

## Lampiran 1. Surat Pernyataan Kerjasama dengan Pihak lain



PEMERINTAH PROVINSI JAWA TENGAH  
DINAS PENDIDIKAN DAN KEBUDAYAAN  
**SEKOLAH MENENGAH KEJURUAN NEGERI 1  
PURWOKERTO**

Jalan dr. Soeparno No.29 Purwokerto Timur Kode Pos 53111 Telp. (0281) 637132  
Fax. (0281) 637132 Web Site : www.smkn1purwokerto.sch.id Email : admin@smkn1purwokerto.sch.id

### SURAT KETERANGAN

Nomor : 423.5 / 1085

Yang bertanda tangan di bawah ini Kepala SMK Negeri 1 Purwokerto menerangkan bahwa :

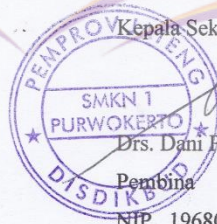
Nama : Fauzan Daffa Ibnu Yusuf  
NIM : 16.12.0073  
Fak./Program Studi. : S1 / System Informasi  
Institusi : AMIKOM Purwokerto

Benar-benar telah melaksanakan Penelitian di SMK Negeri 1 Purwokerto selama 2 (dua) hari mulai 22 dan 23 April 2019 dengan judul **Aplikasi Bursa kerja khusus Sebagai Media Informasi Lowongan Kerja Berbasis Mobail (Study Kasus SMK Negeri 1 Purwokerto).**

Demikian surat keterangan ini dibuat untuk dapat digunakan sebagaimana mestinya.

Purwokerto, 20 September 2019

Kepala Sekolah,



Drs. Dami Priya Widada.

Pembina

NIP. 19680202 199412 1 005

## Lampiran 2. Hasil Wawancara

**Nama** : Drs. Agus Nuryanto

**Jabatan** : Ketua Bursa Kerja Khusus (BKK) SMK Negeri 1 Purwokerto

1. Bagaimana proses penyebaran lowongan kerja yang dilakukan BKK??
  - Dilakukan melalui aplikasi whatsapp untuk menyebarkan lowongan kerja kepada alumni. Dengan mengirimkan informasi lowongan kerja melalui grup alumni..
2. Dari mana BKK mendapatkan lowongan kerja untuk alumni?
  - Dari MOU dengan industri/pihak pemberi lowongan.
3. Bagaimana cara pendaftaran kerjanya untuk lowongan dari BKK?
  - Menggunakan google form dengan cara mengumpulkan biodata yang dibutuhkan, yang nantinya pihak BKK akan mengirim biodata ke pihak pemberi lowongan kerja.
4. Apakah hanya dengan mengumpulkan biodata untuk pendaftaran kerjanya?
  - Tidak, pendaftar nantinya harus mengumpulkan syarat lain yang diminta oleh pihak pemberi lowongan kerja sesuai yang tertulis diinformasi lowongan kerja, persyaratan itu dibawa pada saat tes.
5. Untuk apa biodata dikumpulkan secara online?
  - Biodata digunakan untuk mengumpulkan data pendaftar lowongan kerja yang nantinya akan dikirimkan ke pihak pemberi lowongan kerja sebagai data pendaftar yang berasal dari SMK N 1 Purwokerto dan juga sebagai catatan BKK siapa saja yang akan mendaftar lowongan kerja tersebut, karena lowongan kerja dari BKK itu tidak semua orang bisa mendaftar, tergantung MoU antara pihak pemberi lowongan dan BKK lainnya.

6. Apa kendala yang dialami BKK dalam menyebarkan informasi lowongan kerja?
- Kendalanya adalah tidak efisien dalam menyebarkan lowongan kerja, harus satu persatu dikirim ke grup alumni, sedangkan tiap tahun grup alumni semakin banyak, bahkan kadang harus mengirim informasi ke grup kelas alumni. Dan setiap mengirim informasi lowongan kerja kami membuat google form untuk yang akan mendaftar atau merekap alumni yang mendaftar lewat whatsapp.
7. Apakah BKK membutuhkan sistem untuk kendala tersebut?
- Sekiranya kami membutuhkan sistem yang bisa membuat informasi lowongan kerja dapat terpusat menjadi 1 sistem agar memudahkan dalam menyebarkan informasi lowongan kerja, dan membutuhkan sistem untuk pendaftaran dimana kami tidak perlu membuat google form untuk tiap lowongan kerja dan untuk merekap data pendaftar untuk tiap lowongan kerja.

