



# PENGEMBANGAN SISTEM KONSULTASI DAN PENCARIAN PENGACARA BERBASIS WEBSITE

SKRIPSI

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

Entin Endah Cahyati

NIM: 155150200111078



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2020





# PENGESAHAN

PENGEMBANGAN SISTEM KONSULTASI DAN PENCARIAN PENGACARA BERBASIS

WEBSITE

SKRIPSI

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Entin Endah Cahyati

NIM: 155150200111078

Skripsi ini telah diuji dan dinyatakan lulus pada

23 Juli 2020

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing 1

Dosen Pembimbing 2

Nurudin Santoso, S.T., M.T

Lutfi Fanani, S.Kom., M.T., M.Sc.

NIP: 197409162000121001

NIK: 2016078902171001

Mengetahui

Ketua Jurusan Teknik Informatika



Achmad Basuki, S.T., M.MG, Ph.D

NIP: 197411182003121002





## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 23 Juli 2020



Entin Endah Cahyati

NIM: 155150200111078





## PRAKATA

Puji syukur atas kehadiran Allah SWT yang telah memberikan berkah, rahmat, dan hidayah untuk penulis sehingga skripsi dengan judul “Pengembangan Sistem Konsultasi dan Pencarian Pengacara Berbasis Website” dapat Penulis selesaikan. Melalui prakata ini, penulis ingin menyampaikan ucapan terima kasih kepada berbagai pihak, karena telah membantu dan memberikan dorongan dalam menyelesaikan penyusunan skripsi ini. Untuk itu penulis ingin mengucapkan terima kasih kepada:

1. Kedua orang tua penulis, yaitu Bapak Idiatmoko Cahyono dan Ibu Hariyati serta Adik penulis yaitu M. Andra Dwi Alvaro juga seluruh keluarga besar yang telah memberikan dukungan, doa, dan motivasi yang tidak ada habisnya.
2. Bapak Nurudin Santoso, S.T., M.T. selaku dosen pembimbingan 1 yang telah memberikan ilmu, masukan, dan saran yang bermanfaat dalam menyelesaikan skripsi ini.
3. Bapak Lutfi Fanani, S.Kom, M.T., M.Sc. selaku dosen pembimbing 2 yang telah memberikan ilmu, masukan, dan saran yang bermanfaat dalam menyelesaikan skripsi ini.
4. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D, Bapak Dr. Eng. Herman Tolle, S.T, MT, Bapak Agus Wahyu Widodo, S.T, M.Cs, dan Bapak Drs. Muh. Arif Rahman, M.Kom selaku Dekan, Wakil Dekan 1, Wakil Dekan 2, dan Wakil Dekan 3 Fakultas Ilmu Komputer, Universitas Brawijaya.
5. Bapak Achmad Basuki, S.T, M.MG, Ph.D dan Bapak Adhitya Bhawiyuga, S.Kom, M.Sc selaku Ketua Jurusan Teknik Informatika dan Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer, Universitas Brawijaya.
6. Seluruh Dosen Fakultas Ilmu Komputer yang telah memberikan ilmu yang bermanfaat bagi penulis.
7. Bapak Yayan Riyanto, S.H, M.H dan Ibu Rafiq Aswindadesovi, S.H yang sudah bersedia menjadi narasumber penulis.
8. Sahabat penulis Anindita Maya Pramudina yang selama ini jadi tempat keluh kesah penulis dan selalu memberikan semangat pada penulis.
9. Teman-teman penulis yaitu Nurul, Desty, Yuni, Elsa, Amel, Jojor, dan teman lainnya yang tidak bisa disebutkan satu-satu disini yang selalu membantu dan memberikan semangat kepada penulis.
10. Semua pihak yang tidak bisa penulis sebutkan yang telah membantu dalam pengerjaan skripsi dan juga yang telah memberikan semangat dan motivasi kepada penulis.





Penulis menyadari jika dalam penyusunan skripsi ini masih terdapat banyak kekurangan. Untuk itu penulis sangat membutuhkan kritik dan saran yang membangun. Semoga penelitian skripsi ini dapat bermanfaat untuk kita semua.

