

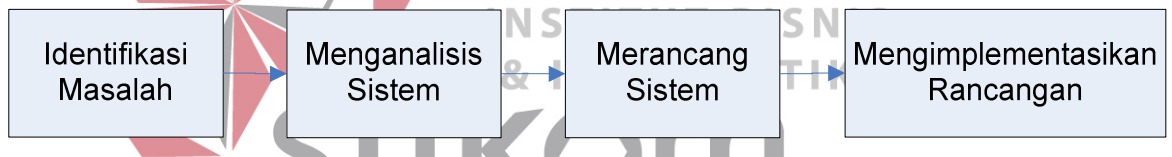
## BAB III

### ANALISIS DAN PERANCANGAN SISTEM

#### 1.1 Analisis Sistem

Analisis sistem merupakan tahap pertama dalam mengembangkan perangkat lunak dengan metode *waterfall*. Pada tahap ini dilakukan pengumpulan fakta, menganalisis fakta-fakta tersebut, kemudian mengomunikasikannya melalui laporan analisis sistem. Temuan-temuan tersebut digambarkan menggunakan diagram-diagram tertentu untuk memperjelas alur atau proses yang ada.

Langkah-langkah yang dilakukan dalam menyelesaikan masalah digambarkan pada gambar blok diagram berikut ini:



Gambar 3.1 Tahap-tahap Analisis Sistem

Tahap-tahap analisis dimulai dari identifikasi masalah kemudian dilakukan usaha dalam menemukan kelemahan atau kekurangan yang ada pada sistem yang lama kemudian menentukan dan merumuskan kebutuhan informasi pengguna sistem sehingga dapat ditentukan solusi untuk memecahkan masalah tersebut.

##### 3.1.1 Identifikasi Masalah

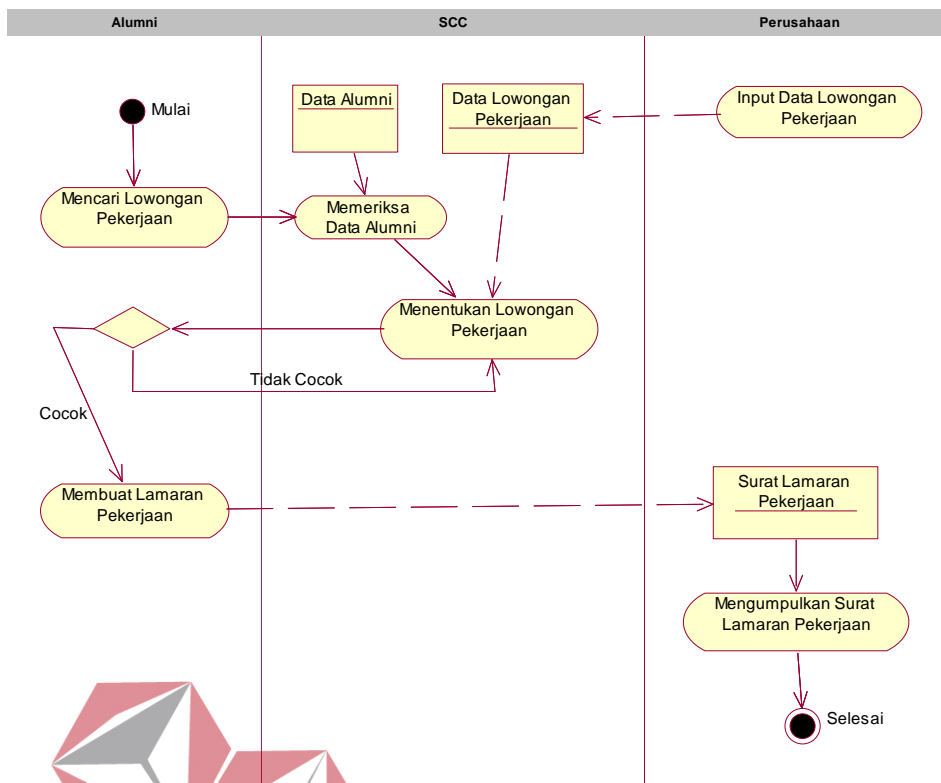
STIKOM Career Center sebagai jembatan antara perusahaan/instansi/industri pencari kerja dan para pencari kerja khususnya alumni STIKOM Surabaya menyediakan berbagai macam lowongan kerja. Bagi lulusan yang belum atau masih mencari lowongan kerja dapat menghubungi SCC

untuk mendapatkan rekomendasi lowongan kerja yang baik. Selain itu, SCC juga secara aktif mencari lulusan berdasarkan permintaan dari pengguna lulusan atau perusahaan.

Lulusan yang sedang mencari lowongan pekerjaan dapat menghubungi SCC dengan cara datang langsung ke kantor SCC atau melalui telepon. Kemudian SCC akan memberikan rekomendasi lowongan pekerjaan bagi lulusan tersebut berdasarkan pertimbangan-pertimbangan tertentu dari SCC. Selain itu, lulusan juga dapat memberikan surat lamaran pekerjaan, serta berkas-berkas yang lainnya yang diperlukan untuk melamar sebuah lowongan pekerjaan melalui SCC.

Selain lulusan yang mencari lowongan pekerjaan, SCC juga secara aktif mencari lulusan berdasarkan permintaan dari pengguna lulusan. Dengan data-data yang dimiliki, SCC menentukan lulusan yang cocok dengan kriteria permintaan dari pengguna lulusan untuk dijadikan rekomendasi bagi perusahaan tersebut. Perusahaan-perusahaan pencari lulusan dari STIKOM Surabaya juga dapat mendaftarkan dirinya melalui *website* SCC. Melalui *website* resmi SCC, perusahaan dapat memasukkan kriteria-kriteria lulusan yang diharapkan.

Lowongan pekerjaan yang diinputkan oleh perusahaan di dalam *website* SCC tersebut juga dapat diakses oleh lulusan. Para lulusan juga dapat memperoleh informasi yang lengkap melalui *website* SCC. Lulusan yang tertarik terhadap lowongan tertentu dapat mengirimkan berkas-berkas lamaran yang diperlukan melalui *website* SCC.



Gambar 3.2 Activity Diagram Alumni Mencari Lowongan Pekerjaan

Website yang digunakan SCC saat ini hanya mampu memberikan rekapitulasi tentang data terbaru alumni dan data tentang lowongan pekerjaan yang diinputkan oleh perusahaan pencari tenaga kerja. Kelemahan dari website yang ada saat ini adalah website tersebut tidak mampu dijadikan tempat khusus oleh SCC untuk mempublikasikan agendanya.

SCC juga membutuhkan data-data tentang prestasi kerja alumni yang didapatkan di dalam dunia kerja. Selama ini SCC belum mampu merekam data tersebut menggunakan sistem yang ada sehingga SCC tidak dapat memberikan penghargaan kepada alumni yang berprestasi di dalam dunia kerja.

Selain itu, STIKOM Surabaya perlu untuk mengevaluasi kurikulum, fasilitas, dan aspek-aspek pendukung proses pembelajaran lainnya apakah sudah sesuai dengan kebutuhan dunia kerja. Namun SCC tidak dapat menyediakan

informasi tersebut karena sistem yang ada saat ini tidak memungkinkan SCC untuk melakukan pelacakan alumni (*tracer study*). Sedangkan hal-hal tentang *tracer study* telah diatur oleh Badan Akreditasi Nasional Perguruan Tinggi dalam standarnya yang ke-3 (tiga).

Untuk melakukan *tracer study*, Direktorat Jenderal Pendidikan Tinggi (DIKTI) telah menyediakan standar kuesioner yang dapat digunakan oleh perguruan tinggi. Kuesioner tersebut terbagi menjadi beberapa bagian yaitu: transisi alumni ke dunia kerja, tingkat penguasaan kompetensi alumni setelah lulus, kontribusi perguruan tinggi terhadap kompetensi alumni, dan evaluasi alumni terhadap aspek-aspek pendukung proses pembelajaran selama kuliah.

Namun masalahnya, jumlah alumni STIKOM Surabaya adalah ribuan dan berasal dari berbagai kota dan pulau. Penyebaran kuesioner secara konvensional akan memakan biaya dan waktu yang besar, karena STIKOM Surabaya harus menanggung biaya pengiriman pulang dan pergi ke kuesioner-kuesioner tersebut kepada para lulusan yang ditentukan sebagai responden *tracer study*.

Apabila kebutuhan-kebutuhan tersebut tidak terpenuhi, maka dalam mempublikasikan agendanya SCC akan menghubungi alumni satu per satu. Hal tersebut tentunya juga akan memakan waktu dan biaya yang relatif besar. Selain itu, para alumni juga tidak dapat saling berkomunikasi dengan alumni yang lainnya. Apabila SCC tidak mampu membuktikan kinerja lulusan di masyarakat sesuai dengan profesi dan bidang ilmu, STIKOM Surabaya tidak dapat memenuhi standar ke-3 (tiga) pada standar akreditasi program studi sarjana yang diatur oleh Badan Akreditasi Nasional Perguruan Tinggi. SCC dan ketua program studi juga

tidak memiliki informasi yang dapat dijadikan acuan dalam memperbaiki aspek-aspek yang menunjang proses belajar mengajar mahasiswa.

### 3.1.2 Melakukan Studi Pustaka

Dari identifikasi masalah yang ditemukan kemudian dicari pustaka sebagai dasar dalam analisis dan perancangan sistem sesuai teori yang ada. Untuk menentukan metode pemecahan masalah, digunakan baik buku maupun jurnal-jurnal terkait yang membahas tentang komunitas dan pelacakan alumni.

### 3.1.3 Menganalisis Kebutuhan Pengguna

Dari analisis kebutuhan pengguna didapatkan beberapa kebutuhan berdasarkan pengguna berikut ini:

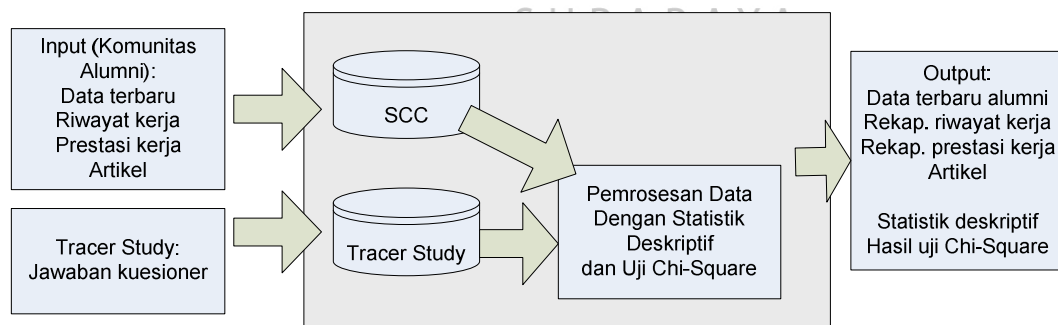
1. Alumni membutuhkan sistem yang memungkinkannya untuk tetap terhubung dengan alumni yang lain. Selain itu, alumni juga dapat melakukan pencarian data dan mengakses data riwayat kerja alumni yang lain.
2. Pengurus alumni sebagai administrator *website* komunitas alumni memerlukan rekapitulasi artikel-artikel yang ditulis oleh alumni dan komentar-komentar yang ada di dalamnya. Selain itu pengurus alumni juga membutuhkan fasilitas untuk mengirimkan *email* secara massal kepada alumni untuk keperluan tertentu.
3. SCC membutuhkan informasi tentang prestasi kerja yang diraih oleh alumni di dalam dunia kerja. Selain itu, karena SCC juga bertugas untuk menjadi wadah bagi pengembangan kualitas mahasiswa dan alumni, SCC membutuhkan informasi tentang: transisi alumni ke dunia kerja, tingkat penguasaan alumni terhadap kompetensi pada saat lulus, kontribusi perguruan

tinggi terhadap kompetensi alumni, dan evaluasi alumni terhadap aspek-aspek pendukung proses pembelajaran selama kuliah.

4. Ketua program studi sebagai penyusun kurikulum dan aspek-aspek pendukung proses pembelajaran lainnya membutuhkan informasi tentang: transisi alumni ke dunia kerja, tingkat penguasaan alumni terhadap kompetensi pada saat lulus, kontribusi perguruan tinggi terhadap kompetensi alumni, dan evaluasi alumni terhadap aspek-aspek pendukung proses pembelajaran selama kuliah.

### 3.2 Perancangan Sistem

Perancangan sistem di sini akan memberikan gambaran tentang langkah-langkah dalam membangun sistem yang akan dibuat. Gambaran umum mengenai sistem yang akan dibangun dapat dilihat pada gambar 3.3. Blok diagram tersebut menjelaskan tentang apa saja yang menjadi input, proses, dan output yang akan diperlukan sistem.



Gambar 3.3 Blok Diagram Website Komunitas Alumni dan *Tracer Study*

Pada gambar di atas dijelaskan bahwa masukan yang diperlukan oleh sistem antara lain adalah: data terbaru, riwayat kerja, prestasi kerja, artikel, dan jawaban kuesioner *tracer study*. Data-data tersebut akan disimpan pada basis data

yang berbeda. Data terbaru alumni, riwayat kerja alumni, prestasi kerja alumni, dan artikel yang dituliskan oleh alumni akan disimpan ke dalam database SCC. Sedangkan jawaban kuesioner *tracer study* oleh alumni akan disimpan ke dalam database *tracer study*.

Masukan yang dibutuhkan untuk *tracer study* antara lain adalah: pencarian kerja dan transisi ke dunia kerja, tingkat penguasaan alumni terhadap kompetensi pada saat lulus, kontribusi perguruan tinggi terhadap kompetensi alumni, dan peran perguruan tinggi terhadap hal-hal pendukung proses belajar mengajar pada saat kuliah.

Selanjutnya data-data yang didapatkan dari hasil kegiatan *tracer study* akan diproses menggunakan statistik deskriptif dan uji *chi-square*. Setiap data-data yang didapatkan dari sebuah pertanyaan pada kuesioner *tracer study* akan disajikan dalam bentuk grafik serta penghitungan rata-rata untuk beberapa kategori. Kemudian hasil uji *chi-square* ditampilkan dengan bentuk yang biasa digunakan dalam penelitian.

Input yang digunakan untuk uji *chi-square* antara lain adalah indeks prestasi kumulatif (IPK), jurusan alumni. Data tersebut didapatkan dari database yang ada di SCC. Sedangkan yang didapatkan dari kegiatan *tracer study* adalah waktu yang dihabiskan alumni untuk mendapatkan pekerjaan pertama, jenis perusahaan/instansi alumni bekerja, pendapatan dari pekerjaan utama, pendapatan dari lembur dan tips, dan pendapatan dari pekerjaan lain.

Data-data tersebut akan diuji menggunakan *chi-square* untuk mengukur hubungannya. *Output* yang dihasilkan adalah informasi tentang seberapa kuat hubungan antara kedua variabel uji yang disajikan melalui nilai *chi-square* hitung.

Kemudian untuk menentukan apakah variabel tersebut saling mempengaruhi, nilai *chi-square* hitung dibandingkan dengan nilai *chi-square* tabel yang didapatkan dari tabel distribusi *chi-square*. Apabila nilai *chi-square* hitung lebih besar dari nilai *chi-square* tabel, maka dapat ditarik kesimpulan bahwa variabel yang diuji memiliki hubungan yang signifikan.

Berikut ini adalah contoh kasus penghitungan uji *chi-square* yang digunakan untuk mencari hubungan antara indeks prestasi kumulatif (IPK) dengan penghasilan alumni dari pekerjaan utama. Sebagai contoh diketahui apabila data yang didapatkan dari kegiatan *tracer study* terhadap seratus orang alumni adalah:

Tabel 3.1 Data Hasil *Tracer Study*

Alumni dengan IPK 2 sampai dengan 2,49 yang penghasilannya $\leq$ 1 juta diketahui sebanyak 4 orang.
Alumni dengan IPK 2 sampai dengan 2,49 yang penghasilannya antara 1 sampai dengan 2,5 juta diketahui sebanyak 2 orang.
Alumni dengan IPK 2 sampai dengan 2,49 yang penghasilannya antara 2,5 sampai dengan 5 juta diketahui sebanyak 5 orang.
Alumni dengan IPK 2 sampai dengan 2,49 yang penghasilannya antara 5 sampai dengan 7,5 juta diketahui sebanyak 1 orang.
Alumni dengan IPK 2 sampai dengan 2,49 yang penghasilannya antara 7,5 sampai dengan 10 juta diketahui sebanyak 4 orang.
Alumni dengan IPK 2,5 sampai dengan 2,99 yang penghasilannya $\leq$ 1 juta diketahui sebanyak 2 orang.
Alumni dengan IPK 2,5 sampai dengan 2,99 yang penghasilannya antara 1 sampai dengan 2,5 juta diketahui sebanyak 5 orang.
Alumni dengan IPK 2,5 sampai dengan 2,99 yang penghasilannya antara 2,5 sampai dengan 5 juta diketahui sebanyak 2 orang.
Alumni dengan IPK 2,5 sampai dengan 2,99 yang penghasilannya antara 5 sampai dengan 7,5 juta diketahui sebanyak 5 orang.
Alumni dengan IPK 2,5 sampai dengan 2,99 yang penghasilannya antara 7,5 sampai dengan 10 juta diketahui sebanyak 12 orang.
Alumni dengan IPK 2,5 sampai dengan 2,99 yang penghasilannya antara 10 sampai dengan 12,5 juta diketahui sebanyak 1 orang.
Alumni dengan IPK 2,5 sampai dengan 2,99 yang penghasilannya $\geq$ 15 juta diketahui sebanyak 1 orang.
Alumni dengan IPK 3 sampai dengan 3,49 yang penghasilannya antara 1 sampai dengan 2,5 juta diketahui sebanyak 3 orang.
Alumni dengan IPK 3 sampai dengan 3,49 yang penghasilannya antara 2,5 sampai dengan 5 juta diketahui sebanyak 1 orang.



Alumni dengan IPK 3 sampai dengan 3,49 yang penghasilannya antara 5 sampai dengan 7,5 juta diketahui sebanyak 4 orang.
Alumni dengan IPK 3 sampai dengan 3,49 yang penghasilannya antara 7,5 sampai dengan 10 juta diketahui sebanyak 6 orang.
Alumni dengan IPK 3 sampai dengan 3,49 yang penghasilannya antara 10 sampai dengan 12,5 juta diketahui sebanyak 3 orang.
Alumni dengan IPK 3 sampai dengan 3,49 yang penghasilannya antara 12,5 sampai dengan 15 juta diketahui sebanyak 2 orang.
Alumni dengan IPK 3 sampai dengan 3,49 yang penghasilannya $\geq$ 15 juta diketahui sebanyak 6 orang.
Alumni dengan IPK 3,5 sampai dengan 4 yang penghasilannya antara 1 sampai dengan 2,5 juta diketahui sebanyak 1 orang.
Alumni dengan IPK 3,5 sampai dengan 4 yang penghasilannya antara 2,5 sampai dengan 5 juta diketahui sebanyak 2 orang.
Alumni dengan IPK 3,5 sampai dengan 4 yang penghasilannya antara 5 sampai dengan 7,5 juta diketahui sebanyak 6 orang.
Alumni dengan IPK 3,5 sampai dengan 4 yang penghasilannya antara 7,5 sampai dengan 10 juta diketahui sebanyak 9 orang.
Alumni dengan IPK 3,5 sampai dengan 4 yang penghasilannya antara 10 sampai dengan 12,5 juta diketahui sebanyak 4 orang.
Alumni dengan IPK 3,5 sampai dengan 4 yang penghasilannya antara 12,5 sampai dengan 15 juta diketahui sebanyak 5 orang.
Alumni dengan IPK 3,5 sampai dengan 4 yang penghasilannya antara $\geq$ 15 juta diketahui sebanyak 4 orang.

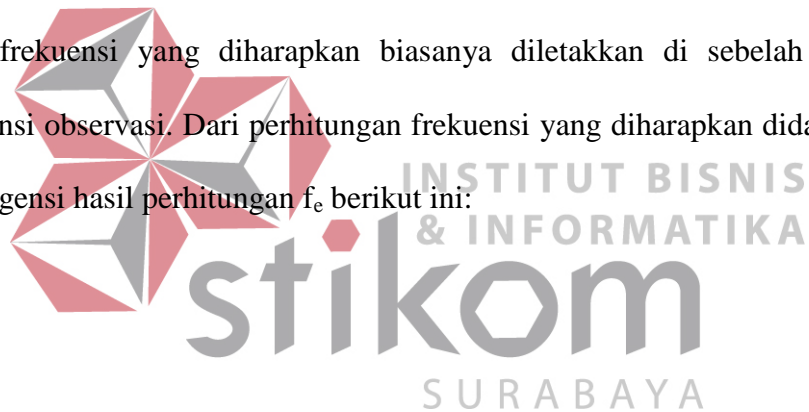
Langkah berikutnya adalah membuat uji hipotesis sesuai dengan variabel-variabel yang diuji. Dalam hal ini  $H_0$  adalah tidak terdapat hubungan yang signifikan antara indeks prestasi kumulatif (IPK) alumni dengan pendapatan penghasilan alumni yang didapatkan dari pekerjaan utama. Sedangkan  $H_1$ nya adalah terdapat hubungan yang signifikan antara indeks prestasi kumulatif (IPK) alumni dengan pendapatan penghasilan alumni yang didapatkan dari pekerjaan utama. Hipotesis ini nantinya akan digunakan untuk mengambil kesimpulan setelah membandingkan antara nilai *chi-square* yang didapatkan dari hasil perhitungan dan nilai *chi-square* yang didapatkan dari tabel distribusi *chi-square*.

