

APLIKASI TRACER STUDY BERBASIS WEB PADA UNIVERSITAS DIPONEGORO

Yosua Alvin Adi Soetrisno^{*,1)}, Abdul Syakur²⁾, Mujid F. Amin³⁾, Maman Somantri⁴⁾, Denis⁵⁾

^{1,2,4,5)} Departemen Teknik Elektro, Fakultas Teknik, Universitas Diponegoro
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

³⁾ Departemen Bahasa & Sastra, Universitas Diponegoro
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*,1)}E-mail: yosua@live.undip.ac.id

E-mail: ²⁾syakur@elektro.undip.ac.id, ³⁾moejid70@gmail.com, ⁴⁾mmsomantri@live.undip.ac.id,

⁵⁾denisginting@elektro.undip.ac.id

Abstrak

Penelitian ini bertujuan untuk melakukan integrasi sistem informasi alumni yang selama ini masih dijalankan secara terpisah oleh masing-masing pihak akademik departemen di Universitas Diponegoro. Sistem informasi alumni sebelumnya mengalami kendala yaitu tidak adanya standarisasi pengisian *form* alumni, sehingga tersedia bermacam-macam versi. Aplikasi yang dirancang pada penelitian ini dapat menjadi media yang bisa menghubungkan antar alumni di setiap fakultas dan juga meregulasi untuk setiap alumni dapat mengisi data di aplikasi ini. Metode pengembangan sistem yang dipakai dalam mengembangkan sistem ini adalah model V. Informasi yang dimasukkan ke aplikasi dapat mempermudah pengumpulan data primer untuk keperluan akreditasi. Pengembangan aplikasi ini berkontribusi juga pada modifikasi dari *library Data Table* yang sebelumnya tidak bisa menampilkan data secara fleksibel dalam pengaksesan dengan bentuk *server side*. Kebaruan dari sisi basis data adalah pembuatan *trigger* otomatis yang menghitung jumlah rerata gaji pertama dan waktu dalam mendapatkan pekerjaan pertama. Pada aplikasi ini dilakukan pengujian *alpha* menggunakan metode *blackbox testing* dan dalam rangka penambahan fungsi dan fitur menggunakan metode penyebaran kuesioner. Dari lulusan tahun 2017 ada 4521 alumni yang telah mengisi data dengan baik, dan ada peningkatan di tahun 2018 sebanyak 6487 yang menandakan bahwa aplikasi ini dapat menjadi sarana yang memadai untuk memenuhi kebutuhan data alumni yang terintegrasi.

Kata kunci: sistem informasi, alumni, model V, data table, trigger, blackbox testing

Abstract

This research aims to integrate alumni information systems which are still being run separately by each academic department at Diponegoro University. Previous alumni information systems had experienced problems, which is the absence of standardization of alumni form filling, and there is many form of application. Applications designed in this research can become a media that connect alumni in each faculty and also regulate for each alumni to fill in the data in this application. The system development method used in developing this system is model V. Information entered into the application can facilitate the collection of primary data for accreditation purposes. The development of this application also contributed to the modification of the Data Table library which previously could not display data flexibly in server side access mode. Novelty from the database side is the creation of an automatic trigger that calculates the average amount of first salary and time in getting the first job. In this application alpha testing is done using the blackbox testing method and in order to add functions and features using the questionnaire distribution method. From 2017 graduates there are 4521 alumni who have filled in the data well, and there is an increase in 2018 of 6487 indicating that this application can be an adequate means to meet the needs of integrated alumni data.

Keyword: information system, alumni, V model, data table, trigger, blackbox testing

1. Pendahuluan

Salah satu faktor yang menjadi penilaian performa perguruan tinggi adalah faktor alumni. Alumni menjadi penting sebagai indikator apakah ilmu dan keterampilan yang telah diterima semasa kuliah dapat diaplikasikan dengan baik dan juga dapat membantu lulusan dapat

terserap di bursa kerja maupun berwirusaha sendiri. Data masa tunggu, kesesuaian pekerjaan, dan gaji pertama alumni juga menjadi faktor penilaian dalam borang akreditasi standar 3 pada setiap fakultas dan departemen yang ada pada suatu Universitas.

Kontribusi setiap alumni akan sangat penting dalam membuat profil data yang baik. Keterikatan alumni dan juga fungsionalitas dari aplikasi *tracer study* perlu diteliti lebih lanjut, agar setiap alumni bisa berkontribusi dan merasa terbantu dengan adanya aplikasi tersebut. Aplikasi tersebut dapat berfungsi dengan baik ketika ada kesatuan visi dan komitmen bersama untuk membentuk profil data dengan baik, karena memang keperluan masing-masing fakultas akan sangat beragam. Aplikasi *tracer study* ini disiapkan untuk menjadi satu wadah yang bisa menjawab keperluan dari masing-masing fakultas dan departemen yang ada. Kegiatan untuk menghimpun informasi alumni dikenal dengan *tracer study*. *Tracer study* biasanya dilakukan setiap akreditasi akan berlangsung dengan data 1-4 tahun ke belakang. *Tracer study* juga dilaksanakan tiap tahun untuk melihat perkembangan lulusan tahun sebelumnya. Bentuk umum dari *tracer study* adalah menanyakan kapan diterima bekerja setelah lulus, apakah setelah lulus langsung bekerja [1], apakah pekerjaan sesuai dengan bidang studi yang diambil dan bagaimanakah nilai kepuasan pengguna dari sisi *stakeholder* terhadap alumni [2].

Universitas Diponegoro sampai saat ini memiliki 13 fakultas dan 112 program studi, mulai dari pendidikan D3 sampai S3. Berdasarkan data Biro Administrasi Akademik pada tahun 2018 ada 9922 lulusan, sedangkan yang terdaftar pada aplikasi telah mencapai 6487 alumni. Hal ini menunjukkan bahwa ada 65 % alumni yang telah mendaftar. Kendala yang dihadapi dalam melakukan pengisian data alumni adalah adanya beberapa sistem, sehingga perlu beberapa kali melakukan pengisian. Dengan adanya aplikasi ini diharapkan data yang dimasukkan ketika mengisi data wisuda, dan juga pengisian data di masing-masing fakultas bisa terintegrasi di aplikasi ini.

Aplikasi *tracer study* yang telah dirancang terdiri dari dua versi yaitu versi *web* dan versi *mobile* untuk sistem operasi *Android*. Aplikasi versi *web* dirancang dengan tema responsif dan juga menggunakan interaksi yang disesuaikan dengan kebutuhan penggunanya. Aplikasi yang dirancang dengan tema responsif dapat menyesuaikan komponennya dengan ukuran layar yang tersedia [3].

Aplikasi pencarian data alumni dirancang dengan menggunakan *library Data Table*. *Data Table* merupakan *library* yang menyediakan fitur *searching* dan *pagination* yang bisa dikustomisasi. Fitur *searching* ini penting untuk mencari alumni dengan kriteria tertentu [4]. Fitur *pagination* penting untuk membatasi *request* ke *server* dalam satu halaman, ketika ada ribuan data yang harus ditampilkan [5].

Data Table pada sistem ini dirancang dengan menggunakan metode *server side scripting* sehingga bisa menampilkan data yang cukup besar dengan total sampai

periode ini ada 132.545 alumni. Teknik *server side scripting* perlu dilakukan untuk mengalihkan semua permintaan ke *SQL engine* yang ada di *server*. *SQL engine* dapat menangani basis data yang besar karena memang dirancang dengan tujuan seperti itu [6]. *Request* dari setiap halaman akan di-*refresh* dengan menggunakan *AJAX (Asynchronous Java xScript)*, setiap kali halaman berpindah [7].

Aplikasi ini dapat diakses dari manapun jika tersedia koneksi internet. Aplikasi *web* ini juga dijadikan dasar dalam membuat aplikasi berbasis *Android* karena semua *request* di *Android* akan diarahkan *request*-nya oleh *web services* [7] berbasis *PHP (PHP Hypertext Preprocessor)* yang juga telah dirancang dalam aplikasi ini.

