduino IDE. The data was acquired using a program in the Python language that implements a library called PySerial. The library reads Arduino with a Baudrate of 57600 Baud which was set in the previous step. Then the program divides the waves into several types of signals needed for further research.

The types of signal waves needed in this research can be explained in the following table:

**Table 1 Signal Type**

No	Signal name	Information
1	Meditation	a numerical value that describes a person's current level of meditation or relaxation state, based on analysis of their brain wave patterns.
2	Attention	a numerical value that reflects a person's current level of concentration or focus, resulting from analysis of their brain activity.
3	Beta	The average value of the beta signal is defined as a number that reflects the intensity of brain activity related to the user's concentration, analytical thinking and focus.

The signal data processed by the program will first be saved as a file with the extension .csv so that it can be read again. After that the data will be sent to a database that uses the Firebase platform. This platform provides real-time data storage that will accommodate output data from Python programs based on the signal retrieval process from Neurosky.

b) Reconstruction

During the data acquisition process carried out by the program, the data is reconstructed into data that can be read and visualized by other devices. The process starts from wavelet transformation of the original data. Wavelet is a transformation that works by dividing a signal into different frequency and time components. The transformation was carried out using the PyWT library. This library converts the original data into wavelet coefficients, namely cA (Approximation coefficient) and cD (Detail coefficient) which can be explained in the following table:

**Table 2 Types of Coefficients**

No	Types of Coefficients	Information
1	Approximation Coefficient	A rough representation or refined approach to original data that shows data that represents the overall or macro

		characteristics of an original data
2	Detailed Coefficients	The result of measuring the difference between the original data and rough approximations that show detailed information on wavelet coefficients

Then the data that has been divided into approximation coefficients and detail coefficients is inverted using IDWT. The results of the inverse are filtered using a Gaussian filter to reduce noise or other effects that could damage the reconstructed data. Then the Gaussian filter data is converted into a suitable format for further processing

Next, the data that has been processed using Gaussian is reprocessed using the Orthogonal Matching Pursuit (OMP) algorithm which is created with appropriate parameter settings. This setting consists of a collection of wavelet coefficients that do not have a value of zero. The resulting data from the OMP algorithm processing is the optimal data combination from the wavelet basis that will be used or analyzed in accordance with this research.

C. Android Based Application

An application is created in four steps: building a cloud server, designing a user interface (UI), designing a block editor using MIT App Inventor, and testing the program.

1. Cloud Server

The cloud server used is Firebase; a real-time Firebase cloud server is created that functions to store data from the application and receive data from sensors for the application. Firebase realtime will be connected to MIT App Inventor by adding "firebaseadb1" to the palette provided by MIT App Inventor by entering "firebaseetoken" and "firebaseurl".

2. Application

The UI creation process is carried out with five main pages: the login page, sign-up page, dashboard, heart rate sensor data, and EEG sensor data. The UI creation process uses all the components available in the palette. The section for displaying EEG graphic data requires a separate extension, which can be found on the MIT App Inventor community. The extension used is ChartMakerPlus1," where the graph is displayed from Google Charts. The process of creating a block editor is the process of connecting blocks into a puzzle, which will become commands for applications such as displaying graphics, sending data to Firebase, and others.

