

## LAMPIRAN

1. Lembar bimbingan
2. *Source code* aplikasi *android*
3. *Source code* PHP

## 1. Source code aplikasi android

### a. MainActivity.java

```
1 package project.pi.monitoring;
2
3 import android.support.v4.app.Fragment;
4 import android.support.v4.app.FragmentManager;
5 import android.support.v4.app.FragmentManagerAdapter;
6 import android.support.v4.app.NotificationCompat;
7 import android.support.v4.app.NotificationManagerCompat;
8 import android.support.v4.view.ViewPager;
9 import android.support.v7.app.ActionBar;
10 import android.support.v7.app.AppCompatActivity;
11 import android.app.Notification;
12 import android.app.NotificationManager;
13 import android.app.PendingIntent;
14 import android.content.Context;
15 import android.content.Intent;
16 import android.graphics.drawable.ColorDrawable;
17 import android.os.Bundle;
18 import android.view.Menu;
19 import android.view.MenuItem;
20
21
22 public class MainActivity extends AppCompatActivity{
23
24     ViewPager viewPager = null;
25     ActionBar actionBar;
26
27     private static Context context;
28
29     private static final int NOTIFY_ME_ID=3626;
30
31     @Override
32     protected void onCreate(Bundle savedInstanceState) {
33         super.onCreate(savedInstanceState);
34         setContentView(R.layout.mainactivity_layout);
35
36         actionBar = getSupportActionBar();
37         actionBar.setBackgroundDrawable(new ColorDrawable(getResources()
38             .getColor(R.color.colorActionBar)));
39
40
41         viewPager = (ViewPager)findViewById(R.id.pager);
42
43         FragmentManager fragmentManager = getSupportFragmentManager();
44         viewPager.setAdapter(new MyAdapter(fragmentManager));
45
46
47     }
48
49
50     @Override
51     public void onBackPressed() {
52         // TODO Auto-generated method stub
53         moveTaskToBack(true);
54     }
55
56
57
58
```

```

59     @Override
60     public boolean onCreateOptionsMenu(Menu menu) {
61         // Inflate the menu; this adds items to the action bar if it is present.
62         //getMenuInflater().inflate(R.menu.main, menu);
63         return true;
64     }
65
66     @Override
67     public boolean onOptionsItemSelected(MenuItem item) {
68         // Handle action bar item clicks here. The action bar will
69         // automatically handle clicks on the Home/Up button, so long
70         // as you specify a parent activity in AndroidManifest.xml.
71
72         /*
73         int id = item.getItemId();
74         if (id == R.id.action_settings) {
75             return true;
76         }
77         */
78
79         return super.onOptionsItemSelected(item);
80     }
81 }
82 }
83
84 class MyAdapter extends FragmentPagerAdapter{
85
86     public MyAdapter(FragmentManager fm) {
87         super(fm);
88         // TODO Auto-generated constructor stub
89     }
90
91     class MyAdapter extends FragmentPagerAdapter{
92
93     public MyAdapter(FragmentManager fm) {
94         super(fm);
95         // TODO Auto-generated constructor stub
96     }
97
98     @Override
99     public Fragment getItem(int i) {
100         // TODO Auto-generated method stub
101
102         Fragment fragment=null;
103
104         if(i==0){
105             fragment = new Fragment_current();
106         }
107         if(i==1){
108             fragment = new Fragment_graph();
109         }
110         if(i==2){
111             fragment = new Fragment_cctv();
112         }
113
114         return fragment;
115     }
116
117     @Override
118     public int getCount() {
119         // TODO Auto-generated method stub
120         return 3;
121     }

```

```
112     @Override
113     public int getCount() {
114         // TODO Auto-generated method stub
115         return 3;
116     }
117
118     @Override
119     public CharSequence getPageTitle(int position) {
120         // TODO Auto-generated method stub
121
122         if(position==0){
123             return "Temperatur Saat Ini";
124         }
125         if(position==1){
126             return "Data Grafik";
127         }
128         if(position==2){
129             return "Monitoring Ruangan";
130         }
131
132
133         return null;
134     }
135 }
136
137
138
139 }
```