Tujuan komprehensif dari pengembangan aplikasi *tracer study* adalah untuk memudahkan alumni dalam memberikan kontribusi tanpa harus terkendala mengisi data dari jaringan internal Universitas. Tujuan lain dari pengembangan *tracer study* adalah juga untuk membentuk komponen data pada borang akreditasi. Keberadaan aplikasi *tracer study* yang terpusat dan bisa dimanfaatkan oleh masing-masing departemen akan sangat membantu dalam membuat profil masing-masing departemen. Dalam aplikasi ini akan dirancang juga beberapa tingkat hak akses sesuai dengan fungsi *filtering* informasi yang dibutuhkan untuk masing-masing departemen karena data bersifat rahasia dan tidak bisa digunakan oleh pihak lain secara sembarangan.

Beberapa penelitian dilakukan untuk mengembangkan sistem informasi alumni dengan tujuan untuk meningkatkan pelacakan alumni dan untuk menghimpun semua data alumni di satu pusat data. Penelitian tentang sistem alumni sebelumnya dilakukan di Universitas Jenderal Soedirman Purwokerto [2]. Penelitian tersebut menggunakan metode *waterfall* untuk mengembangkan *software*-nya.

Hasil dari penelitian tersebut adalah bahwa dengan *blackbox testing*, fitur-fitur yang ada seperti grafik survey dan pemetaan alumni telah berjalan dengan baik. Selain *blackbox testing* ada metode kuesioner *User Acceptance Test* untuk menguji apakah aplikasi sudah bisa diterima dan dipakai oleh alumni dengan tingkat keberhasilan 85.14%. Fitur – fitur yang menjadi masukan dan bisa dikembangkan dalam sistem *database* Universitas Diponegoro ini adalah adanya fitur diskusi dan ada fitur kuesioner untuk mengukur kualitas lulusan.

Pada penelitian di Institut Teknologi Padang [8] sistem dikembangkan karena sebelumnya penanganan data masih bersifat manual. Pada penelitian di Sekolah Tinggi Teknologi Garut dan Universitas Bengkulu [9] dipakai metode pengembangan *Rapid Application Development (RAD)* dengan pengujian *black box testing* [10].

Pengembangan yang dilakukan dengan metode RAD ini menggunakan teknik *prototyping* dari *Entity Relationship Diagram (ERD)* dan *Data Flow Diagram (DFD)* yang telah dirancang sebelumnya, kemudian diciptakan langsung menjadi *source code* dengan menggunakan *tools*. Teknologi yang digunakan bukan menggunakan *web programming* seperti *PHP* dan *database* seperti *MySQL*. Fitur yang berbeda pada aplikasi ini adalah penelusuran alumni, profil alumni yang bisa dibuat menjadi *Curriculum Vitae*, serta mengakomodasi kuesioner untuk borang akreditasi. Fitur yang menjadi masukan pada penelitian Ridwan [10] adalah perlu ditambahkan sistem informasi geografis yang menunjukkan persebaran alumni. Fitur tersebut dikembangkan pada penelitian ini karena bisa digunakan untuk mencari alumni pada kota terdekat.

Pada penelitian Gani [11] dibuat sistem informasi alumni menggunakan *framework Code Igniter (CI)*. *Framework* digunakan karena memiliki *security layer* di mana bisa dilakukan *filtering data*, untuk mencegah eksploitasi celah keamanan. Eksploitasi yang bisa dicegah dengan *framework* adalah *Cross Site Scripting* dan *SQL injection*. Pengujian sistem yang menggunakan *framework CI* ini menggunakan *tools* seperti *web threat analysis*, dapat menghasilkan peringatan jenis rendah (*Low Level*) [11].

Kualitas perangkat lunak pada penelitian Gani diuji dengan *usability testing* yang menyatakan bahwa 79% sistem bisa digunakan dengan mudah. Pada penelitian Gani tidak dibahas mengenai penggunaan *Data Table* untuk menampilkan data tiap halaman dan mengurangi beban *request data*, namun dibahas bahwa penggunaan kompresi *GNU-Zip (GZIP)* dapat mengurangi beban *Hyper-Text Transport Protocol (HTTP) request*, ketika memuat komponen-komponen tampilan dan *library* yang digunakan pada sistem. Pada penelitian sistem yang dikembangkan di Universitas Diponegoro ini tidak menggunakan kompresi *GZIP*, karena ada beberapa *library* yang harus di-*share* di antara objek-objek pada program.

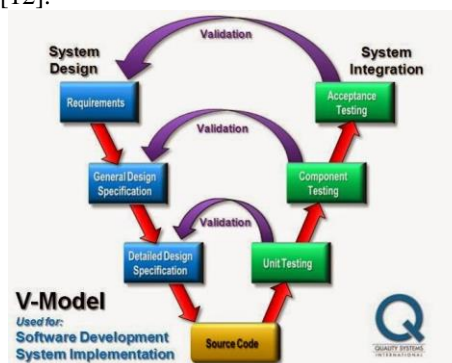
Kontribusi pada penelitian ini adalah untuk mengembangkan fitur-fitur yang belum diakomodasi pada penelitian Lutfi [2] dan Ridwan [10]. Fitur tersebut adalah fitur diskusi, fitur kuesioner, dan juga fitur pencarian alumni berdasarkan peta geografis. Kontribusi lain adalah menggunakan fitur *Data Table* untuk memperbaiki proses pengaksesan data pada tiap halaman, untuk data yang cukup besar, yang diproses pada sisi server. Penelitian ini tidak menggunakan *framework* seperti pada penelitian Gani [11], karena metode pengembangannya akan menjadi lebih leluasa dalam mengakses metode yang disediakan daripada menggunakan *framework*.

2. Metode

Metode pengembangan yang digunakan untuk membangun sistem ini adalah model V. Model V merupakan turunan dari model *waterfall* [12]. Tahap yang ada pada pengembangan model V hampir mirip dengan model yang ada pada model *waterfall*. Pada model *waterfall* proses dijalankan dari awal sampai akhir, sedangkan pada model V proses akan dilakukan secara berulang setiap kali ada kebutuhan baru.

Setelah fase implementasi dilakukan maka langkah selanjutnya adalah kembali ke fase perancangan untuk melakukan penambahan fitur. Proses pengembangan disesuaikan dan bergantung pada proses verifikasi pada langkah sebelumnya, sebelum maju ke langkah selanjutnya. Gambar 1 akan menjelaskan proses validasi yang dilakukan dalam model V.

Produk yang dihasilkan pada setiap fase harus divalidasi sebelum maju ke langkah selanjutnya. Pada model V, *developer* dan *tester* bekerja secara paralel. Pada model V ada dua pengujian sistem yang dilakukan berdasarkan *High Level Document (HLD)* dan *Low Level Document (LLD)* [12].



Gambar 1. Model V

Setiap implementasi *coding* selesai maka dilakukan pengujian unit, pengujian integrasi, dan pengujian sistem. Pada model V ada relasi antara tahap *development* dan tahap *testing* [12]. Adaptasi model V pada penelitian ini adalah penggunaan validasi untuk setiap fase yang menghubungkan antara perancangan sistem dan integrasi sistem. Setiap ada satu modul baru yang diintegrasikan maka akan langsung divalidasi dengan perancangan yang telah ada sebelumnya.

Pada pengembangan tahap awal dibangun fitur-fitur dasar yang bisa dimanfaatkan oleh alumni. Fitur tersebut adalah untuk mendaftar sebagai alumni dan melakukan *update data* alumni, ketika alumni sudah mendapatkan pekerjaan (dalam hal ini melakukan perubahan masa tunggu dan gaji pertama yang diterima).

Fitur awal yang dikembangkan antara lain adalah untuk *filtering* data alumni berdasarkan kriteria pencarian tertentu. Kriteria tersebut misalnya mencari data alumni dari Fakultas Teknik pada Departemen Teknik Elektro untuk lulusan angkatan 2018.

Fitur lain yang dikembangkan adalah sistem komunikasi yang memungkinkan alumni melakukan *posting* informasi. Informasi yang bisa dibagi adalah lowongan pekerjaan maupun kerja sama dengan alumni. Fitur komentar disediakan untuk menanggapi informasi tersebut layaknya fitur *wall* pada media sosial.