Purwokerto, 17 Desember 2019

Ketua BKK SMK Negeri 1 Purwokerto



**Drs. Agus Nuryanto**

### Lampiran 3. Kuesioner *Acceptance Testing*



**UNIVERSITAS AMIKOM PURWOKERTO**  
**Jl. Letjen Pol Sumarto Watumas Purwanegara Purwokerto Utara,**  
**Telp/Fax. (0281) 623321/623196**

---

**Responden**  
**Yang terhormat,**

Dalam rangka hasil penelitian yang menghasilkan solusi berupa hasil aplikasi *Bursa Kerja Khusus SMK Negeri 1 Purwokerto* guna memusatkan informasi lowongan kerja dan memudahkan pendaftaran lowongan kerja bagi pengguna, penulis berusaha mengumpulkan data tentang penilaian kepuasan siswa dan alumni SMK Negeri 1 Purwokerto terhadap aplikasi *Bursa Kerja Khusus SMK Negeri 1 Purwokerto*. Besar harapan penulis agar responden bersedia untuk menjawab pernyataan-pernyataan yang sudah tersedia dalam angket ini. Atas kesediaannya, penulis mengucapkan terima kasih.

**KUIISIONER ACCEPTANCE TESTING APLIKASI**  
**BURSA KERJA KHUSUS SMK NEGERI 1 PURWOKERTO**

Tahun 2019

**Identitas Responden**

Nama : .....

Status : Alumni/Siswa

Jenis Kelamin : Laki-laki/Perempuan

Email : .....

**Petunjuk Pengisian**

1. Tujuan angket ini dilakukan dengan tujuan untuk mengukur penilaian terhadap aplikasi *Bursa Kerja Khusus SMK Negeri 1 Purwokerto* sebagai upaya untuk memusatkan informasi lowongan kerja dan memudahkan pendaftaran lowongan kerja bagi pengguna.
2. Pilihlah salah satu dari alternatif yang disediakan dengan cara memberi tanda centang (✓) pada kolom yang tersedia, yaitu :

Sangat Setuju (SS)	= 5
Setuju (S)	= 4
Ragu-ragu (RG)	= 3
Tidak setuju (TS)	= 2
Sangat tidak setuju (STS)	= 1

**Berikan tanda (✓) Centang jawaban yang sesuai dengan jawaban Anda !**

No.	Pertanyaan	Jawaban				
		SS	ST	RG	TS	STS
1.	Apakah anda pengguna <i>Smartphone</i> Android					
2.	Apakah anda tahu tentang Bursa Kerja Khusus					
3.	Apakah anda tahu tentang tugas dari Bursa Kerja Khusus SMK					
4.	Aplikasi <i>Bursa Kerja Khusus SMK Negeri 1 Purwokerto</i> dapat dioperasikan dengan mudah					
5.	Desain dan tata letak menu pada aplikasi <i>Bursa Kerja Khusus SMK Negeri 1 Purwokerto</i> mudah dipahami					
6.	Fitur dalam aplikasi <i>Bursa Kerja Khusus SMK Negeri 1 Purwokerto</i> sudah lengkap					
7.	Aplikasi ini dapat mempermudah menemukan informasi lowongan kerja dari Bursa Kerja Khusus SMK					
8.	Aplikasi layak digunakan dan di implementasikan					
9.	Fitur-fitur dalam aplikasi <i>Bursa Kerja Khusus SMK Negeri 1 Purwokerto</i> masih terdapat <i>error</i>					
10.	Aplikasi ini perlu dikembangkan					

Apabila Saudara/i menemukan kekurangan dalam aplikasi ini, mohon untuk memberikan saran pengembangan selanjutnya.

.....  
 .....

Purwokerto, ..... Desember 2019

Responden,

( ..... )