Langkah selanjutnya adalah dengan membuat tabel kontingensi untuk data-data tersebut. Tabel kontingensi dapat dilihat pada tabel berikut ini:

Tabel 3.2 Tabel Kontingensi Hasil *Tracer Study*

IPK	Penghasilan dari Pekerjaan Utama								Total
	$\leq 1$	$1 < X < 2,5$	$2,5 < X < 5$	$5 < X < 7,5$	$7,5 < X < 10$	$10 < X < 12,5$	$12,5 < X < 15$	$\geq 15$	
2 - 2,49	4	2	5	1	4	0	0	0	16
2,5 - 2,99	2	5	2	5	12	1	0	1	28
3 - 3,49	0	3	1	4	6	3	2	6	25
3,5 - 4	0	1	2	6	9	4	5	4	31
<b>Total</b>	6	11	10	16	31	8	7	11	100

Setelah tabel kontingensi dibuat, langkah selanjutnya adalah mencari nilai frekuensi yang diharapkan dengan menggunakan rumus yang telah ditentukan. Nilai frekuensi yang diharapkan biasanya diletakkan di sebelah kanan nilai frekuensi observasi. Dari perhitungan frekuensi yang diharapkan didapatkan tabel kontingensi hasil perhitungan  $f_e$  berikut ini:



Tabel 3.3 Tabel Kontingensi dan Hasil Perhitungan  $f_e$

IPK	Penghasilan dari Pekerjaan Utama																Total	
	<= 1		1 < X < 2,5		2,5 < X < 5		5 < X < 7,5		7,5 < X < 10		10 < X < 12,5		12,5 < X < 15		>= 15			
	$f_0$	$f_e$	$f_0$	$f_e$	$f_0$	$f_e$	$f_0$	$f_e$	$f_0$	$f_e$	$f_0$	$f_e$	$f_0$	$f_e$	$f_0$	$f_e$	$f_0$	$f_e$
2 - 2,49	4	0	2	2	5	2	1	3	4	5	0	1	0	1	0	2	16	15
2,5 - 2,99	2	2	5	3	2	3	5	4	12	9	1	2	0	2	1	3	28	28
3 - 3,49	0	2	3	3	1	2	4	4	6	8	3	2	2	2	6	3	25	25
3,5 - 4	0	2	1	3	2	3	6	5	9	10	4	2	5	2	4	3	31	31
<b>Total</b>	<b>6</b>	<b>6</b>	<b>11</b>	<b>11</b>	<b>10</b>	<b>10</b>	<b>16</b>	<b>16</b>	<b>31</b>	<b>31</b>	<b>8</b>	<b>8</b>	<b>7</b>	<b>7</b>	<b>11</b>	<b>11</b>	<b>100</b>	<b>100</b>

Dari tabel tersebut nilai *chi-square* hitung bisa didapatkan dari penghitungan dengan rumus *chi-square*. Berikut ini adalah contoh perhitungan yang digunakan untuk mencari nilai *chi-square* hitung berdasarkan tabel kontingensi di atas:

$$\chi_{hitung}^2 = \left[ \frac{\sum (f_0 - f_e)^2}{f_e} \right]$$

$$\chi_{hitung}^2 = \frac{(4-0)^2}{0} + \frac{(2-2)^2}{2} + \frac{(0-2)^2}{2} + \frac{(0-2)^2}{2} + \frac{(2-2)^2}{2} + \frac{(5-3)^2}{3} + \frac{(3-3)^2}{3} + \frac{(1-3)^2}{3} + \frac{(5-2)^2}{2} + \frac{(2-3)^2}{3} + \frac{(1-2)^2}{2} + \frac{(2-3)^2}{3} + \frac{(1-3)^2}{3} + \frac{(5-4)^2}{4} + \frac{(4-4)^2}{4} + \frac{(6-5)^2}{5} + \frac{(4-5)^2}{5} + \frac{(12-9)^2}{9} + \frac{(6-8)^2}{8} + \frac{(9-10)^2}{10} + \frac{(0-1)^2}{1} + \frac{(1-2)^2}{2} + \frac{(3-2)^2}{2} + \frac{(4-2)^2}{2} + \frac{(0-1)^2}{1} + \frac{(0-2)^2}{2} + \frac{(2-2)^2}{2} + \frac{(5-2)^2}{2} + \frac{(0-2)^2}{2} + \frac{(1-3)^2}{3} + \frac{(6-3)^2}{3} + \frac{(4-3)^2}{3}$$

$$\chi_{hitung}^2 = 35,73024$$

Setelah *chi-square* hitung ditemukan, langkah selanjutnya adalah mendapatkan nilai *chi-square* tabel dari tabel distribusi *chi-square* dengan menggunakan rumus berikut ini:

$$\begin{aligned} Df &= (\text{Baris}-1)(\text{Kolom}-1) \\ &= (4-1)(8-1) \\ &= 21 \end{aligned}$$

Maka didapatkan  $\chi^2_{\text{tabel}} = 32,67056$ . Kemudian nilai tersebut dibandingkan dengan nilai *chi-square* hitung. Apabila nilai *chi-square* hitung lebih besar dari nilai *chi-square* tabel, maka  $H_0$  ditolak.

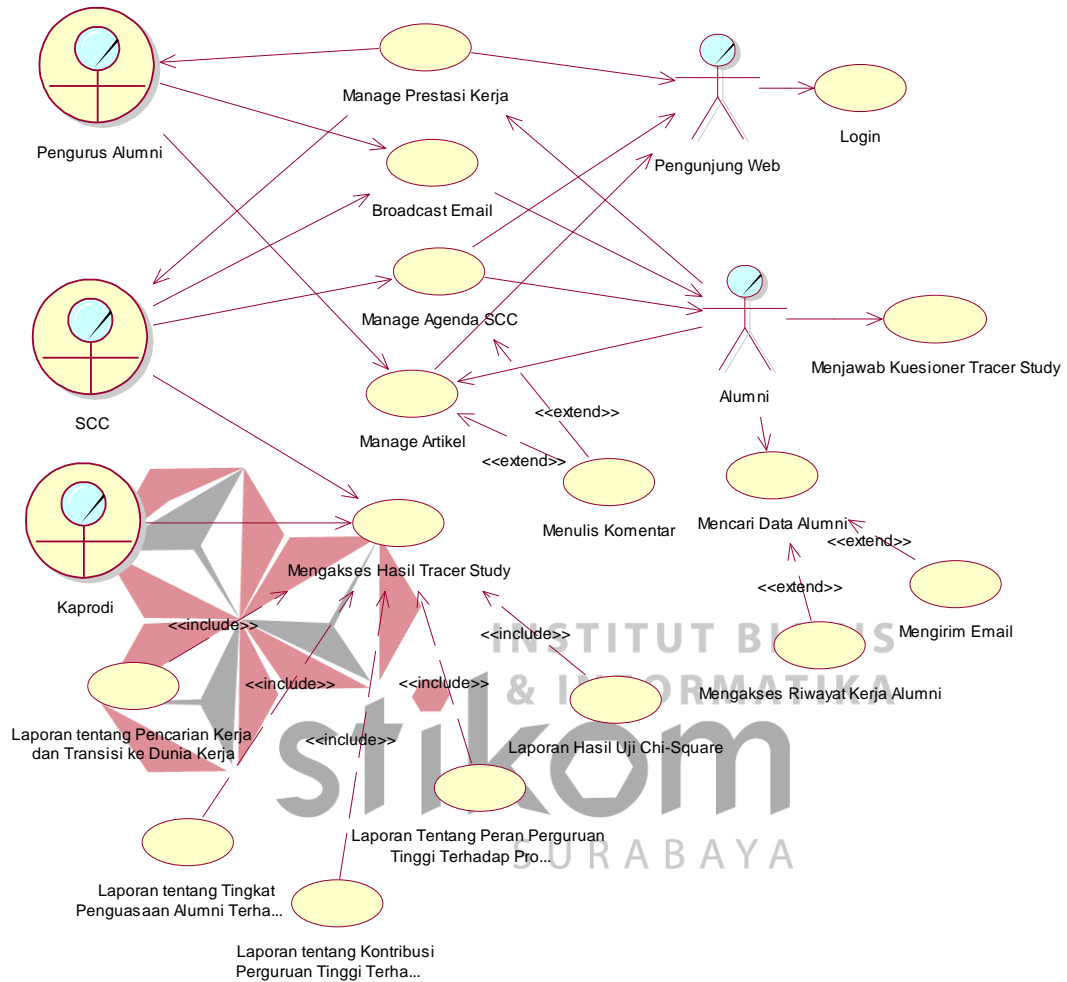
Dari hasil perhitungan tersebut didapatkan bahwa *chi-square* hitung lebih besar dari *chi-square* tabel. Maka  $H_0$  ditolak, artinya terdapat hubungan yang signifikan antara indeks prestasi kumulatif (IPK) dengan penghasilan alumni yang didapatkan dari pekerjaan utama.

### 3.2.1 Use Case Diagram

Diagram *use case* bisnis menunjukkan interaksi antara *use case* bisnis dan aktor bisnis. *Use case* bisnis mempresentasikan proses bisnis yang dilakukan, sedangkan aktor bisnis mempresentasikan peranan (roles) yang dimainkan pada aktivitas bisnis. Peranan yang dilakukan oleh seseorang atau sesuatu di dalam ruang lingkup bisnis disebut pekerja bisnis (Sholiq, 2010:20).

Seperti yang terlihat pada gambar 3.4, terdapat tiga pekerja bisnis yaitu SCC, pengurus alumni, dan kaprodi. Pengunjung *web* dan alumni berada di luar organisasi sehingga disebut sebagai aktor bisnis. Menerima hasil *tracer study* yang diinputkan oleh alumni diperankan oleh kaprodi dan SCC. Selain itu SCC juga berperan dalam *manage* agenda SCC dan *broadcast email*. Sedangkan

pengurus alumni berperan dalam *manage* artikel dan *broadcast email*. Peran alumni adalah menjawab kuesioner *tracer study*, *manage* prestasi kerja, *manage* artikel, dan mencari data alumni.



Gambar 3.4 Use Case Diagram Website Komunitas Alumni dan Tracer Study

Pengunjung *web* dapat mengakses konten *website* komunitas alumni seperti prestasi kerja alumni, agenda SCC, dan artikel-artikel yang dituliskan baik oleh pengurus alumni atau alumni. Pengunjung *web* dapat melakukan login untuk mendapatkan akses dan peran yang lainnya. Dari login yang dilakukan oleh pengunjung *web* dapat dibedakan apakah pengunjung tersebut adalah pekerja bisnis atau aktor bisnis.

### 3.2.2 Flow of Events Use Case Login

Proses login dibutuhkan untuk mengidentifikasi pengunjung *web* apakah dia termasuk sebagai alumni atau pengurus alumni atau SCC atau kaprodi. Proses login juga dibutuhkan untuk memberikan akses-akses tertentu sesuai dengan peran pengguna di dalam sistem.

Tabel 3.4 *Flow of Events* Login

Nama Use Case	Login	
<b>Kebutuhan terkait</b>	Proses untuk menentukan apakah seorang pengunjung website tersebut adalah alumni atau pengurus alumni atau SCC atau ketua program studi. Masing-masing jenis pengguna memiliki <i>username</i> dan <i>password</i> yang tersimpan pada tabel yang berbeda. <i>Username</i> dan <i>password</i> tersebut harus verifikasi oleh sistem agar sistem dapat memastikan bahwa pengguna yang login benar-benar pengguna yang berhak atas fungsi-fungsi tertentu yang ada dalam sistem.	
<b>Tujuan</b>	Menangani proses login pengunjung <i>website</i> .	
<b>Prasyarat</b>	Tidak ada	
<b>Kondisi akhir Sukses</b>	Pengguna yang berhasil melakukan login dapat mengakses menu-menu yang telah disediakan menurut peran pengguna tersebut.	
<b>Kondisi akhir gagal</b>	Apabila login yang dilakukan oleh pengguna gagal, maka pengguna harus menginputkan kembali <i>username</i> dan <i>password</i> yang mereka miliki dengan benar.	
<b>Aktor utama</b>	Pengunjung web	
<b>Aktor sekunder</b>	Tidak Ada	
<b>Pemicu</b>	Aktor pengunjung <i>web</i> menekan tautan login.	
<b>Alur utama</b>	<b>Langkah</b>	<b>Aksi</b>
	1	Membuka halaman utama <i>website</i> komunitas alumni.
	2	Menekan tautan login
	3	Menampilkan halaman login dengan inputan yang diperlukan adalah <i>username</i> dan <i>password</i> .
	4	Mengisi <i>username</i> , <i>password</i> , dan memilih jenis pengguna kemudian menekan tombol login.
	5	Mengirimkan <i>username</i> , <i>password</i> , dan data jenis pengguna ke <i>server</i> .
	6	Melakukan koneksi ke database.
	7	Mengambil <i>password</i> berdasarkan <i>username</i> yang telah diinputkan.

	8	Melakukan verifikasi <i>password</i> .
	9	Memasang <i>session</i> login dengan nilai <i>true</i> .
	10	Mengarahkan pengunjung web ke halaman utama.
	11	<i>Use case</i> berakhir.
<b>Alur perluasan</b>	<b>Langkah</b>	<b>Aksi Percabangan</b>
	8.1	Menampilkan pesan login gagal, kemudian aktor harus kembali ke langkah ke-4.

### 3.2.3 Flow of Events Menjawab Kuesioner Tracer Study

*Flow of events* menjawab kuesioner *tracer study* menjelaskan tentang alur kejadian yang dilakukan oleh alumni dalam menjawab kuesioner *tracer study*. Kuesioner *tracer study* terdiri dari 4 (empat) bagian yang harus diisi. Bagian yang telah diisi tidak akan dapat dibuka lagi oleh pengguna yang telah mengisikan bagian tersebut.

Tabel 3.5 *Flow of Events* Menjawab Kuesioner *Tracer Study*

Nama Use Case	Menjawab Kuesioner Tracer Study
<b>Kebutuhan terkait</b>	Pengumpulan data-data yang akan digunakan dalam proses membuat laporan berupa statistik deskriptif dan sebagai inputan untuk uji <i>chi-square</i> . Terdapat 4 bagian kuesioner yang harus dijawab oleh alumni. Setiap bagian memiliki kategori yang berbeda. Dan bagian yang telah diisi oleh alumni tidak dapat dibuka kembali oleh alumni yang telah mengisikan bagian kuesioner tersebut.
<b>Tujuan</b>	Mendapatkan data sebagai inputan untuk proses pembuatan laporan statistik deskriptif dan uji <i>chi-square</i> .
<b>Prasyarat</b>	Tidak ada
<b>Kondisi akhir Sukses</b>	Sistem akan menampilkan pesan terima kasih untuk alumni yang telah selesai mengisi semua kuesioner <i>tracer study</i> .
<b>Kondisi akhir gagal</b>	Semua <i>field</i> yang ada pada kuesioner <i>tracer study</i> bersifat <i>mandatory</i> atau harus diisi. Maka apabila pengguna belum mengisikan jawaban untuk sebuah pertanyaan maka sistem tidak akan menyimpan data tersebut.
<b>Aktor utama</b>	Alumni
<b>Aktor sekunder</b>	SCC dan Kaprodi

<b>Pemicu</b>	Aktor pengunjung <i>web</i> berhasil melakukan login sebagai alumni atau aktor alumni menekan tautan <i>Tracer Study</i> .	
<b>Alur utama</b>	<b>Langkah</b>	<b>Aksi</b>
	1	Membuka halaman utama <i>website</i> komunitas alumni.
	2	Menekan tautan <i>Tracer Study</i>
	3	Melakukan koneksi ke dalam database
	4	Memeriksa apakah alumni tersebut telah mengisi kuesioner.
	5	Menampilkan halaman kuesioner <i>tracer study</i> .
	6	Alumni mengisikan data yang diperlukan dalam kuesioner <i>tracer study</i> tersebut.
	7	Alumni menekan tombol <i>simpan</i> .
	8	Mengirimkan jawaban alumni ke server.
	9	Menyimpan jawaban alumni ke dalam database.
	10	Mengarahkan pengunjung <i>web</i> ke halaman pesan terima kasih.
	11	<i>Use case</i> berakhir.
<b>Alur perluasan</b>	<b>Langkah</b>	<b>Aksi Percabangan</b>
	7.1	Menampilkan pesan kepada alumni bahwa alumni belum melengkapi data yang diperlukan. Setelah alumni mengkonfirmasi pesan kembali ke langkah yang ke-6.
	10.1	Memeriksa apakah alumni telah mengisi seluruh kuesioner <i>tracer study</i> . Apabila belum, maka arahkan ke halaman bagian kuesioner yang belum diisi dan kembali ke langkah yang ke-5.

### 3.2.4 *Flow of Events Manage Prestasi Kerja*

*Manage* prestasi kerja dilakukan oleh alumni. Alumni dapat menambah dan menghapus data prestasi kerjanya melalui halaman prestasi kerja saya. Kemudian prestasi kerja tersebut akan ditampilkan kepada SCC dan pengurus alumni sebagai rekapitulasi prestasi kerja.



Tabel 3.6 *Flow of Events Manage Prestasi Kerja*

Nama Use Case	<i>Manage Prestasi Kerja</i>	
<b>Kebutuhan terkait</b>	Pengumpulan data-data yang akan digunakan dalam proses membuat laporan rekapitulasi prestasi kerja.	
<b>Tujuan</b>	Mendapatkan data prestasi kerja sebagai acuan dalam menentukan alumni yang berhak mendapatkan penghargaan atas prestasinya di dalam dunia kerja.	
<b>Prasyarat</b>	Tidak ada	
<b>Kondisi akhir Sukses</b>	Data prestasi kerja akan tersimpan ke dalam <i>database</i> dengan indikasi prestasi yang telah diinputkan muncul pada daftar prestasi kerja alumni.	
<b>Kondisi akhir gagal</b>	Field-field yang harus diisi belum diisi oleh alumni maka sistem akan menampilkan pesan bahwa alumni harus melengkapi data yang dibutuhkan.	
<b>Aktor utama</b>	Alumni	
<b>Aktor sekunder</b>	SCC, Pengurus Alumni, dan Pengunjung <i>Web</i>	
<b>Pemicu</b>	Aktor alumni menekan tautan prestasi kerja saya.	
<b>Alur utama</b>	<b>Langkah</b>	<b>Aksi</b>
	1	Membuka halaman utama <i>website</i> komunitas alumni.
	2	Menekan tautan Prestasi Kerja Saya
	3	Mengarahkan alumni ke halaman prestasi kerja
	4	Alumni mengisikan data-data prestasi kerja yang diperlukan.
	5	Alumni menekan tombol simpan.
	6	Mengirimkan data-data yang diinputkan oleh alumni ke <i>server</i> .
	7	Menyimpan data prestasi kerja alumni ke dalam database.
	8	Mengarahkan alumni ke halaman Prestasi Kerja Saya.
9	<i>Use case</i> berakhir.	
<b>Alur perluasan</b>	<b>Langkah</b>	<b>Aksi Percabangan</b>
	4.1	Alumni menghapus data dengan menekan tautan hapus. Kemudian sistem akan menampilkan pesan konfirmasi kepada alumni apakah benar-benar akan menghapus data prestasinya. Setelah alumni mengkonfirmasi pesan tersebut kembali ke langkah 3.
5.1	Memeriksa apakah alumni telah mengisi seluruh data prestasi kerja yang diperlukan. Apabila belum, maka tampilkan pesan bahwa data yang diperlukan belum lengkap kemudian	

		kembali ke langkah yang ke-4.
--	--	-------------------------------

### 3.2.5 Flow of Events Broadcast Email

Berikut ini adalah *flow of events* untuk *use case broadcast email*. Broadcast email digunakan oleh SCC atau pengurus alumni untuk mengirimkan email secara massal kepada para alumni. SCC atau pengurus alumni dapat mengirimkan email ke para alumni berdasarkan penyaringan data yang diinginkan.

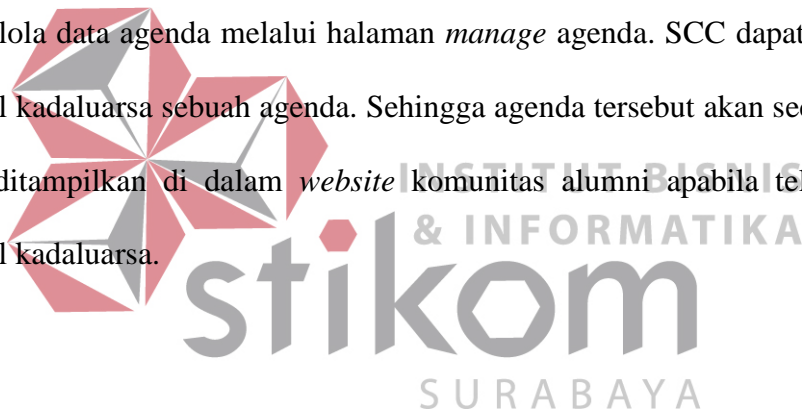
Tabel 3.7 *Flow of Events Broadcast Email*

Nama Use Case	Broadcast Email	
<b>Kebutuhan terkait</b>	Mengirimkan email secara massal kepada alumni.	
<b>Tujuan</b>	Mengumumkan kegiatan <i>tracer study</i> dan keperluan pemberitaan secara massal yang lainnya.	
<b>Prasyarat</b>	Tidak ada	
<b>Kondisi akhir Sukses</b>	Tampil pesan bahwa email telah terkirim kepada alumni berdasarkan penyaringan data yang ditentukan oleh aktor.	
<b>Kondisi akhir gagal</b>	<ul style="list-style-type: none"> <li>• Penampilan pesan error karena terdapat data email yang tidak valid yang diinputkan oleh alumni.</li> <li>• Penampilan pesan error karena konfigurasi email yang tidak tepat.</li> </ul>	
<b>Aktor utama</b>	SCC dan pengurus alumni	
<b>Aktor sekunder</b>	Alumni	
<b>Pemicu</b>	Aktor SCC atau pengurus alumni menekan tautan <i>broadcast email</i> .	
<b>Alur utama</b>	<b>Langkah</b>	<b>Aksi</b>
	1	Menekan tautan Broadcast Email.
	2	Mengarahkan aktor ke halaman broadcast email.
	3	Aktor menentukan penyaringan data berdasarkan jurusan alumni dan tahun lulusan alumni.
	4	Aktor mengisikan subjek email dan isi email.
	5	Aktor menekan tombol kirim.
	6	Mengirimkan subjek email dan isi email yang diinputkan oleh aktor ke <i>server</i> .
	7	Melakukan koneksi database
8	Membaca data email yang ada di dalam	

		database berdasarkan kriteria penyaringan yang dilakukan aktor.
	9	Mengirimkan email satu per satu berdasarkan hasil kueri.
	10	<i>Use case</i> berakhir.
<b>Alur perluasan</b>	<b>Langkah</b>	<b>Aksi Percabangan</b>
	5.1	Memeriksa apakah alumni telah mengisi data yang diperlukan untuk pengiriman email. Apabila belum, maka tampilkan pesan bahwa data yang diperlukan belum lengkap kemudian kembali ke langkah yang ke-4.