## b. Fragment\_current.java

```
1 package project.pi.monitoring;
2
3 import java.util.ArrayList;
4 import java.util.List;
5 import java.util.Timer;
6 import java.util.TimerTask;
7
8 import org.apache.http.NameValuePair;
9 import org.json.JSONArray;
10 import org.json.JSONException;
11 import org.json.JSONObject;
12
13 import android.app.Notification;
14 import android.app.NotificationManager;
15 import android.app.PendingIntent;
16 import android.app.ProgressDialog;
17 import android.content.Context;
18 import android.content.Intent;
19 import android.net.NetworkInfo.DetailedState;
20 import android.os.AsyncTask;
21 import android.os.Bundle;
22 import android.os.Handler;
23 import android.support.annotation.Nullable;
24 import android.support.v4.app.Fragment;
25 import android.support.v4.app.NotificationCompat;
26 import android.view.LayoutInflater;
27 import android.view.MotionEvent;
28 import android.view.View;
29 import android.view.ViewGroup;
30 import android.widget.ScrollView;
31 import android.widget.TextView;
32
33     View LayoutView;
34     TextView txtTitle, txtTemperature, txtTitleLastUpdate, txtLastUpdate, txtTap, txtConvert, txtTip;
35     ScrollView scrLView;
36
37     //progress dialog
38     private ProgressDialog pDialog;
39
40     //JSON parser Object
41     JSONParser jParser = new JSONParser();
42
43     //url untuk mendapatkan data
44     private static String urlCurrentTemp = "http://portalbaru.unsada.ac.id/home/layanan/suhu_server/android/current_temperature.php";
45
46     JSONArray currentTemperature = null;
47
48     //declare variable
49     private static final String TAG_STATUS = "status";
50     private static final String TAG_TEMPERATURE = "last_temperature";
51     private static final String TAG_LASTTEMP = "lasttemp";
52     private static final String TAG_LASTTEMPF = "lasttempf";
53     private static final String TAG_LASTTEMPK = "lasttempk";
54     private static final String TAG_LASTTEMPRE = "lasttempre";
55     private static final String TAG_LASTTIME = "lasttime";
56     private static final String TAG_LASTDATE = "lastdate";
57
58     //setting refresh interval
59     public static final int INTERVAL = 1000 * 60 * 10;
60
61     //declare id for notification
62     private static final int NOTIFY_ME_ID=3626;
```

```

67 //set temperature limit rule
68 public float limitTemperature = 27;
69 public float nowTemperature = 0;
70 public float prevTemperature = 0;
71
72 public float warningTemperature = (float)limitTemperature - 2;
73 public float overTemperature = (float)limitTemperature + 1;
74
75 //notif status
76 Boolean notifStatCool = false;
77 Boolean notifStatWarm = false;
78 Boolean notifStatHot = false;
79 Boolean notifStatOver = false;
80
81 //update count
82 public int updateCount = 6;
83
84
85 //declare other variable
86 Handler handler;
87 Timer timer;
88 TimerTask task;
89
90
91 @Override
92 public void onCreate(@Nullable Bundle savedInstanceState) {
93     // TODO Auto-generated method stub
94     super.onCreate(savedInstanceState);
95
96     handler = new Handler();
97     timer = new Timer();
98
99
100 }
101
102
103
104 @Override
105 @Nullable
106 public View onCreateView(LayoutInflater inflater,
107     @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
108     // TODO Auto-generated method stub
109
110     // start get widget from xml
111     LayoutView = inflater.inflate(R.layout.fragment_current_layout, container, false);
112
113     scrollView = (ScrollView)LayoutView.findViewById(R.id.scrollView1);
114
115     txtTitle = (TextView)LayoutView.findViewById(R.id.txtTitleTemperature);
116     txtTemperature = (TextView)LayoutView.findViewById(R.id.txtTemperature);
117     txtTitleLastUpdate = (TextView)LayoutView.findViewById(R.id.txtTitleLastUpdate);
118     txtLastUpdate = (TextView)LayoutView.findViewById(R.id.txtDateLastUpdate);
119     txtTap = (TextView)LayoutView.findViewById(R.id.txtTapRefresh);
120     txtConvert = (TextView)LayoutView.findViewById(R.id.txtConvertTemperature);
121     txtTip = (TextView)LayoutView.findViewById(R.id.txtTipTodo);
122     txtTip.setVisibility(View.GONE);
123     txtTip.setText("");