#### Lampiran 4. Tabulasi Kuesioner

Respon den	Pernyataan										Juml ah	Rata- rata
	1	2	3	4	5	6	7	8	9	10		
1	5	2	2	4	4	4	5	5	4	5	40	4,00
2	5	4	4	5	5	4	5	5	2	4	43	4,30
3	5	4	4	4	5	4	5	5	3	5	44	4,40
4	5	4	3	4	5	4	4	4	3	5	41	4,10
5	5	5	4	5	4	5	5	5	2	5	45	4,50
6	5	5	5	4	4	4	5	5	2	5	44	4,40
7	5	4	4	4	5	5	5	5	1	5	43	4,30
8	5	4	4	4	4	5	5	5	1	5	42	4,20
9	5	5	5	5	5	4	5	5	2	1	42	4,20
10	5	3	3	5	5	4	5	5	3	5	43	4,30
11	5	3	3	5	5	4	5	5	3	5	43	4,30
12	3	3	3	4	4	4	4	4	3	5	37	3,70
13	5	4	4	4	4	4	4	4	4	4	41	4,10
14	5	5	4	4	4	4	4	3	4	5	42	4,20
15	5	2	2	5	5	3	3	5	1	5	36	3,60
16	5	3	3	5	4	4	5	5	2	5	41	4,10
17	5	5	4	4	5	4	5	4	3	5	44	4,40
18	5	2	2	2	2	2	4	3	3	5	30	3,00
19	5	5	4	5	5	4	5	5	5	5	48	4,80
20	5	4	4	5	5	4	4	4	2	5	42	4,20
21	5	4	3	3	4	4	5	2	3	5	38	3,80
22	5	5	5	5	5	5	5	5	2	5	47	4,70
23	5	5	5	4	4	5	5	5	2	5	45	4,50
24	5	5	5	5	5	4	4	5	2	5	45	4,50
25	5	5	5	4	4	4	5	5	2	5	44	4,40
26	5	4	5	5	5	5	5	5	5	5	49	4,90
27	5	5	5	5	5	5	5	5	5	5	50	5,00
28	5	3	3	3	3	3	3	3	3	5	37	3,70
29	5	3	3	3	3	3	3	3	3	4	36	3,60
30	5	4	5	5	5	5	5	5	5	5	49	4,90
31	4	3	3	3	3	3	4	4	3	5	35	3,50
32	5	2	2	3	3	3	4	4	3	5	34	3,40
33	5	3	4	3	3	4	5	3	3	3	36	3,60
34	5	4	3	4	3	4	3	4	3	4	37	3,70
35	5	4	3	5	4	3	5	4	3	5	41	4,10
36	5	4	5	4	5	3	3	5	5	5	44	4,40
37	5	4	4	4	4	4	4	5	4	5	43	4,30
38	5	3	4	3	5	4	4	5	4	5	42	4,20

39	5	5	3	4	4	4	5	4	3	4	41	4,10
40	5	3	4	4	3	3	5	3	4	5	39	3,90
Jumlah	197	154	150	166	169	158	179	175	120	189		
Rata-rata	4,925	3,85	3,75	4,15	4,225	3,95	4,475	4,375	3	4,725		



## Lampiran 5. Kode Program

### Kode Program ApiService

```
public interface ApiService {  
  
    // @TIPEMETHOD("API_END_POINT")  
    @GET("tampil_loker.php")  
    Call<ResponseLoker> request_show_all_loker();  
    // <ModelData> nama_method()  
  
}
```

### Kode Program InitRetrofit

```
public class InitRetrofit {  
    public static String API_URL =  
    "http://aimerstudio.tech/bkk/upload_loker/";  
  
    public static Retrofit setInit() {  
        return new Retrofit.Builder().baseUrl(API_URL)  
            .addConverterFactory(GsonConverterFactory.create())  
            .build();  
    }  
  
    public static ApiService getInstance() {  
        return setInit().create(ApiServices.class);  
    }  
}
```