### 3.2.6 *Flow of Events Manage Agenda SCC*

Dalam mempublikasikan agenda-agendanya kepada alumni, SCC dapat mengelola data agenda melalui halaman *manage* agenda. SCC dapat menentukan tanggal kadaluarsa sebuah agenda. Sehingga agenda tersebut akan secara otomatis tidak ditampilkan di dalam *website* komunitas alumni apabila telah melewati tanggal kadaluarsa.



Tabel 3.8 *Flow of Events Manage Agenda SCC*

Nama Use Case	<i>Manage Agenda SCC</i>	
<b>Kebutuhan terkait</b>	Mengelola data agenda SCC yang akan ditampilkan pada <i>website</i> komunitas alumni.	
<b>Tujuan</b>	Mempublikasikan agenda-agenda SCC kepada para alumni melalui <i>website</i> komunitas alumni.	
<b>Prasyarat</b>	Tidak ada	
<b>Kondisi akhir Sukses</b>	Data agenda yang diinputkan tersimpan ke dalam <i>database</i> dan ditampilkan pada <i>website</i> komunitas alumni berdasarkan tanggal kadaluarsanya.	
<b>Kondisi akhir gagal</b>	<i>Field-field</i> yang dibutuhkan untuk mengisi sebuah <i>form</i> agenda belum terpenuhi sehingga sistem akan menampilkan pesan kepada aktor dan data agenda tidak akan disimpan ke dalam <i>database</i> .	
<b>Aktor utama</b>	SCC	
<b>Aktor sekunder</b>	Alumni dan Pengunjung <i>Web</i>	
<b>Pemicu</b>	Aktor SCC menekan tautan <i>Manage Agenda</i> .	
<b>Alur utama</b>	<b>Langkah</b>	<b>Aksi</b>
	1	Menekan tautan <i>Manage Agenda</i> .
	2	Mengarahkan aktor ke halaman manage agenda.
	3	Aktor mengisi data-data yang diperlukan untuk membuat sebuah agenda seperti judul agenda, deskripsi singkat, isi agenda, dan tanggal kadaluarsa.
	4	Aktor menekan tombol simpan.
	5	Mengirimkan data-data agenda yang telah diinputkan oleh aktor ke <i>server</i> .
	6	Melakukan koneksi database.
	7	Menyimpan data agenda ke dalam database.
	8	Membaca data agenda yang ada di dalam database.
	9	Mengarahkan aktor ke halaman manage agenda.
	10	<i>Use case</i> berakhir.
<b>Alur perluasan</b>	<b>Langkah</b>	<b>Aksi Percabangan</b>
	2.1	Menekan tombol edit agenda untuk mengubah data agenda yang ada di dalam database.
	2.2	Mengarahkan pengguna kepada halaman untuk mengedit agenda.
	2.3	Mengubah data-data agenda yang diperlukan.
	2.4	Menekan tombol simpan.
	2.5	Mengirimkan data-data agenda ke server.
	2.6	Melakukan koneksi database

	2.7	Menyimpan data-data yang dimasukkan aktor ke dalam database.
	2.8	Mengarahkan aktor ke halaman manage agenda.
	4.1	Memeriksa apakah alumni telah mengisi data agenda yang diperlukan. Apabila belum, maka tampilkan pesan bahwa data yang diperlukan belum lengkap kemudian kembali ke langkah yang ke-3.

### 3.2.7 Flow of Events Manage Artikel

Alumni dapat menulis artikel yang dapat ditampilkan pada *website* komunitas alumni dan mengelola artikel-artikel tersebut. Pengurus alumni dapat mengelola artikel-artikel yang ditulis oleh alumni.

Tabel 3.9 Flow of Events Manage Artikel

Nama Use Case	Manage Artikel	
<b>Kebutuhan terkait</b>	Mengelola artikel yang akan ditampilkan pada <i>website</i> komunitas alumni.	
<b>Tujuan</b>	Memungkinkan alumni untuk dapat saling berbagi informasi dengan alumni yang lainnya.	
<b>Prasyarat</b>	Tidak ada	
<b>Kondisi akhir Sukses</b>	Data artikel yang diinputkan tersimpan ke dalam <i>database</i> dan ditampilkan pada <i>website</i> komunitas alumni.	
<b>Kondisi akhir gagal</b>	<i>Field-field</i> yang dibutuhkan untuk mengisi sebuah <i>form</i> artikel belum terpenuhi sehingga sistem akan menampilkan pesan kepada aktor dan data artikel tidak akan disimpan ke dalam <i>database</i> .	
<b>Aktor utama</b>	Pengurus Alumni dan Alumni	
<b>Aktor sekunder</b>	Tidak ada	
<b>Pemicu</b>	Aktor pengurus alumni atau alumni menekan tautan <i>Manage Artikel</i> .	
<b>Alur utama</b>	<b>Langkah</b>	<b>Aksi</b>
	1	Menekan tautan <i>Manage Artikel</i> .
	2	Mengarahkan aktor ke halaman manage artikel.
	3	Aktor mengisi data-data yang diperlukan untuk membuat sebuah artikel seperti judul artikel, deskripsi singkat, dan isi artikel.
	4	Aktor menekan tombol simpan.

	5	Mengirimkan data-data artikel yang telah diinputkan oleh aktor ke <i>server</i> .
	6	Melakukan koneksi database.
	7	Menyimpan data artikel ke dalam database.
	8	Membaca data artikel yang ada di dalam database.
	9	Mengarahkan aktor ke halaman <i>manage</i> artikel.
	10	<i>Use case</i> berakhir.
<b>Alur perluasan</b>	<b>Langkah</b>	<b>Aksi Percabangan</b>
	2.1	Menekan tombol edit artikel untuk mengubah data artikel yang ada di dalam database.
	2.2	Mengarahkan pengguna kepada halaman untuk mengedit artikel.
	2.3	Mengubah data-data artikel yang diperlukan.
	2.4	Menekan tombol simpan.
	2.5	Mengirimkan data-data artikel ke server.
	2.6	Melakukan koneksi database
	2.7	Menyimpan data-data yang dimasukkan aktor ke dalam database.
	2.8	Mengarahkan aktor ke halaman <i>manage</i> artikel.
	4.1	Memeriksa apakah alumni telah mengisi data artikel yang diperlukan. Apabila belum, maka tampilkan pesan bahwa data yang diperlukan belum lengkap kemudian kembali ke langkah yang ke-3.

### 3.2.8 *Flow of Events* Mencari Data Alumni

Alumni dapat mencari data alumni lainnya melalui halaman pencarian alumni. Data yang ditampilkan antara lain adalah jurusan, tahun lulus, dan riwayat kerja alumni tersebut. Selain itu, melalui halaman pencarian alumni, alumni dapat mengirimkan email kepada alumni yang berhasil ditemukan pada halaman hasil pencarian alumni.

Tabel 3.10 *Flow of Events* Mencari Data Alumni

<b>Nama Use Case</b>	<b>Mencari Data Alumni</b>
----------------------	----------------------------

<b>Kebutuhan terkait</b>	Mencari data alumni berdasarkan nama, jurusan, dan tahun lulus alumni tersebut.	
<b>Tujuan</b>	Memungkinkan alumni untuk dapat saling berkomunikasi.	
<b>Prasyarat</b>	Tidak ada	
<b>Kondisi akhir sukses</b>	Tampil tabel hasil pencarian alumni berdasarkan nama, jurusan, dan tahun lulus yang diinputkan oleh alumni.	
<b>Kondisi akhir gagal</b>	Kueri tidak dapat menemukan data yang dicari.	
<b>Aktor utama</b>	Alumni	
<b>Aktor sekunder</b>	Tidak ada.	
<b>Pemicu</b>	Aktor alumni menekan tautan pencarian alumni.	
<b>Alur utama</b>	<b>Langkah</b>	<b>Aksi</b>
	1	Menekan tautan pencarian alumni.
	2	Mengarahkan aktor ke halaman pencarian alumni.
	3	Aktor memasukkan nama alumni, dan penyaringan data berdasarkan jurusan alumni dan tahun lulusan alumni.
	4	Menekan tombol cari.
	5	Mengirimkan kriteria pencarian ke server.
	6	Melakukan koneksi ke database
	7	Menjalankan kueri berdasarkan kriteria-kriteria pencarian yang telah dimasukkan.
	8	Membaca data alumni yang ada di dalam database.
	9	Menampilkan data hasil pencarian ke halaman hasil pencarian alumni.
10	<i>Use case</i> berakhir.	
<b>Alur perluasan</b>	<b>Langkah</b>	<b>Aksi Percabangan</b>
	5.1	Memeriksa apakah alumni telah mengisi nama minimal 3 (tiga) karakter. Apabila belum maka sistem akan menampilkan pesan peringatan kepada alumni untuk menginputkan nama.

### 3.2.9 Flow of Events Mengakses Hasil Tracer Study

Halaman hasil *tracer study* menyediakan data-data yang didapatkan dari kegiatan *tracer study*. Data-data tersebut disajikan dalam bentuk grafik pai dan *bar*. Kemudian untuk data seperti Indeks Prestasi Kumulatif (IPK), jurusan, penghasilan dan sebagainya digunakan untuk uji *chi-square*. Halaman ini juga memungkinkan SCC dan ketua program studi untuk mengetahui sampel yang

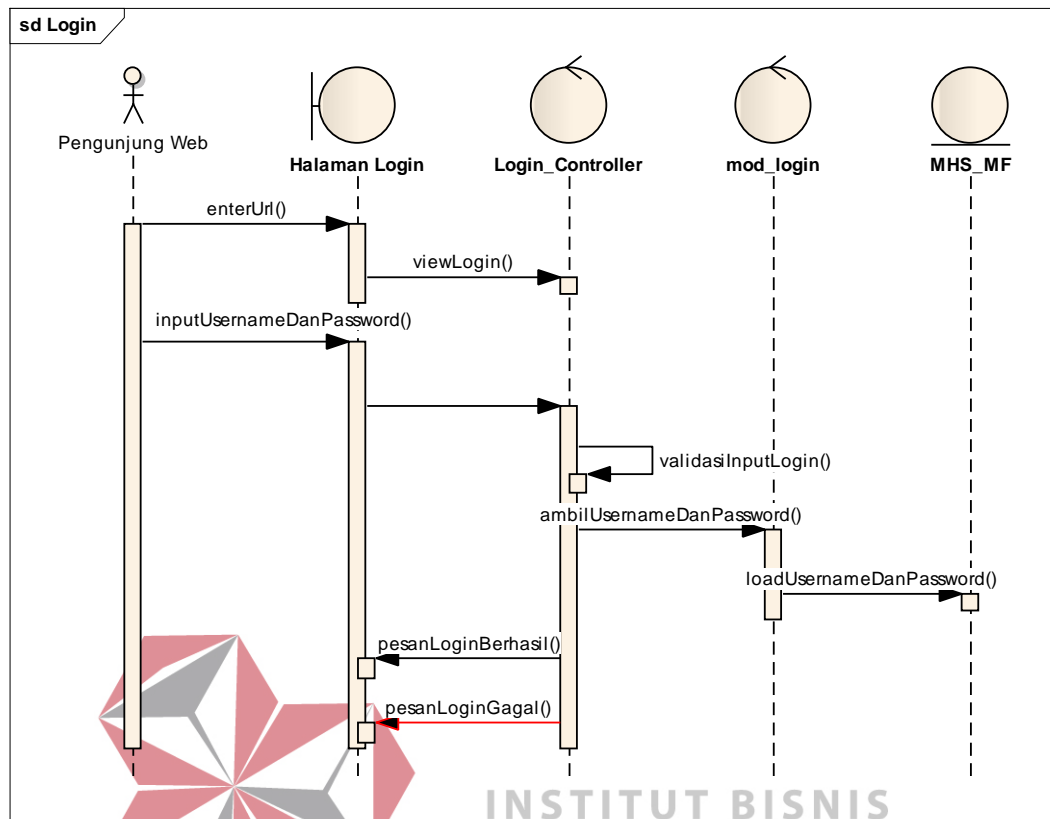
ideal untuk setiap jurusan berdasarkan jumlah populasi mereka. Penghitungan sampel tersebut menggunakan metode sampel berstrata.

Tabel 3.11 *Flow of Events* Mengakses Hasil *Tracer Study*

Nama Use Case	Mengakses Hasil <i>Tracer Study</i>	
<b>Kebutuhan terkait</b>	Menyajikan data hasil <i>tracer study</i> dengan statistik deskriptif dan uji <i>chi-square</i> .	
<b>Tujuan</b>	Memberikan informasi yang mendukung manajemen dalam mengambil keputusan terkait dengan aspek-aspek pendukung proses pembelajaran di STIKOM Surabaya.	
<b>Prasyarat</b>	Tidak ada	
<b>Kondisi akhir sukses</b>	Grafik-grafik ter-render dengan sukses.	
<b>Kondisi akhir gagal</b>	Apabila kueri tidak menghasilkan data maka sistem akan menampilkan pesan bahwa data kosong.	
<b>Aktor utama</b>	SCC dan ketua program studi	
<b>Aktor sekunder</b>	Tidak ada.	
<b>Pemicu</b>	Aktor alumni menekan tautan hasil <i>tracer study</i> .	
<b>Alur utama</b>	<b>Langkah</b>	<b>Aksi</b>
	1	Menekan tautan hasil <i>tracer study</i> .
	2	Mengarahkan aktor ke halaman hasil <i>tracer study</i> .
	3	Aktor memilih penyaringan data berdasarkan jurusan dan tahun lulusan alumni (responden).
	4	Menekan tombol <i>filter</i> .
	5	Mengirimkan kriteria pencarian ke server.
	6	Melakukan koneksi ke database
	7	Menjalankan kueri berdasarkan kriteria-kriteria pencarian yang telah dimasukkan.
	8	Membaca data hasil <i>tracer study</i> yang ada di dalam <i>database</i> .
	9	Menampilkan hasil kueri dan perhitungan ke dalam bentuk grafik dan langkah-langkah pengujian <i>chi-square</i> .
	10	<i>Use case</i> berakhir.
<b>Alur perluasan</b>	<b>Langkah</b>	<b>Aksi Percabangan</b>
	3.1	Apabila aktor yang tidak memilih jenis penyaringan data, maka seluruh data yang ada pada database dari hasil <i>tracer study</i> akan diproses dalam membuat grafik dan perhitungan <i>chi-square</i> .



### 3.2.10 Sequence Diagram Login



Gambar 3.5 Sequence Diagram Login

Pengunjung *web* yang akan login dimulai dari halaman awal di mana pengunjung memasukkan alamat *web* ke dalam peramban internet. Setelah itu *controller* login membaca halaman *view*. Kemudian pengunjung memasukkan *username* dan *password* ke dalam halaman login. *Username* dan *password* yang dimasukkan oleh pengunjung divalidasi di dalam *controller* login.

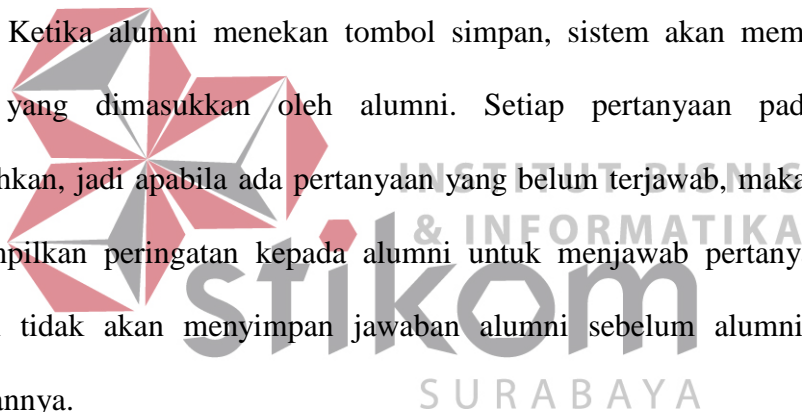
Pada saat memvalidasi *username* dan *password* yang dimasukkan oleh pengunjung, *controller* login meminta data *username* dan *password* dari login model. Login model mengambil data dari tabel pengguna, kemudian mengirimkannya kepada *controller* login. Di dalam *controller* login, data yang didapatkan dari login model dimasukkan ke dalam sebuah variabel. Kemudian variabel tersebut divalidasi dengan input yang telah dimasukkan oleh pengguna.

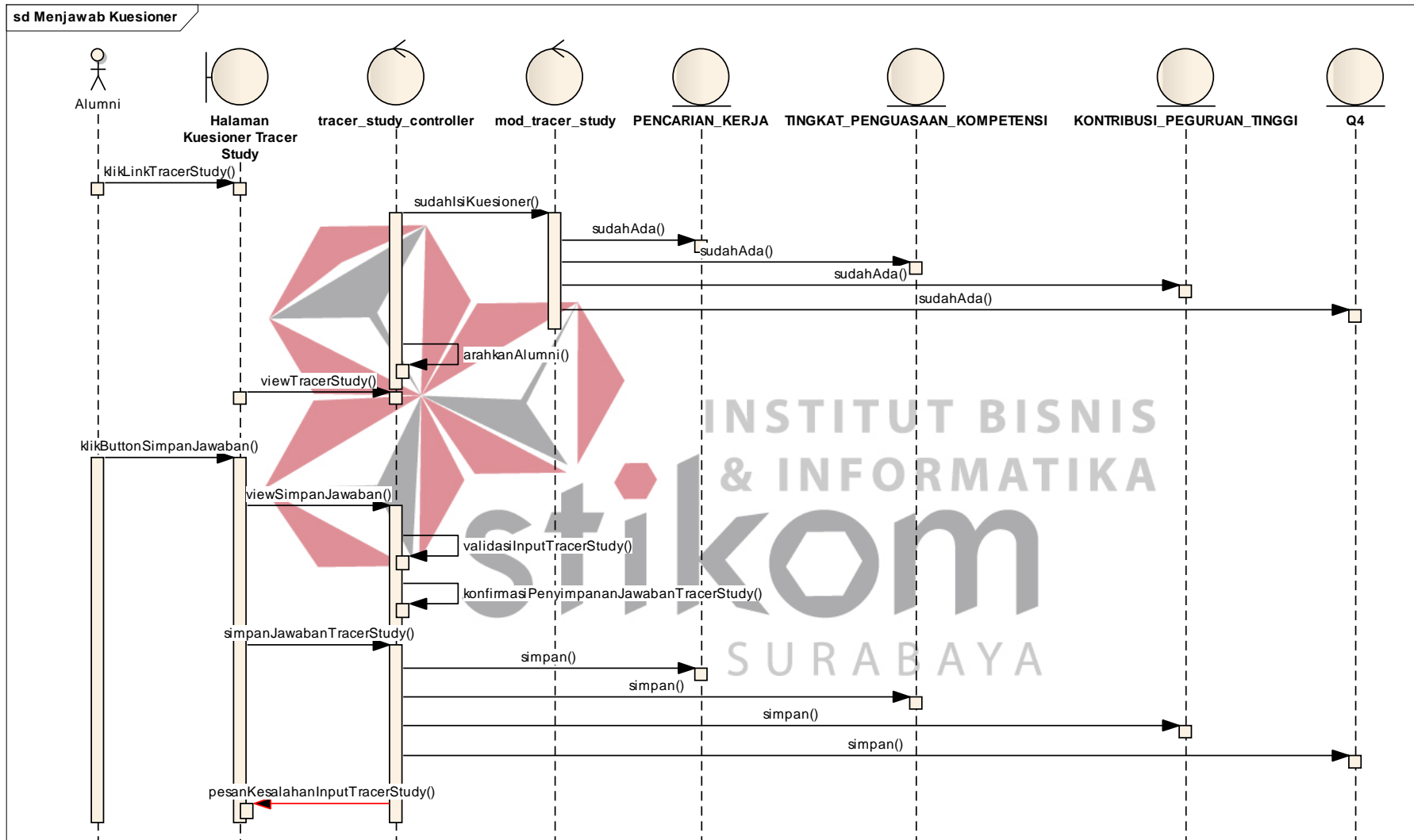
Apabila nilai yang diinputkan pengunjung dan variabel tersebut sesuai maka pengunjung berhasil login. Apabila validasi tidak berhasil, maka sistem akan menampilkan pesan kepada pengguna bahwa login gagal.

