

BAB III METODE PENGEMBANGAN

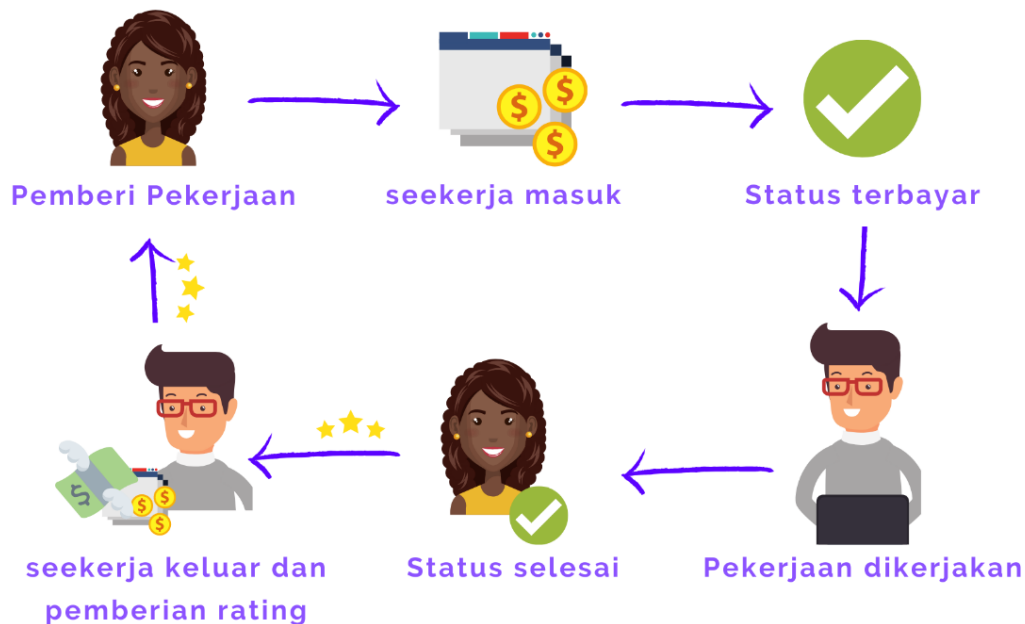
3.1 Fase Perencanaan Kebutuhan (*Requirement Planning Phase*)

Pertemuan antar pengguna dan analis bertemu untuk mengidentifikasi objek dari sistem dan kebutuhan informasi dari objek tersebut. Analisis pada fase ini melibatkan tim **seekerja** itu sendiri yang terdiri atas 3 orang yakni Niky Ayu Lestari bertindak sebagai ketua tim yang bertugas mengambil keputusan terkait kemajuan **seekerja**, mengorganisir kebutuhan dan rencana pengembangan dalam tim, serta memastikan tim mampu berkinerja sesuai tujuan dan waktu yang tersedia selama pengembangan bisnis, sekaligus sebagai *Hacker* yang bertanggung jawab pada pengembangan teknologi mulai dari analisis hingga pada implementasi sistem pada website dan aplikasi **seekerja**. Sementara 2 lainnya yakni Astri Winati bertindak sebagai *Hustler* yang bertanggung jawab terhadap konsep dan strategi pemasaran, serta melakukan analisis pasar guna memastikan bahwa **seekerja** diterima oleh pasar dan Nadia Latifa bertindak sebagai *Hipster* yang bertanggung jawab pada desain antarmuka laman web **seekerja** bersama dengan Niky Ayu Lestari.

Dibutuhkan keterlibatan intens dari kedua pihak tersebut, tidak hanya penandatanganan pada dokumen. Pengguna dari **seekerja** adalah mahasiswa dan ibu-ibu yang telah menikah berusia 24-27 tahun di Yogyakarta. Kedua persona tersebut terbentuk atas hasil survei pasar dan pengguna yang dilakukan tersebar melalui media online ke dan melalui berbagai kalangan di beberapa daerah di Indonesia.

Pada survei tersebut dijelaskan maksud dan tujuan hadirnya **seekerja** itu sendiri, hasilnya kedua kategori pengguna tersebutlah yang mendominasi untuk tertarik pada hadirnya **seekerja**. Oleh karena itu dibentuklah persona atas kedua kategori pengguna tersebut. Selain menunjukkan adanya peluang dan target pasar **seekerja**, hasil survei tersebut pun digunakan sebagai analisis strategi pemasaran serta penentuan kategori pekerjaan yang akan tersedia pada **seekerja**, namun tidak menutup kemungkinan bahwa pengguna **seekerja** terutama sebagai pencari pekerjaan datang dari berbagai kalangan lainnya.

Orientasi dari fase ini adalah untuk memecahkan masalah bisnis, sedangkan fokusnya adalah untuk mencapai tujuan bisnis tersebut. Tahapan ini dilakukan dengan mengidentifikasi tujuan dan kebutuhan informasi dari sistem **seekerja** yang akan dibuat.