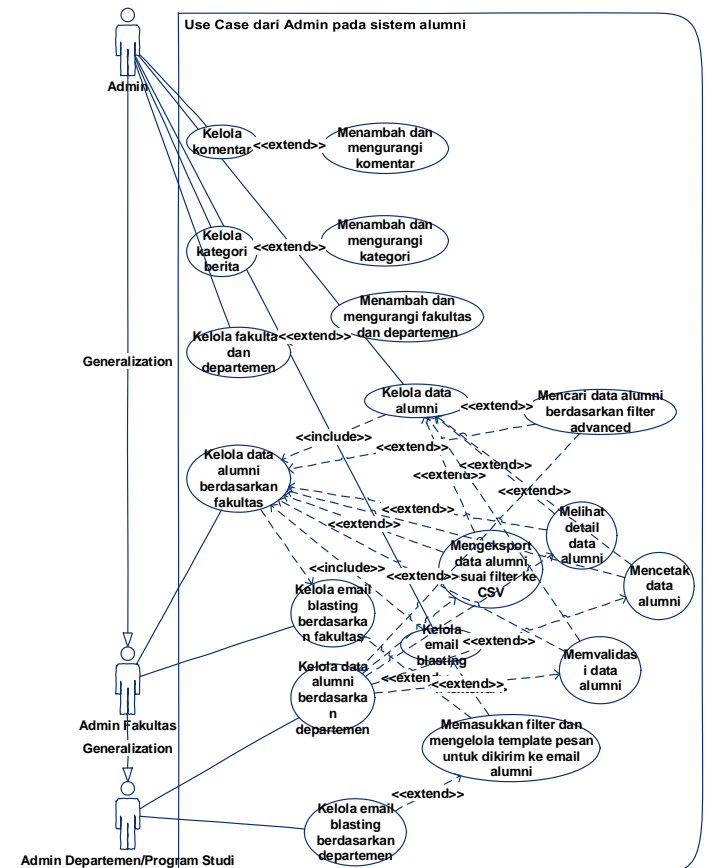
Fitur yang disediakan langsung dicoba oleh pengguna untuk mendapatkan masukan perbaikan dari segi fitur dan interaksi. Pada tahap pertama *launching* aplikasi, ada masukan untuk bisa melakukan *filtering* data terhadap semua *field* yang diisi pada form dan juga supaya disediakan pemisahan hak akses dalam tingkat departemen dan fakultas.

Fitur tersebut kemudian dikembangkan dan seterusnya dicoba kembali oleh pengguna guna memperoleh respon dalam pengembangan dari segi yang lain. Fitur baru yang disarankan adalah adanya fitur *export* data dan juga penyempurnaan fungsi *register* dan *update* data. Setiap sistem telah selesai diperbaiki akan langsung dicoba oleh pengguna, demi mendapatkan respon yang lebih luas, dalam segi fungsionalitas program [13]. Hal inilah yang menjadi dasar modifikasi dari model V awal yang terdapat pada Gambar 1.

Aplikasi ini dirancang dengan menggunakan bahasa pemrograman *PHP*. Basis data yang digunakan pada sistem ini adalah *MySQL* dengan *engine MyISAM*. *Engine MyISAM* dipilih, karena memiliki skema yang lebih sederhana dan secara performa lebih cepat daripada *engine InnoDB*. *MyISAM* memiliki proses yang lebih cepat dalam membaca data, sehingga secara fungsionalitas cocok dengan fungsi aplikasi alumni ini. Aplikasi alumni ini dirancang untuk bisa menampilkan hasil pencarian alumni atau hasil grafis pada penyajian data alumni. *MyISAM* tidak memiliki peraturan relasi khusus, seperti pada *InnoDB*, sehingga bekerja dengan lebih ringan. *MyISAM* mendukung sistem *indexing* sehingga bisa digunakan untuk melakukan pencarian data secara lebih cepat. Kapasitas yang tampungan data pada *MyISAM* juga lebih besar daripada *InnoDB*.

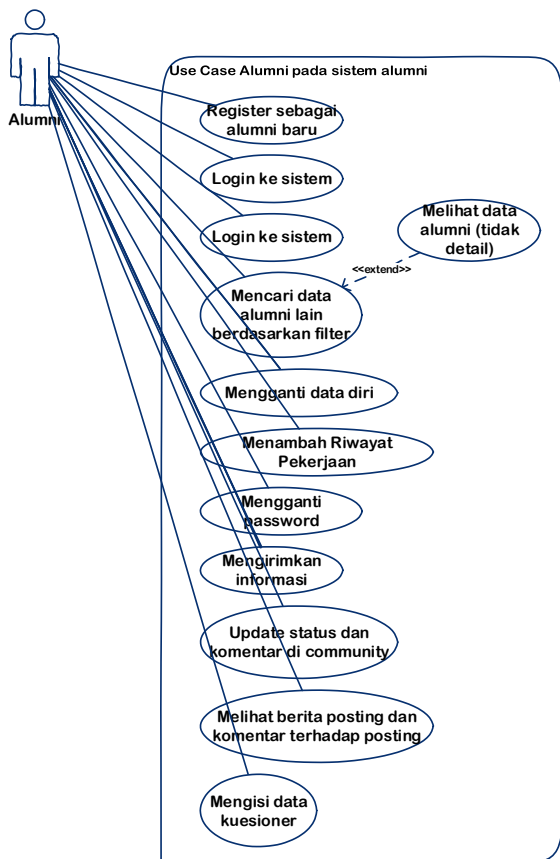
Sistem operasi yang digunakan untuk menjalankan *web server* berbasis *LAMP (Linux Apache MySQL PHP)* adalah *Ubuntu 15.04*. Spesifikasi prosesor yang dipakai adalah *Intel(R) Xeon(R) CPU E5-2695 v2*, dengan *memory* sebesar 4GB, dan dengan *harddisk* sebesar 40 GB *SCSI*.

Perancangan sistem diawali dengan analisis kebutuhan. Analisis kebutuhan digunakan untuk memahami peranan setiap pengguna dalam satu proses aplikasi. Setelah memahami peranan pengguna, maka bisa dibangun struktur informasi yang akan digunakan sesuai dengan hak akses dan tugas pengguna pada aplikasi. Analisis kebutuhan diawali dengan analisis fungsionalitas yang ditunjukkan pada Gambar 2, Gambar 3, dan Gambar 4. Gambar 2 merupakan *use case diagram* dari pengguna yang berperan sebagai *administrator* pada sistem.



Gambar 2. Use Case Administrator pada sistem alumni

Pada Gambar 2 bisa dilihat bahwa *administrator* untuk mencari dan melakukan *editing* data alumni dibuat dalam beberapa tingkatan. Tingkatan tersebut adalah *administrator* umum, *administrator* fakultas, dan *administrator* departemen, bahkan sampai ke program studi. Peran yang berbeda dari *administrator* umum adalah bahwa mereka bisa mengelola kategori berita, mengelola komentar, dan mengelola fakultas dan departemen. Peran *admin* sebagai pengelola data juga bisa melakukan ekspor data untuk keperluan akreditasi.