### **3.2.11 Sequence Diagram Menjawab Kuesioner Tracer Study**

Setiap alumni yang telah login ke dalam sistem dapat menjawab kuesioner *tracer study* dengan menekan tautan *tracer study*. Sistem akan mengirimkan halaman kuesioner *tracer study* untuk dijawab oleh alumni. Kemudian alumni menjawab setiap pertanyaan yang diberikan di dalam kuesioner tersebut dan menekan tombol simpan.

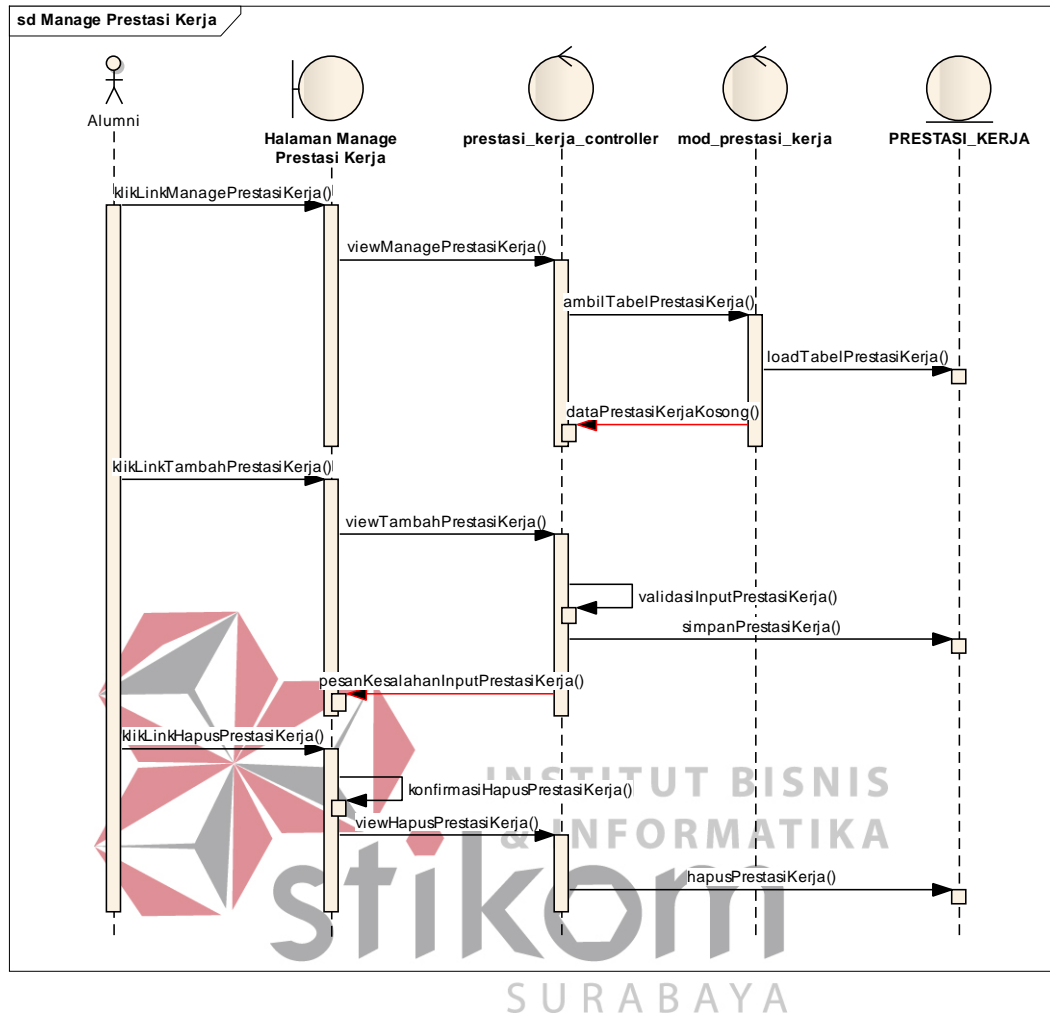
Ketika alumni menekan tombol simpan, sistem akan memeriksa setiap input yang dimasukkan oleh alumni. Setiap pertanyaan pada kuesioner dibutuhkan, jadi apabila ada pertanyaan yang belum terjawab, maka sistem akan menampilkan peringatan kepada alumni untuk menjawab pertanyaan tersebut. Sistem tidak akan menyimpan jawaban alumni sebelum alumni melengkapi jawabannya.





Gambar 3.6 Sequence Diagram Menjawab Kuesioner Tracer Study

### 3.2.12 Sequence Diagram Manage Prestasi Kerja



Gambar 3.7 Sequence Diagram Manage Prestasi Kerja

Alumni menambah, dan menghapus data prestasi kerja melalui halaman *manage* prestasi kerja. *Controller* prestasi kerja akan meminta data prestasi kerja kepada model prestasi kerja. Kemudian model prestasi kerja mengambil data dari tabel prestasi kerja dari dalam *database*. Setelah itu data yang didapatkan dari model prestasi kerja ditampilkan pada halaman *manage* prestasi kerja. Apabila belum ada data prestasi kerja di dalam *database* maka sistem akan menampilkan pesan kepada alumni bahwa data prestasi kerja masih kosong.

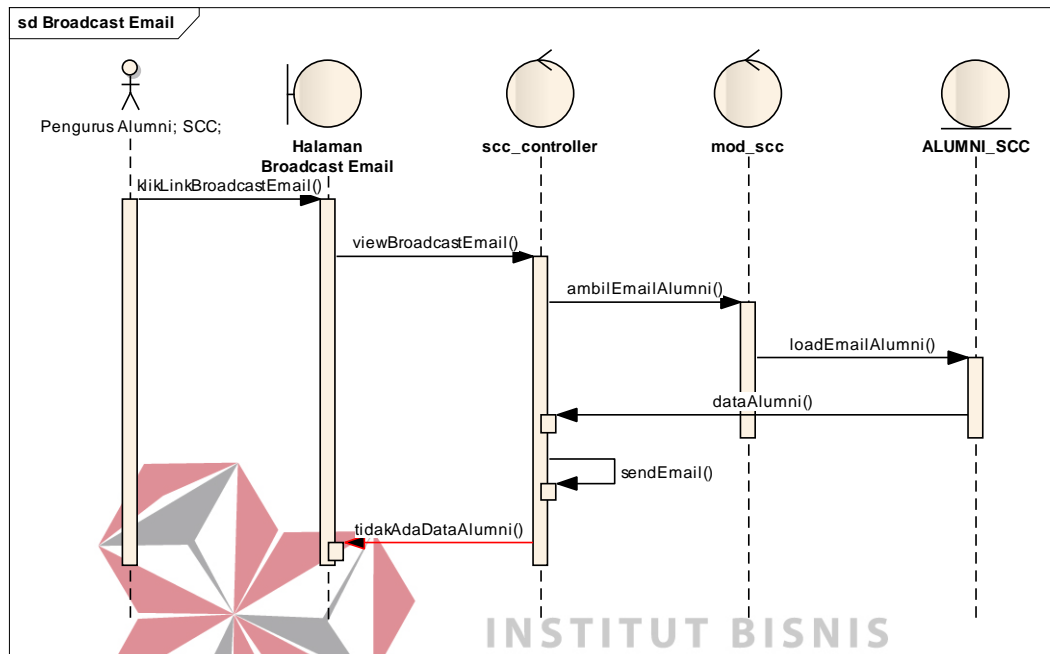
Melalui halaman *manage* prestasi kerja alumni dapat menambah data prestasi kerjanya. Alumni memasukkan data-data prestasi kerja yang diperlukan dan menekan tombol simpan. Kemudian sistem akan memeriksa data prestasi kerja yang dimasukkan oleh alumni. Apabila data tersebut sesuai dengan ketentuan validasi, maka sistem akan menyimpannya ke dalam tabel prestasi kerja.

Alumni menghapus data prestasi kerja melalui halaman ini. Ketika alumni menekan tombol hapus, sistem akan menampilkan pesan konfirmasi kepada alumni apakah benar-benar akan menghapus data prestasi kerja tersebut. Apabila alumni mengonfirmasi penghapusan maka sistem akan menghapus data prestasi kerja alumni dari dalam *database*.

### 3.2.13 *Sequence Diagram Broadcast Email*

*Broadcast* email dilakukan oleh pengurus alumni dan SCC dengan menekan tautan *broadcast* email. Kemudian *controller* broadcast email akan menampilkan halaman *broadcast* email yang berisi *form* untuk mengirim email ke banyak alamat. Pengurus alumni dan SCC dapat menentukan tujuan pengiriman tersebut berdasarkan jurusan dan tahun lulusan alumni. Pengurus alumni dan SCC mengisi subjek email dan isi email kemudian menekan tombol kirim. Kemudian *controller* broadcast email akan meminta data email kepada *broadcast* email model dengan beberapa parameter yang telah diberikan. Kemudian *broadcast* email model mengembalikan data email kepada *controller* broadcast email. Di dalam *controller* broadcast email, data yang didapatkan dari *broadcast* email model dimasukkan ke dalam variabel *array*. Kemudian dari variabel tersebut *controller* broadcast email mengirimkan email satu per satu kepada

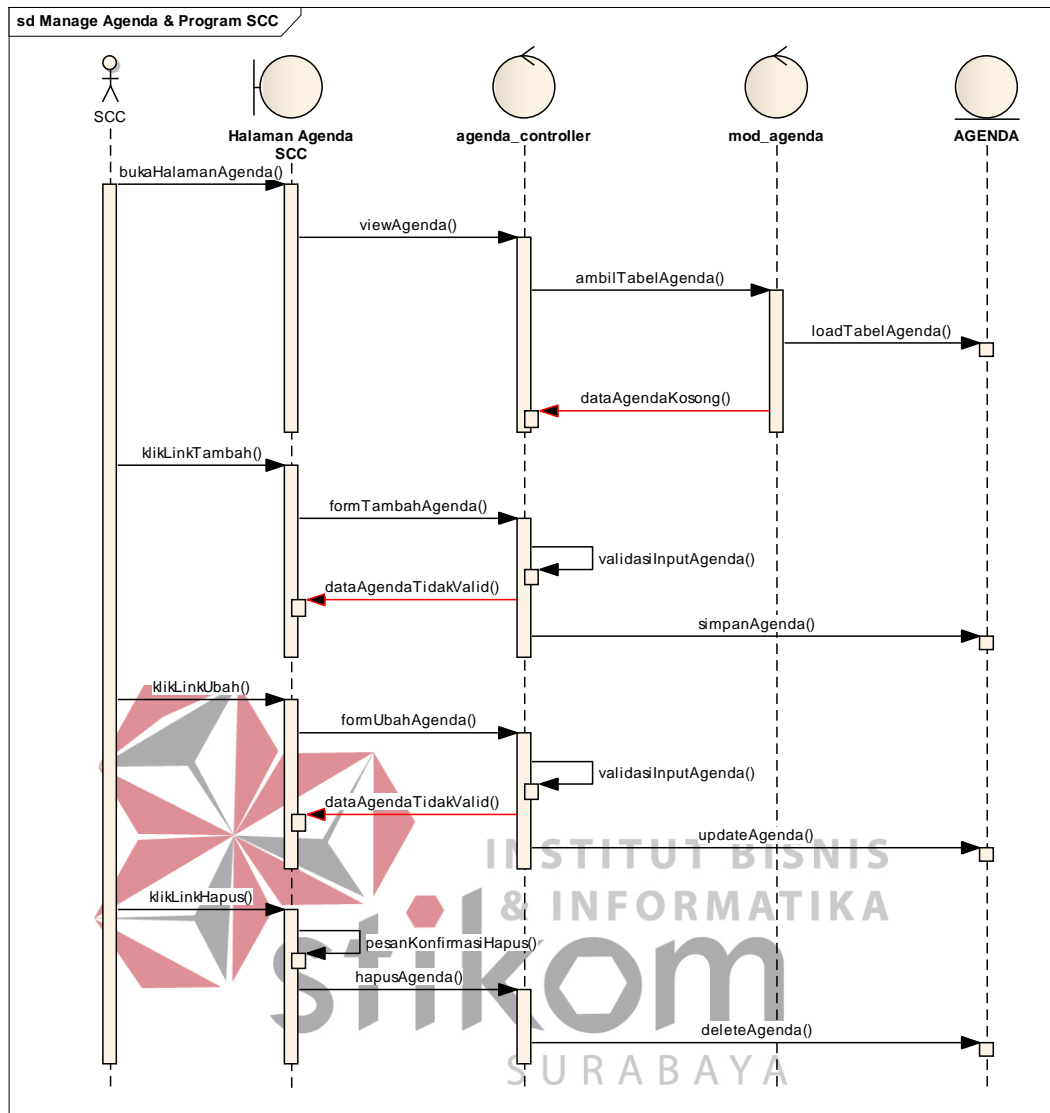
alamat yang terdapat pada variabel *array* tersebut. Apabila data email tidak ditemukan maka sistem akan menampilkan pesan bahwa data alamat email kosong.



Gambar 3.8 Sequence Diagram Broadcast Email

### 3.2.14 Sequence Diagram Manage Agenda SCC

SCC mengelola data agenda dan programnya melalui halaman agenda dan program SCC dengan menekan tautan *manage* agenda dan program SCC. Kemudian *controller* agenda menampilkan halaman agenda yang berisi tabel agenda. Tabel agenda didapatkan dari agenda model. Agenda model membaca data yang ada di dalam tabel agenda kemudian mengembalikannya kepada *controller* agenda. Apabila agenda model tidak menemukan data di dalam tabel agenda maka agenda model akan mengembalikan pesan bahwa data agenda kosong.



Gambar 3.9 *Sequence Diagram Manage Agenda SCC*

SCC menambah data agenda yang akan ditampilkan di dalam website komunitas alumni dengan menekan tautan tambah agenda. Kemudian *controller* agenda akan menampilkan halaman agenda yang berisi *form* untuk menambah data agenda. Setelah SCC mengisi data-data agenda yang diperlukan dan menekan tombol simpan, sistem akan memeriksa data yang dimasukkan oleh SCC. Apabila data tersebut sesuai dengan ketentuan validasi agenda maka *controller* agenda akan menyimpan data agenda tersebut ke dalam tabel agenda. Apabila data tersebut

tidak sesuai dengan ketentuan validasi agenda, maka sistem akan menampilkan pesan kesalahan kepada SCC.

SCC memperbarui data agenda dengan menekan tombol edit. Kemudian *controller* agenda akan meminta data agenda sesuai dengan parameter yang dikirimkan dan menampilkannya ke dalam halaman yang berisi *form* untuk memperbarui data agenda. Setelah SCC mengisi data-data agenda yang diperlukan dan menekan tombol simpan, sistem akan memeriksa data-data baru yang diinputkan oleh SCC. Apabila data-data tersebut sesuai dengan ketentuan validasi maka *controller* agenda akan menyimpan pembaruan data agenda tersebut ke dalam tabel agenda. Apabila data yang diberikan tidak sesuai dengan ketentuan validasi maka sistem akan menampilkan pesan peringatan kepada SCC terkait dengan data yang diinputkan tersebut.

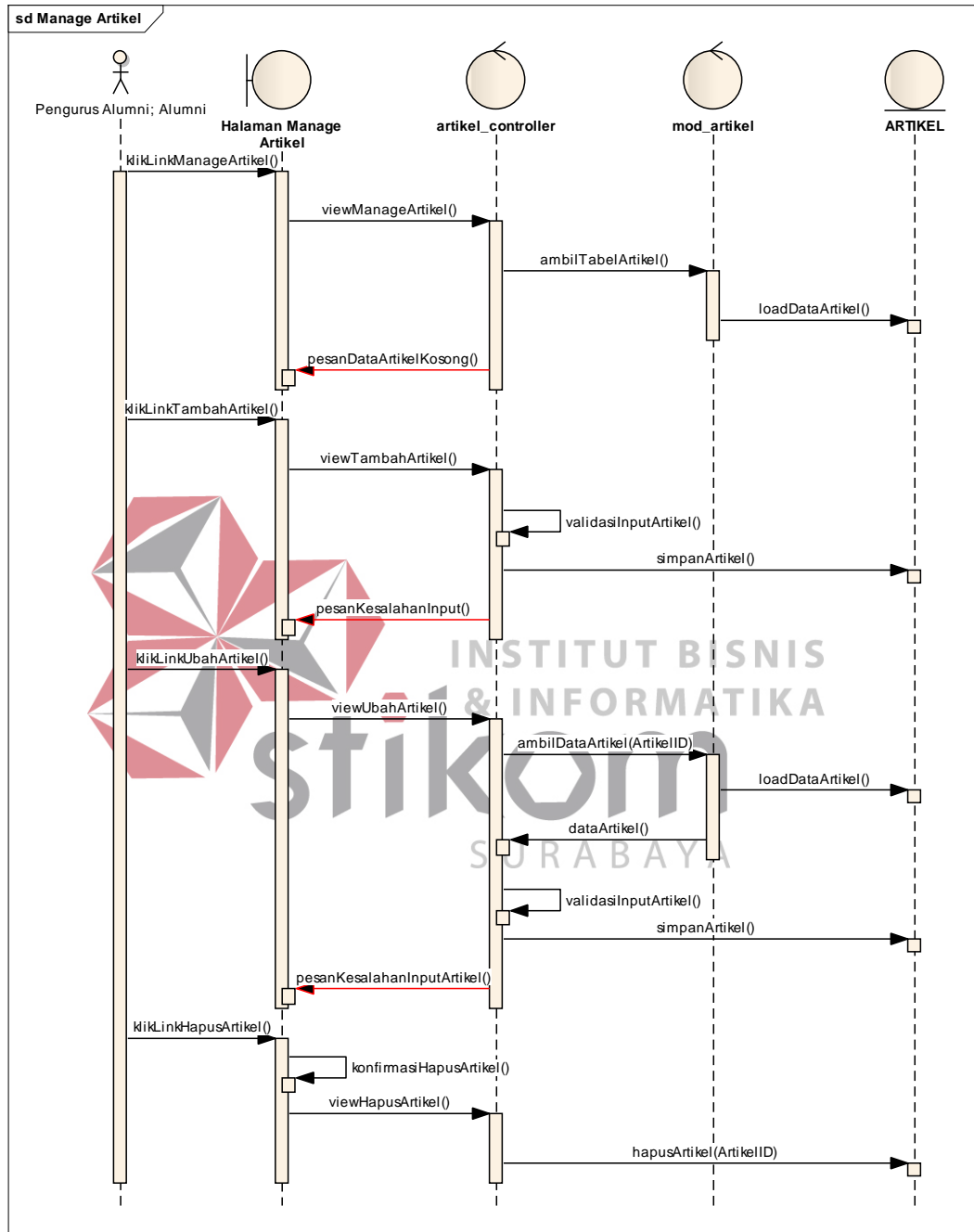
Penghapusan data agenda dilakukan dengan menekan tombol hapus. Maka sistem akan menampilkan pesan konfirmasi kepada SCC apakah benar-benar akan menghapus data tersebut. Apabila SCC mengonfirmasi penghapusan data agenda, maka *controller* agenda akan menghapus data agenda dari dalam database.

### **3.2.15 Sequence Diagram Manage Artikel**

Pengurus alumni mengelola artikel yang dimasukkan oleh alumni melalui halaman *manage* artikel. Melalui halaman ini pengurus alumni dapat menambah, mengubah dan menghapus artikel yang dituliskan oleh alumni. Ketika pengurus alumni menekan tautan *manage* artikel, *controller* artikel meminta data artikel-artikel yang dituliskan oleh alumni dari artikel model kemudian mengembalikan halaman *manage* artikel yang berisi tabel artikel. Apabila tidak terdapat artikel di



dalam tabel artikel maka sistem akan menampilkan pesan bahwa data artikel masih kosong.



Gambar 3.10 *Sequence Diagram Manage Artikel*

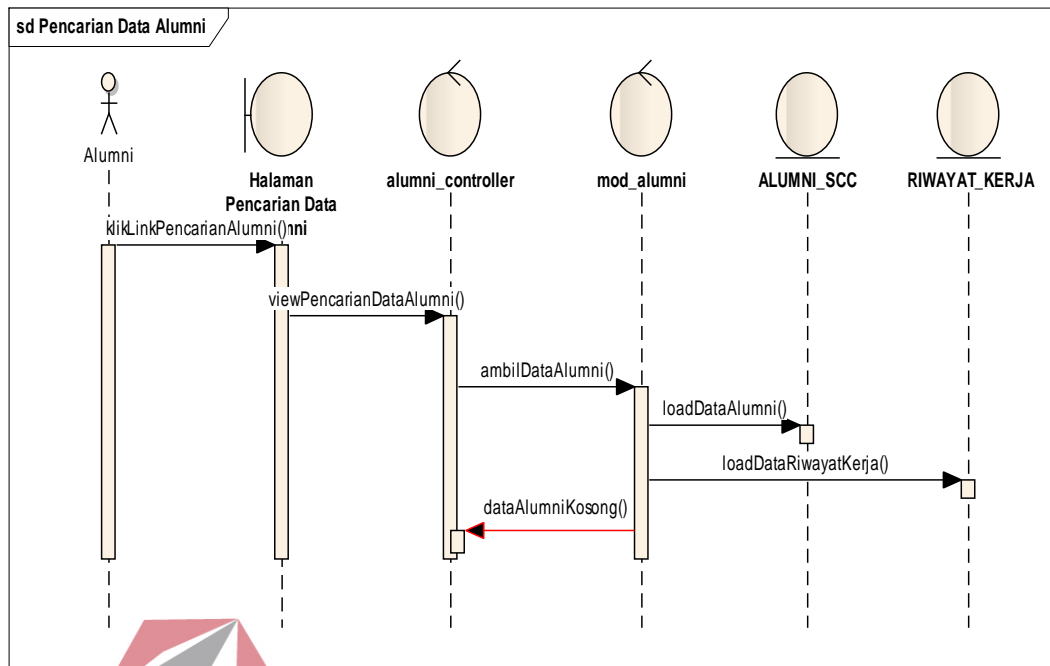
Pengurus alumni menambah artikel dengan menekan tombol tambah artikel. Kemudian *controller* artikel mengarahkan pengurus alumni kepada

halaman yang berisi *form* untuk mengisi data-data artikel. Kemudian pengurus alumni mengisi data artikel seperti judul artikel, deskripsi singkat artikel, dan isi artikel. Ketika pengurus alumni menekan tombol simpan, maka sistem akan memeriksa data-data yang telah dimasukkan. Apabila data tersebut sesuai dengan ketentuan validasi, maka *controller* artikel menyimpan data-data tersebut ke dalam tabel artikel. Apabila data yang dimasukkan pengurus alumni tidak sesuai dengan ketentuan validasi maka sistem akan menampilkan pesan kesalahan pada pengurus alumni.