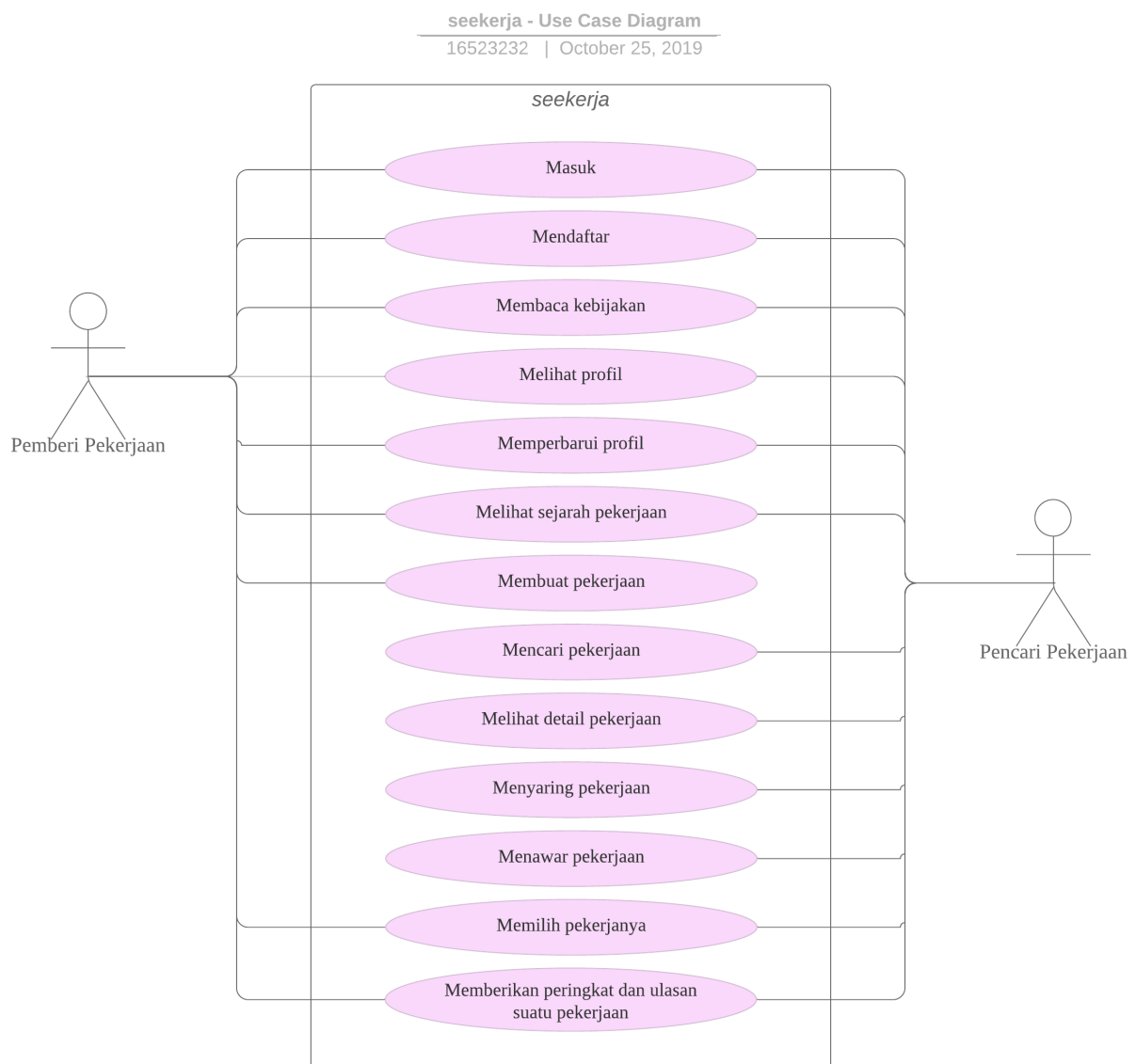
Gambar 3.1 Alur pembayaran sistem Escrow.

Dalam mencapai tujuan bisnis **seekerja**, penting merancang perlindungan kepada para pengguna guna membangun kepercayaan kepada **seekerja**. Perlindungan ini dibangun melalui beberapa aspek, di antaranya penggunaan sistem pembayaran Escrow berupa penggunaan rekening bersama yang disediakan oleh **seekerja** sehingga pemberi pekerjaan diwajibkan untuk membayarkan upah pekerjaan sebelum pekerjaan tersebut dilakukan oleh pencari pekerja, begitupun pencari pekerjaan yang baru bisa menerima upah setelah pekerjaan dinyatakan selesai oleh pemberi pekerjaan.

Sistem pembayaran Escrow pada **seekerja** seperti yang terlihat pada Gambar 3.1 Alur pembayaran sistem Escrow. mengharuskan pemberi pekerjaan untuk mentransfer sejumlah upah terhadap suatu pekerjaan yang telah disepakati bersama pencari pekerjaan terpilih ke rekening bersama yang disediakan oleh **seekerja**. Apabila pembayaran telah terkonfirmasi, maka status dari pekerjaan tersebut akan berubah dari “Pembayaran” yang artinya harus menunggu pembayaran menjadi “Sekarang”. Status pekerjaan “Sekarang” mengartikan bahwa pekerjaan dapat mulai dikerjakan oleh pencari pekerjaan karena telah terjamin pembayaran atas upah pekerjaan tersebut. Apabila pekerjaan telah selesai dilakukan maka pemberi pekerjaan dapat merubah status pekerjaan tersebut menjadi “Selesai” melalui laman web **seekerja**.

Adanya fitur *two-review* berupa pemberian *rating* dan ulasan yang dilakukan oleh kedua pihak yaitu pemberi dan pencari pekerjaan ini dapat dijadikan sebagai salah satu bahan

pertimbangan bagi pemberi pekerjaan dalam memilih penerima pekerjaan nantinya. Begitupun pencari pekerja yang dapat memanfaatkan fitur tersebut untuk meyakinkan diri bahwa pemberi pekerjaan adalah pihak yang dapat dipercaya dan bertanggungjawab atas pekerjaan yang akan diberikan. Pemberian *rating* dan ulasan ini dilakukan setelah pekerjaan dinyatakan selesai oleh pemberi pekerjaan yang kemudian juga dilakukan oleh pencari pekerjaan setelah menerima upah atas pekerjaan tersebut. *Rating* dan ulasan yang telah diberikan oleh kedua pihak tersebut akan ditampilkan pada menu pekerjaan “Selesai”.



Gambar 3.2 Use Case Diagram seekerja.