Malang, 23 Juli 2020

Penulis  
entincahya96@gmail.com



## ABSTRAK

**Entin Endah Cahyati, Pengembangan Sistem Konsultasi dan Pencarian Pengacara Berbasis Website**

**Pembimbing: Nurudin Santoso, S.T., M.T dan Lutfi Fanani, S.Kom., M.T., M.Sc.**

Hukum adalah kaidah atau aturan dalam bentuk tertulis maupun tidak tertulis, yang tersusun menjadi suatu sistem yang menentukan apa yang boleh dan tidak boleh dilakukan oleh manusia dalam kehidupan bermasyarakat. Keberadaan hukum bertujuan untuk melindungi setiap individu dari penyalahgunaan kekuasaan serta untuk menegakkan keadilan. Dengan adanya hukum di suatu negara, maka setiap orang di negara tersebut berhak mendapatkan keadilan dan pembelaan di depan hukum yang berlaku. Hukum ini juga bertujuan untuk menjaga dan mencegah orang agar tidak menjadi hakim atas dirinya sendiri. Untuk melindungi dan mencegah penyalahgunaan hukum, kita memerlukan seorang yang paham tentang permasalahan yang ada pada ruang lingkup hukum yaitu pengacara. Memilih pengacara yang sesuai dengan kasus yang kita miliki dengan kasus-kasus yang telah ditangani oleh pengacara. Sebelum menentukan cocok atau tidaknya kasus kita dengan pengacara dibutuhkan konsultasi terlebih dahulu mengenai kasus yang kita hadapi. Saat ini banyak yang membutuhkan pengacara tapi sulit untuk mendapatkan pengacara. Banyak penyebab orang sulit menemukan pengacara, salah satunya ketidakcocokan antara pengacara dan klien. Untuk mengetahui cocok atau tidaknya pengacara dengan klien, klien harus konsultasi terlebih dahulu dengan pengacara. Untuk mempermudah pencarian dan konsultasi pengacara maka disediakan sistem yang dapat digunakan untuk berkonsultasi dan melakukan pencarian pengacara. Proses untuk mengembangkan penelitian ini dengan menggunakan metode Rapid Application Development (RAD) dengan pendekatan Object-Oriented (OO) dan berbasis website. Tahapan yang digunakan dalam mengembangkan sistem ini adalah studi pustaka, analisis kebutuhan, perancangan, implementasi, pengujian, dan penarikan kesimpulan dan saran. Penelitian ini menggunakan 4 pengujian yaitu pengujian unit, pengujian integrasi, pengujian validasi, dan pengujian compatibility. Pengujian unit, integrasi, dan validasi menghasilkan 100% valid. Sedangkan untuk pengujian compatibility menghasilkan sistem dapat berjalan dengan baik pada 6 peramban yang berbeda.

Kata kunci: pengacara, hukum, pengembangan perangkat lunak, *rapid application development*



## ABSTRACT

**Entin Endah Chayati, Development of Web-Based System for Consultation and Lawyer Search**

**Supervisors: Nurudin Santoso, S.T., M.T dan Lutfi Fanani, S.Kom., M.T., M.Sc.**

*Law is rules in written or unwritten form, which are arranged into a system that determines what can and cannot be done by humans in social life. The existence of law aims to protect every individual from abuse of power and to uphold justice. With the existence of law in a country, every person in the country is entitled to justice and defense in the face of prevailing laws. This law also aims to protect and prevent people from becoming judges of themselves. To protect and prevent misuses of the law, we need someone who understands the problem that exist within the scope of the law, namely lawyers. Choose a lawyer that matches the case that we have with the cases that have been handled by the lawyer. Before determining the suitability of our cases with a lawyer, a consultation is needed about the case we are facing. Today many people need lawyers but to get it is difficult. Many causes people find it difficult to find a lawyer, which one is the mismatch between lawyer and client. To find out whether a lawyer is suitable or not with a client, first the client must consult with the lawyer. To facilitate the search and consultation of lawyers, a system is provided that can be used to consult and search for lawyers. The process for developing this research is using the Rapid Application Development (RAD) method with an Object-Oriented (OO) approach and website-based. The stages used in developing this system are literature study, needs analysis, design, implementation, testing, and drawing conclusion and suggestion. This research uses 4 test namely unit testing, integration testing, validation testing, and compatibility testing. Unit, integration, and validation testing yield 100% valid. Ad for compatibility testing, the system works well on 6 different browsers.*