```

```

125 //end of get widget from xml
126
127 /*-----*/
128
129 //start set font size
130
131 txtTitle.setTextSize(12 * getResources().getDisplayMetrics().density);
132 txtTemperature.setTextSize(55 * getResources().getDisplayMetrics().density);
133 txtConvert.setTextSize(8 * getResources().getDisplayMetrics().density);
134 txtTip.setTextSize(10 * getResources().getDisplayMetrics().density);
135 txtTitleLastUpdate.setTextSize(7 * getResources().getDisplayMetrics().density);
136 txtLastUpdate.setTextSize(9 * getResources().getDisplayMetrics().density);
137 txtTap.setTextSize(5 * getResources().getDisplayMetrics().density);
138
139 //end of set font size
140
141 /*-----*/
142
143
144 //function for refresh
145 scrollView.setOnTouchListener(new View.OnTouchListener() {
146
147     @Override
148     public boolean onTouch(View v, MotionEvent event) {
149         // TODO Auto-generated method stub
150
151         switch (event.getAction()) {
152             case MotionEvent.ACTION_UP:
153                 v.performClick();
154                 new loadCurrentTemperature().execute();
155                 break;
156             break;
157
158             default:
159                 break;
160         }
161         return true;
162     }
163 });
164
165
166 return LayoutView;
167
168 }
169
170 @Override
171 public void onActivityCreated(@Nullable Bundle savedInstanceState) {
172     // TODO Auto-generated method stub
173     super.onActivityCreated(savedInstanceState);
174
175     ;
176 }
177
178 @Override
179 public void onResume() {
180     // TODO Auto-generated method stub
181     super.onResume();
182
183     timer.cancel();
184

```

```

185     task = new TimerTask() {
186         @Override
187         public void run() {
188             handler.post(new Runnable() {
189                 public void run() {
190                     new loadCurrentTemperature().execute();
191                 }
192             });
193         }
194     };
195     timer = new Timer();
196     timer.schedule(task, 0, INTERVAL); //it executes this every 1000ms
197
198 }
199
200
201 @Override
202 public void onPause() {
203     // TODO Auto-generated method stub
204     super.onPause();
205 }
206
207
208 //method for notification
209 public void temperatureNotif (String message){
210
211     final Intent toIntent = new Intent();
212     toIntent.setClass(getActivity(),MainActivity.class);
213     PendingIntent i = PendingIntent.getActivity(getActivity(), 0, toIntent, NotificationCompat.FLAG_ONGOING_EVENT);
214
215     NotificationCompat.Builder note = new NotificationCompat.Builder(getActivity())
216         .setContentTitle("Notifikasi Temperatur")
217         .setContentText("Suhu Ruangan: " + nowTemperature)
218         .setStyle(new NotificationCompat.BigTextStyle().bigText("Suhu Ruangan: " + nowTemperature +
219             getResources().getString(R.string.dummyCelsius) + " \n" +message))
220         .setSmallIcon(R.drawable.ic_launcher)
221         .setTicker(message)
222         .setAutoCancel(true)
223         .setContentIntent(i)
224         .setDefaults(Notification.DEFAULT_SOUND);
225
226
227     final NotificationManager notifManager = (NotificationManager)getActivity().getSystemService(Context.NOTIFICATION_SERVICE);
228     notifManager.notify(NOTIFY_ME_ID, note.build());
229
230 }
231
232 //method for notification
233 public void changeNotifStat(Boolean statCool, Boolean statWarm, Boolean statHot, Boolean statOver){
234
235     notifStatCool = statCool;
236     notifStatWarm = statWarm;
237     notifStatHot = statHot;
238     notifStatOver = statOver;
239
240 }
241
242
243 //Background AsyncTask untuk mengambil data dari mysql
244 class loadCurrentTemperature extends AsyncTask<String, String, String>{
245
246