Pada tahapan ini pun dibuat sebuah *Use Case Diagram* guna menggambarkan kebutuhan dari pengguna tersebut dengan bantuan *tools online* bernama Lucidchart. Seorang pengguna

dapat menjadi dua peran sekaligus dalam satu akun yaitu pemberi dan pencari pekerjaan. Pengguna dalam hal ini adalah orang yang telah melakukan pendaftaran dan memiliki akun pada **seekerja**.

Kebutuhan pengguna pada **seekerja** digambarkan melalui *Use Case Diagram* seperti yang terlihat pada Gambar 3.2 *Use Case Diagram seekerja*, terdiri dari masuk, mendaftar, membaca kebijakan, melihat profil, memperbaharui profil, melihat sejarah pekerjaan yang pernah dilakukan dan dibagikannya, mencari pekerjaan, melihat detail pekerjaan, menyaring pekerjaan sesuai kategori yang diinginkan, menawar atas suatu pekerjaan dengan memberikan deskripsi penawaran dan upah yang diinginkan, memilih pekerjaan, dan yang terakhir adalah memberikan peringkat atau rating beserta ulasan antar kedua pihak atas suatu pekerjaan yang telah dilakukan.

Dari diagram inilah penulis dapat membuat desain berupa gambaran sistem, struktur data, hingga pada perancangan antarmuka guna pengembangan sistem ke depannya.

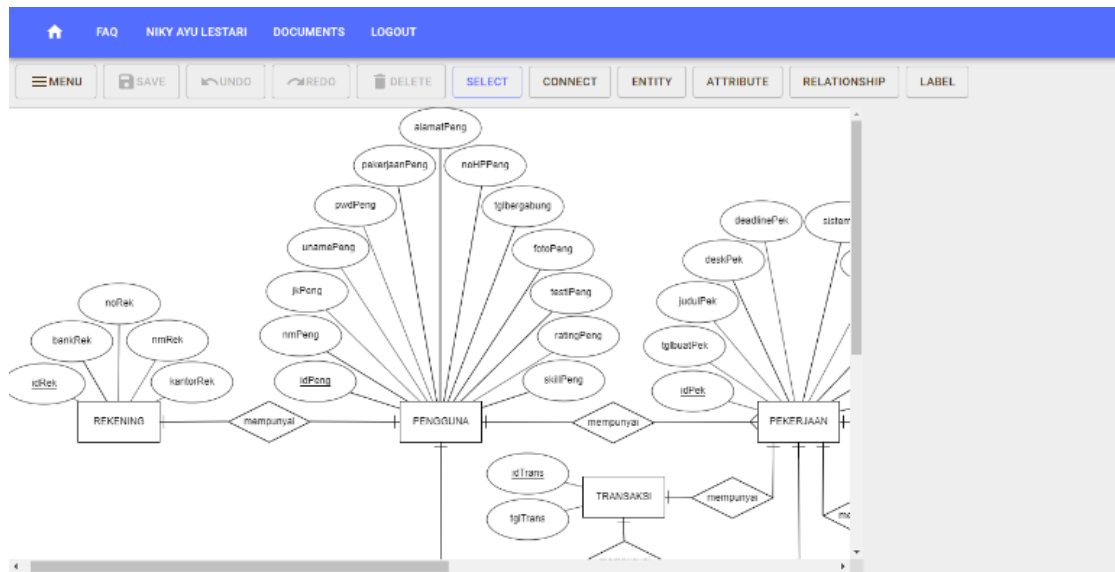
3.2 Workshop Desain RAD (*RAD Design Workshop*)

Pada tahap ini pengguna dapat merespon prototipe dan analisis menyempurnakan modul desain apabila masih terdapat ketidaksesuaian berdasarkan respon tersebut. Hasilnya adalah spesifikasi yang meliputi gambaran sistem secara umum, struktur data, dan lainnya.

3.2.1 Iterasi ke-1

Dari hasil pada tahapan sebelumnya, maka dilakukan konfirmasi kepada *stakeholder* yang dalam hal ini adalah anggota tim untuk merespon hasil dan menyempurnakan apabila masih ada ketidaksesuaian. Hasilnya adalah rancangan pada fase perencanaan kebutuhan diterima sehingga dapat berlanjut pada tahapan ini.

Untuk iterasi pertama pada tahapan ini dibuat gambaran desain basis data yang mampu menampung data dari hasil tahapan sebelumnya. Pada iterasi ini memanfaatkan *tools* guna mempercepat proses pengembangan, di antaranya ERDPlus terkait pembuatan *Entity Relationship Diagram* (ERD), Skema Relasi, sekaligus generasi skrip basis data yang dalam hal ini menggunakan *Structured Query Language* (SQL).



Gambar 3.3 Membuat *Entity Relationship Diagram* (ERD) menggunakan ERDPlus.

Pembuatan *Entity Relationship Diagram* (ERD) pada iterasi ini menghasilkan 7 entitas yang saling terhubung melalui beberapa relasi seperti yang terlihat pada Gambar 3.3, di antaranya entitas REKENING yang memiliki atribut idRek sebagai atribut uniknya, adapun atribut bankRek, noRek, nmRek, dan kantorRek guna menyimpan informasi terkait rekening yang digunakan oleh pengguna.

Entitas PENGGUNA yang terhubung dengan entitas REKENING memiliki atribut idPeng sebagai atribut uniknya, atribut nmPeng, jkPeng, unamePeng, pwdPeng, pekerjaanPeng, alamatPeng, dan noHPPeng guna menyimpan data diri pengguna, atribut tglbergabung guna mengetahui sejak kapan pengguna bergabung dengan **seekerja** untuk digunakan pada kebutuhan mendatang, atribut fotoPeng guna menyimpan file foto dari pengguna, atribut testiPeng guna menyimpan kesan atau testimoni pengguna selama menggunakan **seekerja**, atribut ratingPeng guna menyimpan kalkulasi dari peringkat atau *rating* yang dimiliki pengguna terhadap pekerjaan yang dilakukan ataupun pekerjaan yang diberikan, dan atribut skillPeng guna mengetahui keahlian yang dimiliki oleh pengguna untuk mendapat rekomendasi pekerjaan-pekerjaan yang sesuai menurut algoritma sistem.