Alumni mengubah data artikel dengan menekan tombol edit yang ada pada halaman *manage* artikel. Kemudian sistem akan mengarahkan pengurus alumni kepada halaman yang berisi *form* untuk mengubah data artikel. Setelah pengurus alumni menekan tombol simpan, maka sistem akan memeriksa data artikel yang telah dimasukkan. Apabila data tersebut sesuai dengan ketentuan validasi, maka *controller* artikel akan memperbarui data artikel tersebut ke dalam tabel artikel. Apabila data yang dimasukkan tidak sesuai dengan ketentuan validasi, maka sistem akan menampilkan pesan kesalahan kepada pengurus alumni.

Pengurus alumni menghapus data artikel yang ada pada tabel artikel dengan menekan tombol hapus. Ketika pengurus alumni menekan tombol hapus, sistem akan menampilkan pesan konfirmasi penghapusan data artikel. Apabila pengurus alumni mengonfirmasi penghapusan data tersebut, maka *controller* artikel akan menghapus data artikel dari dalam *database*.

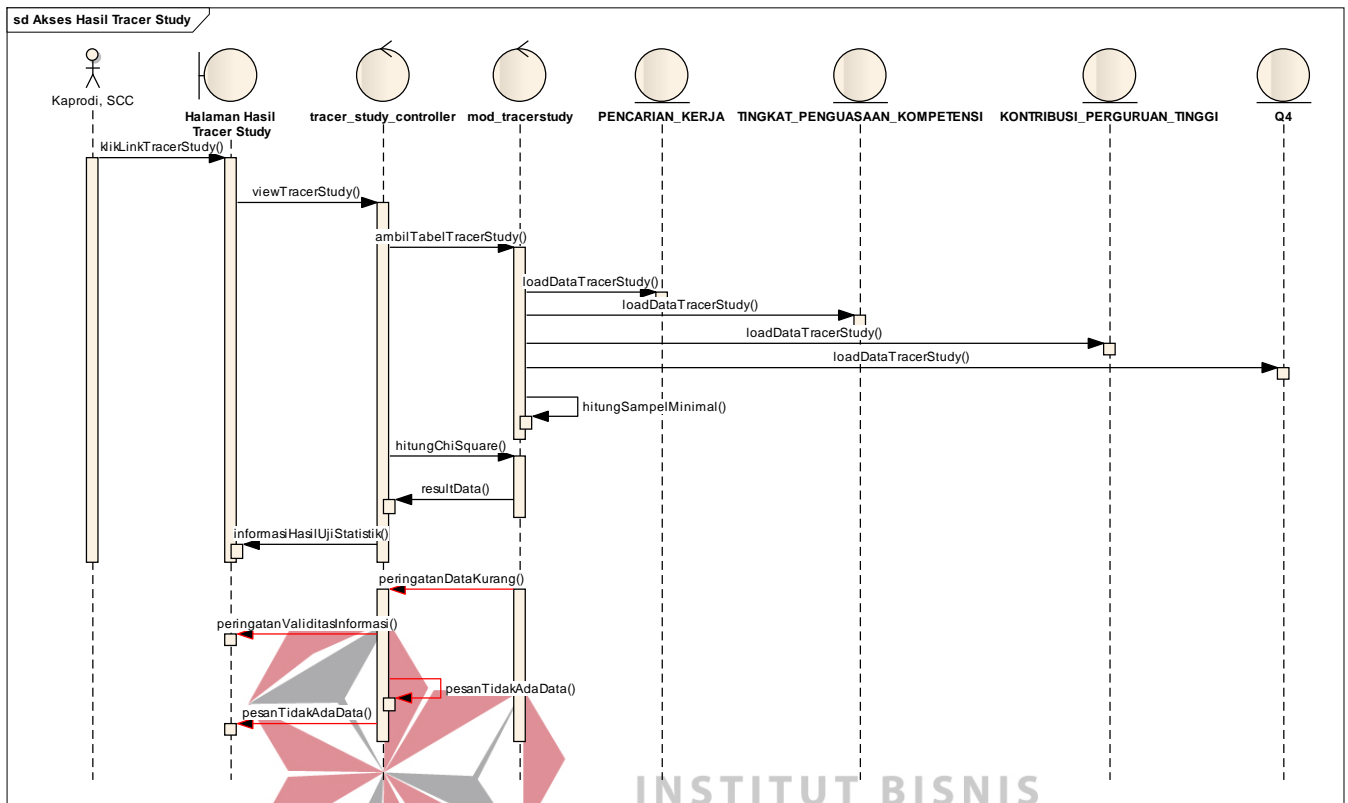
### 3.2.16 Sequence Diagram Mencari Data Alumni



Gambar 3.11 Sequence Diagram Mencari Data Alumni

Pencarian alumni dilakukan pada halaman pencarian data alumni. *Controller* alumni akan mengirimkan halaman pencarian data alumni setelah pengguna menekan tautan pencarian alumni. Setelah pengguna memasukkan nama dan menekan tombol cari, maka sistem akan memeriksa inputan pengguna. Apabila input yang dimasukkan pengguna sesuai dengan ketentuan validasi pencarian data alumni, maka *controller* alumni akan meminta data dari model alumni berdasarkan parameter-parameter yang telah ditentukan pada halaman pencarian alumni. Alumni model akan meminta data dari dalam *database* alumni dan mengembalikannya kepada *controller* alumni. Apabila tidak terdapat data yang sesuai maka sistem akan menampilkan pesan kepada pengguna bahwa data alumni yang dicari tidak ditemukan.

### 3.2.17 Sequence Diagram Mengakses Hasil Tracer Study



Gambar 3.12 Sequence Diagram Mengakses Hasil Tracer Study

Kaprodi dan SCC mengakses halaman hasil *tracer study* dengan menekan tautan hasil *tracer study*. Setelah itu controller *tracer study* meminta data hasil *tracer study* kepada *tracer study* model. *Tracer study* model membaca data yang ada di dalam *database* kemudian mengembalikannya kepada *controller tracer study*. Data yang didapatkan dari *tracer study* model disimpan ke dalam sebuah variabel array di dalam *controller tracer study*. Data tersebut kemudian dihitung sesuai dengan prosedur penghitungan jumlah sampel dan uji *chi-square*. Kemudian informasi hasil pengolahan data tersebut dimasukkan kedalam sebuah variabel kalimat yang nantinya ditampilkan kepada kaprodi dan SCC. Data-data lain yang didapatkan dari hasil fungsi *aggregate* kueri ditampilkan secara grafik pada halaman hasil *tracer study*.

### 3.2.18 *Class Diagram*

Diagram kelas digunakan untuk menampilkan kelas-kelas atau paket-paket dalam sistem dan realai antar mereka. Diagram kelas adalah alat perancangan terbaik untuk tim pengembang perangkat lunak. Diagram kelas membantu tim pengembang mendapatkan pola kelas-kelas dalam sistem, struktur sistem sebelum menuliskan kode program, dan membantu untuk memastikan bahwa sistem adalah rancangan terbaik dari beberapa alternatif rancangan (Sholiq, 2010).

Berdasarkan perancangan sistem pada diagram *use case*, dibutuhkan *class-class* untuk membangun dan mendukung jalannya aplikasi *web*. Hubungan antar kelas tersebut digambarkan pada sebuah *class diagram*. *Class diagram* dari sistem yang dibangun ditampilkan secara terpisah sesuai dengan relasi-relasi yang ada untuk mempermudah pembacaan.

*Class-class* yang akan dibuat di dalam *website* komunitas alumni dan *tracer study* ini akan didesain dengan konsep *Model-View-Controller* (MVC). *Class model* merupakan *class* yang menangani operasi-operasi yang berhubungan dengan entitas atau *database*. *Class view* merupakan *class* yang menangani penampilan informasi ke dalam layar kepada pengguna. Sedangkan *class controller* merupakan *business logic* dari aplikasi yang dibuat.

## A. Class Model

*Class model* akan menangani operasi-operasi yang berhubungan dengan database. Operasi-operasi yang ada dalam *class model* akan dipanggil langsung oleh *class controller*.

### A.1 Class MHS\_MF

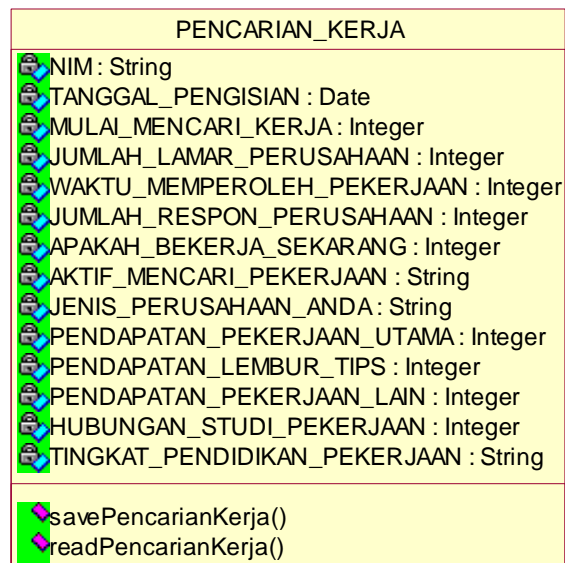
*Class MHS\_MF* berfungsi untuk menangani pembacaan data-data mahasiswa dalam database. Atribut-atribut yang dimiliki oleh class *MHS\_MF* antara lain adalah NIM, NAMA, ALAMAT, SEX, TGL\_LAHIR, PIN, dan TELP. *Class MHS\_MF* juga digunakan untuk mendapatkan PIN alumni yang akan melakukan login ke dalam *website*.



Gambar 3.13 Notasi Kelas MHS\_MF

### A.2 Class PENCARIAN\_KERJA

*Class PENCARIAN\_KERJA* berfungsi untuk menangani data-data yang didapatkan dari responden yang menjawab kuesioner bagian 1 (satu). Selain itu, *class* ini juga menangani permintaan data yang digunakan untuk menampilkan informasi dalam bentuk statistik deskriptif dan uji *chi-square*.

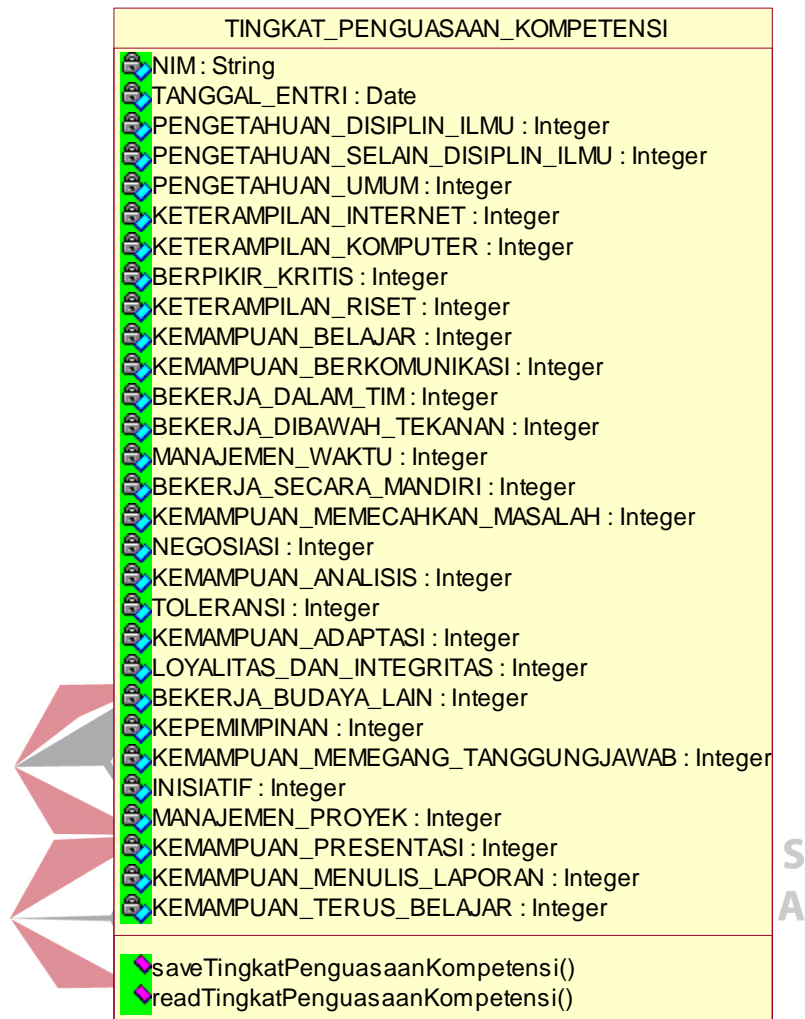


Gambar 3.14 Notasi Kelas Pencarian Kerja

### A.3 *Class* TINGKAT\_PENGUASAAN\_KOMPETENSI

*Class* ini berfungsi untuk menangani data-data yang didapatkan dari responden yang mengisi kuesioner tracer study bagian 2 (dua). Selain itu, *class* ini juga berfungsi untuk menangani permintaan data yang akan digunakan dalam menyajikan informasi dalam bentuk statistik deskriptif.

Setiap atribut yang ada pada *class* TINGKAT\_PENGUASAAN\_KOMPETENSI merupakan data-data yang didapatkan dari setiap pertanyaan pada kuesioner bagian 2 (dua). Melalui *class* ini pengembang dapat melakukan penyimpanan data-data responden ke dalam database. Operasi-operasi yang terdapat pada *class* ini antara lain: `saveTingkatPenguasaanKompetensi` yang digunakan untuk menyimpan dan `readTingkatPenguasaanKompetensi` yang digunakan untuk membaca.



Gambar 3.15 Notasi Kelas TINGKAT\_PENGUASAAN\_KOMPETENSI

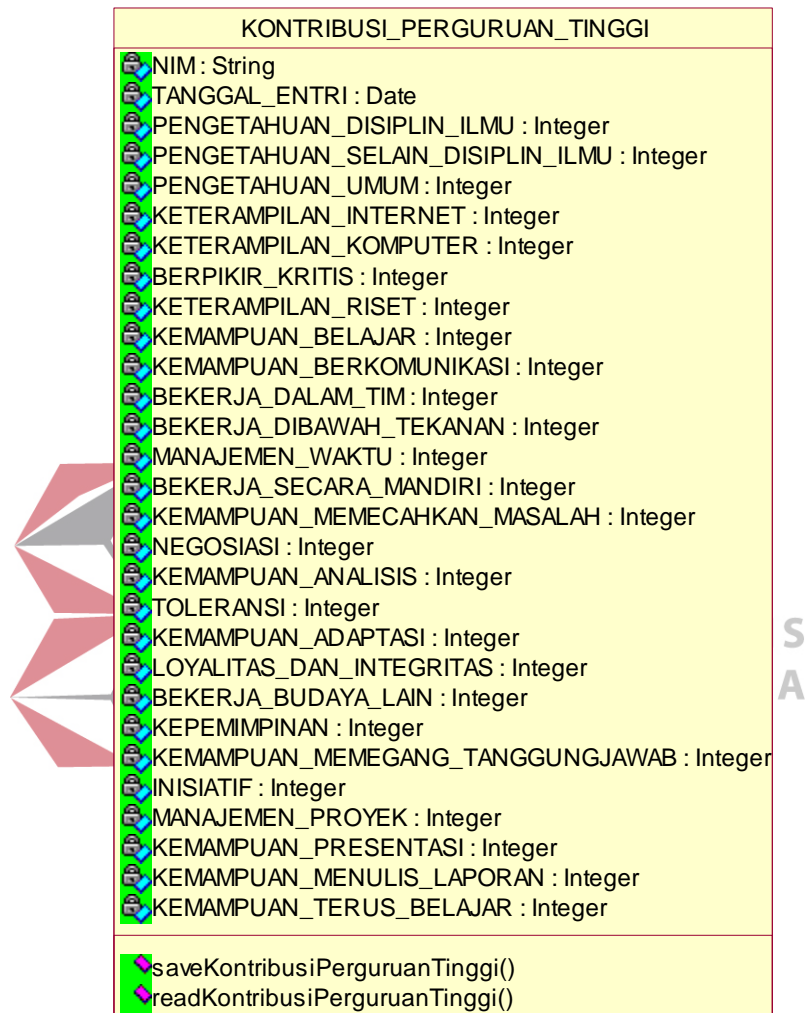
#### A.4 *Class* KONTRIBUSI\_PERGURUAN\_TINGGI

*Class* ini berfungsi untuk menangani data-data yang didapatkan dari responden yang mengisi kuesioner tracer study bagian 3 (tiga). Selain itu, *class* ini juga berfungsi untuk menangani permintaan data yang akan digunakan dalam menyajikan informasi dalam bentuk statistik deskriptif.

Setiap atribut yang ada pada *class* KONTRIBUSI\_PERGURUAN\_TINGGI merupakan data-data yang didapatkan dari setiap pertanyaan pada kuesioner bagian 3 (tiga). Operasi-operasi yang



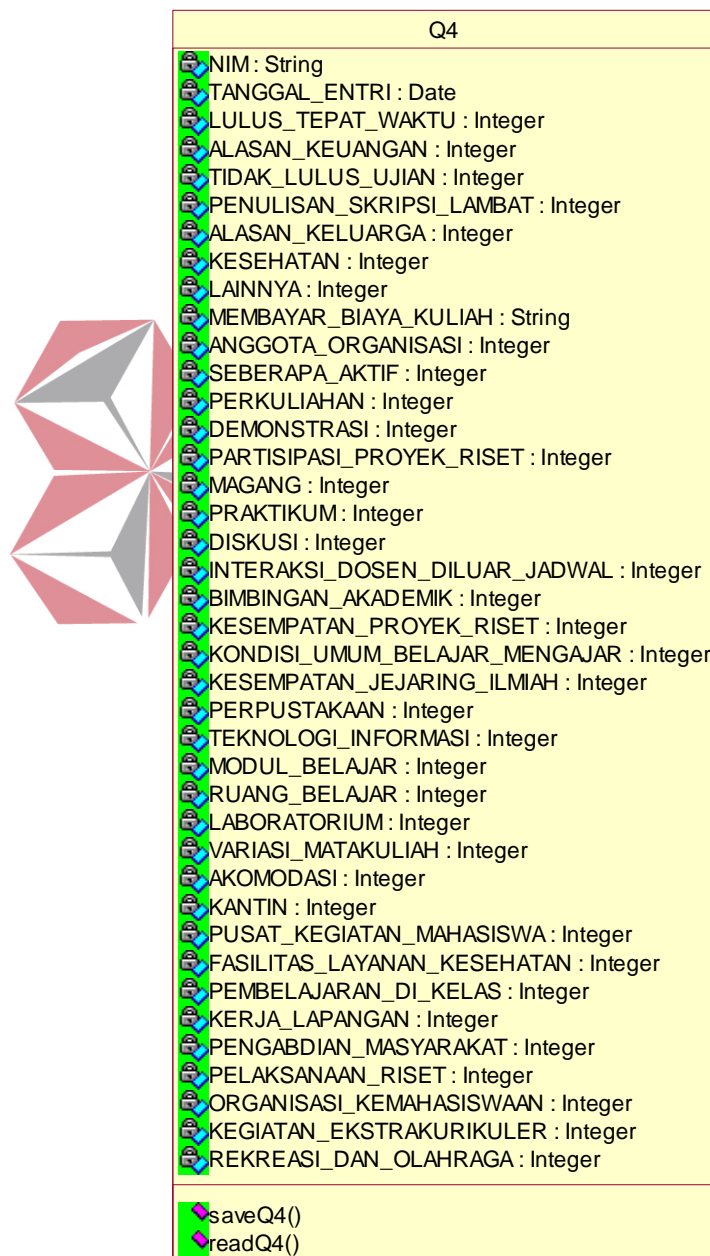
terdapat pada class ini antara lain: `saveKontribusiPerguruanTinggi` yang digunakan untuk menyimpan dan `readKontribusiPerguruanTinggi` yang digunakan untuk membaca.



Gambar 3.16 Notasi Kelas `KONTRIBUSI_PERGURUAN_TINGGI`

### A.5 Class Q4

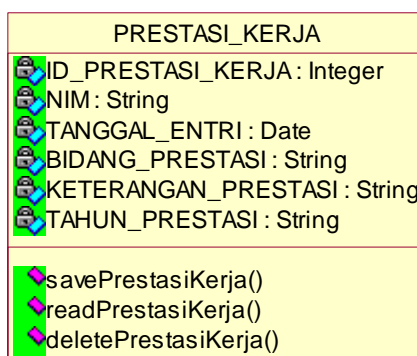
*Class* ini berfungsi untuk menangani data-data yang didapatkan dari responden yang mengisi kuesioner tracer study bagian 3 (empat). Selain itu, *class* ini juga berfungsi untuk menangani permintaan data yang akan digunakan dalam menyajikan informasi dalam bentuk statistik deskriptif.