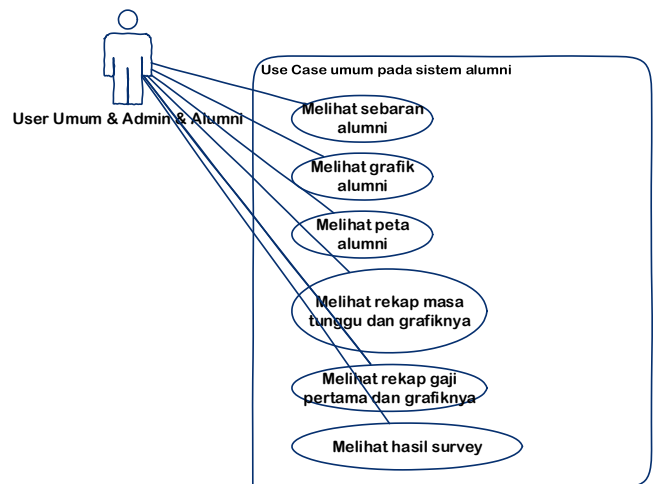


Gambar 3. Use Case Alumni pada sistem alumni

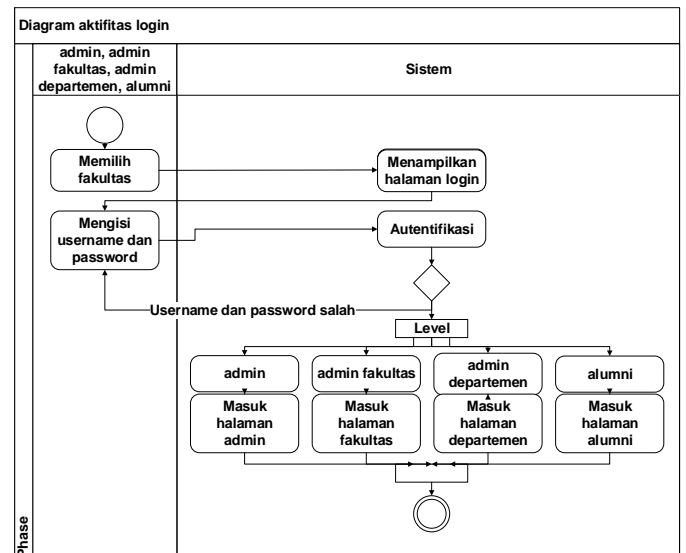
Pada Gambar 3 menunjukkan bahwa sebagai alumni ada beberapa aksi yang bisa dilakukan. Peran dari alumni pada pengembangan awal adalah untuk *register* ke sistem, karena data alumni belum terintegrasi dengan Biro Administrasi Akademik. Setelah berhasil registrasi ke sistem, maka alumni bisa melakukan *update* data diri dan melakukan *edit* riwayat, berinteraksi di komunitas, dan melihat berita seputar alumni yang di-*posting* oleh alumni yang ada. Untuk keperluan akreditasi, maka sebagai alumni juga harus mengisi data kuesioner yang sudah disesuaikan formatnya dengan borang akreditasi.

Pada Gambar 4 dijelaskan mengenai aksi yang bisa dilakukan sebagai pengguna umum aplikasi. Pengguna umum bisa melihat grafik, data, dan persebaran dari alumni, dan melihat rata-rata masa tunggu dan gaji pertama alumni dari masing-masing fakultas, serta hasil dari kuesioner.

Diagram aktifitas dapat menjelaskan secara rinci proses yang dilakukan dari masing-masing *use case diagram*, yang telah dijelaskan sebelumnya. Diagram aktifitas dibagi menjadi diagram aktifitas *login*, diagram kelola data alumni, kelola kategori berita dan komentar, kelola fakultas dan departemen, proses *updating* data alumni, dan proses melihat grafik yang ada pada sistem alumni.



Gambar 4. Use Case pengguna umum pada sistem alumni

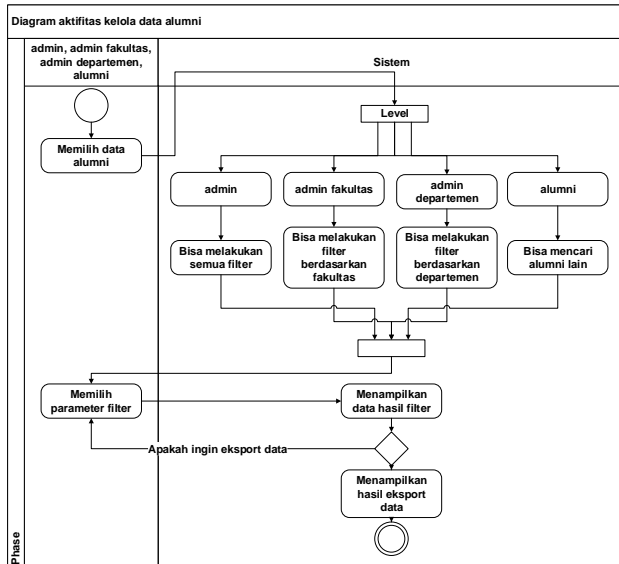


Gambar 5. Diagram aktifitas login

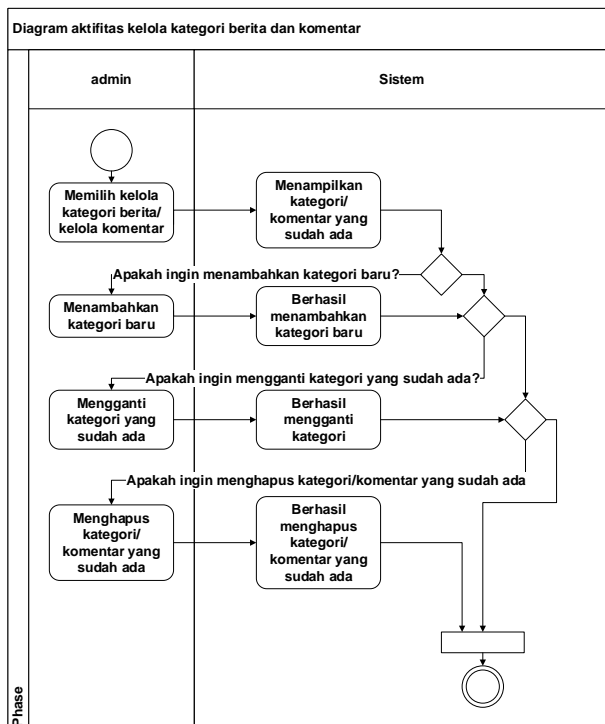
Diagram aktifitas *login* pada Gambar 5 merupakan aktifitas yang bisa dilakukan oleh semua *user*. Diagram ini terdiri dari aktifitas memilih fakultas dan kemudian *user* bisa mengisi *username* dan *password*. Pengguna dibedakan menjadi beberapa tingkat yaitu, *admin* pusat, *admin* fakultas, *admin* departemen, dan alumni. Tingkat pengguna tersebut digunakan untuk melakukan *filter* data mahasiswa dalam lingkup fakultas atau lingkup departemen. *Admin* fakultas atau departemen hanya bisa mengakses data fakultas atau departemen tertentu.

Diagram aktifitas kelola data alumni pada Gambar 6 menjelaskan mengenai proses *filtering* data alumni untuk mendapatkan data alumni yang lebih spesifik berdasarkan kriteria yang diinginkan. Diagram aktifitas kelola data alumni mengikuti hak akses yang ada untuk masing-masing *admin*. Parameter *filter* yang bisa dimasukkan mengikuti masing-masing *field* yang disediakan pada

form registrasi. Parameter *filter* di tiap *field* bisa dipadukan, misalnya *user* bisa melakukan pencarian mahasiswa lulusan tahun 2018 yang bekerja di Semarang.



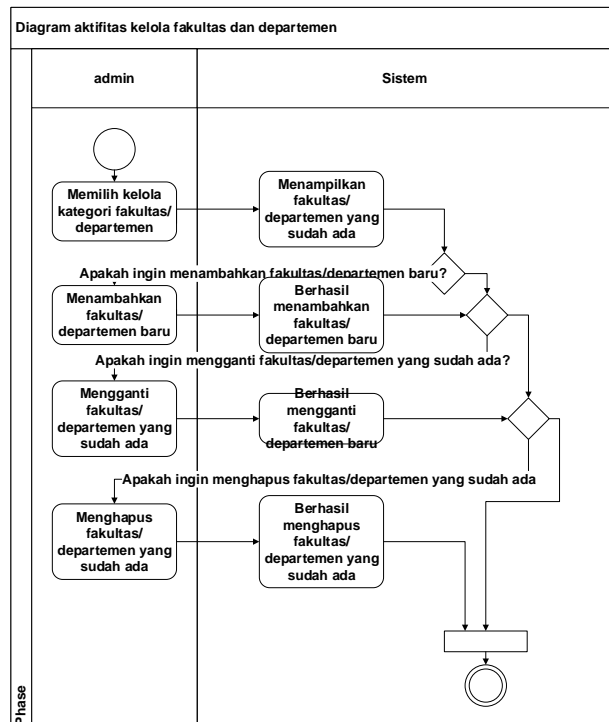
Gambar 6. Diagram aktifitas kelola data alumni



Gambar 7. Diagram aktifitas kelola kategori berita dan komentar

Diagram aktifitas kelola kategori berita dan komentar pada Gambar 7 merupakan fasilitas yang dimiliki oleh *admin* pusat. Diagram tersebut terdiri dari aktifitas untuk membuat kategori dan komentar baru, melakukan *edit*

kategori dan komentar, serta menghapus kategori dan komentar.