Selanjutnya, entitas lain yang terhubung pada entitas PENGGUNA adalah entitas PEKERJAAN dan TAWARAN. Untuk entitas PEKERJAAN memiliki atribut idPek sebagai atribut uniknya, atribut tglbuatPek guna mengetahui kapan pekerjaan tersebut dibuat, atribut judulPek dan deskPek guna menyimpan judul dan penjabaran yang menjelaskan pekerjaan tersebut, atribut deadlinePek guna menyimpan tenggat waktu atas pekerjaan tersebut, atribut

sistemPek guna mengetahui jenis pekerjaan berupa pekerjaan yang dapat dilakukan secara jarak jauh, atukah mengharuskan kehadiran pencari pekerjaan secara langsung dalam penyelesaiannya, atribut upahPek guna menyimpan besaran upah yang akan diberikan atas pekerjaan tersebut, atribut skillPek guna mengetahui keahlian yang menjadi kebutuhan dalam penyelesaian pekerjaan, atribut statusPek guna mengetahui status dari pekerjaan tersebut, atribut tglselesaiPek guna menyimpan waktu diselesaikannya pekerjaan tersebut.

Status pekerjaan pada **seekerja** terbagi menjadi 4 sesi, yakni Penawaran, Menunggu Pembayaran, Pengerjaan, dan Selesai. Penawaran artinya pekerjaan tengah dalam proses ditawarkan atau ditampilkan sebagai pekerjaan yang tersedia dan siap ditawarkan oleh para pencari pekerjaan, belum mendapat pencari pekerjaan yang tepat atau disepakati. Pada proses penawaran ini, pekerjaan masih bisa dibatalkan selama belum terpilih pencari pekerjaannya. Selanjutnya adalah status Menunggu Pembayaran yang artinya pencari pekerjaan atas pekerjaan tersebut telah dipilih, pencari pekerjaan terpilih menunggu dan waktunya pemberi pekerjaan untuk melakukan pembayaran upah atas pekerjaan tersebut ke rekening **seekerja** karena pembayaran pada **seekerja** menggunakan sistem Escrow guna menjamin keamanan kedua pihak.

Setelah pembayaran upah terkonfirmasi oleh **seekerja**, kemudian status akan berubah menjadi status selanjutnya yakni Pengerjaan, pada status inilah pencari pekerjaan terpilih dipersilahkan untuk mulai mengerjakan pekerjaan yang disepakati sesuai dengan tenggat waktu yang disepakati. Apabila pekerjaan telah dianggap selesai oleh pemberi pekerjaan, maka pemberi pekerjaan dipersilahkan untuk memberikan *rating* beserta ulasan terhadap kinerja atau hasil pekerjaan dari pencari pekerjaan tersebut yang kemudian pekerjaan akan dianggap selesai oleh sistem **seekerja**.

Adapun entitas TAWARAN yang memiliki atribut idTaw sebagai atribut uniknya, atribut tglTaw guna mengetahui kapan tawaran tersebut dibuat, atribut deskTaw guna memberikan penjabaran terkait tawaran tersebut sekaligus sebagai media mempromosikan diri kepada pemberi pekerjaan, atribut wktTaw guna mengetahui berapa lama waktu yang ditawarkan pencari pekerjaan untuk menyelesaikan pekerjaan tersebut, atribut upahTaw tentu saja guna mengetahui berapa upah yang diinginkan oleh pencari pekerjaan sesuai dengan tenggat waktu atau promosi diri yang disampaikan, biasanya semakin sedikit waktu dalam menyelesaikan suatu pekerjaan maka semakin tinggi pula upah yang ditawarkan oleh pencari pekerjaan.

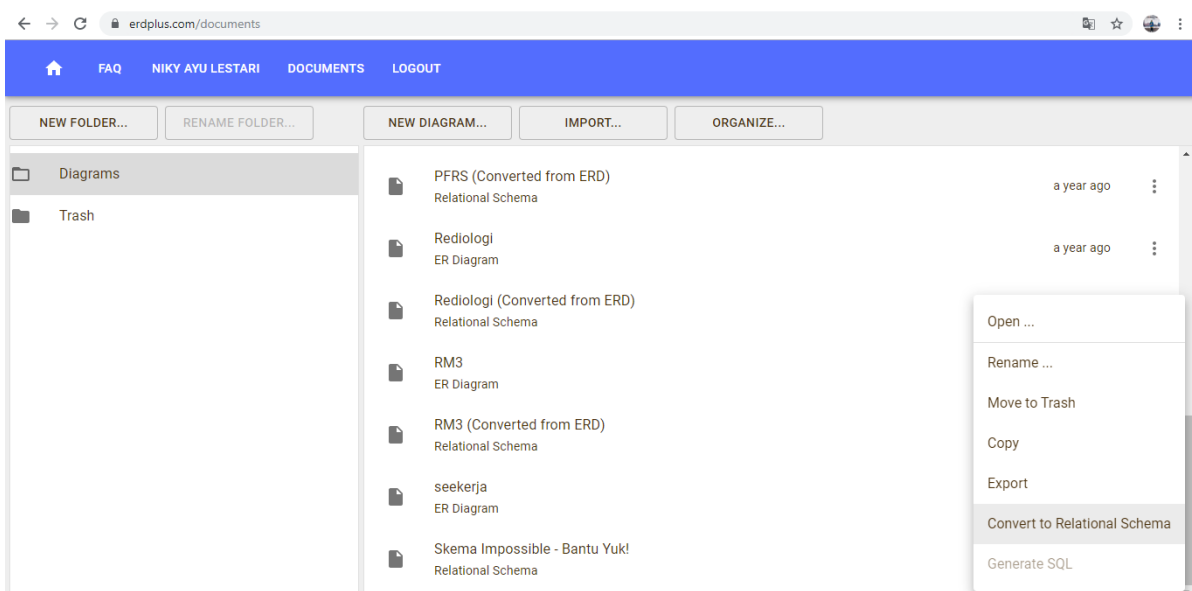
Entitas TRANSAKSI memiliki atribut idTrans sebagai atribut uniknya dan atribut tglTrans guna menyimpan tanggal terjadinya transaksi. Transaksi dalam hal ini adalah