```



```

246
247 //menyimpan ke variabel
248 String lasttemp = null;
249 String lasttempf = null;
250 String lasttempk = null;
251 String lasttempre = null;
252 String lasttime = null;
253 String lastdate = null;
254
255 Boolean alert = false ;
256
257 //sebelum memulai proses background tampilkan progress dialog
258 @Override
259 protected void onPreExecute() {
260     // TODO Auto-generated method stub
261     super.onPreExecute();
262     pDialog = new ProgressDialog(getActivity());
263     pDialog.setMessage(getString(R.string.currentPleaseWait));
264     pDialog.setIndeterminate(false);
265     pDialog.setCancelable(false);
266     pDialog.show();
267
268 }
269
270
271 //Mengambil Data Temperatur terakhir
272 @Override
273 protected String doInBackground(String... params) {
274
275     List<NameValuePair> dataToSend = new ArrayList<NameValuePair>();
276
277     JSONObject json = jParser.makeHttpRequest(urlCurrentTemp, "POST", dataToSend);
278
279     --
280
281     //Log.d("Current Temperature: ", json.toString());
282
283     try {
284
285         //checking tag status
286         int status = json.getInt(TAG_STATUS);
287
288         if (status == 1){
289
290             alert = false;
291
292             //last temperature found
293             //mengambil array last_temperature
294             currentTemperature = json.getJSONArray(TAG_TEMPERATURE);
295
296             //looping semua obyek json
297             for(int i = 0; i < currentTemperature.length(); i++){
298
299                 JSONObject c = currentTemperature.getJSONObject(i);
300
301                 //menyimpan ke variabel
302                 lasttemp = c.getString(TAG_LASTTEMP);
303                 lasttempf = c.getString(TAG_LASTTEMPF);
304                 lasttempk = c.getString(TAG_LASTTEMPK);
305                 lasttempre = c.getString(TAG_LASTTEMPRE);
306                 lasttime = c.getString(TAG_LASTTIME);
307                 lastdate = c.getString(TAG_LASTDATE);
308
309             }