Gambar 8. Diagram aktifitas kelola fakultas dan departemen

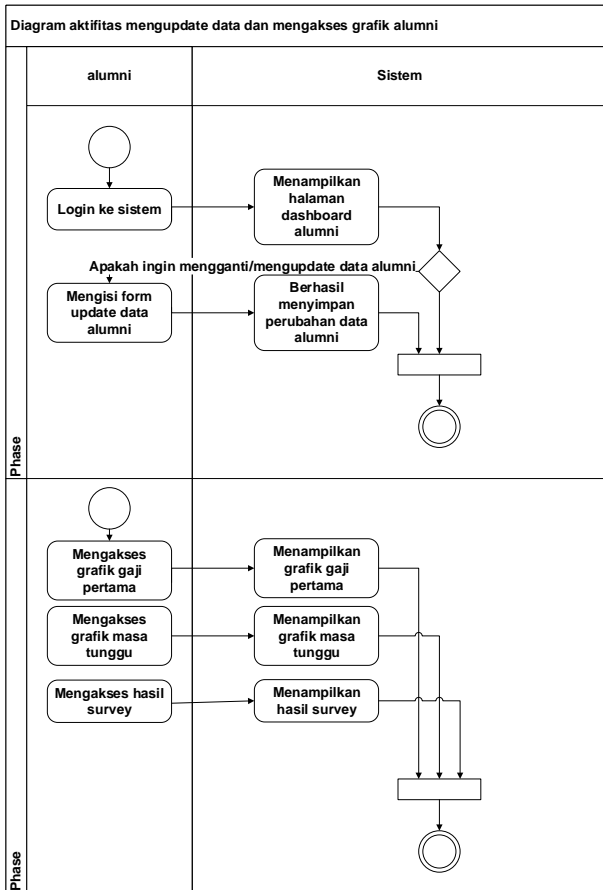
Diagram aktifitas kelola kategori berita dan komentar pada Gambar 8 terdiri dari aktifitas untuk membuat fakultas dan departemen baru, melakukan *edit* fakultas dan departemen yang sudah tersedia, serta menghapus fakultas dan departemen yang sudah tidak tersedia lagi.

Diagram aktifitas *update* data dapat dilakukan oleh setiap alumni. Selain itu alumni bisa mengakses data alumni berupa grafik masa tunggu dan pekerjaan tanpa perlu *login* ke aplikasi. *Update* data yang bisa dilakukan adalah *update* status bekerja, dan juga *update* keterangan riwayat pekerjaan. Aktifitas pengaksesan data alumni antara lain adalah melihat grafik berupa sebaran alumni, masa tunggu alumni mendapatkan pekerjaan, dan juga gaji rata-rata untuk pekerjaan pertama yang diperoleh alumni.

Library Data Table mempunyai fitur untuk melakukan *filtering* tingkat lanjut dengan memasukkan setiap *field* yang ingin dicari menggunakan *AJAX*. Tanpa menggunakan *library* ini, maka pencarian harus dilakukan dengan mengkombinasikan banyak parameter 'and' pada *query SQL*.

Parameter yang harus di-*setting* pada *Data Table* adalah *bServerSide* di-*set* menjadi *True* agar bisa melakukan proses *server side*. Proses *server side* ini berguna bila data yang ada di *database* sangatlah besar atau lebih dari

ratusan ribu. Tanpa menggunakan parameter *server side* maka proses *loading* data akan berhenti di suatu baris, karena penggunaan *memory* pada masing-masing *browser* juga terbatas.



Gambar 9. Diagram aktifitas mengupdate data dan mengakses data alumni

Parameter “dom” penting untuk memberi fitur yang bisa diakses pada proses *filter* dan *sorting* tabel. Dengan mengaktifkan fitur ‘lfrtip’ maka tabel bisa di-*filter* dan di-*sorting* secara bersamaan.

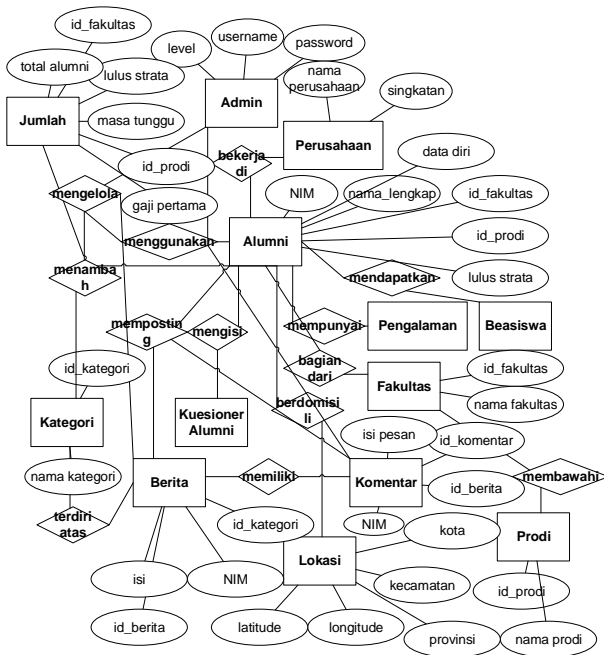
Untuk mengirim masing-masing parameter dari *field* pada fitur pencarian digunakan fungsi “aoData.push()” untuk mengirim parameter “fnServerParams”. Parameter “fnServerParams” merupakan parameter bawaan dari *Data Table* yang digunakan untuk memasukkan kriteria pencarian. Berikut adalah penggalan *source code* untuk melakukan *filtering* pada *Data Table*.

```

$(# kirim_filter ).on( 'click', function() {
dataAlumni.fnDestroy();
$(# alumni ).dataTable({
"bProcessing": true,
"bServerSide": true,
"sAjaxSource": "dataalumni.php",
"dom": "lfrtip",
"fnServerParams": function ( aoData ) {
aoData.push( { "name": "nim", "value": $( "#nim" ).val() },
{ "name": "nm_lengkap", "value": $( "#nm_lengkap" ).val() },
...
});
"aoColumns": [
null, null, null, null, null, null, null, null, null, null,
{ "bSearchable": true, "bVisible": false },
{ "bSearchable": true, "bVisible": false },
null, null,
{ "bSearchable": true, "bVisible": false },
{ "bSearchable": true, "bVisible": false },
{ "bSearchable": true, "bVisible": false },
{ "bSearchable": true, "bVisible": false },
null,
{ "bSearchable": true, "bVisible": false },
{ "bSearchable": true, "bVisible": false },
{ "bSearchable": true, "bVisible": false },
{ "bSearchable": true, "bVisible": false },
null
],
"language": {
"processing": "<span class='glyphicon glyphicon-refresh glyphicon-refresh-animate'></span> Loading data, Please wait..."
},});});
    
```

Pada bagian “aoColumns” maksimal diisi oleh 13 tabel dan bila lebih dari itu terjadi *error*. *Data Table* bisa melakukan pencarian dan *filtering*, dengan mengatur beberapa parameter yang harus dirubah menjadi parameter “get”. Parameter tersebut diantaranya “iDisplayStart”, “iDisplayLength”, “iSortingCols”, dan “sSearch”. Parameter “iDisplayStart” diisi dengan dari mulai baris mana *query* dilakukan dan pasangannya adalah “iDisplayLength” yang digunakan untuk membatasi menampilkan data menggunakan *query* “limit”. Parameter “iSortingCols” menunjukkan kolom mana saja yang bisa diurutkan. Parameter “sSearch” digunakan untuk memberikan parameter “where” dalam proses seleksi data.

Selain perancangan diagram aktifitas maka juga dilakukan perancangan diagram relasi entitas untuk menggambarkan relasi data di antara entitas yang terbentuk pada *database*. Gambar 10 merupakan *Entity Relationship Diagram* untuk keseluruhan sistem *database* alumni.