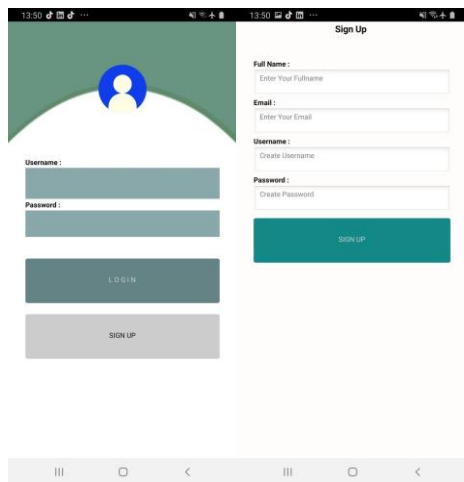


Figure 2 Login and Signup Page

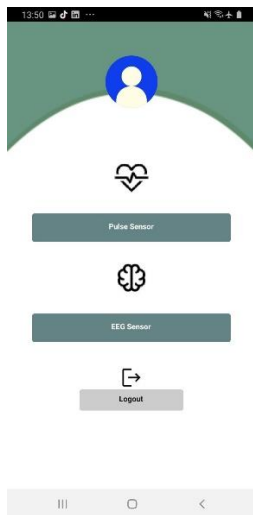


Figure 3 Dashboard Page

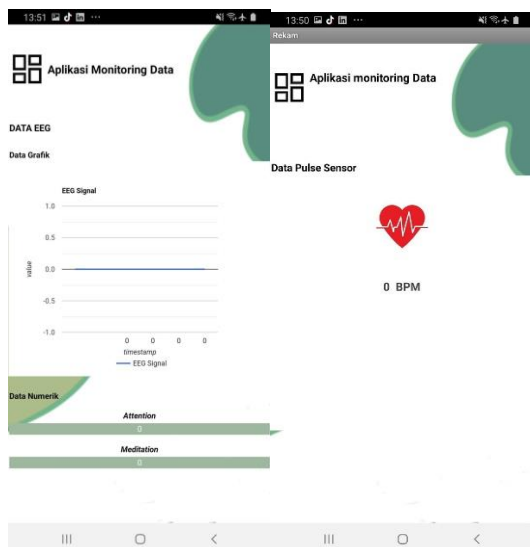


Figure 4 Data Page

### III. RESULT AND DISCUSSION

The test will be carried out in calm and relaxed conditions using the device within 2-3 minutes of the heart

rate sensors and EEG sensors by obtaining 32 data from it. One of the test result can be seen in Figure 5 that successfully displayed the EEG graphic data, number of attention and meditation, and the heartrate data.

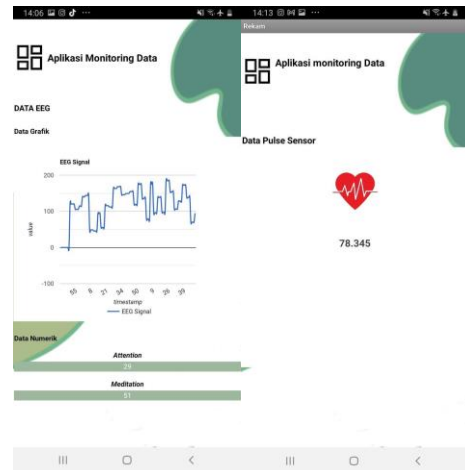


Figure 5 Test Result

#### A. Sensor Pulse & Scheduling

Firstly, when connecting the heart rate detection sensor, we can see that by connecting the Pulse Sensor to the ESP 32, by attaching the sensor to the user's finger, the user can display results in the form of BPM data.

Second, in the scheduling process or scheduling data delivery, we can see that by managing the data delivery process, measurement results can be more accurate. This scheduling process uses the priority scheduling method. Where each Pulse Sensor sends data five times, first after the data has been collected, the data is averaged. So you can get good results.

B. The test results in this research include test results with raw data, reconstructed data into wavelet values, graphic results based on calculation results between raw data and reconstruction data and Mean Absolute Error (MAE) calculation results for raw data calculations with reconstruction data along with the results of Signal-to-Noise Ratio (SNR) value from the reconstructed value.

##### 1) Raw data testing results

Table 3 Raw Data Table

No	Beta signal raw data
1	177.5
2	98
3	168
4	143.5
5	244
6	115
7	88.5
8	138.5
9	70
10	60
11	107
12	157
13	122
14	82.5
15	150.5

16	218
----	-----

The table shows that the raw data from the beta signal accumulation test results have varying amplitude values. This data has a minimum value of 60 and a maximum value of 244. However, this value can be higher or lower depending on the condition of the Neurosky Mindwave device user.

2) Results of testing reconstruction data

The data that was displayed in the previous results was then reconstructed into wavelet data using PyWT which was explained in CD4. Then the results are then filtered using the Gaussian algorithm and then processed with the OMP algorithm. The results of the reconstruction data can be shown in the following table.