```

```

310 }else{
311
312     alert = true;
313
314 }
315
316 } catch (JSONException e) {
317     e.printStackTrace();
318 }
319
320     return null;
321
322 }
323
324
325 //setelah selesai proses background, hilangkan progress dialog
326
327 @Override
328 protected void onPostExecute(String result) {
329
330     //dismiss progress dialog
331     pDialog.dismiss();
332
333     //update tampilan dari background thread
334     getActivity().runOnUiThread(new Runnable() {
335
336         @Override
337         public void run() {
338
339             //update data from variable to TextView
340             txtTemperature.setText(lasttemp + getResources().getString(R.string.dummyCelsius));
341             txtConvert.setText(lasttempf + getResources().getString(R.string.dummyFahrenheit) + " | "
342                 + lasttempk + getResources().getString(R.string.dummyKelvin) + " | "
343                 + lasttempre + getResources().getString(R.string.dummyReaumur));
344             txtLastUpdate.setText(lastdate + " - " + lasttime);
345
346             nowTemperature = Float.parseFloat(lasttemp);
347
348             //notification condition
349             if(prevTemperature <= nowTemperature){
350
351                 if(nowTemperature < limitTemperature && nowTemperature >= warningTemperature){
352                     txtTemperature.setTextColor(getResources().getColor(R.color.colorTemperaturWarm));
353                     txtTip.setVisibility(View.VISIBLE);
354                     txtTip.setTextColor(getResources().getColor(R.color.colorTemperaturWarm));
355                     txtTip.setText(getResources().getString(R.string.currentTipWarm));
356                     if(notifStatWarm.equals(true)) temperatureNotif(getResources().getString(R.string.notifWarmUp));
357                     changeNotifStat(true, false, true, true);
358                 }else if (nowTemperature > limitTemperature && nowTemperature >= overTemperature) {
359                     txtTemperature.setTextColor(getResources().getColor(R.color.colorTemperaturHot));
360                     txtTip.setVisibility(View.VISIBLE);
361                     txtTip.setTextColor(getResources().getColor(R.color.colorTemperaturHot));
362                     txtTip.setText(getResources().getString(R.string.currentTipHot));
363                     if(notifStatOver.equals(true)) temperatureNotif(getResources().getString(R.string.notifOver));
364                     changeNotifStat(true, true, true, false);
365                 }else if (nowTemperature >= limitTemperature && nowTemperature > warningTemperature){
366                     txtTemperature.setTextColor(getResources().getColor(R.color.colorTemperaturHot));
367                     txtTip.setVisibility(View.VISIBLE);
368                     txtTip.setTextColor(getResources().getColor(R.color.colorTemperaturHot));
369                     txtTip.setText(getResources().getString(R.string.currentTipHot));
370                     if(notifStatHot.equals(true)) temperatureNotif(getResources().getString(R.string.notifHot));

```

```

370         if(notifStatHot.equals(true))temperatureNotif(getResources().getString(R.string.notifHot));
371         changeNotifStat(true, true, false, true);
372     }
373
374
375     }else if(prevTemperature >= nowTemperature){
376
377         if(nowTemperature < limitTemperature && nowTemperature >= warningTemperature){
378             txtTemperature.setTextColor(getResources().getColor(R.color.colorTemperaturWarm));
379             if(notifStatWarm.equals(true))temperatureNotif(getResources().getString(R.string.notifWarmDown));
380             txtTip.setVisibility(View.VISIBLE);
381             txtTip.setTextColor(getResources().getColor(R.color.colorTemperaturWarm));
382             txtTip.setText(getResources().getString(R.string.currentTipWarmDown));
383             changeNotifStat(true, false, true, true);
384         }else if (nowTemperature < limitTemperature && nowTemperature < warningTemperature) {
385             txtTemperature.setTextColor(getResources().getColor(R.color.colorTemperaturNormal));
386             txtTip.setVisibility(View.GONE);
387             txtTip.setText("");
388             if(notifStatCool.equals(true))temperatureNotif(getResources().getString(R.string.notifCool));
389             changeNotifStat(false, true, true, true);
390         }
391     }
392     prevTemperature = nowTemperature;
393
394     updateCount = updateCount + 1;
395     if(updateCount >= 5) {
396         changeNotifStat(false, true, true, true);
397         updateCount = 0;
398     }
399 }
400 });
401
402 }