Gambar 10 Entity Relationship Diagram

Pada Gambar 10 disajikan tabel lengkap dari seluruh entitas yang ada. Selain tabel inti disiapkan juga tabel *view* yang digunakan untuk mempercepat proses perhitungan gaji rata-rata dan juga masa tunggu mendapatkan pekerjaan pertama. Alumni memiliki *primary key* Nomor Induk Mahasiswa sebagai pembeda yang unik. Tabel inti ini memiliki relasi dengan beberapa tabel lain seperti tabel fakultas dan tabel program studi. Setiap melakukan *insert* data di tabel alumni ini, maka akan terjadi *trigger* data ke tabel lain, yaitu tabel jumlah yang tersaji pada Tabel 1. Perintah *trigger* yang dilakukan untuk mengisi Tabel 1 adalah sebagai berikut.

```

BEGIN
DECLARE vrow_count int;DECLARE vmasa_tunggu int;
DECLARE vrentang_gaji_1 int;DECLARE vrentang_gaji_2 int;
DECLARE vrentang_gaji_3 int;DECLARE vrentang_gaji_4 int;
DECLARE vomset_1 int;DECLARE vomset_2 int;
DECLARE vomset_3 int;DECLARE vomset_4 int;
SELECT COUNT(nim) INTO vrow_count FROM tbl_alumni WHERE
id_fakultas = NEW.id_fakultas and tbl_alumni.lulus_strata =
NEW.lulus_strata and tbl_alumni.id_prodi = NEW.id_prodi;
SELECT sum(masa_tunggu) INTO vmasa_tunggu FROM tbl_alumni
WHERE id_fakultas = NEW.id_fakultas and tbl_alumni.lulus_strata =
NEW.lulus_strata and tbl_alumni.id_prodi = NEW.id_prodi;
IF(NEW.gaji_pertama = 1) THEN
SELECT COUNT(gaji_pertama) INTO vrentang_gaji_1 FROM
tbl_alumni WHERE id_fakultas = NEW.id_fakultas and
tbl_alumni.lulus_strata = NEW.lulus_strata and tbl_alumni.id_prodi =
NEW.id_prodi and tbl_alumni.gaji_pertama = NEW.gaji_pertama;
UPDATE tbl_jumlah
SET tbl_jumlah.rentang_gaji_1 = vrentang_gaji_1
WHERE id_fakultas = NEW.id_fakultas and lulus_strata =
NEW.lulus_strata and id_prodi = NEW.id_prodi;
ELSEIF (NEW.gaji_pertama = 2) THEN
...
END IF;
IF(NEW.omset = 1) THEN

```

```

SELECT COUNT(omset) INTO vomset_1 FROM tbl_alumni WHERE
id_fakultas = NEW.id_fakultas and tbl_alumni.lulus_strata =
NEW.lulus_strata and tbl_alumni.id_prodi = NEW.id_prodi and
tbl_alumni.omset = NEW.omset;
UPDATE tbl_jumlah
SET tbl_jumlah.omset_1 = vomset_1
WHERE id_fakultas = NEW.id_fakultas and lulus_strata =
NEW.lulus_strata and id_prodi = NEW.id_prodi;
ELSEIF (NEW.omset = 2) THEN
...
END IF;
UPDATE tbl_jumlah
SET tbl_jumlah.total_alumni = vrow_count
WHERE id_fakultas = NEW.id_fakultas and lulus_strata =
NEW.lulus_strata and id_prodi = NEW.id_prodi;
UPDATE tbl_jumlah
SET tbl_jumlah.total_masa_tunggu = vmasa_tunggu
WHERE id_fakultas = NEW.id_fakultas and lulus_strata =
NEW.lulus_strata and id_prodi = NEW.id_prodi;
END

```

Trigger yang dibuat membantu proses *query* data yang memiliki perhitungan. Setiap alumni dari fakultas dan departemen tertentu bertambah, maka jumlah alumni di tabel *view* juga ditambah dengan menggunakan *trigger*. Tabel jumlah alumni merupakan tabel *view* yang membantu dalam melakukan perhitungan tiap fakultas dan departemen tanpa perlu melakukan “select count” langsung di tabel alumni. Setiap ada alumni baru yang masuk akan di-*filter* berdasarkan “ID” fakultas dan program studi. Tabel jumlah alumni digunakan untuk menampilkan tabel masa tunggu dan gaji pertama alumni.

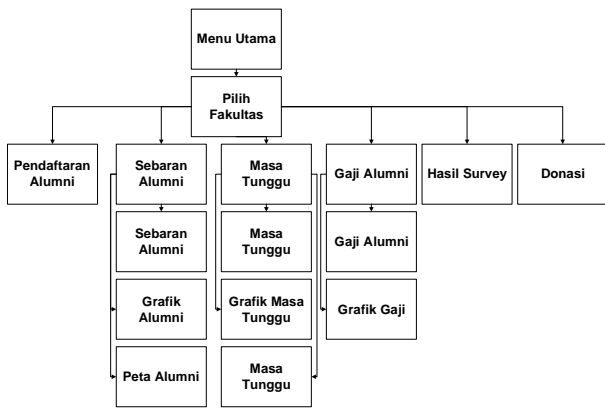
Berikut merupakan isi dari tabel *view* “jumlah_alumni” yang menyimpan jumlah total alumni dan total masa tunggu. Untuk rentang gaji dan rentang omset karena datanya bukan berupa data numeris, maka dilakukan perhitungan terhadap modus atau jumlah terbanyak rentang gaji pada opsi rentang 1, 2, 3, atau 4.

Tabel 1 Tabel *view* jumlah_alumni

Name	Type
id_fakultas	int(10)
lulus_strata	varchar(2)
id_prodi	int(10)
total_alumni	int(11)
total_masa_tunggu	int(11)
rentang_gaji_1	double(10,2)
rentang_gaji_2	double(10,2)
rentang_gaji_3	double(10,2)
rentang_gaji_4	double(10,2)
omset_1	double(10,2)
omset_2	double(10,2)
omset_3	double(10,2)
omset_4	double(10,2)

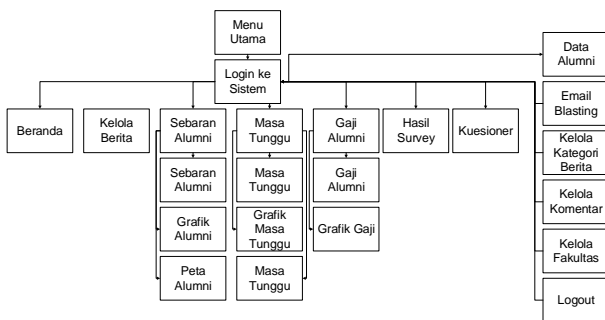
3. Hasil dan Analisis

Selain perancangan ERD juga ada perancangan struktur menu. Struktur menu untuk user secara umum dijelaskan pada Gambar 11. *User* secara umum bisa mengakses menu-menu yang ada di sebelah atas.



Gambar 11. Menu User Umum

Selain menu *user* umum ada juga menu utama setelah *user* login. Setelah *user* berhasil *login* akan muncul menu tambahan di sebelah kiri. Menu tambahan ketika *login* dijelaskan pada Gambar 12.

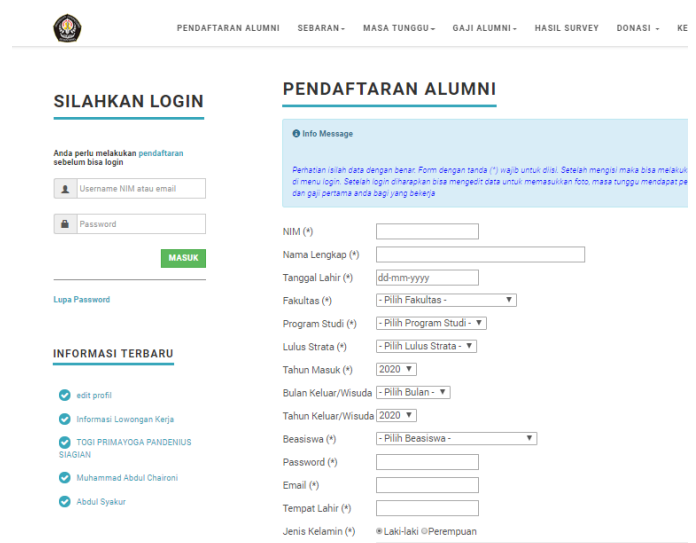


Gambar 12. Menu User Setelah Login

Pada saat pengguna masuk ke sistem maka diberi pilihan untuk memilih fakultas seperti pada Gambar 13. Setelah masuk ke halaman utama maka peserta dihadapkan pilihan untuk melakukan registrasi bila belum pernah membuat akun di sistem dan diarahkan ke menu *login* bila sudah pernah melakukan registrasi ke sistem seperti pada Gambar 14.