terjadinya kesepakatan antar pemberi pekerjaan untuk memilih pencari pekerjaan atas pekerjaan tersebut.

Entitas REVIEW memiliki atribut idRev sebagai atribut uniknya, atribut bintangRevBer guna menyimpan *rating* dari pemberi pekerjaan, atribut deskRevBer guna menyimpan ulasan yang diberikan oleh pencari pekerjaan kepada pemberi pekerjaan terkait suatu pekerjaan tertentu, atribut bintangRevCar guna menyimpan *rating* dari pencari pekerjaan, atribut deskRevCar guna menyimpan ulasan yang diberikan oleh pemberi pekerjaan kepada pencari pekerjaan terkait suatu pekerjaan tertentu, dan yang terakhir yakni entitas KATEGORI yang memiliki atribut idKat sebagai atribut unik, atribut nmKat guna menyimpan nama dari masing-masing kategori, dan atribut logoKat guna menyimpan nama file dari logo yang digunakan.

Setelah membuat *Entity Relationship Diagram* (ERD), selanjutnya diagram tersebut dikonversikan menjadi skema relasi guna mengetahui pemodelan dari basis data beserta kunci-kuncinya yang meliputi *primary key* dan atribut tamu yang menjadi *foreign key* sebagai hasil dari hubungan antar entitas pada ERD sebelumnya.

Oleh karena pengembangan ini memanfaatkan salah satu *tools online* yang dapat dimanfaatkan dalam proses pengembangan dengan metode RAD bernama ERDPlus, maka pembuatan skema relasi pun dapat dengan mudah dilakukan yaitu dengan mengkonversi ERD menjadi skema relasi secara langsung melalui menu pada ERDPlus.



Gambar 3.4 Konversi ERD ke Skema Relasi melalui ERDPlus.

Hasil dari konversi ERD ke skema relasi melalui ERDPlus seperti yang terlihat pada Gambar 3.4 , mampu mempersingkat proses pengembangan sehingga dapat menghasilkan skema relasi dari ERD sebelumnya tanpa harus membuatnya dari halaman kosong atau menggunakan *tools* yang berbeda lagi. Hasil konversi pada ERD **seekerja** tersebut menghasilkan 7 tabel sesuai dengan entitas pada ERD, yakni tabel TRANSAKSI dengan idTrans sebagai *primary key*, tabel TAWARAN dengan idTaw sebagai *primary key*, tabel PENGGUNA dengan idPeng sebagai *primary key*, tabel REKENING dengan idRek sebagai *primary key*, tabel PEKERJAAN dengan idPek sebagai *primary key*, tabel REVIEW dengan idRev sebagai *primary key*, dan tabel KATEGORI dengan idKat sebagai *primary key*.

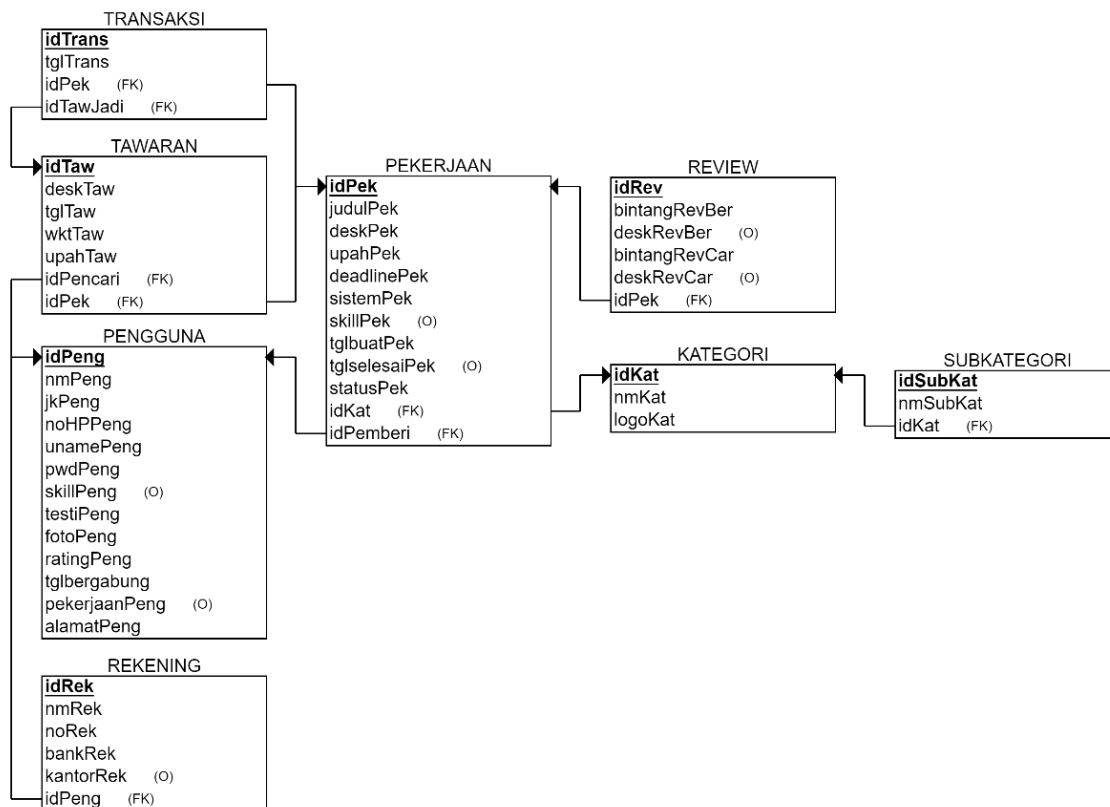
Tabel TRANSAKSI memiliki 2 *foreign key*, di antaranya idPek yang merupakan atribut tamu dari hasil relasi dengan tabel PEKERJAAN dan idTaw yang selanjutnya diubah menjadi idTawJadi untuk mendeskripsikan bahwa idTaw tersebut merupakan idTaw untuk tawaran yang telah ditentukan oleh pemberi pekerjaan sebagai tawaran yang diterima dan memutuskan pencari pekerjaannya.