```

## 2. Sourcecode PHP

### a. Current\_temperature.php

```
1 <!-- konten yang berisi status temperatur terakhir, di include pada halaman temperatur -->
2 <!-- konten hanya akan muncul jika halaman temperatur dipilih -->
3 <?php
4
5 //menampilkan konten jika halaman temperatur dipilih
6 if($page=="temp"){
7
8 //koneksi ke database
9 include "connect.php";
10
11 //query untuk mendapatkan value temperatur terakhir
12 $query = "SELECT waktu, temperatur FROM pi_temperature ORDER BY waktu DESC LIMIT 1";
13 $sql = mysqli_query($connect,$query);
14 $result = mysqli_fetch_array($sql,MYSQLI_BOTH);
15
16 //memindahkan data yang sudah diambil dari database ke variable lain
17 $lasttemp = $result['temperatur']; //nilai temperatur asli dalam satuan Celsius
18 $lasttempf = ($lasttemp * 9/5) + 32; //mengkonversi nilai temperatur asli menjadi satuan Fahrenheit
19 $lasttempk = $lasttemp + 273.15; //mengkonversi nilai temperatur asli menjadi satuan Kelvin
20 $lasttempre = $lasttemp * 0.8; //mengkonversi nilai temperatur asli menjadi satuan Reaumur
21
22 //mengubah format waktu dan tanggal
23 $datetime = strtotime($result['waktu']);
24 $lasttime = date('H:i',$datetime);
25 $lastdate = date('d/M/Y',$datetime);
26
27
28 //sendmail code
29 /*$_SESSION['last_temp'] = $lasttemp;
30 $_SESSION['notifcount'] = $_SESSION['notifcount'] + 1;
31
32 if($_SESSION['notifcount']>5){
33     $_SESSION['notifcount']=1;
34 }
35
36 echo $_SESSION['notifcount'];
37 echo "<px>";
38 echo $_SESSION['last_temp'];
39 echo "<px>";
40 */
41 //include "send_mail.php";
42 //send mail end
43
44 -?>
45
46 <!-- div untuk data temperatur terakhir yang diambil -->
47 <!-- Menampilkan data -->
48
49 <div class="row">
50 <div>
51 <center>
52 <h2>Temperatur Ruang Server</h2>
53 <p style="font-size:6em"><?php echo $lasttemp; ?> °C</p>
54 <p><?php echo $lasttempf . " °F "; <?php echo $lasttempk . " °K "; <?php echo $lasttempre . " °Ré"></p>
55 <h4 class="info">Update Terakhir : <?php echo $lastdate . " - " . $lasttime . " WIB"; ?></h4>
56
57 <br/>
58 <hr>
59
60 </center>
61 </div>
```

```
63 | </div>
64 |
65 | <!-- div untuk judul grafik pada halaman temperatur -->
66 | <!-- Judul utama grafik -->
67 |
68 | <div class="row well well-sm">
69 |     <center><h2>Data Grafik Temperatur Ruang Server</h2></center>
70 | </div>
71 |
72 | <?php
73 |
74 |     mysqli_close($connect);
75 |
76 | }
77 |
78 | ?>
```