**Keywords :** *lawyers, law, software development, rapid application development*





## DAFTAR ISI

|  |      |
|--|------|
| PENGESAHAN .....   | ii   |
| PERNYATAAN ORISINALITAS .....                                | iii  |
| PRAKATA .....  | iv   |
| ABSTRAK .....  | vi   |
| ABSTRACT .....   | vii  |
| DAFTAR ISI .....   | viii |
| DAFTAR TABEL .....   | xiii |
| DAFTAR GAMBAR .....  | xx   |
| DAFTAR LAMPIRAN .....  | xxii |
| BAB 1 PENDAHULUAN .....                                      | 1    |
| 1.1 Latar Belakang .....                                     | 1    |
| 1.2 Rumusan Masalah .....                                    | 2    |
| 1.3 Tujuan .....   | 2    |
| 1.4 Manfaat .....  | 2    |
| 1.5 Batasan Masalah .....                                    | 3    |
| 1.6 Sistematika Pembahasan .....                             | 3    |
| BAB 2 LANDASAN KEPUSTAKAAN .....                             | 5    |
| 2.1 Kajian Pustaka .....                                     | 5    |
| 2.2 <i>Work Breakdown Structure</i> (WBS) .....              | 6    |
| 2.3 <i>Software Development Life Cycle</i> (SDLC) .....      | 6    |
| 2.3.1 <i>Rapid Application Development</i> (RAD) Model ..... | 6    |
| 2.4 Analisis Kebutuhan .....                                 | 7    |
| 2.5 <i>Unified Modelling Language</i> (UML) .....            | 8    |
| 2.5.1 <i>Use Case Diagram</i> .....                          | 8    |
| 2.5.2 <i>Sequence Diagram</i> .....                          | 9    |
| 2.5.3 <i>Class Diagram</i> .....                             | 10   |
| 2.6 <i>Model View Controller</i> (MVC) .....                 | 11   |
| 2.7 CodeIgniter .....  | 12   |
| 2.8 Bahasa Pemrograman .....                                 | 12   |
| 2.8.1 HTML .....   | 12   |





|   |           |
|---|-----------|
| 2.8.2 PHP .....                                 | 13        |
| 2.8.3 CSS .....                                 | 13        |
| 2.8.4 JavaScript .....                          | 13        |
| 2.9 MySQL .....                                 | 14        |
| 2.10 Pengujian Perangkat Lunak .....            | 14        |
| 2.10.1 Pengujian <i>White-Box</i> .....         | 14        |
| 2.10.2 Pengujian <i>Black-Box</i> .....         | 14        |
| 2.11 Aplikasi Penunjang .....                   | 15        |
| 2.11.1 <i>Sublime Text</i> .....                | 15        |
| 2.11.2 XAMPP .....                              | 15        |
| <b>BAB 3 METODOLOGI PENELITIAN .....</b>        | <b>16</b> |
| 3.1 Studi Pustaka .....                         | 16        |
| 3.2 Rekayasa Kebutuhan .....                    | 17        |
| 3.3 Perancangan Sistem .....                    | 17        |
| 3.4 Implementasi Sistem .....                   | 19        |
| 3.5 Pengujian Sistem .....                      | 19        |
| 3.6 Kesimpulan dan Saran .....                  | 20        |
| <b>BAB 4 REKAYASA KEBUTUHAN .....</b>           | <b>21</b> |
| 4.1 Rekayasa Kebutuhan .....                    | 21        |
| 4.1.1 Gambaran Umum Sistem .....                | 21        |
| 4.1.2 Identifikasi Aktor .....                  | 21        |
| 4.2 Spesifikasi Kebutuhan .....                 | 22        |
| 4.2.1 Aturan Penomoran .....                    | 22        |
| 4.2.2 Kebutuhan Fungsional .....                | 23        |
| 4.2.3 Kebutuhan Non Fungsional .....            | 41        |
| 4.3 Pemodelan Kebutuhan .....                   | 41        |
| 4.3.1 <i>Use Case Diagram</i> .....             | 41        |
| 4.3.2 <i>Use Case Scenario</i> .....            | 44        |
| <b>BAB 5 PERANCANGAN DAN IMPLEMENTASI .....</b> | <b>76</b> |
| 5.1 Perancangan .....                           | 76        |
| 5.1.1 Perancangan <i>Sequence Diagram</i> ..... | 76        |
| 5.1.2 Perancangan <i>Class Diagram</i> .....    | 79        |