### Kode Program LokerItem

```
public class LokerItem {  
    @SerializedName("kode_loker")  
    private String kodeLoker;  
    @SerializedName("tgl_arsip")  
    private String tglArsip;  
    @SerializedName("tgl_terbit")  
    private String tglTerbit;  
    @SerializedName("perusahaan")  
    private String perusahaan;  
    @SerializedName("jurusan")  
    private String jurusan;  
    @SerializedName("id")  
    private String id;  
    @SerializedName("tanggal")  
    private String tanggal;  
    @SerializedName("judul")  
    private String judul;  
    @SerializedName("isi_loker")  
    private String isiLoker;  
    @SerializedName("gambar")  
    private String gambar;  
  
    public void setKodeLoker(String kodeLoker) {  
        this.kodeLoker = kodeLoker;    }  
    public String getKodeLoker() {  
        return kodeLoker;    }  
}
```



```

public void setTglArsip(String tglArsip){
    this.tglArsip = tglArsip; }
public String getTglArsip(){
    return tglArsip; }
public void setTglTerbit(String tglTerbit){
    this.tglTerbit = tglTerbit; }
public String getTglTerbit(){
    return tglTerbit; }
public void setPerusahaan(String perusahaan){
    this.perusahaan = perusahaan; }
public String getPerusahaan(){
    return perusahaan; }
public void setJurusan(String jurusan){
    this.jurusan = jurusan; }
public String getJurusan(){
    return jurusan; }
public void setId(String id){
    this.id = id; }
public String getId(){
    return id; }
public void setTanggal(String tanggal){
    this.tanggal = tanggal; }
public String getTanggal(){
    return tanggal; }
public void setJudul(String judul){
    this.judul = judul; }
public String getJudul(){
    return judul; }
public void setIsLoker(String isiLoker){
    this.isiLoker = isiLoker; }
public String getIsiLoker(){
    return isiLoker; }
public void setGambar(String gambar){
    this.gambar = gambar; }
public String getGambar(){
    return gambar; }

@Override
public String toString(){
    return
        "LokerItem{" +
        "kode_loker = '" + kodeLoker + '\'' +
        ",tgl_arsip = '" + tglArsip + '\'' +
        ",tgl_terbit = '" + tglTerbit + '\'' +
        ",perusahaan = '" + perusahaan + '\'' +
        ",jurusan = '" + jurusan + '\'' +
        ",id = '" + id + '\'' +
        ",tanggal = '" + tanggal + '\'' +
        ",judul = '" + judul + '\'' +
        ",isi_loker = '" + isiLoker + '\'' +
        ",gambar = '" + gambar + '\'' +
        "}";
}
}

```

## Kode Program ResponseLoker

```

public class ResponseLoker{
    @SerializedName("loker")
    private List<LokerItem> loker;
    @SerializedName("status")

```

```

private boolean status;
public void setLoker(List<LokerItem> loker) {
    this.loker = loker;
}
public List<LokerItem> getLoker() {
    return loker;
}
public void setStatus(boolean status) {
    this.status = status;
}
public boolean isStatus() {
    return status;
}

@Override
public String toString() {
    return
        "ResponseLoker{" +
        "loker = '" + loker + '\\\'' +
        ", status = '" + status + '\\\'' +
        "}";
}
}

```

### Kode Program Adapter

```

public class AdapterBerita extends
    RecyclerView.Adapter<AdapterBerita.MyViewHolder> {
    Context context;
    List<LokerItem> loker;
    private HashMap<String,String> user = new HashMap<>();
    private DatabaseHandler db;
    public AdapterBerita(Context context, List<LokerItem> data_loker) {
        // Inisialisasi
        this.context = context;
        this.loker = data_loker;
    }
    @Override
    public MyViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
        View view = LayoutInflater.from(context).
            inflate(R.layout.list_loker_thumbnail, parent, false);
        MyViewHolder holder = new MyViewHolder(view);
        return holder;
    }

    @Override
    public void onBindViewHolder(MyViewHolder holder, final int position)
    {
        final String status = user.get("status");
        holder.tvJudul.setText(loker.get(position).getJudul());
        holder.tvTglTerbit.setText(loker.get(position).getTanggal());
        holder.tvOleh.setText("Oleh : "+
            loker.get(position).getPerusahaan());
        final String urlGambarLoker = "" +
            loker.get(position).getGambar();
        Picasso.with(context).load(urlGambarLoker).
            into(holder.ivGambarLoker);
        holder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

```

```

        Intent varIntent = new Intent(context,
DetailActivity.class);
        varIntent.putExtra("ID",
loker.get(position).getId());
        varIntent.putExtra("JUDUL_LOKER",
            loker.get(position).getJudul());
        varIntent.putExtra("TANGGAL_LOKER",
            loker.get(position).getTanggal());
        varIntent.putExtra("PERUSAHAAN_LOKER",
            loker.get(position).getPerusahaan());
        varIntent.putExtra("GAMBAR_LOKER", urlGambarLoker);
        varIntent.putExtra("ISI_LOKER",
            loker.get(position).getIsiLoker());
        varIntent.putExtra("KODE_LOKER",
            loker.get(position).getKodeLoker());
        context.startActivity(varIntent);
        ((LowonganKerja context).finish());
    });
}

@Override
public int getItemCount() {
    return loker.size();
}

public class MyViewHolder extends RecyclerView.ViewHolder {
    // Deklarasi widget
    ImageView ivGambarLoker;
    TextView tvJudul, tvTglTerbit, tvOleh;

    public MyViewHolder(View itemView) {
        super(itemView);
        // inisialisasi widget
        ivGambarLoker = (ImageView)
            itemView.findViewById(R.id.ivPosterLoker);
        tvJudul = (TextView)
itemView.findViewById(R.id.tvJudulLoker);
        tvTglTerbit = (TextView)
            itemView.findViewById(R.id.tvTglTerbit);
        tvOleh = (TextView) itemView.findViewById(R.id.tvOleh);
    }
}