## b. Get\_xml.php

```
1 <?php
2
3 //GET identifier dari ajax untuk menentukan query yang akan digunakan
4 if(isset($_GET['val'])){
5     $interval = $_GET['val'];
6 }else{
7     $interval = "";
8 }
9
10 //koneksi ke database
11 include "connect.php";
12
13 //menentukan table
14 $pi_table = "pi_temperature";
15
16 /*****
17 *
18 * Query untuk mengambil sumber data grafik
19 *
20 * query akan dijalankan sesuai GET yang didapat.
21 * 12h => untuk mengambil data temperatur saat ini hingga 2 jam kebelakang
22 * 124h => mengambil data temperatur selama 24 jam terakhir dimulai dari waktu saat ini
23 * 13d => mengambil data 3 hari terakhir dari waktu saat ini
24 * 11w => mengambil data 1 minggu terakhir dari waktu saat ini
25 * 11m => mengambil data rata-rata temperatur per hari selama 1 bulan terakhir dari waktu sekarang
26 * 13m => mengambil data rata-rata temperatur per minggu selama 3 bulan terakhir dari waktu sekarang
27 *
28 *
29 *****/
30
31 if($interval === "" || $interval == "12h"){
32     $query = "SELECT waktu, temperatur FROM $pi_table WHERE waktu >= now() - INTERVAL 2 HOUR ORDER BY waktu" ;
33 }
34 else if($interval == "124h"){
35     $query = "SELECT waktu, temperatur FROM $pi_table WHERE waktu >= now() - INTERVAL 1 DAY ORDER BY waktu" ;
36 }
37 else if($interval == "13d"){
38     $query = "SELECT waktu, temperatur FROM $pi_table WHERE waktu >= now() - INTERVAL 3 DAY ORDER BY waktu" ;
39 }
40 else if($interval == "11w"){
41     $query = "SELECT waktu, temperatur FROM $pi_table WHERE waktu >= now() - INTERVAL 7 DAY ORDER BY waktu" ;
42 }
43 else if($interval == "11m"){
44     $query = "SELECT DATE(waktu) as waktu, ROUND(AVG(temperatur),2) as temperatur FROM $pi_table WHERE waktu >= now() - INTERVAL 1 MONTH GROUP BY DAY(waktu)
45     ORDER BY waktu" ;
46 }
47 else if($interval == "13m"){
48     $query = "SELECT DATE(waktu) as waktu, ROUND(AVG(temperatur),2) as temperatur FROM $pi_table WHERE waktu >= NOW() - INTERVAL 3 MONTH GROUP BY WEEK(waktu)
49     ORDER BY waktu" ;
50 }
51 //menjalankan query yang sudah disiapkan sebelumnya
52 $result = mysqli_query($connect,$query);
53
54 //menentukan zona waktu
55 date_default_timezone_set("Asia/Bangkok");
56
57 echo "<?xml?>\n";
58 echo "<meta>\n";
59 echo "<legend>\n";
60 echo "<entry>Temperatur Server</entry>\n";
61 echo "</legend>\n";
62 echo "</meta>\n";
63 echo "<data>\n";
64
65 while ($row = mysqli_fetch_array($result,MYSQLI_BOTH)) {
66
67     $time = strtotime($row['waktu']);
68
69     echo "<row>\n";
70     echo "<t>.$time.</t>\n";
71     echo "<v>.$row['temperatur'].</v>\n";
72     echo "</row>\n";
73 }
74
75 echo "</data>\n";
76 echo "</?xml?>";
77
78 //print_r($data);
79
80 //menutup koneksi ke database
81 mysqli_close($connect);
82
83 ?>
```



## LEMBAR BIMBINGAN SKRIPSI

### TEKNIK INFORMATIKA – DARMA PERSADA

**NIM** : 2010230106  
**NAMA LENGKAP** : Achmad Sayuti  
**DOSEN PEMBIMBING** : Adam Arif Budiman, S.T., M.Kom.  
**JUDUL SKRIPSI** : PERANCANGAN SISTEM MONITORING SUHU MENGGUNAKAN RASPBERRY PI BERBASIS WEB DAN ANDROID PADA RUANG SERVER UNIVERSITAS DARMA PERSADA

No.	Pertemuan	Tanggal	Pokok Bahasan	Paraf
1.	I	5 Mei 2015	Penyerahan Bab I	
2.	II	15 Mei 2015	Uji coba perangkat, <i>Raspberry Pi</i> dan sensor suhu	
3.	III	29 Mei 2015	Penyerahan Bab II	
4.	IV	12 Juni 2015	Menambahkan webcam ke dalam sistem monitoring , Revisi Bab II, penyerahan Bab III	
5.	V	3 Juli 2015	Uji coba awal untuk mengetahui sistem sudah berfungsi, Revisi Bab III	
6.	VI	31 Juli 2015	Penyerahan Bab IV dan Bab V	
7.	VII	21 Agustus 2015	Revisi Bab IV dan Bab V, Persetujuan sidang isi	
8.	VIII	31 Agustus 2015	Revisi Sidang isi, Persetujuan sidang akhir	

**Jakarta, 2 September 2015**  
**Dosen Pembimbing**

**Adam Arif Budiman, S.T., M.Kom.**