Gambar 3.17 Notasi Kelas Q4

### A.6 Class PRESTASI\_KERJA

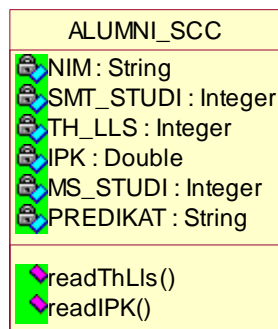
*Class* PRESTASI\_KERJA menangani data-data prestasi kerja yang diinputkan oleh alumni. *Class-class* ini menyediakan atribut seperti ID\_PRESTASI\_KERJA, NIM, TANGGAL\_ENTRI, BIDANG\_PRESTASI, KETERANGAN\_PRESTASI, dan TAHUN\_PRESTASI. Untuk operasinya *class* ini menyediakan `savePrestasiKerja`, `readPrestasiKerja`, dan `deletePrestasiKerja`.



Gambar 3.18 Notasi Kelas PRESTASI\_KERJA

#### A.7 *Class* ALUMNI\_SCC

*Class* ALUMNI\_SCC menangani data-data terbaru yang diinputkan alumni melalui *website* SCC. Atribut-atribut yang dimiliki antara lain adalah NIM, SMT\_STUDI, TH\_LLS, IPK, MS\_STUDI, dan PREDIKAT. Kemudian operasi-operasi yang dimiliki antara lain adalah `readThLls` dan `readIPK`. Selain menangani data alumni, kelas ini juga berfungsi sebagai dasar penyaringan data yang dilakukan pada halaman hasil *tracer study*.



Gambar 3.19 Notasi Kelas ALUMNI\_SCC

### A.8 Class ARTIKEL

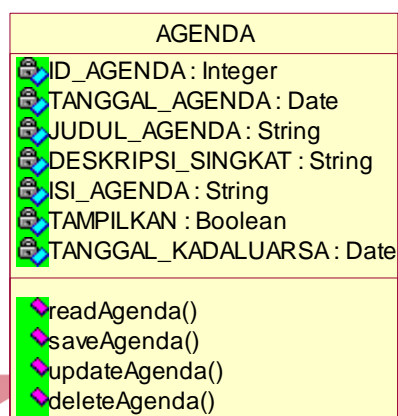
*Class* ARTIKEL digunakan untuk menangani data-data artikel. Atribut-atribut yang ada pada kelas ini antara lain adalah ID\_ARTIKEL, TANGGAL\_ARTIKEL, JUDUL\_ARTIKEL, DESKRIPSI\_SINGKAT, ISI\_ARTIKEL, dan TAMPILKAN.



Gambar 3.20 Notasi Kelas ARTIKEL

### A.9 *Class* AGENDA

*Class* AGENDA digunakan untuk menangani data-data agenda. Atribut-atribut yang ada pada kelas ini antara lain adalah ID\_AGENDA, TANGGAL\_AGENDA, JUDUL\_AGENDA, DESKRIPSI\_SINGKAT, ISI\_AGENDA, TAMPILKAN, dan TANGGAL\_KADALUARSA.



Gambar 3.21 Notasi Kelas AGENDA

### A.10 *Class* RIWAYAT\_KERJA



Gambar 3.22 Notasi Kelas RIWAYAT\_KERJA

*Class* RIWAYAT\_KERJA digunakan untuk menangani data riwayat kerja alumni yang diinputkan melalui *website* SCC. Atribut-atribut yang ada antara lain adalah NIM, ID\_PERUSH, TAHUN\_MASUK, TAHUN\_KELUAR, dan

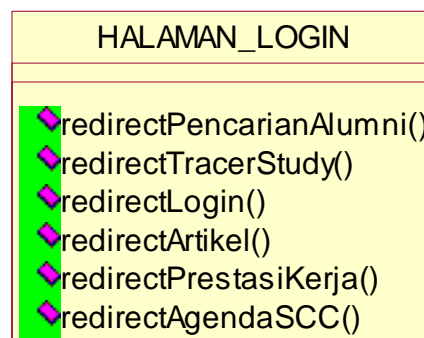
POSISIJAB. Operasi-operasi yang ada pada class ini antara lain adalah readIDPerush, readTahunMasuk, readTahunKeluar, dan readPosisiJab.

## B. Class View

*Class view* merupakan *class-class* yang menangani penampilan informasi di layar kaca kepada pengguna yang berinteraksi dengan sistem. *Class view* membentuk suatu user interface yang digunakan untuk menampilkan dan menginputkan data yang diperoleh dari luar sistem.

### B.1 Class HALAMAN\_LOGIN

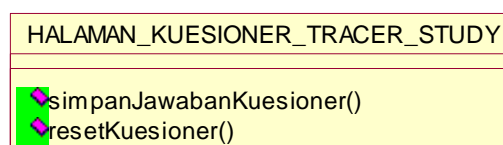
Kelas HALAMAN\_LOGIN adalah kelas yang akan ditampilkan ketika pengguna pertama kali masuk ke dalam sistem. Melalui halaman ini, pengguna dapat bernavigasi ke halaman-halaman yang lain melalui menu yang telah disediakan. Operasi-operasi yang dimiliki kelas HALAMAN\_LOGIN antara lain adalah redirectPencarianAlumni, redirectTracerStudy, redirectLogin, redirectArtikel, redirectPrestasiKerja, dan redirectAgendaSCC.



Gambar 3.23 Notasi Kelas HALAMAN\_LOGIN

### B.2 Class HALAMAN\_KUESIONER\_TRACER\_STUDY

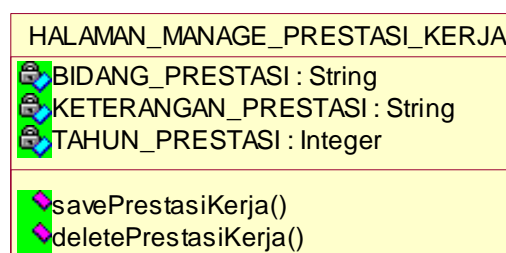
Kelas `HALAMAN_KUESIONER_TRACER_STUDY` digunakan untuk mendapatkan data-data dari alumni. Operasi-operasi yang dimiliki kelas ini antara lain adalah `simpanJawabanKuesioner` dan `resetKuesioner`. `simpanJawabanKuesioner` digunakan untuk mengirimkan data yang telah dimasukkan oleh alumni ke *server*. Sedangkan `resetKuesioner` digunakan untuk mengembalikan kondisi kuesioner pada keadaan semula.



Gambar 3.24 Notasi Kelas `HALAMAN_KUESIONER_TRACER_STUDY`

### B.3 *Class* `HALAMAN_MANAGE_PRESTASI_KERJA`

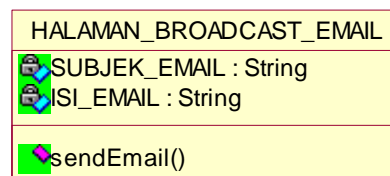
Kelas `HALAMAN_MANAGE_PRESTASI_KERJA` digunakan untuk mengelola prestasi kerja oleh alumni. Melalui halaman ini alumni dapat menambahkan data prestasi kerja baru. Selain itu alumni juga dapat menghapus data prestasi kerjanya yang ada di dalam database.



Gambar 3.25 Notasi Kelas `HALAMAN_MANAGE_PRESTASI_KERJA`

#### B.4 *Class* HALAMAN\_BROADCAST\_EMAIL

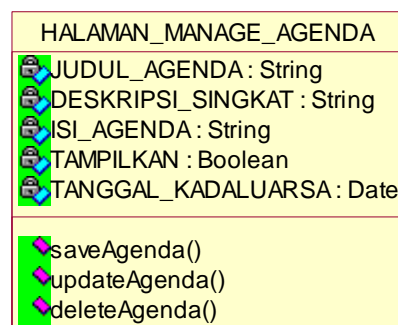
Kelas HALAMAN\_BROADCAST\_EMAIL digunakan untuk memberikan tampilan kepada pengguna yang akan mengirimkan email secara massal. Kelas ini memiliki atribut SUBJEK\_EMAIL dan ISI\_EMAIL. Dan memiliki operasi sendEmail.



Gambar 3.26 Notasi Kelas HALAMAN\_BROADCAST\_EMAIL

#### B.5 *Class* HALAMAN\_MANAGE\_AGENDA\_SCC

Kelas HALAMAN\_MANAGE\_AGENDA\_SCC digunakan SCC untuk mengelola data agendanya di dalam database. Melalui kelas ini, SCC dapat melakukan penambahan data, pengubahan, dan penghapusan. Atribut-atribut yang dimiliki kelas ini antara lain adalah JUDUL\_AGENDA, DESKRIPSI\_SINGKAT, ISI\_AGENDA, TAMPILKAN, dan TANGGAL\_KADALUARSA.

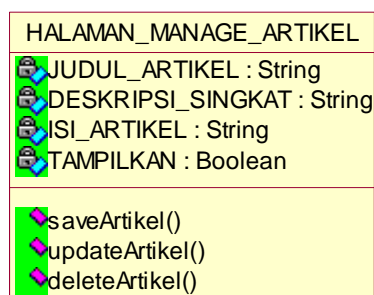


Gambar 3.27 Notasi Kelas HALAMAN\_MANAGE\_AGENDA

#### B.6 *Class* HALAMAN\_MANAGE\_ARTIKEL



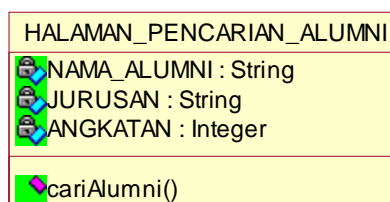
Kelas `HALAMAN_MANAGE_ARTIKEL` digunakan untuk mengelola data artikel yang ada di dalam *database*. Pengguna dapat melakukan penambahan, perubahan, dan penghapusan data artikel melalui kelas ini. Atribut-atribut yang dimiliki kelas ini antara lain adalah `JUDUL_ARTIKEL`, `DESKRIPSI_SINGKAT`, `ISI_ARTIKEL`, dan `TAMPILKAN`.



Gambar 3.28 Notasi Kelas `HALAMAN_MANAGE_ARTIKEL`

### B.7 *Class* `HALAMAN_PENCARIAN_DATA`

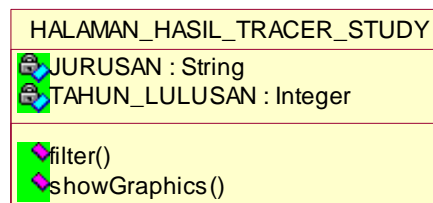
Kelas `HALAMAN_PENCARIAN_DATA` digunakan untuk mencari data alumni yang ada di dalam *database*. Pengguna yang menggunakan kelas ini dapat melakukan pencarian alumni berdasarkan nama alumni, jurusan alumni, dan angkatan alumni tersebut. Operasi yang dilakukan oleh kelas ini hanya `cariAlumni`.



Gambar 3.29 Notasi Kelas `HALAMAN_PENCARIAN_DATA`

### B.8 *Class* `HALAMAN_HASIL_TRACER_STUDY`

Kelas `HALAMAN_HASIL_TRACER_STUDY` bertugas untuk menampilkan hasil pengolahan data yang disajikan menggunakan grafik. Pengguna juga dapat melakukan penyaringan data yang akan ditampilkan berdasarkan jurusan alumni atau tahun lulusan alumni. Atribut-atribut yang dimiliki kelas ini antara lain adalah `JURUSAN` dan `TAHUN_LULUSAN`.



Gambar 3.30 Notasi Kelas `HALAMAN_TRACER_STUDY`

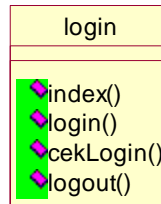
### C. *Class Controller*

Kelas *controller* di sini adalah kelas-kelas yang mengatur jalannya *website*. Beberapa pengecekan input pengguna juga dilakukan di dalam *controller*. Kelas-kelas *controller* juga berfungsi sebagai perantara antara tampilan *user interface* dengan pengolahan data di dalam *database*.

#### C.1 *Class login*

Kelas login digunakan untuk mengatur jalannya login yang dilakukan oleh pengunjung *web*. Salah satu operasi yang juga dilakukan oleh kelas ini adalah mengubah status login pengguna menjadi *false* dengan memanggil fungsi `logout`. *Username* dan *password* yang dimasukkan oleh pengunjung *web* dikirimkan kepada fungsi `cekLogin`. Kemudian fungsi tersebut akan memvalidasi masukan pengunjung *web*. Apabila *username* dan *password* yang dimasukkan sesuai maka fungsi `cekLogin` akan mengubah status login pengunjung *web* tersebut menjadi

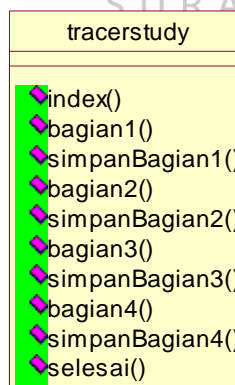
*true*. Setelah itu, fungsi cekLogin juga akan mengarahkan pengguna ke halaman berikutnya sesuai dengan peran pengguna tersebut.



Gambar 3.31 Notasi Kelas Login

## C.2 *Class tracerstudy*

Kelas *tracerstudy* digunakan untuk mengatur alumni yang akan mengisi kuesioner *tracerstudy*. Kelas ini juga berfungsi untuk menyimpan data yang didapatkan dari alumni yang mengisi kuesioner. Operasi yang dilakukan oleh kelas ini terdiri dari bagian dan *simpanBagian*. Operasi bagian berfungsi untuk menampilkan kuesioner yang akan diisi oleh alumni. Operasi *simpanBagian* berfungsi untuk menangkap inputan alumni dari kuesioner-kuesioner tersebut.

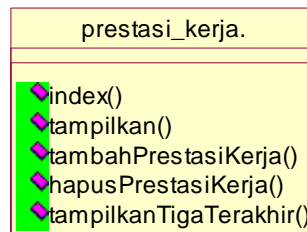


Gambar 3.32 Notasi Kelas *tracerstudy*

## C.3 *Class prestasi\_kerja*

Kelas *prestasi\_kerja* digunakan untuk pengolahan data prestasi kerja alumni. Fungsi *tampilkanTigaTerakhir* digunakan untuk menampilkan data yang

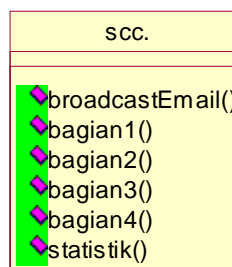
terakhir dimasukkan oleh alumni. Data tersebut akan ditampilkan pada halaman awal dan beberapa halaman lainnya sebagai jalan pintas untuk menampilkan prestasi kerja alumni lebih detail.



Gambar 3.33 Notasi Kelas prestasi\_kerja

#### C.4 *Class scc*

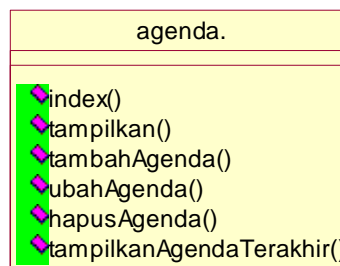
Kelas scc juga digunakan sebagai kelas yang menangani aktor yang berperan sebagai pengurus alumni, SCC, dan kaprodi. Peran-peran seperti *broadcast* email, akses hasil *tracer study* dapat dilakukan melalui kelas ini. Operasi-operasi yang dilakukan kelas ini antara lain adalah broadcastEmail, bagian1, bagian2, bagian3, bagian4, dan statistik. Fungsi statistik berfungsi untuk menampilkan halaman hasil uji statistik.



Gambar 3.34 Notasi Kelas scc

#### C.5 *Class agenda*

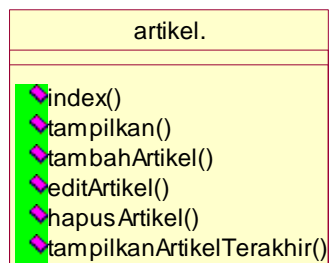
Kelas agenda berfungsi untuk mengatur pengolahan data agenda. Melalui kelas ini SCC dapat menambah, mengubah, dan menghapus data agenda. Fungsi `tampilkanAgendaTerakhir` adalah untuk menampilkan data agenda yang terakhir diinputkan oleh SCC. Kelas agenda memiliki operasi-operasi antara lain `index`, `tampilkan`, `tambahAgenda`, `ubahAgenda`, `hapusAgenda`, dan `tampilkanAgendaTerakhir`.



Gambar 3.35 Notasi Kelas agenda

### C.6 *Class artikel*

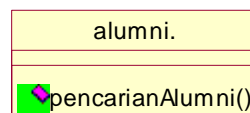
Kelas artikel berfungsi untuk mengatur pengolahan data artikel. Fungsi-fungsi yang dimiliki antara lain adalah `index`, `tampilkan`, `tambahArtikel`, `editArtikel`, `hapusArtikel`, dan `tampilkanArtikelTerakhir`. Fungsi `tampilkanArtikelTerakhir` berfungsi untuk menampilkan data artikel yang diinputkan terakhir.



Gambar 3.36 Notasi Kelas artikel

### C.7 Class alumni

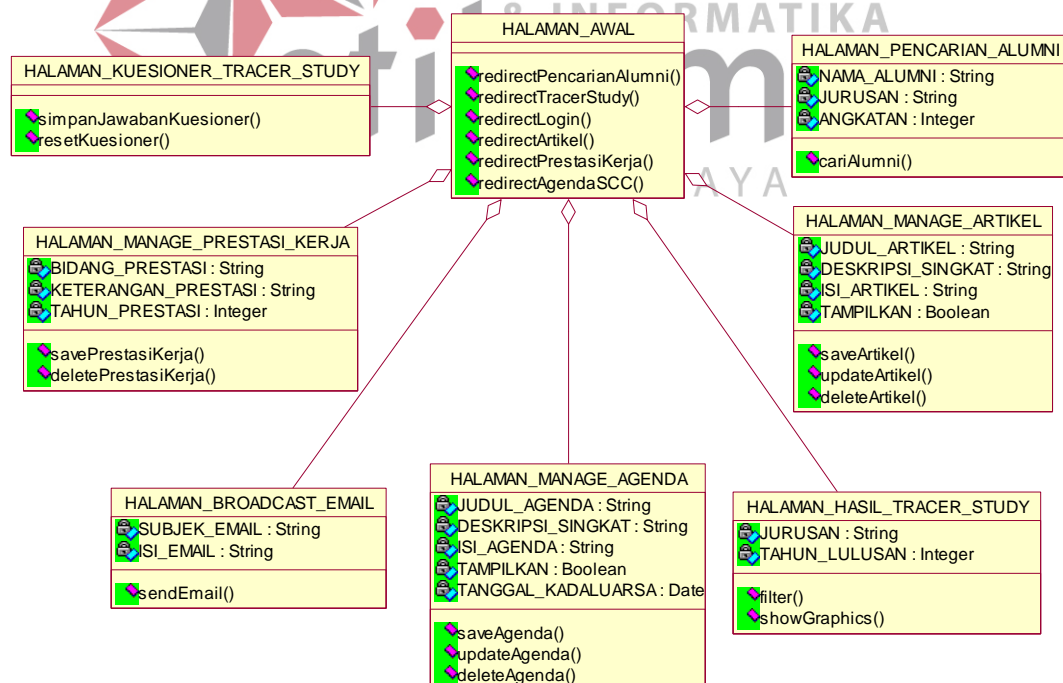
Kelas alumni hanya memiliki fungsi pencarianAlumni. Pencarian alumni dipanggil setelah pengunjung *web* mengisi kriteria pencarian kemudian menekan tombol cari pada halaman pencarian alumni.



Gambar 3.37 Notasi Kelas alumni

### D. Relasi Class Diagram Tampilan

Relasi *class diagram* tampilan menjelaskan tentang relasi antar *class view*. Pada gambar tersebut dijelaskan keterkaitan antar *class view* yang satu dengan *class view* yang lain di dalam *website* komunitas alumni dan *tracer study*.



Gambar 3.38 Relasi Class Diagram Tampilan

### E. Relasi *Class Diagram* Proses dan Entitas

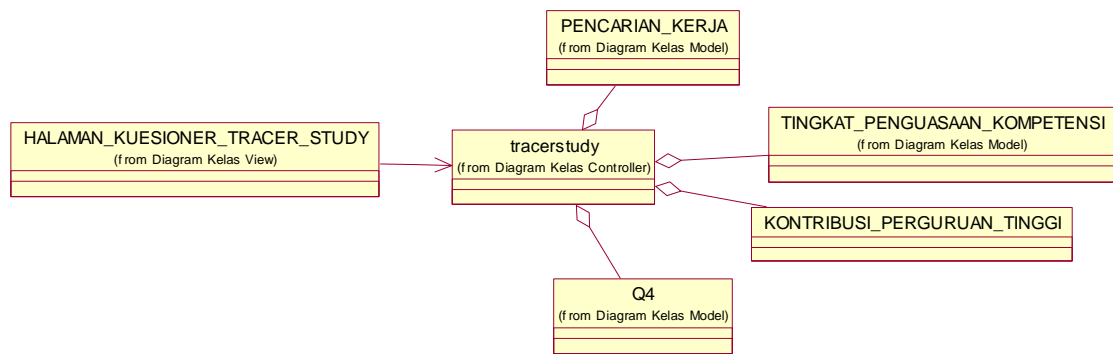
Relasi *class diagram* proses dan entitas menjelaskan tentang interaksi antara proses-proses yang terjadi di dalam sistem terhadap entitas tertentu. Entitas menggambarkan tabel-tabel yang ada di dalam *database*. Berikut ini adalah diagram-diagram yang menggambarkan relasi antar proses dan entitas yang terjadi di dalam sistem.