|   |            |
|---|------------|
| 5.1.3 Perancangan Basis Data.....   | 83         |
| 5.1.4 Perancangan Komponen.....   | 100        |
| 5.1.5 Perancangan Antarmuka.....  | 103        |
| 5.2 Implementasi.....   | 109        |
| 5.2.1 Spesifikasi Sistem.....   | 110        |
| 5.2.2 Implementasi Basis Data.....  | 110        |
| 5.2.3 Implementasi Komponen.....  | 111        |
| 5.2.4 Implementasi Antarmuka.....   | 114        |
| <b>BAB 6 PENGUJIAN.....</b>   | <b>119</b> |
| 6.1 Pengujian Unit.....   | 119        |
| 6.1.1 Pengujian Unit Tambah Ulasan.....                                       | 119        |
| 6.1.2 Pengujian Unit Kirim Pertanyaan.....                                    | 121        |
| 6.1.3 Pengujian Unit Cari Pengacara.....                                      | 123        |
| 6.2 Pengujian Integrasi.....  | 126        |
| 6.2.1 Pengujian Integrasi <i>Method</i> ambil_data() klas<br>C_Pengacara..... | 126        |
| 6.3 Pengujian Validasi.....   | 128        |
| 6.3.1 Pengujian Validasi <i>Login</i> .....                                   | 129        |
| 6.3.2 Pengujian Validasi Registrasi.....                                      | 131        |
| 6.3.3 Pengujian Validasi <i>Logout</i> .....                                  | 134        |
| 6.3.4 Pengujian Validasi Tambah Info Hukum.....                               | 135        |
| 6.3.5 Pengujian Validasi Edit Info Hukum.....                                 | 136        |
| 6.3.6 Pengujian Validasi Hapus Info Hukum.....                                | 138        |
| 6.3.7 Pengujian Validasi Kirim Portofolio.....                                | 139        |
| 6.3.8 Pengujian Validasi Edit Portofolio.....                                 | 141        |
| 6.3.9 Pengujian Validasi Lihat Daftar Pertanyaan.....                         | 142        |
| 6.3.10 Pengujian Validasi Jawab Pertanyaan.....                               | 143        |
| 6.3.11 Pengujian Validasi Lihat Bookingan Klien.....                          | 144        |
| 6.3.12 Pengujian Validasi Terima Bookingan Klien.....                         | 145        |
| 6.3.13 Pengujian Validasi Tolak Bookingan Klien.....                          | 146        |
| 6.3.14 Pengujian Validasi Lihat Ulasan Klien.....                             | 147        |
| 6.3.15 Pengujian Validasi Lihat Riwayat Klien.....                            | 148        |





|  |     |
|--|-----|
| 6.3.16 Pengujian Validasi Laporkan Klien.....              | 148 |
| 6.3.17 Pengujian Validasi Lihat Jasa Pengacara .....       | 150 |
| 6.3.18 Pengujian Validasi Terima Jasa Pengacara.....       | 151 |
| 6.3.19 Pengujian Validasi Tolak Jasa Pengacara .....       | 151 |
| 6.3.20 Pengujian Validasi Tambah Jadwal Kerja .....        | 152 |
| 6.3.21 Pengujian Validasi Edit Jadwal Kerja .....          | 154 |
| 6.3.22 Pengujian Validasi Hapus Jadwal Kerja .....         | 156 |
| 6.3.23 Pengujian Validasi Pilih Jasa Pengacara .....       | 157 |
| 6.3.24 Pengujian Validasi Batalkan Jasa Pengacara .....    | 158 |
| 6.3.25 Pengujian Validasi Pilih Konsultasi Tatap Muka..... | 160 |
| 6.3.26 Pengujian Validasi Lihat Bookingan Saya .....       | 161 |
| 6.3.27 Pengujian Validasi Lihat Riwayat Aktivitas .....    | 162 |
| 6.3.28 Pengujian Validasi Beri Ulasan .....                | 162 |
| 6.3.29 Pengujian Validasi Laporkan Pengacara.....          | 164 |
| 6.3.30 Pengujian Validasi Kirim Pertanyaan .....           | 167 |
| 6.3.31 Pengujian Validasi Lihat Jawaban Pertanyaan.....    | 168 |
| 6.3.32 Pengujian Validasi Validasi Pengacara .....         | 169 |
| 6.3.33 Pengujian Validasi Tolak Pengacara .....            | 169 |
| 6.3.34 Pengujian Validasi Lihat Portofolio Pengacara ..... | 170 |
| 6.3.35 Pengujian Validasi Lihat Laporan Info Hukum .....   | 171 |
| 6.3.36 Pengujian Validasi Hapus Laporan Info Hukum .....   | 171 |
| 6.3.37 Pengujian Validasi Lihat Laporan Pengacara .....    | 173 |
| 6.3.38 Pengujian Validasi Hapus Pengacara .....            | 173 |
| 6.3.39 Pengujian Validasi Lihat Laporan Klien .....        | 174 |
| 6.3.40 Pengujian Validasi Hapus Klien .....                | 175 |
| 6.3.41 Pengujian Validasi Beri Komentar .....              | 176 |
| 6.3.42 Pengujian Validasi Laporkan Info Hukum .....        | 177 |
| 6.3.43 Pengujian Validasi Lihat Profil .....               | 179 |
| 6.3.44 Pengujian Validasi Edit Profil .....                | 179 |
| 6.3.45 Pengujian Validasi Lihat Daftar Pengacara .....     | 181 |
| 6.3.46 Pengujian Validasi Lihat Profil Pengacara .....     | 182 |
| 6.3.47 Pengujian Validasi Cari Pengacara .....             | 183 |