Tabel TAWARAN pun memiliki 2 *foreign key* yang berasal dari relasi dengan tabel PENGGUNA yakni idPeng yang kemudian diubah menjadi idPencari guna mendeskripsikan bahwa pengguna tersebut adalah pencari pekerjaan yang melakukan tawaran tersebut dan idPek untuk mengetahui tawaran berasal dari pekerjaan yang mana.

Tabel REKENING pun memiliki *foreign key* yang berasal dari relasi dengan tabel PENGGUNA yakni idPeng guna mengetahui pemilik dari rekening yang dimaksud serta memiliki satu kolom *optional* yang akan menjadi kolom beratribut *NULL* pada basis data yang artinya boleh tidak diisi atau dikosongkan. Sementara untuk tabel PENGGUNA tidak memiliki *foreign key* namun memiliki 2 kolom *optional* yakni skillPeng dan pekerjaanPeng.

Tabel PEKERJAAN memiliki 2 *foreign key* yang berasal dari relasi dengan tabel PENGGUNA yakni idPeng yang kemudian diubah menjadi idPemberi guna mengetahui pengguna yang memberikan pekerjaan tersebut dan idKat yang merupakan hasil relasi dengan tabel KATEGORI guna mengetahui kategori dari pekerjaan tersebut. Tabel ini pun memiliki 2 kolom *optional* yakni skillPek guna mengetahui keahlian yang dibutuhkan pada pekerjaan tersebut dan tglselesaiPek berupa tanggal yang akan terisi otomatis setelah pekerjaan selesai.

Tabel REVIEW pun memiliki 2 kolom *optional* yakni deskRevBer dan deskRevCar yang menyimpan ulasan untuk pemberi dan pencari pekerjaan atas suatu pekerjaan tertentu, serta memiliki *foreign key* yang berasal dari relasi dengan tabel PEKERJAAN yakni idPek. Sementara untuk tabel KATEGORI tidak memiliki *foreign key* ataupun kolom *optional*.



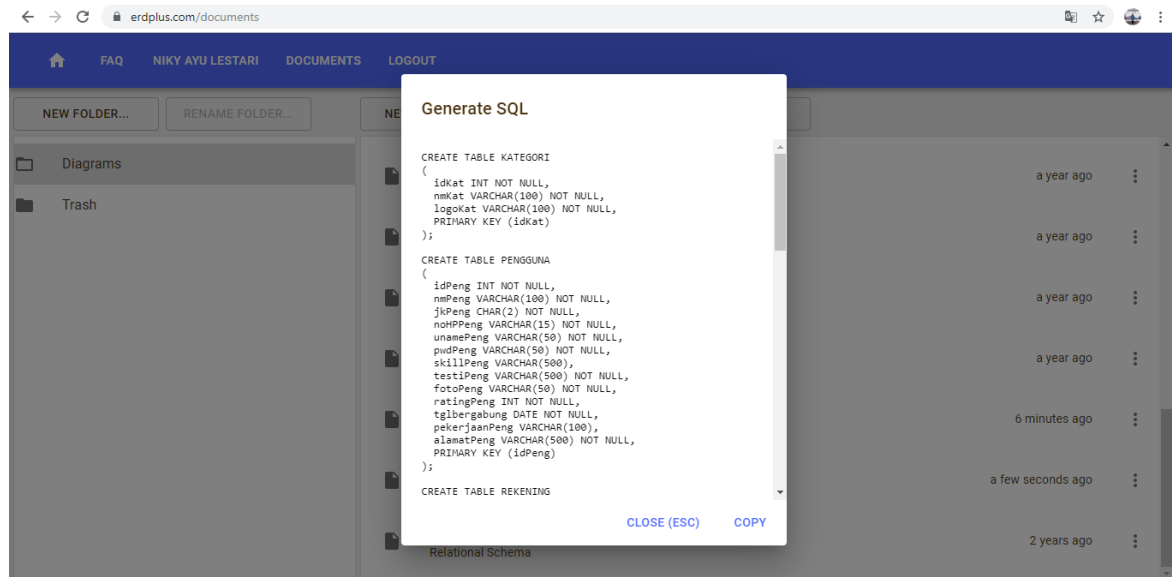
Gambar 3.6 Skema Relasi seekerja.

Hasil skema relasi seperti yang terlihat pada Gambar 3.6 disempurnakan yakni dengan mendeklarasikan tipe data beserta ukurannya. Oleh karena menggunakan *tools online* ERDPlus, skema relasi dapat secara otomatis dilakukan generasi menjadi skrip SQL seperti yang terlihat pada Gambar 3.7 .



Gambar 3.7 Generasi skema relasi ke skrip SQL.