```

## Kode Program Lowongan Kerja

```

public class LowonganKerja extends AppCompatActivity {

    Context c;
    private RecyclerView recyclerView;
    private SessionManager session;
    private DatabaseHandler db;
    private ProgressDialog pDialog;
    private HashMap<String, String> user = new HashMap<>();

    ImageView btnAK, btnAP, btnPM, btnPBS, btnMM, btnTKJ, btnRPL, btnFM;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.list_loker);
        recyclerView = (RecyclerView) findViewById(R.id.rvListLoker);
    }
}

```

```

        // RecyclerView harus pakai Layout manager
        recyclerView.setLayoutManager(new LinearLayoutManager(this));
        // Eksekusi method
        tampilLoker();
    }

    private void tampilLoker() {
        ApiService api = InitRetrofit.getInstance();
        // Siapkan request
        Call<ResponseLoker> beritaCall = api.request_show_all_loker();
        // Kirim request
        beritaCall.enqueue(new Callback<ResponseLoker>() {
            @Override
            public void onResponse(Call<ResponseLoker> call,
                Response<ResponseLoker> response) {
                // Pasikan response Sukses
                if (response.isSuccessful()){
                    Log.d("response api", response.body().toString());
                    List<LokerItem> data_loker = response.body().getLoker();
                    boolean status = response.body().isStatus();
                    if (status){
                        AdapterBerita adapter = new
AdapterBerita(LowonganKerja.this,
                            data_loker);
                        recyclerView.setAdapter(adapter);
                    } else {
                        // kalau tidak true
                        Toast.makeText(LowonganKerja.this,
                            "Tidak ada Lowongan untuk saat ini",
                            Toast.LENGTH_SHORT).show();
                    }
                }
            }
            @Override
            public void onFailure(Call<ResponseLoker> call, Throwable t) {
                t.printStackTrace();
            }
        });
    }
}

```

## Kode Program Detail Lowongan Kerja

```

public class DetailActivity extends AppCompatActivity {

    // Deklarasi
    ImageView ivGambarBerita;
    TextView tvTglTerbit, tvPenulis, tvkodeloker, txtjudul, teksLoker;
    Button btnDaftarKerja, btnHapusLoker, btnUbahLoker, btnFavorit;

    String kode_loker, id, judul_loker, tanggal_loker, perusahaan_loker,
isi_loker,
        gambar_loker, txtLoker, ambilKode;

    private SessionManager session;
    private DatabaseHandler db;
    private HashMap<String, String> user = new HashMap<>();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.detail_loker);
    }
}

```

```

// Inisialisasi
ivGambarBerita = (ImageView) findViewById(R.id.ivGambarBerita);
tvTglTerbit = (TextView) findViewById(R.id.tvTglTerbit);
txtjudul = (TextView) findViewById(R.id.txtjudul);
tvPenulis = (TextView) findViewById(R.id.tvPerusahaan);
tvkodeloker = (TextView) findViewById(R.id.tvKodeLoker);
teksLoker = (TextView) findViewById(R.id.teksLoker);

id = getIntent().getStringExtra("ID");
gambar_loker = getIntent().getStringExtra("GAMBAR_LOKER");
judul_loker = getIntent().getStringExtra("JUDUL_LOKER");
tanggal_loker = getIntent().getStringExtra("TANGGAL_LOKER");
perusahaan_loker =
getIntent().getStringExtra("PERUSAHAAN_LOKER");
isi_loker = getIntent().getStringExtra("ISI_LOKER");
txtLoker = getIntent().getStringExtra("ISI_LOKER");
kode_loker = getIntent().getStringExtra("KODE_LOKER");

btnDaftarKerja = (Button) findViewById(R.id.btnDaftar);
btnDaftarKerja.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ambilKode = tvkodeloker.getText().toString().trim();
        Intent intent = new Intent(getApplicationContext(),
            DaftarLoker.class);
        intent.putExtra("Data-Kode", ambilKode);
        startActivity(intent);
    }
});
showDetailBerita();
}

private void showDetailBerita() {
    txtjudul.setText(judul_loker);
    tvPenulis.setText("Oleh : " + perusahaan_loker);
    tvTglTerbit.setText(tanggal_loker);
    tvkodeloker.setText(kode_loker);
    teksLoker.setText(txtLoker);
    // Untuk gambar berita
    Picasso.with(this).load(gambar_loker).into(ivGambarBerita);
}