|   |     |
|---|-----|
| 6.3.48 Pengujian Validasi Lihat Info Hukum .....                                | 184 |
| 6.3.49 Pengujian Validasi Tambah Hasil Pertemuan Konsultasi<br>Tatap Muka ..... | 185 |
| 6.3.50 Pengujian Validasi Tambah Progres Kasus .....                            | 187 |
| 6.3.51 Pengujian Validasi Edit Progres Kasus .....                              | 188 |
| 6.3.52 Pengujian Validasi Hapus Progres Kasus .....                             | 190 |
| 6.3.53 Pengujian Validasi Lihat Progres Kasus .....                             | 191 |
| 6.3.54 Pengujian Validasi Lihat Hasil Pertemuan Konsultasi<br>Tatap Muka .....  | 192 |
| 6.4 Pengujian <i>Compatibility</i> .....  | 192 |
| BAB 7 penutup .....   | 194 |
| 7.1 Kesimpulan .....  | 194 |
| 7.2 Saran .....   | 195 |
| DAFTAR REFERENSI .....  | 196 |
| LAMPIRAN A HASIL WAWANCARA .....  | 198 |
| LAMPIRAN B ROADMAP SKRIPSI .....  | 201 |





## DAFTAR TABEL

|   |    |
|---|----|
| Tabel 2.1 Notasi pada <i>Use Case Diagram</i> .....               | 9  |
| Tabel 2.2 Notasi pada <i>Sequence Diagram</i> .....               | 10 |
| Tabel 2.3 Simbol <i>Class Diagram</i> .....                       | 11 |
| Tabel 4.1 Identifikasi Pemangku Kepentingan .....                 | 22 |
| Tabel 4.2 Kebutuhan Fungsional Iterasi 1 .....                    | 23 |
| Tabel 4.3 Kebutuhan Fungsional Iterasi 2 .....                    | 38 |
| Tabel 4.4 Daftar Kebutuhan Non Fungsional .....                   | 41 |
| Tabel 4.5 <i>Use Case Scenario Login</i> .....                    | 44 |
| Tabel 4.6 <i>Use Case Scenario Registrasi</i> .....               | 45 |
| Tabel 4.7 <i>Use Case Scenario Logout</i> .....                   | 45 |
| Tabel 4.8 <i>Use Case Scenario Tambah Info Hukum</i> .....        | 46 |
| Tabel 4.9 <i>Use Case Scenario Edit Info Hukum</i> .....          | 47 |
| Tabel 4.10 <i>Use Case Scenario Hapus Info Hukum</i> .....        | 47 |
| Tabel 4.11 <i>Use Case Scenario Kirim Portofolio</i> .....        | 48 |
| Tabel 4.12 <i>Use Case Scenario Edit Portofolio</i> .....         | 48 |
| Tabel 4.13 <i>Use Case Scenario Lihat Daftar Pertanyaan</i> ..... | 49 |
| Tabel 4.14 <i>Use Case Scenario Jawab Pertanyaan</i> .....        | 50 |
| Tabel 4.15 <i>Use Case Scenario Lihat Bookingan Klien</i> .....   | 50 |
| Tabel 4.16 <i>Use Case Scenario Terima Bookingan Klien</i> .....  | 51 |
| Tabel 4.17 <i>Use Case Scenario Tolak Bookingan Klien</i> .....   | 51 |
| Tabel 4.18 <i>Use Case Scenario Lihat Ulasan Klien</i> .....      | 52 |
| Tabel 4.19 <i>Use Case Scenario Lihat Riwayat Klien</i> .....     | 52 |
| Tabel 4.20 <i>Use Case Scenario Laporkan Klien</i> .....          | 53 |
| Tabel 4.21 <i>Use Case Scenario Lihat Jasa Pengacara</i> .....    | 53 |
| Tabel 4.22 <i>Use Case Scenario Terima Jasa Pengacara</i> .....   | 54 |
| Tabel 4.23 <i>Use Case Scenario Tolak Jasa Pengacara</i> .....    | 55 |
| Tabel 4.24 <i>Use Case Scenario Tambah Jadwal Kerja</i> .....     | 55 |
| Tabel 4.25 <i>Use Case Scenario Edit Jadwal Kerja</i> .....       | 56 |
| Tabel 4.26 <i>Use Case Scenario Hapus Jadwal Kerja</i> .....      | 56 |
| Tabel 4.27 <i>Use Case Scenario Pilih Jasa Pengacara</i> .....    | 57 |