Pada gambar 3.39 dijelaskan bahwa kelas HALAMAN\_LOGIN akan memanggil fungsi yang ada pada kelas login dengan parameter *username* dan *password*. Kemudian kelas login meminta pin yang ada di dalam kelas MHS\_MF sesuai dengan *username* yang diberikan. Pin yang didapatkan dari kelas MHS\_MF kemudian dicocokkan dengan pin yang dimasukkan oleh pengunjung *web*.



Gambar 3.39 *Class Diagram* Login

Untuk proses menjawab kuesioner *tracer study*, kelas-kelas yang terlibat digambarkan pada gambar 3.40. Melalui kelas *controller* tracerstudy, HALAMAN\_KUESIONER\_TRACERSTUDY mengirimkan data-data yang didapatkan dari responden ke server. Dari kelas *controller* tracerstudy kemudian dikirimkan kepada kelas-kelas model untuk disimpan ke dalam *database*.



Gambar 3.40 *Class Diagram* Menjawab Kuesioner *Tracer Study*

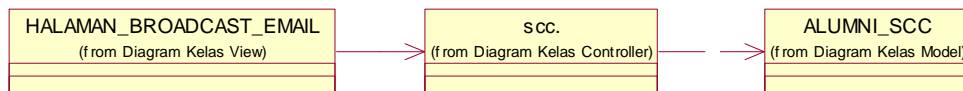
Pada gambar 3.41, entitas-entitas yang dilibatkan pada proses manage prestasi kerja adalah PRESTASI\_KERJA dan MHS\_MF. Entitas MHS\_MF dibutuhkan untuk mengidentifikasi pemilik data prestasi kerja tersebut. Relasi antara kelas *controller* prestasi\_kerja dengan entitas PRESTASI\_KERJA berupa agregasi karena seluruh atribut yang berada pada entitas PRESTASI\_KERJA digunakan dalam kelas *controller* prestasi\_kerja.



Gambar 3.41 *Class Diagram* Manage Prestasi Kerja

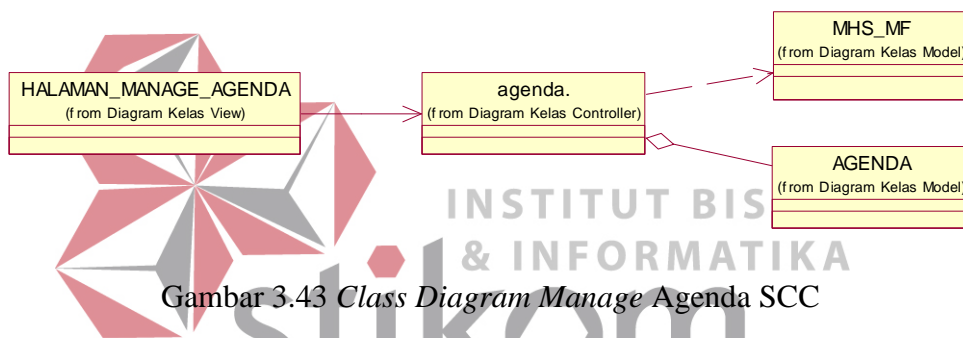
Gambar 3.42 menjelaskan tentang hubungan antara kelas model, kelas *view*, dan kelas *controller* pada proses *broadcast* email. Entitas yang terlibat adalah ALUMNI\_SCC, karena data alumni yang terbaru disimpan di dalam tabel tersebut. Kelas *controller* scc dependent terhadap kelas ALUMNI\_SCC karena membutuhkan alamat email terbaru dari alumni untuk mengirimkan email.





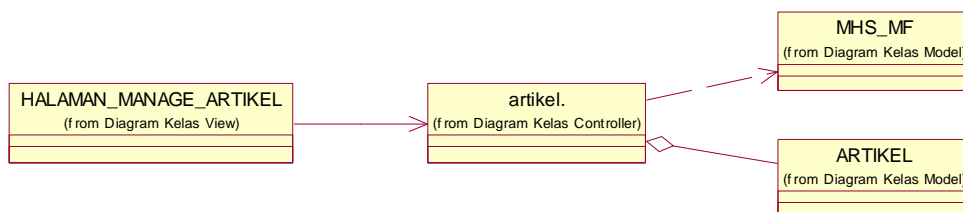
Gambar 3.42 *Class Diagram Broadcast Email*

Hubungan antar kelas pada proses manage agenda SCC digambarkan pada gambar 3.43. Pada gambar tersebut dijelaskan bahwa kelas-kelas yang terlibat antara lain adalah HALAMAN\_MANAGE\_AGENDA, kelas *controller* agenda, MHS\_MF, dan AGENDA. Seluruh atribut yang ada pada kelas model AGENDA digunakan pada kelas *controller* agenda sehingga hubungan antara kelas tersebut adalah agregasi.



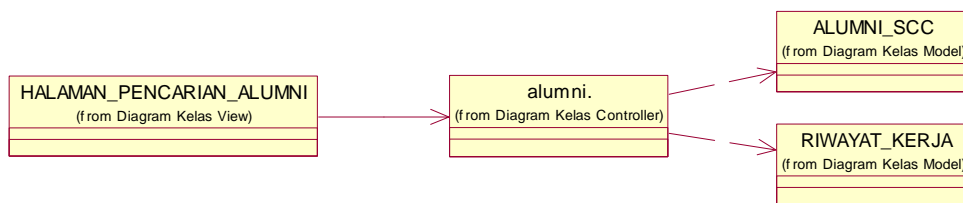
Gambar 3.43 *Class Diagram Manage Agenda SCC*

Gambar 3.44 menjelaskan tentang kelas-kelas yang terlibat pada proses *manage* artikel. Kelas-kelas yang terlibat adalah HALAMAN\_MANAGE\_ARTIKEL, kelas *controller* artikel, MHS\_MF, ARTIKEL. Kelas *controller* artikel menggunakan seluruh atribut yang ada pada kelas ARTIKEL sehingga hubungan antar kelas tersebut adalah agregasi.



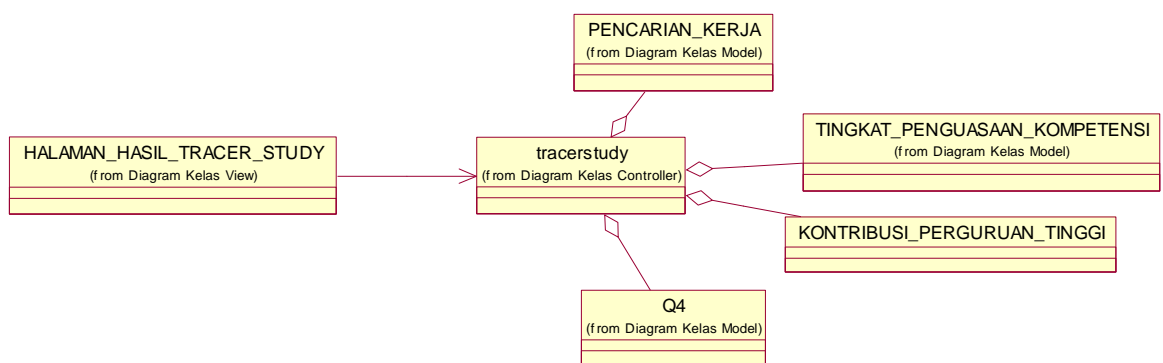
Gambar 3.44 *Class Diagram Manage Artikel*

Gambar 3.45 menjelaskan tentang kelas-kelas yang terlibat di dalam proses mencari data alumni. Kelas-kelas yang terlibat antara lain adalah HALAMAN\_PENCARIAN\_ALUMNI, kelas *controller* alumni, ALUMNI\_SCC, dan RIWAYAT\_KERJA. Kelas RIWAYAT\_KERJA dibutuhkan, karena melalui halaman ini pengunjung web dapat menampilkan riwayat kerja alumni.



Gambar 3.45 *Class Diagram* Mencari Data Alumni

Untuk menampilkan hasil pengolahan data-data yang didapatkan dari *tracer study*, kelas-kelas yang dilibatkan digambarkan pada gambar 3.46. kelas-kelas yang dilibatkan pada proses ini hampir sama dengan kelas-kelas yang dilibatkan dalam proses menjawab kuesioner *tracer study*. Sejumlah 4 (empat) kelas model dilibatkan untuk menangani permintaan data yang ada di dalam *database*.

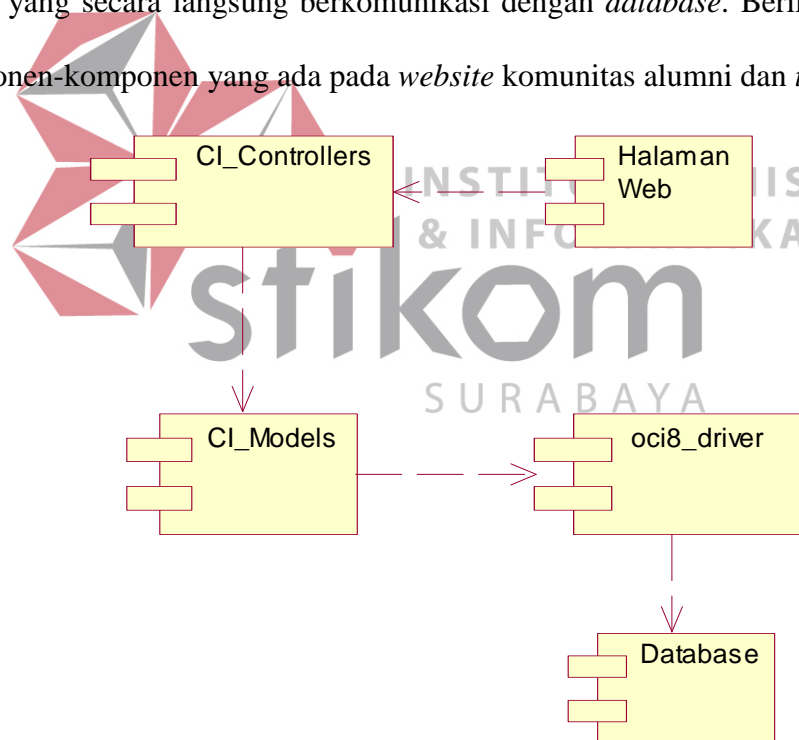


Gambar 3.46 *Class Diagram* Mengakses Hasil *Tracer Study*

### 3.2.19 Diagram Komponen

Diagram komponen menggambarkan modul fisik dari kode. Garis-garis yang menghubungkan antara komponen yang satu dengan komponen yang lainnya menunjukkan dependensi komponen tersebut. Setiap komponen diperlukan agar program yang dibangun dapat berjalan dengan baik.

Halaman *web* berkomunikasi secara langsung dengan server melalui *controller*. Komponen *controller* dalam mendapatkan atau menyimpan data ke dalam database melalui *model*. Sedangkan untuk dapat mengakses data yang ada di dalam *database*, komponen *model* membutuhkan bantuan dari komponen *driver* yang secara langsung berkomunikasi dengan *database*. Berikut ini adalah komponen-komponen yang ada pada *website* komunitas alumni dan *tracer study*.



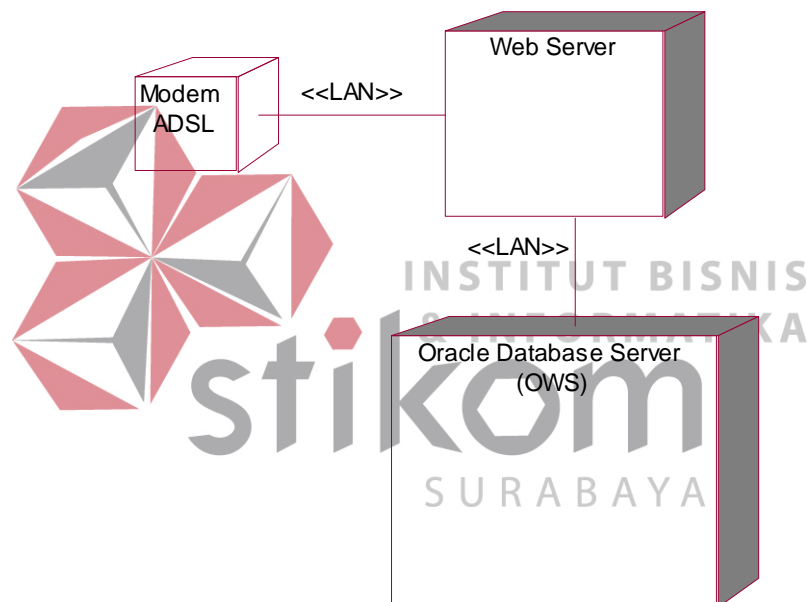
Gambar 3.47 Diagram Komponen

Gambar di atas menjelaskan bahwa komponen halaman web bergantung kepada kelas *controller*. Kelas *controller* bergantung kepada kelas model dalam meminta data yang diperlukan dari dalam *database*. Kelas model mengirimkan

kueri yang akan dijalankan melalui komponen `oci8_driver`. Kemudian `oci8_driver` merupakan kelas yang bertanggung jawab dalam komunikasi dengan *database*.

### 3.2.20 Diagram Deployment

Diagram *deployment* menjelaskan tentang segala hal yang berkaitan dengan penyebaran fisik aplikasi. Diagram deployment berisi prosesor-prosesor, peralatan-peralatan, proses-proses, dan hubungan antar prosesor atau peralatan tersebut.



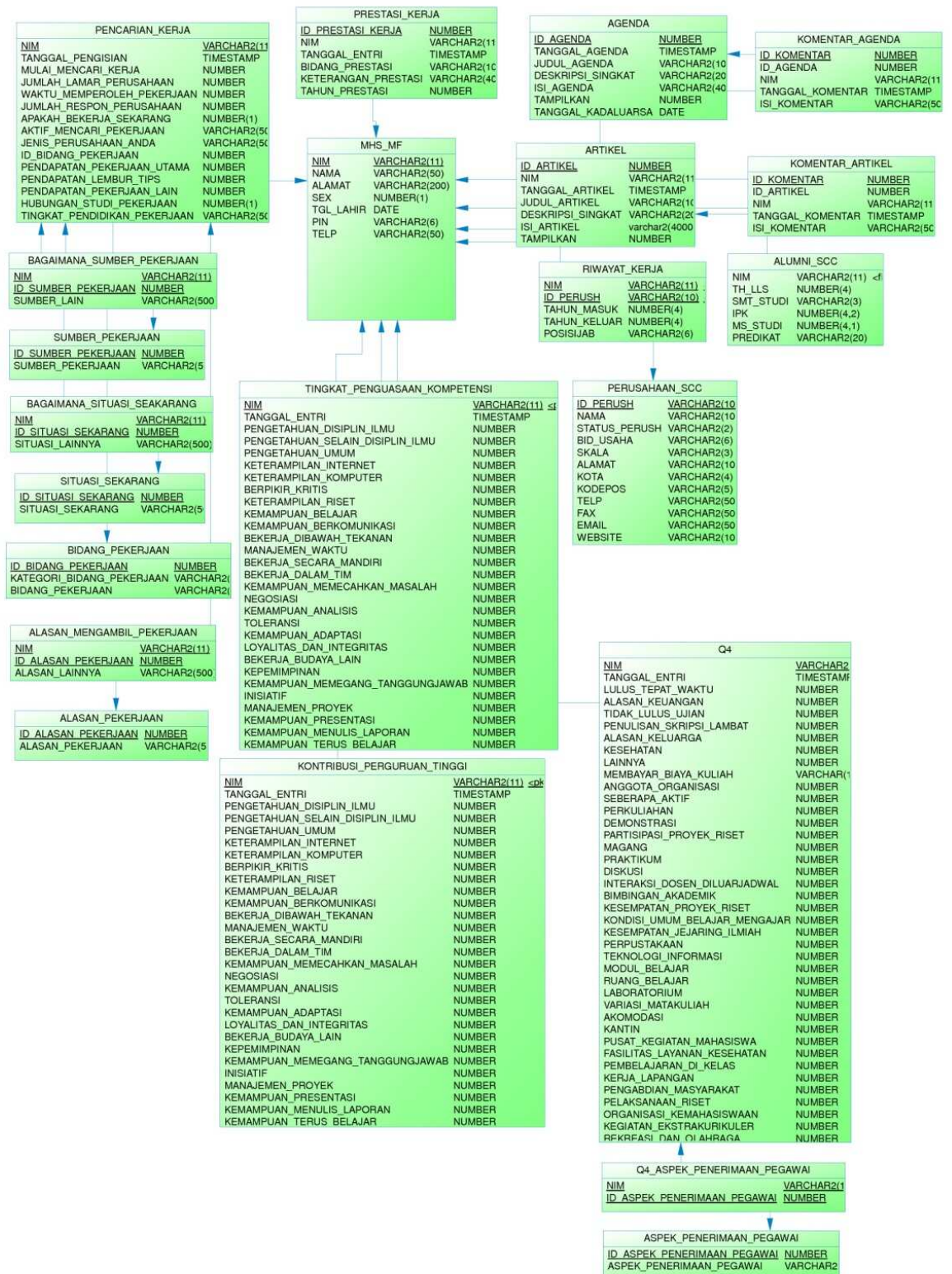
Gambar 3.48 Diagram *Deployment*

Diagram *deployment* di atas menjelaskan bahwa kode-kode yang telah ditulis dipasang pada sebuah *web server*. *Web server* tersebut terkoneksi dengan internet melalui sebuah modem. Sedangkan *database* yang digunakan terdapat pada *server* yang berbeda. Setiap perangkat tersebut dihubungkan dengan kabel.

### 3.2.21 Physical Data Model

*Physical data model* menggambarkan tentang relasi tabel-tabel yang ada di dalam database. Beberapa tabel yang digunakan diambil dari sistem yang telah ada karena sistem yang baru akan diintegrasikan dengan sistem yang sudah ada. Desain database dapat dilihat pada gambar 3.49.





Gambar 3.49 Physical Data Model

### 3.3 Desain Input/Output

Desain input/output digunakan untuk menampilkan rancangan sesuai dengan kebutuhan pengguna. Fungsi-fungsi sebuah *form* dan bagian-bagiannya dijelaskan di sini. Melalui desain input/output pengembang dapat mengetahui tampilan *website* yang akan dibangun.

#### 3.3.1 Desain Input/Output Halaman Login

<a href="#">Beranda</a>	<a href="#">Pencarian Alumni</a>	<a href="#">Lowongan Pekerjaan</a>	<a href="#">Tracer Study</a>
-------------------------	----------------------------------	------------------------------------	------------------------------

Username:

Password:

Login sebagai:

Footer

Gambar 3.50 Desain Input/Output Halaman Login

Halaman login digunakan pengunjung untuk masuk ke dalam sistem dengan memasukkan *username* dan *password*. Alumni, SCC, pengurus alumni, dan kaprodi masuk ke dalam sistem melalui halaman ini. Halaman ini juga dapat

menampilkan pesan kesalahan kepada pengguna apabila *username* dan *password* yang dimasukkan salah.

### 3.3.2 Desain Input/Output Halaman Menjawab Kuesioner *Tracer Study*

The image shows a web-based questionnaire interface. At the top, there are five navigation tabs: 'Keluarga Alumni', 'Bagian 1', 'Bagian 2', 'Bagian 3', and 'Bagian 4'. Below the tabs is the main content area titled '[JUDUL KUESIONER]'. It contains four numbered questions:

- 1. XXXXXXXXXXXXXXX: A text input field.
- 2. XXXXXXXXXXXXXXX: Four checkboxes. The first, second, and fourth are checked (green checkmarks). The third is unchecked (empty box).
- 3. XXXXXXXXXXXXXXX: A dropdown menu.
- 4. XXXXXXXXXXXXXXX: Three radio buttons. The third one is selected (green dot).

At the bottom of the form, there are two buttons: 'Simpan' (Save) and 'Reset'.

Gambar 3.51 Desain Input/Output Halaman Kuesioner *Tracer Study*

Halaman kuesioner merupakan halaman yang digunakan alumni untuk mengikuti kegiatan *tracer study*. Isi halaman ini disesuaikan dengan kuesioner yang ada dari DIKTI. Ketika alumni menekan tombol simpan sistem akan memeriksa data yang dimasukkan oleh pengguna. Apabila data yang dimasukkan alumni tidak valid, maka sistem akan menampilkan pesan kesalahan input kepada alumni. Apabila seluruh data yang telah dimasukkan oleh alumni telah valid, maka sistem akan menyimpan data-data tersebut kemudian mengarahkan alumni kepada halaman kuesioner yang selanjutnya.



### 3.3.3 Desain Input/Output Halaman Manage Prestasi Kerja

Gambar 3.52 Rancangan Halaman Prestasi Kerja

Alumni memasukkan data prestasi kerja melalui halaman ini. Data prestasi kerja yang dibutuhkan antara lain adalah bidang prestasi, tahun prestasi, dan keterangan prestasi. Data prestasi yang dimasukkan alumni melalui halaman ini akan ditampilkan pada halaman awal.

### 3.3.4 Desain Input/Output Halaman *Broadcast Email*

Halaman ini digunakan oleh pengurus alumni dan SCC untuk mengirimkan email secara massal kepada alumni. Pengurus alumni dapat mengirimkan email kepada alumni tertentu berdasarkan jenis penyaringan yang dipilih. Penyaringan data yang dapat dilakukan oleh pengurus alumni dan SCC adalah berdasarkan jurusan alumni dan tahun lulusan alumni.