```

## Kode Program Daftar Lowongan Kerja

```

public class DaftarLoker extends AppCompatActivity {

    Button btnDaftarLoker;
    EditText edtNama, edtTptLahir, edtTglLahir, edtAsal, edtJurusan,
    edtTahun,
    edtTinggi, edtBerat, edtAlamat, edtEmail, edtNoHp, edtKode;
    String getNama, getTptLahir, getTglLahir, getAsal, getJurusan,
    getTahun,
    getTinggi, getBerat, getAlamat, getEmail, getNoHp;

    String id_kode;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}

```

```

setContentView(R.layout.daftar_loker);
getWindow().setSoftInputMode(WindowManager.LayoutParams.
    SOFT_INPUT_STATE_ALWAYS_HIDDEN);

edtAlamat = (EditText) findViewById(R.id.dAlamat);
edtNama = (EditText) findViewById(R.id.dNama);
edtTptLahir = (EditText) findViewById(R.id.dTptLahir);
edtTglLahir = (EditText) findViewById(R.id.dTglLahir);
edtAsal = (EditText) findViewById(R.id.dAsal);
edtJurusan = (EditText) findViewById(R.id.dJurusan);
edtTahun = (EditText) findViewById(R.id.dTahun);
edtTinggi = (EditText) findViewById(R.id.dTinggi);
edtBerat = (EditText) findViewById(R.id.dBerat);
edtEmail = (EditText) findViewById(R.id.dEmail);
edtNoHp = (EditText) findViewById(R.id.dNoHp);
edtKode = (EditText) findViewById(R.id.dKodeLoker);

id_kode = getIntent().getStringExtra("Data-Kode");
edtKode.setText(id_kode);

btnDaftarLoker = (Button) findViewById(R.id.dDaftarLoker);
btnDaftarLoker.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if ((getNama = edtNama.getText().toString()).isEmpty()) {
            Toast.makeText(DaftarLoker.this, "Nama Harus diisi",
                Toast.LENGTH_LONG).show();
            return;
        }
        if ((getTptLahir =
            edtTptLahir.getText().toString()).isEmpty()) {
            Toast.makeText(DaftarLoker.this, "Tempat Lahir Harus
            diisi",
                Toast.LENGTH_LONG).show();
            return;
        }
        if ((getTglLahir =
            edtTglLahir.getText().toString()).isEmpty()) {
            Toast.makeText(DaftarLoker.this, "Tanggal Lahir Harus
            diisi",
                Toast.LENGTH_LONG).show();
            return;
        }
        if ((getAsal = edtAsal.getText().toString()).isEmpty())
        {
            Toast.makeText(DaftarLoker.this, "Asal Sekolah Harus
            diisi",
                Toast.LENGTH_LONG).show();
            return;
        }
        if ((getJurusan =
            edtJurusan.getText().toString()).isEmpty()) {
            Toast.makeText(DaftarLoker.this, "Jurusan Harus
            diisi",
                Toast.LENGTH_LONG).show();
            return;
        }
        if ((getTahun =
            edtTahun.getText().toString()).isEmpty()) {
            Toast.makeText(DaftarLoker.this, "Tahun Lulus Harus
            diisi",
                Toast.LENGTH_LONG).show();
            return;
        }
        if ((getTinggi =
            edtTinggi.getText().toString()).isEmpty()) {
            Toast.makeText(DaftarLoker.this, "Tinggi Badan Harus
            diisi",