Table 4 Data from reconstruction results

No	Reconstructed data
1	177.5
2	121.89
3	164.26
4	147.12
5	238.21
6	148.33
7	90.65
8	125.62
9	73.78
10	66.79
11	104.3
12	139.27
13	123.81
14	96.18
15	146.7
16	193.91

The data contained in the table shows that the results of the beta signal amplitude reconstruction have a slight difference and these results have detailed values from the raw data with an accuracy of 0.01.

3) Results of Mean Absolute Error and Signal-to-Noise Ratio values

The error value from the results of the reconstruction data to the results from the raw data is calculated again using the Mean Absolute Error (MAE) calculation which has the following formula:

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_i - x|$$

With the Xi value being the value of the raw data and the X value being the value of the reconstructed data. Based on this formula, the MAE results are 9.98625 with the following error ratio.

Table 5 Data Error Ratio

No	Error ratio in data
1	0
2	23.89
3	3.74
4	3.62

5	5.79
6	33.33
7	2.15
8	12.88
9	3.78
10	6.79
11	2.7
12	17.73
13	1.81
14	13.68
15	3.8
16	24.09

Then, the results of the mean absolute error are used to calculate the Signal-to-Noise Ratio (SNR) value from the reconstruction data. The formula for the SNR value is as follows.

$$SNR = \frac{P_{signal}}{P_{noise}}$$

The Psignal value shows the value of each reconstruction data, while Pnoise is the value of the Mean Absolute Error (MAE). Based on this formula, an SNR result can be obtained from the reconstruction data which can be shown in the following table.

Tabel III.1 Nilai SNR Data Rekonstruksi

No	Reconstructed data SNR value
1	-7,502040768
2	-9,134343571
3	-7,838706157
4	-8,317307179
5	-6,224424451
6	-8,281734374
7	-10,42034626
8	-9,003436451
9	-11,31463783
10	-11,74690991
11	-9,811181257
12	-8,555448586
13	-9,066467106
14	-10,16317661
15	-8,329723203
16	-7,118022278

Based on this table, an average SNR value for the reconstruction data was obtained of 8 decibels.

C. Application

Android application can display pulse sensor data and EEG sensor data with both graphic and numeric data in real time illustrated in Figure 5. Application performance testing is carried out via Firebase Lab Test. Testing was carried out on 10 devices with different API levels. Performance of android application with the average score of time to initial display 1s 223ms, slow UI thread 4%, slow draw commands 18% and slow bitmap uploads 1%. The highest total RAM value with a range from 137K KiB to 242K KiB (140MB to 247MB) and the highest total CPU value with a range from 10%–25%.

#### IV. CONCLUSION

The results of the test showed that EEG sensor pulse sensor data were successfully displayed, and brain activity data was produced in the form of "attention" and "meditation" with a range of 0–100% and beta ( $\beta$ ) wave type EEG signals with an average value obtained of 00–200 V. The priority scheduling method obtained an average delay of 33.268 ms, throughput of 3.172 kbps, and packet loss of 0.18%. As a result of the margin of error, the average noise value with reconstructed data is 13%. So the margin of error is relatively low and stable for gathering and reconstructing data. Performance of Android applications with the average score of time to initial display of 1s 223 ms, slow UI thread 4%, slow draw commands 18%, and slow bitmap uploads 1%. The highest total RAM value with a range from 137K KiB to

242K KiB (140MB to 247MB) and the highest total CPU value with a range from 10%–25%. Based on the overall test results, it can be inferred that the system equipment as a whole is considered suitable for use.

#### REFERENCES

- [1] [1] "Brain health," World Health Organization, <https://www.who.int/health-topics/brain-health> (accessed Sep. 11, 2023).
- [2] Kementerian Kesehatan Republik Indonesia, <https://www.kemkes.go.id/downloads/resources/download/pusdatin/infodatin/Infodatin-Lansia-2022.pdf> (accessed Oct. 30, 2022).
- [3] 1910104114 Putri Lestari, S. SiT. Sri Subiyatun, and S. ST. Istri Utami, "Literature review Gambaran Autis Pada anak usia 0–10 Tahun," DIGILIB UNISAYOGYA, <http://digilib.unisayogya.ac.id/5268/> (accessed 2023).