Gambar 13. Menu memilih Fakultas



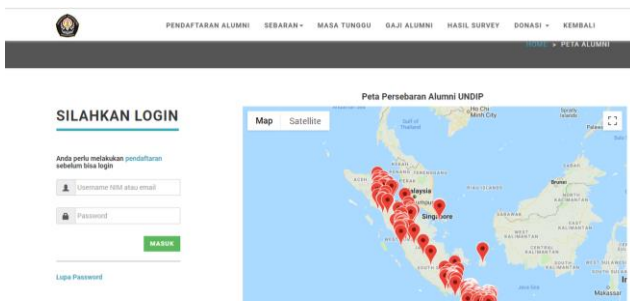
Gambar 14. Menu registrasi dan login ke sistem

Pada aplikasi terdapat informasi jumlah seluruh alumni yang sudah mendaftar. Jumlah alumni yang sudah mendaftar bisa dilihat lebih detail lagi untuk masing-masing fakultas, dan juga menurut strata pendidikan alumni seperti pada Gambar 15. Selain jumlah alumni yang mendaftar ada juga rekapitulasi masa tunggu rata-rata alumni dan rata-rata gaji pertama ketika alumni mendapatkan pekerjaan.

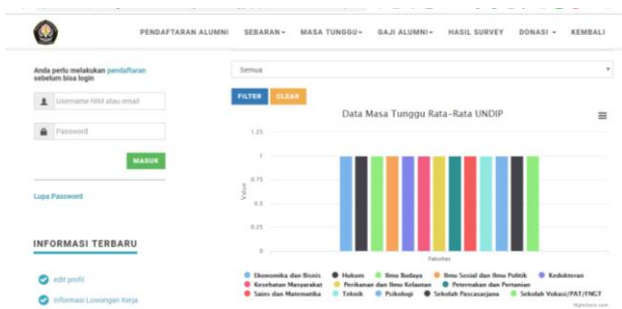
Detail	Fakultas	Jumlah
DETAIL	Ekonomika dan Bisnis	20906
DETAIL	Hukum	13874
DETAIL	Ilmu Budaya	9785
DETAIL	Ilmu Sosial dan Ilmu Politik	14107
DETAIL	Kedokteran	12458
DETAIL	Kesehatan Masyarakat	7313
DETAIL	Perikanan dan Ilmu Kelautan	7221

Gambar 15. Jumlah alumni yang sudah mendaftar ke sistem

Pada aplikasi terdapat *library Google Map* untuk menampilkan lokasi dari masing-masing alumni berdasarkan domisili tempat tinggalnya. Peta persebaran alumni ini bisa digunakan untuk memetakan di mana alumni paling banyak menyebar seperti pada Gambar 16. Selain dalam bentuk *list*, persebaran alumni juga tersedia dengan pilihan *filter* masing-masing fakultas dan departemen. *List* dari persebaran alumni bisa dilihat pada Gambar 17.

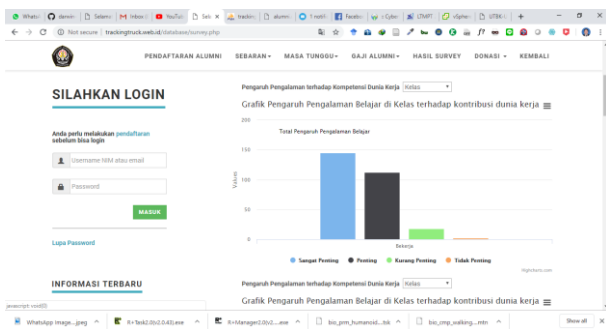


Gambar 16. Jumlah alumni yang sudah mendaftar ke sistem



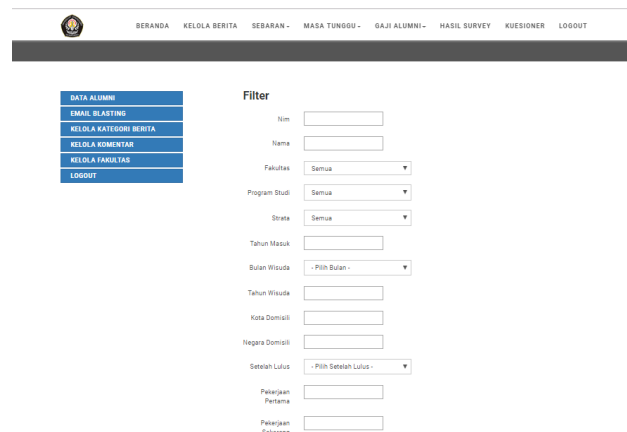
Gambar 17. Data masa tunggu dengan filter

Selain grafik dari persebaran alumni juga bisa dilihat grafik dari hasil survey yang dilakukan di aplikasi oleh alumni. Grafik ini digunakan untuk memperoleh faktor yang penting, yang bisa diaplikasikan oleh alumni ke bidang pekerjaan. Grafik dari hasil form survey bisa dilihat pada Gambar 18.

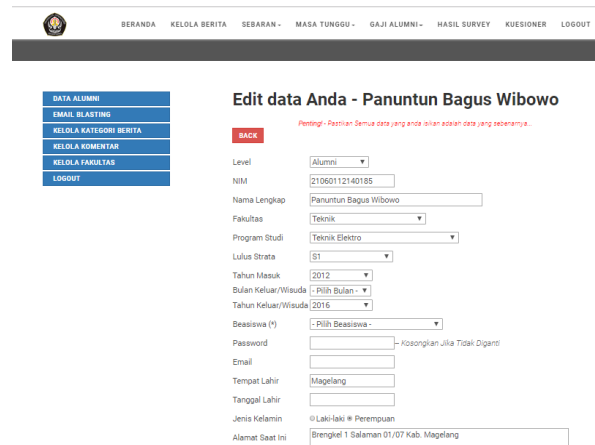


Gambar 18. Grafik Hasil Survey

Filter data yang bisa ditampilkan pada aplikasi secara lengkap adalah *field-field* yang diisikan pada *form* pendaftaran. Filter ini dilakukan dengan menggunakan logika “and”, sehingga bisa melakukan banyak perpaduan dalam melakukan pencarian spesifik seperti pada Gambar 19. Setelah melakukan filter data secara spesifik untuk *admin* pusat, *admin* tingkat fakultas, dan *admin* tingkat departemen bisa melakukan *edit* data alumni bila terjadi kesalahan, atau dari sisi alumni sendiri dapat memperbarui datanya seperti pada Gambar 20.



Gambar 19. Filter data alumni berdasarkan field tertentu



Gambar 20. Edit data diri untuk alumni

Selain melakukan *edit* data alumni, *admin* bisa melakukan pengelolaan kategori berita yang di-*posting* oleh alumni, dan juga melakukan pengelolaan komentar terhadap *posting* yang masuk melalui menu *community*. Alumni bisa menambah status dan melakukan *posting* komentar melalui menu *community*. Pengelolaan data komentar bisa dilihat pada Gambar 21.

Pada Tabel 2 bisa dilihat hasil pengujian fungsional lengkap yang menguji masing-masing unit. Pengujian yang dilakukan tidak hanya dari segi keberhasilan fungsi, namun juga pada keberhasilan *query* yang dilakukan secara konkuren untuk dilihat performa perbandingan penggunaan tabel *view* dan melakukan “select count” secara langsung.

Uji validitas data *output* yang dilakukan adalah *data type check*, *simple range and constraint check*, *code and cross reference check*, dan *structured validation*.

Pada *data type check* dicoba untuk di setiap *form* tidak diisikan data pada *mandatory field* yang diperlukan. Pada tes tersebut, bila belum memasukkan data pada

mandatory field, maka form tidak dapat diproses ke tahap selanjutnya.

Pengujian lain adalah mencoba memasukkan alumni dengan NIM yang sama dengan sebelumnya. Bila ada NIM yang sama, yang sudah pernah dipakai, maka aplikasi tidak memperbolehkan form untuk di-submit.

Pada pengujian simple range and constraint check dicoba untuk memasukkan NIM lebih dari kapasitasnya, hal tersebut membuat form isian akan menjadi merah menandakan tipe data NIM harus diisi sesuai jumlah karakternya.

Pada pengujian code and cross reference check dicoba untuk menggabungkan beberapa pengujian antara data yang dimasukkan dengan kesesuaian jumlah alumni. Ketika melakukan insert data alumni maka pada tabel view dicocokkan dengan query perhitungan langsung yang menghitung jumlah alumni yang masuk, total rata-rata masa tunggu, dan total modus dari gaji. Hasil yang diperoleh adalah bahwa pada jumlah alumni telah cocok bila dibandingkan dengan query perhitungan langsung.