```

```

        Toast.LENGTH_LONG).show();
        return;
    }if ((getBerat =
edtBerat.getText().toString()).isEmpty()) {
        Toast.makeText(DaftarLoker.this, "Berat Badan Harus
diisi",
        Toast.LENGTH_LONG).show();
        return;
    }if ((getAlamat =
edtAlamat.getText().toString()).isEmpty()) {
        Toast.makeText(DaftarLoker.this, "Alamat Harus
diisi",
        Toast.LENGTH_LONG).show();
        return;
    }if ((getEmail =
edtEmail.getText().toString()).isEmpty()) {
        Toast.makeText(DaftarLoker.this, "Email Harus diisi",
        Toast.LENGTH_LONG).show();
        return;
    }if ((getNoHp = edtNoHp.getText().toString()).isEmpty())
{
        Toast.makeText(DaftarLoker.this, "Nomer HP Harus
diisi",
        Toast.LENGTH_LONG).show();
        return;
    }
    daftarLoker();
});
}
}

private void daftarLoker() {
    final String nama = edtNama.getText().toString().trim();
    final String tempat = edtTptLahir.getText().toString().trim();
    final String tgl = edtTglLahir.getText().toString().trim();
    final String asal = edtAsal.getText().toString().trim();
    final String jurusan = edtJurusan.getText().toString().trim();
    final String tahun = edtTahun.getText().toString().trim();
    final String tinggi = edtTinggi.getText().toString().trim();
    final String berat = edtBerat.getText().toString().trim();
    final String alamat = edtAlamat.getText().toString().trim();
    final String email = edtEmail.getText().toString().trim();
    final String nohp = edtNoHp.getText().toString().trim();
    final String kode = edtKode.getText().toString().trim();

class AddEmployee extends AsyncTask<Void, Void, String> {
    private ProgressDialog loading;
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        loading = ProgressDialog.show(DaftarLoker.this,
"Menambahkan...", "Tunggu...", false, false);
    }

    @Override
    protected void onPostExecute(String s) {
        super.onPostExecute(s);
        loading.dismiss();
        Toast.makeText(DaftarLoker.this, s,
Toast.LENGTH_LONG).show();
    }

    @Override

```

```

protected String doInBackground(Void... v) {
    HashMap<String, String> params = new HashMap<>();
    params.put(konfigurasi.KEY_EMP_anama, nama);
    params.put(konfigurasi.KEY_EMP_atempat_lahir, tempat);
    params.put(konfigurasi.KEY_EMP_atgl_lahir, tgl);
    params.put(konfigurasi.KEY_EMP_aasal_sekolahdf, asal);
    params.put(konfigurasi.KEY_EMP_ajurusan, jurusan);
    params.put(konfigurasi.KEY_EMP_atahun_lulus, tahun);
    params.put(konfigurasi.KEY_EMP_tinggi_badan, tinggi);
    params.put(konfigurasi.KEY_EMP_berat_badan, berat);
    params.put(konfigurasi.KEY_EMP_aalamat, alamat);
    params.put(konfigurasi.KEY_EMP_aemail, email);
    params.put(konfigurasi.KEY_EMP_anohp, nohp);
    params.put(konfigurasi.KEY_EMP_akode_loker, kode);

    RequestHandler rh = new RequestHandler();
    String res = rh.sendPostRequest(konfigurasi.
        URL_DaftarLoker, params);
    return res;
}

AddEmployee ae = new AddEmployee();
ae.execute();
}

```

## Kode Program Download Pendaftar

```

<!doctype html>
<?php
require_once 'koneksi.php';
?>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<link rel="stylesheet" href="https://cdn.datatables.net/1.10.20/css/jquery.dataTables.min.css"/>
<link rel="stylesheet" href="https://cdn.datatables.net/buttons/1.6.1/css/buttons.dataTables.min.css"/>
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css"
integrity="sha384-GJzZqFGwb1QTTN6wy59ffF1BuGJpLSa9DkMmp0DgiMDm41YMj70gZWKYbI706tWS" crossorigin="anonymous">
<title>BKK SMK N 1</title>
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
<a class="navbar-brand" href="#">BKK SMK Negeri 1 Purwokerto</a>
</nav>
<div class="container-fluid" style="margin-top:60px">
<div class="table-responsive">
<table id="example" class="display nowrap" style="width:100%">
<thead>
<tr>
<th scope="col">Nama</th>
<th scope="col">Tempat Lahir</th>
<th scope="col">Tanggal Lahir</th>
<th scope="col">Asal Sekolah</th>
<th scope="col">Jurusan</th>
<th scope="col">Tahun Lulus</th>
<th scope="col">Tinggi</th>
<th scope="col">Berat</th>
<th scope="col">Alamat</th>
<th scope="col">Email</th>