Didapatkan sebanyak 102 baris skrip seperti yang terlihat pada Gambar 3.8 yang menghasilkan 8 tabel berdasarkan hasil dari konversi ERD ke skema relasi yang selanjutnya siap diterapkan pada PHPMyAdmin pada tahapan selanjutnya yaitu fase implementasi.

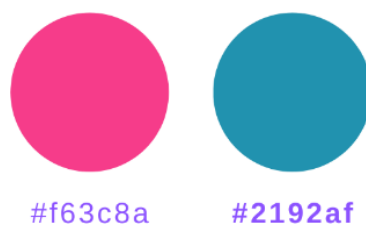


Gambar 3.8 Hasil generasi skema relasi ke skrip SQL.

Pada iterasi ini juga dilakukan mulai dibuat gambaran sistem berupa tampilan antarmuka (*User interface*) beserta aktifitasnya dengan bantuan *tools online* bernama MockingBot.

3.2.3 Iterasi ke-3

Iterasi ketiga dilakukan beberapa penyempurnaan pada antarmuka beserta aktifitasnya. Iterasi ini pun menghasilkan 2 warna dominan seperti yang terlihat pada Gambar 3.9 yang akan menjadi identitas **seekerja**, selain itu terjadi perubahan nama branding yang awalnya kerjain.com menjadi **seekerja.id** dikarenakan ketersediaan domain dan pertimbangan promosi.



Gambar 3.9 Warna dominan seekerja.

Iterasi dilakukan sebanyak tiga kali yang kemudian dilanjutkan pada fase implementasi terhadap hasil akhir dari tahapan workshop desain RAD ini.

3.3 Fase Implementasi (*Implementation Phase*)

Pada fase sebelumnya, pengguna dan analis telah bekerja intens selama workshop desain bisnis dan aspek non-teknik sistem. Begitu aspek ini disepakati dan sistem disempurnakan, selanjutnya sistem baru atau bagian dari sistem yang disempurnakan tadi diperkenalkan kepada tim pengembang untuk diimplementasikan menjadi sebuah sistem berbasis web. Keterlibatan pengguna sangat dibutuhkan untuk memastikan kepuasan pengguna terhadap sistem yang dikembangkan.



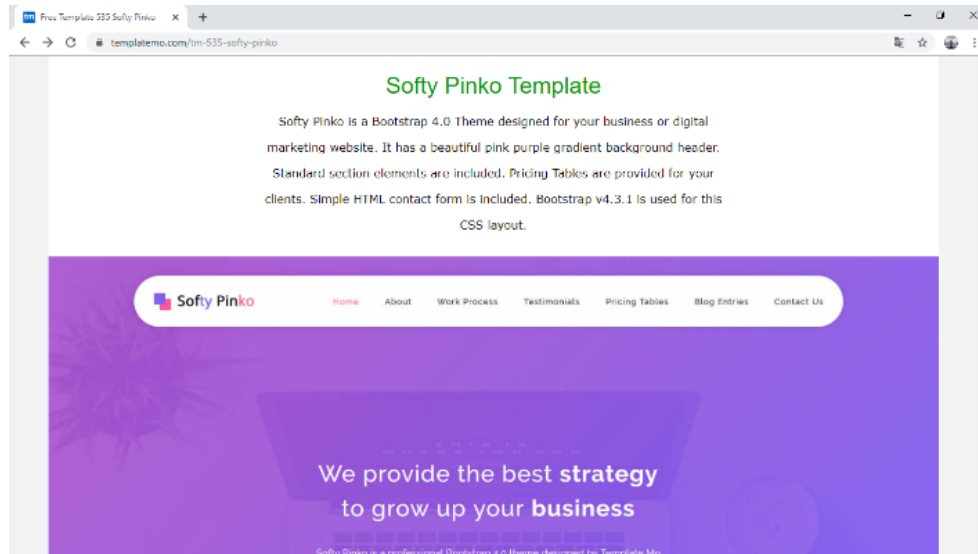
Gambar 3.10 *Tools front-end.*

Pada tahapan ini dilakukan pengembangan *front-end* memanfaatkan skrip *open source* dari Softy Pinko Company (Desain Templatemo). Pada pengembangan *front-end* pun memanfaatkan beberapa *tools* grafis lainnya, di antaranya Paint, Canva, dan Flaticon.



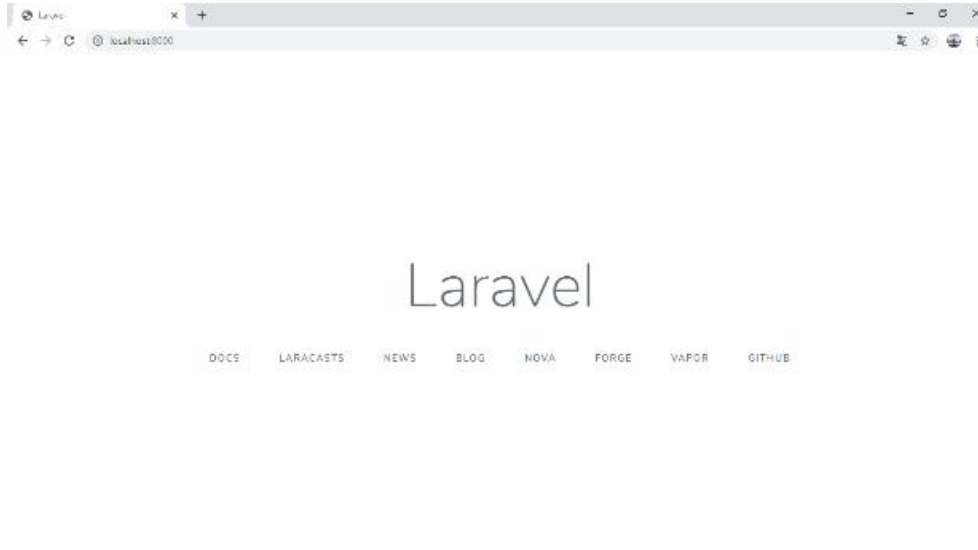
Gambar 3.11 *Tools back-end.*

Selanjutnya dilakukan pengembangan *back-end* dengan menyesuaikan hasil dari ketiga iterasi sebelumnya. Pengembangan pada tahap ini dilakukan berbasis web dengan framework Laravel menggunakan *tools* bernama Sublime Text sebagai *editor* skrip dan Google Chrome sebagai *browser*.