Pada pengujian structured validation semua fungsionalitas diuji. Fungsionalitas dari beberapa modul dibuatkan sebuah alur pengujian yang menguji dari mulai data dimasukkan, data di-edit, data di-filter, serta diberi variasi memasukkan data dengan parameter yang salah. Data dengan parameter yang salah perlu diperlakukan khusus yaitu dengan pengecekan ke sumber pangkalan data yang pada saat ini sedang dikembangkan. Selain itu juga diuji beberapa performance pada sistem seperti pengujian kecepatan query dari konkurensi.



Gambar 21. Pengelolaan data komentar

Pada Tabel 2 disajikan hasil yang diharapkan dari pengujian blackbox dan juga ada beberapa catatan yang menjadi masukan pengujian selanjutnya. Selain dari pengujian blackbox terdapat pengujian query menggunakan concurrent user yang mengakses database secara bersamaan menggunakan tools bawaan dari mysql yaitu "mysqlslap".

Tabel 2. Tabel pengujian fungsi blackbox

Skenario Pengujian	Hasil yang Diharapkan	Diterima	Ditolak	Dengan Catatan
Pengujian Login ke Sistem dengan email dan NIM	Bisa login dengan email dan juga NIM	V		Apakah ada error bila dicoba dengan user lain
Pengujian registrasi ke sistem	Bisa registrasi dengan notifikasi email	V		Mengatur limit pengiriman email
Pengujian mengupdate data diri dari alumni	Data berhasil diupdate dan tidak ada kesalahan pilihan	V		Perlu dicek apakah field bekerja, kuliah di luar negeri, wirausaha sudah sesuai
Pengujian filtering mencari data alumni	Alumni bisa di-filter sesuai dengan parameter	V		Perlu pengecekan apakah data sudah cocok dan sesuai
Pengujian import data alumni	Data alumni ter-import sesuai dengan filter	V		
Pengujian pengisian news feed	News feed sudah terisi dan bisa dikomentari	V		
Pengujian pengisian data riwayat pekerjaan	Data riwayat pekerjaan ditampilkan dengan baik	V		
Pengujian tampilan grafik hasil kuesioner	Grafik ditampilkan dengan baik sesuai filter	V		
Pengujian tampilan tree dari jumlah alumni per prodi	Grafik ditampilkan dengan baik dan jumlah alumni sesuai	V		
Pengujian tampilan grafik masa tunggu dan gaji pertama	Grafik sesuai dan ditampilkan dengan baik	V		

Tabel 3. Tabel pengujian konkurensi query

Konkurensi	Iterasi	Query	Waktu Rata-Rata (s)	Minimal	Maksimal
50	5	select count(nim) from tbl_alumni	2	1.937	2.11
50	5	select total_alumni from tbl_jumlah	0.037	0.031	0.047

Pada Tabel 3 bisa dilihat bahwa *trigger* yang digunakan untuk menambah data di tabel jumlah berhasil meningkatkan performa dari segi pengaksesan *query*. Dengan 50 user mengakses *query* dalam waktu yang bersamaan, maka waktu yang dibutuhkan relatif lebih singkat dengan perbandingan antara 0.037 detik dan 2 detik. Hal ini menunjukkan bahwa teknik tabel *view* yang digunakan akan meningkatkan performa dari segi waktu.

4. Kesimpulan

Berdasarkan pengujian yang dilakukan pada aplikasi dan juga masukan yang berasal dari setiap fakultas dan departemen, dapat disimpulkan bahwa sistem alumni berbasis web yang dirancang dengan PHP ini bisa digunakan dengan baik oleh setiap alumni yang mendaftar. Permasalahan yang terjadi adalah tidak adanya proses verifikasi menggunakan *email* untuk *user* yang mendaftar ke sistem ini. Untuk melakukan verifikasi dibutuhkan *email relay server* yang bisa mengirimkan *email* berjumlah ratusan ribu dalam waktu sehari, karena jumlah alumni hampir mencapai 200 juta. Selain fitur *email*, semua fitur pada aplikasi telah berjalan baik dan dapat memenuhi fungsi yang dibutuhkan untuk melakukan manajemen dan pelacakan alumni.

Aplikasi ini dapat dikembangkan lebih lanjut dengan melakukan integrasi dengan Sistem Informasi Akademik, agar setiap alumni yang lulus bisa langsung terdaftar sebagai alumni. Alumni hanya bisa melakukan *update* data tentang pekerjaan dan tidak perlu registrasi ke aplikasi, bila sudah terintegrasi.

Pengembangan aplikasi ini dalam *platform* khusus *mobile* juga diperlukan untuk mempermudah para pengguna *platform Android* atau *iPhone* untuk melakukan *update* data. Pengembangan yang lain adalah dari sisi kemampuan aplikasi yang dapat menghubungkan *user* agar bisa berinteraksi secara aktif di dalam aplikasi. Pengembangan media sosial khusus diperlukan juga bagi para alumni. Dengan menggunakan media sosial ini bisa menjadi sarana melakukan kerjasama dengan berbagai pihak di luar Undip juga dari rekanan Undip.

Daftar Pustaka

- [1]. P. Wibowo Yunanto, D. Nurhidayat, and R. Wicaksono, "Development of Web-based Information System for Universitas Negeri Jakarta" *KnE Soc. Sci.*, vol. 3, no. 12, p. 453, Mar. 2019.
- [2]. A. A. Lutfi and R. Wahyudi, "Aplikasi Tracer Study Berbasis Website Responsive pada Fakultas Pertanian Universitas Jenderal Soedirman Purwokerto" vol. 13, p. 8, 2017.
- [3]. I. R. I. Astutik and I. Widiaty, "Integrated Management Information System for Curriculum in University" IOP Conf. Ser. Mater. Sci. Eng., vol. 384, p. 012077, Jul. 2018.
- [4]. A. Azis, Y. Sugiarti, N. Kumaladewi, and M. Q. Huda, "Designing and Building an Information System of Career Development and Alumni Based on Android (Case Study: Information Systems Department, Syarif Hidayatullah State Islamic University Jakarta)" in *2018 6th International Conference on Cyber and IT Service Management (CITSM)*, Parapat, Indonesia, 2018, pp. 1–5.
- [5]. L. Kokoszkiwicz *et al.*, "hBrowse - Generic framework for hierarchical data visualisation" in *Proceedings of Science, Munich, Germany*, 2012, p. 9.
- [6]. Asroni, J. Yusanti, and S. Riyadi, "Dashboard-based Alumni Tracer Study Report Using Normalized Data Store Architecture" in *Proceedings of the 2019 8th International Conference on Software and Computer Applications - ICSCA '19*, Penang, Malaysia, 2019, pp. 70–74.
- [7]. Z. Chen, "Using AJAX to Optimize Web Services Performance," in *2009 International Conference on Computational Intelligence and Software Engineering* Wuhan, China, 2009, pp. 1–4.
- [8]. I. Warman and K. N. Saputra, "Sistem Informasi Alumni ITP Menggunakan PHP Dan My SQL" *J. Momentum*, vol. 12, no. 1, pp. 43–50, Feb. 2012.
- [9]. M. Rizka, A. Amri, H. Hendrawaty, and M. Mahdi, "Analisis Dan Perancangan Sistem Informasi Tracer Study Berbasis Web" *J. Infomedia*, vol. 3, no. 2, Dec. 2018.
- [10]. R. Setiawan, "Perancangan Sistem Informasi Pengelolaan Alumni Perguruan Tinggi" vol. 15, no. 02, p. 10, 2018.
- [11]. A. G. Gani, "Analisis Sistem Informasi Pengelolaan Data Alumni Berbasis Codeigniter PHP Framework" in *e-Prints@UNDY*, p. 23, 2018.
- [12]. S. Balaji, "Waterfall vs V-Model vs Agile: A Comparative Study on SDLC" in *International Journal of Information Technology and Business Management*. Vol., no. 1, p. 5, 2012.
- [13]. M. Al-Hashimi, A. Razzaque, and A. M. Hamdan, "Alumni assessment of Management Information Systems job skills and skill gaps" p. 8, 2018.