```



```

        <th scope="col">No HP</th>
        <th scope="col">Kode Loker</th>
    </tr>
</thead>
<tbody>
    <?php
        if(!ISSET($_POST['submit'])){
            $sql = "SELECT * FROM pendaftar";
            $query = mysqli_query($con, $sql);
            $no = 0;
            while ($row = mysqli_fetch_array($query)){
                ?>
            <tr>
                <td><?php echo $row['nama']; ?></td>
                <td><?php echo $row['tempat_lahir']; ?></td>
                <td><?php echo $row['tgl_lahir']; ?></td>
                <td><?php echo $row['asal']; ?></td>
                <td><?php echo $row['jurusan']; ?></td>
                <td><?php echo $row['tahun_lulus']; ?></td>
                <td><?php echo $row['tinggi']; ?></td>
                <td><?php echo $row['berat']; ?></td>
                <td><?php echo $row['alamat']; ?></td>
                <td><?php echo $row['email']; ?></td>
                <td><?php echo $row['nohp']; ?></td>
                <td><?php echo $row['kode_loker']; ?></td>
            </tr>
            <?php
                $no++;
            } } ?>
        </tbody>
    </table>
</div>
</div>

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
    integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous">
</script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.6/dist/umd/popper.min.js"
    integrity="sha384-wHAiFfRm1M4ZbqY1hA6Z0b1glut" crossorigin="anonymous">
</script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/js/bootstrap.min.js"
    integrity="sha384-B0UgLyR+JN6CkvvICOB2joaf5I413gm9GU6Hc1og6Ls7i6U/mkkaduKaBhlAXv9K" crossorigin="anonymous">
</script>
<script>
    $(document).ready(function() {
        $('#example').DataTable( {
            dom: 'Bfrtip',
            buttons: [
                'excel'
            ]
        } );
    });
</script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<script src="https://code.jquery.com/jquery-3.3.1.js"></script>
<script src="https://cdn.datatables.net/1.10.20/js/jquery.dataTables.min.js"></script>
<script src="https://cdn.datatables.net/buttons/1.6.1/js/dataTables.buttons.min.js"></script>
<script src="https://cdn.datatables.net/buttons/1.6.1/js/buttons.flash.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jszip/3.1.3/jszip.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.53/pdfmake.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.53/vfs_fonts.js"></script>
<script src="https://cdn.datatables.net/buttons/1.6.1/js/buttons.html5.min.js"></script>
<script src="https://cdn.datatables.net/buttons/1.6.1/js/buttons.print.min.js"></script>
</body>
</html>

```

## Lampiran 6. Kartu Bimbingan

DOKUMEN KARTU BIMBINGAN SKRIPSI	Nomor Dokumen	PP.05/02/55701/AMIKOMPWT
	Revisi	1
	Halaman	1 dari 1
	Tanggal	18 Mei 2015

**Token : 9049**

**KARTU BIMBINGAN SKRIPSI**

Nama Mahasiswa : FAUZAN DAFFA IBNU YUSUF  
NPM : 16.12.0073  
Program Studi : SISTEM INFORMASI SI  
Judul Skripsi : APLIKASI BURSA KERJA KHUSUS SEBAGAI MEDIA INFORMASI LOWONGAN KERJA DAN PENDAFTARAN BERBASIS MOBILE (STUDI KASUS SMK NEGERI 1 PURWOKERTO)  
Dosen Pembimbing : Mohammad Imron

**Histori Bimbingan Skripsi**

**Bimbingan Ke - : 1**  **A**  
Hari / Tanggal : Jumat / 01 November 2019  
Keterangan : BAB I Revisi, perbaiki penulisan dan referensi difrasekan menurut bahasa penulis

**Bimbingan Ke - : 2**  **A**  
Hari / Tanggal : Senin / 04 November 2019  
Keterangan : BAB I revisi dan BAB II Perbaiki lagi, cari jurnal referensi yang hampir sama dengan penelitiannya

**Bimbingan Ke - : 3**  **A**  
Hari / Tanggal : Kamis / 07 November 2019  
Keterangan : Acc BAB I, dan Revisi BAB II

**Bimbingan Ke - : 4**  **A**  
Hari / Tanggal : Rabu / 13 November 2019  
Keterangan : ACC Bab II, Perbaiki BAB III, pelajari metode yang anda gunakan

**Bimbingan Ke - : 5**  **A**  
Hari / Tanggal : Jumat / 15 November 2019  
Keterangan : Acc BAB II, lanjutkan ke Project

**Bimbingan Ke - : 6**  **A**  
Hari / Tanggal : Kamis / 21 November 2019  
Keterangan : Acc bab III,

**Bimbingan Ke - : 7**  **A**  
Hari / Tanggal : Sabtu / 28 Desember 2019  
Keterangan : Acc Project, dan revisi laporan

**Bimbingan Ke - : 8**  **A**  
Hari / Tanggal : Selasa / 31 Desember 2019  
Keterangan : Acc laporan All, dan siap di ujikan