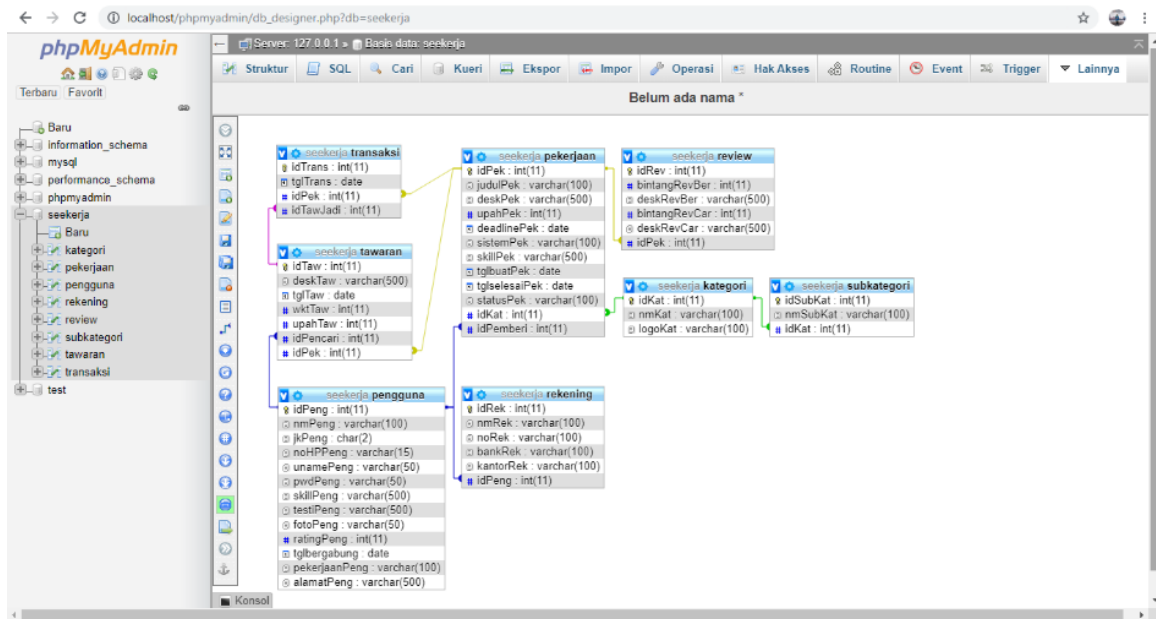
Gambar 3.12 Memanfaatkan skrip *open source* Softy Pinko Company.

Implementasi skrip *open source* pada pengembangan *front-end* seperti yang terlihat pada Gambar 3.12 dilakukan juga dengan bantuan *tools* Xampp dan Sublime Text untuk melihat penerapannya pada framework Laravel.



Gambar 3.13 Tampilan web sebelum diterapkan skrip *open source* Softy Pinko Company.

Setelah pengembangan *front-end* dianggap telah memenuhi kebutuhan **seekerja**, maka pengembangan dilanjutkan pada *back-end* yang dimulai dengan menerapkan skrip SQL hasil generasi pada tahapan sebelumnya ke dalam PHPMYAdmin untuk membuat basis data **seekerja** seperti yang terlihat pada Gambar 3.14.



Gambar 3.14 Basis data seekerja.

Setelah itu dilanjutkan pada pengembangan *back-end* sesuai dengan *Use Case Diagram* yang telah dibuat untuk memastikan bahwa semua kebutuhan pada **seekerja** terpenuhi serta mengacu pada desain antarmuka yang telah dirancang sedemikian rupa.

```

14 public function create(Request $request){
15     $kategori = new Kategori();
16     $kategori->nmKat = $request->input('nmKat');
17     if($request->hasFile('logokat')){
18         $file = $request->file('logokat');
19         $extension = $file->getClientOriginalExtension();
20         $filename = time() . '.' . $extension;
21         $file->move('beranda/assets/images/kategori/', $filename);
22         $kategori->logokat = $filename;
23     }
24     return $request;
25     $kategori->logokat;
26 }
27 $kategori->save();
28 return redirect('/admin/kategori/')->with('sukses','Kategori baru berhasil dibuat!');
29 /*App\Kategori::create($request->all());
30 $kategori = \App\Kategori::find($idkat);
31 if($request->hasFile('logokat')){
32     $request->file('logokat')->move('beranda/assets/images', $request->file('logokat')->getClientOriginalName());
33     $kategori->logokat = $request->file('logokat')->getClientOriginalName();
34     $kategori->save();
35 }
36 return redirect('/admin/kategori/')->with('sukses','Kategori baru berhasil dibuat!');*/
37 }
38
39 public function update(Request $request,$idkat){
40     $kategori = Kategori::find($idkat);
41     $kategori->update([
42         'nmKat' => $request->nmKat
43     ]);
44     if($request->hasFile('logokat')){
45         $request->file('logokat')->move('beranda/assets/images/kategori', $request->file('logokat')->getClientOriginalName());
46         $kategori->update([
47             'logokat' => $request->file('logokat')->getClientOriginalName()
48         ]);
49     }
50     return redirect('/admin/kategori/')->with('sukses','Kategori berhasil diubah.');
```

Gambar 3.15 Pengembangan *back-end*.