HEADER	
<p><b>ARTIKEL</b>  <a href="#">Artikel Baru</a>  <a href="#">Komentar Artikel</a></p> <p><b>AGENDA/PROGRAM</b>  <a href="#">Agenda Baru</a>  <a href="#">Komentar Agenda</a>  <a href="#">Broadcast Email</a></p> <p><b>PRESTASI KERJA ALUMNI</b>  <a href="#">Prestasi</a></p> <p><b>HASIL TRACER STUDY</b>  <a href="#">Bagian 1</a>  <a href="#">Bagian 2</a>  <a href="#">Bagian 3</a>  <a href="#">Bagian 4</a>  <a href="#">Hasil Uji Statistik</a></p>	<p>Filter</p> <p>Jurusan <input type="text"/></p> <p>Lulusan <input type="text"/></p> <p>Subjek Email  <input type="text"/></p> <p>Isi Email</p> <p>Rich Textbox  <input type="text"/></p> <p><input type="button" value="Kirim"/></p>

Gambar 3.53 Desain Input/Output Halaman *Broadcast Email*

Isi email adalah sebuah *rich textbox* sehingga pengurus alumni dan SCC dapat memasukkan huruf dengan format-format tertentu. Selain itu pengurus alumni dan SCC dapat memasukkan gambar ke dalam isi email tersebut. Setelah pengurus alumni dan SCC memasukkan subjek email, isi email, dan menekan tombol kirim maka sistem akan memeriksa data yang dimasukkan oleh pengurus alumni dan SCC. Apabila data-data yang diperlukan dalam mengirim email telah valid maka sistem akan mengirimkan email kepada alumni satu per satu.

### 3.3.5 Desain Input/Output Halaman *Manage Agenda*

Halaman ini berfungsi untuk mengelola data agenda yang dituliskan oleh pengurus alumni. Halaman *manage* artikel hanya dapat diakses oleh pengurus alumni. Melalui halaman ini pengurus alumni dapat menampilkan data agenda yang ada di dalam sistem. Pengurus alumni dapat mengubah dan menghapus data agenda.

HEADER						
<a href="#">ARTIKEL</a> <a href="#">Artikel Baru</a> <a href="#">Komentar Artikel</a>  <a href="#">AGENDA/PROGRAM</a> <a href="#">Agenda Baru</a> <a href="#">Komentar Agenda</a> <a href="#">Broadcast Email</a>  <a href="#">PRESTASI KERJA ALUMNI</a> <a href="#">Prestasi</a>  <a href="#">HASIL TRACER STUDY</a> <a href="#">Bagian 1</a> <a href="#">Bagian 2</a> <a href="#">Bagian 3</a> <a href="#">Bagian 4</a> <a href="#">Hasil Uji Statistik</a>	Tanggal	Judul Agenda	Deskripsi Singkat	Kadaluarsa		
					<a href="#">Edit</a>	<a href="#">Hapus</a>

Gambar 3.54 Desain Input/Output Halaman *Manage* Agenda

### 3.3.6 Desain Input/Output Halaman *Manage* Artikel

HEADER						
<a href="#">ARTIKEL</a> <a href="#">Artikel Baru</a> <a href="#">Komentar Artikel</a>  <a href="#">AGENDA/PROGRAM</a> <a href="#">Agenda Baru</a> <a href="#">Komentar Agenda</a> <a href="#">Broadcast Email</a>  <a href="#">PRESTASI KERJA ALUMNI</a> <a href="#">Prestasi</a>  <a href="#">HASIL TRACER STUDY</a> <a href="#">Bagian 1</a> <a href="#">Bagian 2</a> <a href="#">Bagian 3</a> <a href="#">Bagian 4</a> <a href="#">Hasil Uji Statistik</a>	Tanggal	Judul	Deskripsi Singkat	Tampil		
					<a href="#">Edit</a>	<a href="#">Hapus</a>

Gambar 3.55 Desain Input/Output Halaman *Manage* Artikel

Halaman ini berfungsi untuk mengelola data artikel yang dituliskan oleh alumni. Halaman *manage* artikel hanya dapat diakses oleh pengurus alumni. Melalui halaman ini pengurus alumni dapat menampilkan data artikel yang ada di dalam sistem. Pengurus alumni dapat mengubah dan menghapus data artikel.

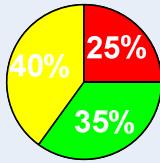
### 3.3.7 Desain Input/Output Halaman Pencarian Alumni

Gambar 3.56 Desain Input/Output Halaman Pencarian Alumni

Halaman pencarian alumni digunakan pengunjung untuk mencari data alumni. Pengunjung dapat melakukan pencarian data alumni berdasarkan nama, jurusan, dan tahun angkatan alumni. Data hasil pencarian yang ditampilkan antara lain adalah NIM, nama, dan jurusan alumni.

Melalui halaman ini juga alumni dapat melihat riwayat pekerjaan alumni tersebut. Selain itu untuk kebutuhan berkomunikasi antar alumni, melalui halaman ini alumni dapat mengirimkan email secara langsung kepada alumni yang dimaksud. Alumni dapat melakukannya dengan menekan tautan kirim email. Kemudian alumni akan diarahkan ke halaman mengirim email.

### 3.3.8 Desain Output Laporan Hasil *Tracer Study*

HEADER											
<p><b>ARTIKEL</b>  <a href="#">Artikel Baru</a>  <a href="#">Komentar Artikel</a></p> <p><b>AGENDA/PROGRAM</b>  <a href="#">Agenda Baru</a>  <a href="#">Komentar Agenda</a>  <a href="#">Broadcast Email</a></p> <p><b>PRESTASI KERJA ALUMNI</b>  <a href="#">Prestasi</a></p> <p><b>HASIL TRACER STUDY</b>  <a href="#">Bagian 1</a>  <a href="#">Bagian 2</a>  <a href="#">Bagian 3</a>  <a href="#">Bagian 4</a>  <a href="#">Hasil Uji Statistik</a></p>	<p><b>Filter</b></p> <p>Jurusan <input type="text"/></p> <p>Lulusan <input type="text"/></p> <p><b>Hasil Tracer Study Alumni</b></p> <table> <tr> <td>Jurusan</td> <td>XXXXXXXXXXXXXXXXXX</td> </tr> <tr> <td>Tahun Lulusan</td> <td>XXXXXXXXXXXXXXXXXX</td> </tr> <tr> <td>XXXXXXXXXXXXXXXXXX</td> <td>XXXXXXXXXXXXXXXXXX</td> </tr> <tr> <td>XXXXXXXXXXXXXXXXXX</td> <td>XXXXXXXXXXXXXXXXXX</td> </tr> <tr> <td>XXXXXXXXXXXXXXXXXX</td> <td>XXXXXXXXXXXXXXXXXX</td> </tr> </table> <p><b>[KETERANGAN GRAFIK]</b></p>  <p>A pie chart with three segments: a yellow segment labeled 40%, a red segment labeled 25%, and a green segment labeled 35%.</p>	Jurusan	XXXXXXXXXXXXXXXXXX	Tahun Lulusan	XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXX
Jurusan	XXXXXXXXXXXXXXXXXX										
Tahun Lulusan	XXXXXXXXXXXXXXXXXX										
XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXX										
XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXX										
XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXX										

Gambar 3.57 Desain Output Laporan Hasil *Tracer Study*

Laporan hasil *tracer study* hanya dapat diakses oleh SCC dan kaprodi. Halaman ini berguna untuk menampilkan hasil pengolahan data dari kuesioner yang telah dijawab oleh alumni. Pada halaman ini, data-data yang diperoleh dari alumni ditampilkan dalam bentuk grafik pai dan batang. SCC dan kaprodi juga dapat menyaring data-data yang akan ditampilkan berdasarkan jurusan alumni dan tahun lulusan alumni.

### 3.4 Rancangan Pengujian

Untuk memastikan apakah sistem yang dirancang telah memenuhi kebutuhan pengguna dilakukan pengujian dengan pendekatan metode *black-box testing*. Dalam uji coba ini dilakukan dengan cara memasukkan data-data tertentu dan penggunaan fungsi-fungsi kontrol yang ada pada suatu *form*. Berikut ini adalah hal-hal yang akan diujikan:

Tabel 3.12 Rencana Pengujian

Requirement yang diuji	Fungsi yang diuji
Proses Login	Melakukan login
Halaman kuesioner <i>tracer study</i>	Input data pada halaman kuesioner <i>tracer study</i>
Halaman <i>manage</i> prestasi kerja alumni	Mengelola data prestasi kerja alumni
Halaman <i>broadcast</i> email	Mengirim email massal
Halaman <i>manage</i> agenda	Mengelola data agenda
Halaman <i>manage</i> artikel	Mengelola data artikel
Pencarian data alumni	Menampilkan data hasil pencarian
Halaman hasil <i>tracer study</i>	Menampilkan hasil pengolahan data <i>tracer study</i>

### 3.4.1 Desain Uji Coba Proses Login

Desain uji coba proses login bertujuan untuk mengetahui apakah proses login dapat berfungsi dengan baik. Uji coba ini dilakukan dengan memasukkan *username* dan *password* yang valid dan tidak valid. Desain uji coba ini dijelaskan pada tabel di bawah ini:

Tabel 3.13 Desain Uji Coba Proses Login

Test Case	Tujuan	Input	Output Diharapkan	Status
1	Pengujian validitas <i>username</i> dan <i>password</i> untuk jenis pengguna alumni.	<i>Username</i> =07390100043; <i>password</i> =386584; dan login sebagai=alumni. Kemudian klik tombol login.	Sistem akan mengarahkan alumni kepada halaman awal dengan menu-menu yang disediakan untuk alumni.	
2	Pengujian validitas <i>username</i> dan <i>password</i> untuk jenis pengguna pengurus alumni.	<i>Username</i> =admin; <i>password</i> =admin; dan login sebagai=alumni. Kemudian klik tombol login.	Sistem langsung mengarahkan pengurus alumni kepada halaman administrator untuk pengurus alumni.	
3	Pengujian validitas <i>username</i> dan	Pengujian validitas <i>username</i> dan <i>password</i> untuk jenis pengguna	Sistem langsung mengarahkan SCC kepada	

<i>Test Case</i>	Tujuan	Input	Output Diharapkan	Status
	<i>password</i> untuk jenis pengguna SCC.	SCC.	halaman administrator untuk SCC.	
4	Pengujian validitas <i>username</i> dan <i>password</i> untuk jenis pengguna kaprodi.	Pengujian validitas <i>username</i> dan <i>password</i> untuk jenis pengguna kaprodi.	Sistem langsung mengarahkan kaprodi kepada halaman administrator untuk kaprodi.	
5	Pengujian terhadap input <i>username</i> dan <i>password</i> yang salah.	<i>Username</i> =andrias; <i>password</i> =ganteng; dan login sebagai=alumni. Kemudian klik tombol login.	Terdapat pesan kesalahan di bawah <i>form</i> login.	

### 3.4.2 Desain Uji Coba Halaman Menjawab Kuesioner *Tracer Study*

Desain uji coba ini dilakukan untuk mengetahui apakah halaman kuesioner *tracer study* sudah sesuai dengan sifat-sifat kuesioner *tracer study* yang ada. Uji coba ini dilakukan dengan menekan kontrol yang ada pada halaman *tracer study* dan memasukkan beberapa data. Tabel berikut ini menjelaskan uji coba yang akan dilakukan:

Tabel 3.14 Desain Uji Coba Halaman Kuesioner *Tracer Study*

<i>Test Case</i>	Tujuan	Input	Output Diharapkan	Status
6	Memastikan bahwa alumni yang dapat mengisi kuesioner hanya alumni yang telah mengupdate datanya melalui <i>website</i> SCC.	Menekan tautan <i>Tracer Study</i> .	Halaman pemberitahuan bahwa alumni harus mengupdate datanya terlebih dahulu melalui <i>website</i> SCC.	
7	Memastikan bahwa proses penyimpanan jawaban responden terhadap kuesioner	Seluruh data yang diperlukan pada sebuah kuesioner.	Pengguna diarahkan kepada kuesioner bagian selanjutnya.	

<i>Test Case</i>	Tujuan	Input	Output Diharapkan	Status
	berjalan dengan baik.			
8	Responden tidak melewati satu pertanyaan yang ada pada kuesioner <i>tracer study</i> .	Masukkan semua data yang diperlukan pada kuesioner kecuali satu pertanyaan.	Sistem menampilkan pesan kepada pengguna bahwa terdapat pertanyaan yang belum terisi.	
9	Menghindari input dari responden yang tidak diinginkan.	Huruf pada <i>field</i> pendapatan.	Pesan peringatan kepada pengguna bahwa input untuk pendapatan harus angka.	
10	Menguji fungsi tombol reset untuk mengembalikan <i>form tracer study</i> kepada keadaan semula.	Menekan tombol reset.	<i>Form tracer study</i> bersih dari jawaban yang telah diinputkan alumni dan kembali kepada keadaan semula.	
11	Memastikan bahwa bagian kuesioner yang telah diisi tidak dapat dibuka untuk menghindari pengisian kuesioner lebih dari satu kali.	Mengisi kuesioner bagian 2 secara lengkap. Kemudian menekan tautan bagian 2.	Sistem akan mengarahkan pengguna kepada bagian kuesioner yang belum diisi.	

S U R A B A Y A

### 3.4.3 Desain Uji Coba Halaman Manage Prestasi Kerja

Uji coba ini dilakukan untuk mengetahui apakah fungsi-fungsi yang terdapat pada halaman prestasi kerja alumni dapat digunakan dengan baik. Uji coba dilakukan dengan menginputkan beberapa data valid dan data yang kosong. Uji coba halaman prestasi kerja dijelaskan pada tabel berikut ini:

Tabel 3.15 Desain Uji Coba Halaman Manage Prestasi Kerja

<i>Test Case</i>	Tujuan	Input	Output Diharapkan	Status
12	Pengujian proses penyimpanan data	Bidang prestasi, dan tahun,	Bertambahnya daftar prestasi kerja yang	



	prestasi kerja.	keterangan. Kemudian klik tombol simpan.	berada pada bagian kanan atas.	
13	Pengujian terhadap proses penghapusan data prestasi kerja.	Klik tautan hapus pada bagian kanan atas.	Berkurangnya daftar prestasi kerja yang berada pada bagian kanan atas.	
14	Menguji respon sistem apabila pengguna memasukkan data kosong.	Klik tombol simpan.	Muncul pesan kesalahan bahwa bidang prestasi tidak boleh kosong.	

#### 3.4.4 Desain Uji Coba Halaman *Broadcast Email*

Uji coba pada halaman ini dilakukan untuk mengetahui apakah halaman broadcast email dapat berfungsi dengan baik. Uji coba dilakukan dengan memilih jenis penyaringan data berdasarkan jurusan atau tahun lulus, kemudian diikuti dengan memasukkan subjek email dan isi email. Tabel berikut ini menjelaskan uji coba yang akan dilakukan:

Tabel 3.16 Desain Uji Coba Halaman *Broadcast Email*

<i>Test Case</i>	Tujuan	Input	Output Diharapkan	Status
15	Memastikan sistem dapat mengirimkan email dengan baik.	Subjek email, isi email, dan klik pada tombol kirim.	Sistem akan menampilkan pesan bahwa email berhasil dikirim.	

#### 3.4.5 Desain Uji Coba Halaman *Manage Agenda*

Halaman *manage agenda* perlu diuji untuk mengetahui apakah halaman tersebut dapat berfungsi dengan baik. Pengujian halaman tersebut dilakukan dengan menggunakan tombol atau tautan yang telah disediakan. Tabel berikut ini menjelaskan uji coba yang akan dilakukan pada halaman *manage agenda*:

Tabel 3.17 Desain Uji Coba Halaman *Manage Agenda*

<i>Test Case</i>	Tujuan	Input	Output Diharapkan	Status
16	Pengujian terhadap fungsi tautan judul agenda.	Klik pada tautan judul agenda.	Sistem mengarahkan pengguna kepada halaman agenda yang dimaksud.	
17	Menguji fungsi tautan edit agenda.	Klik pada tautan edit.	Sistem mengarahkan pengguna kepada halaman mengedit agenda yang dimaksud.	
18	Menguji fungsi tautan hapus agenda.	Klik pada tautan hapus.	Sistem akan menampilkan pesan konfirmasi kepada pengguna apakah benar-benar akan menghapus agenda yang dimaksud.	

### 3.4.6 Desain Uji Coba Halaman *Manage Artikel*

Halaman *manage* artikel digunakan untuk mengelola data artikel yang dituliskan. Uji coba pada halaman ini dilakukan dengan menggunakan tautan-tautan yang telah disediakan apakah dapat berfungsi dengan baik. Uji coba yang akan dilakukan dijelaskan pada tabel berikut ini:

Tabel 3.18 Desain Uji Coba Halaman *Manage Artikel*

<i>Test Case</i>	Tujuan	Input	Output Diharapkan	Status
19	Pengujian terhadap fungsi tautan judul artikel.	Klik pada tautan judul artikel.	Sistem mengarahkan pengguna kepada halaman artikel yang dimaksud.	
20	Menguji fungsi tautan edit artikel.	Klik pada tautan edit.	Sistem mengarahkan pengguna kepada halaman mengedit artikel yang dimaksud.	
21	Menguji fungsi	Klik pada tautan	Sistem akan	

<i>Test Case</i>	Tujuan	Input	Output Diharapkan	Status
	tautan hapus artikel	hapus.	menampilkan pesan konfirmasi kepada pengguna apakah benar-benar akan menghapus artikel yang dimaksud.	

### 3.4.7 Desain Uji Coba Halaman Pencarian Alumni

Desain uji coba halaman pencarian alumni digunakan untuk memastikan apakah halaman pencarian alumni dapat berfungsi dengan baik.

Tabel 3.19 Desain Uji Coba Halaman Pencarian Alumni

<i>Test Case</i>	Tujuan	Input	Output Diharapkan	Status
22	Pengujian hasil data yang memang ada.	Nama alumni=Andrias, jurusan, dan tahun angkatan. Kemudian klik tombol cari.	Tabel alumni hasil pencarian yang berisi NIM, nama alumni, dan jurusan.	Sukses
23	Pengujian hasil data yang tidak ada.	Nama alumni=Ganteng, jurusan, dan tahun angkatan. Kemudian klik tombol cari.	Pesan bahwa data alumni tidak ditemukan.	Sukses

### 3.4.8 Desain Uji Coba Halaman Hasil *Tracer Study*

Halaman mengakses hasil *tracer study* digunakan untuk menampilkan data yang telah diolah. Halaman ini tidak memerlukan input kecuali untuk penyaringan data. Pengujian pada halaman ini dilakukan pada fungsi penyaringan data apakah dapat berfungsi dengan baik. Uji coba yang akan dilakukan dijelaskan pada tabel berikut ini:

Tabel 3.20 Desain Uji Coba Halaman Hasil *Tracer Study*

<i>Test Case</i>	Tujuan	Input	Output Diharapkan	Status
24	Pengujian terhadap fungsi penyaring data berdasarkan jurusan.	Pilih jurusan kemudian klik tombol filter.	Hasil <i>tracer study</i> berdasarkan jurusan yang dipilih.	
25	Pengujian terhadap fungsi penyaring data berdasarkan tahun lulusan.	Pilih tahun lulusan kemudian klik tombol filter.	Hasil <i>tracer study</i> berdasarkan tahun lulusan yang dipilih.	
26	Pengujian terhadap fungsi penyaring data berdasarkan jurusan dan tahun lulusan responden.	Pilih jurusan, pilih tahun lulusan, dan klik tombol filter.	Hasil <i>tracer study</i> berdasarkan jurusan dan tahun lulusan responden.	
27	Pengujian terhadap hasil <i>tracer study</i> tanpa penyaringan data.	Klik tombol filter.	Hasil <i>tracer study</i> dihitung dari semua jurusan dan semua tahun lulusan.	

Uji coba juga perlu dilakukan pada informasi yang dihasilkan, apakah valid atau tidak. Uji coba ini dilakukan pada sebuah grafik yang akan mewakili grafik-grafik yang lain dan sebuah uji *chi-square* apakah aplikasi dapat menghasilkan nilai *chi-square* hitung dengan benar. Data yang digunakan adalah data yang didapatkan dari hasil *tracer study* pada jurusan Manajemen Informatika yang lulus pada tahun 2010.

Tabel 3.21 Desain Uji Coba Output Hasil *Tracer Study*

Test Case	Tujuan	Input	Output Diharapkan	Status
28	Memastikan bahwa grafik pendapatan alumni dari pekerjaan utama menunjukkan jumlah data alumni yang valid.	Hasil <i>tracer study</i> terakhir pada alumni jurusan Manajemen Informatika lulusan tahun 2010. Pada penyaringan	Grafik menunjukkan jumlah data yang sesuai dengan jumlah responden yaitu 14 responden.	

Test Case	Tujuan	Input	Output Diharapkan	Status
		data pilih jurusan Manajemen Informatika dan pilih tahun 2010 pada tahun lulus. Tekan tombol filter.		
29	Memastikan bahwa perhitungan rata-rata untuk data pengetahuan di bidang atau disiplin ilmu bagian 2 (dua) adalah valid.	Hasil <i>tracer study</i> terakhir pada alumni jurusan Manajemen Informatika lulusan tahun 2010. Pada penyaringan data pilih jurusan Manajemen Informatika dan pilih tahun 2010 pada tahun lulus. Tekan tombol filter.	Hasil penghitungan rata-rata yaitu 3,7059 atau jika dibulatkan adalah 4 (empat) yang berarti tinggi.	
30	Memastikan bahwa perhitungan uji <i>chi-square</i> antara indeks prestasi kumulatif (IPK) dengan waktu yang dihabiskan untuk memperoleh pekerjaan menunjukkan nilai <i>chi-square</i> hitung yang valid.	Hasil <i>tracer study</i> terakhir pada alumni jurusan Manajemen Informatika lulusan tahun 2010 sebanyak 17 orang. Pada penyaringan data pilih jurusan DIII Manajemen Informatika dan 2010 pada tahun lulus. Kemudian pada uji statistik pilih pendapatan alumni dari pekerjaan utama pada variabel yang kedua.	Uji <i>chi-square</i> menunjukkan nilai <i>chi-square</i> hitung yang valid yaitu 9.9648264809903.	