|   |    |
|---|----|
| Tabel 4.28 <i>Use Case Scenario</i> Batalkan Jasa Pengacara .....                     | 58 |
| Tabel 4.29 <i>Use Case Scenario</i> Pilih Konsultasi Tatap Muka .....                 | 58 |
| Tabel 4.30 <i>Use Case Scenario</i> Lihat Bookingan Saya .....                        | 59 |
| Tabel 4.31 <i>Use Case Scenario</i> Lihat Riwayat Aktivitas .....                     | 59 |
| Tabel 4.32 <i>Use Case Scenario</i> Beri Ulasan .....                                 | 60 |
| Tabel 4.33 <i>Use Case Scenario</i> Laporkan Pengacara .....                          | 60 |
| Tabel 4.34 <i>Use Case Scenario</i> Kirim Pertanyaan .....                            | 61 |
| Tabel 4.35 <i>Use Case Scenario</i> Lihat Jawaban Pertanyaan .....                    | 62 |
| Tabel 4.36 <i>Use Case Scenario</i> Validasi Pengacara .....                          | 62 |
| Tabel 4.37 <i>Use Case Scenario</i> Tolak Pengacara .....                             | 63 |
| Tabel 4.38 <i>Use Case Scenario</i> Lihat Portofolio Pengacara .....                  | 63 |
| Tabel 4.39 <i>Use Case Scenario</i> Lihat Laporan Info Hukum .....                    | 64 |
| Tabel 4.40 <i>Use Case Scenario</i> Hapus Laporan Info Hukum .....                    | 64 |
| Tabel 4.41 <i>Use Case Scenario</i> Lihat Laporan Pengacara .....                     | 65 |
| Tabel 4.42 <i>Use Case Scenario</i> Hapus Pengacara .....                             | 65 |
| Tabel 4.43 <i>Use Case Scenario</i> Lihat Laporan Klien .....                         | 66 |
| Tabel 4.44 <i>Use Case Scenario</i> Hapus Klien .....                                 | 66 |
| Tabel 4.45 <i>Use Case Scenario</i> Beri Komentar .....                               | 67 |
| Tabel 4.46 <i>Use Case Scenario</i> Diagram Laporkan Info Hukum .....                 | 67 |
| Tabel 4.47 <i>Use Case Scenario</i> Lihat Melihat Profil .....                        | 68 |
| Tabel 4.48 <i>Use Case Scenario</i> Edit Profil .....                                 | 69 |
| Tabel 4.49 <i>Use Case Scenario</i> Lihat Daftar Pengacara .....                      | 69 |
| Tabel 4.50 <i>Use Case Scenario</i> Lihat Profil Pengacara .....                      | 70 |
| Tabel 4.51 <i>Use Case Scenario</i> Mencari Pengacara .....                           | 70 |
| Tabel 4.52 <i>Use Case Scenario</i> Lihat Info Hukum .....                            | 71 |
| Tabel 4.53 <i>Use Case Scenario</i> Hasil Pertemuan Konsultasi Tatap Muka .....       | 71 |
| Tabel 4.54 <i>Use Case Scenario</i> Tambah Progres Kasus .....                        | 72 |
| Tabel 4.55 <i>Use Case Scenario</i> Edit Progres Kasus .....                          | 73 |
| Tabel 4.56 <i>Use Case Scenario</i> Hapus Progres Kasus .....                         | 73 |
| Tabel 4.57 <i>Use Case Scenario</i> Lihat Progres Kasus .....                         | 74 |
| Tabel 4.58 <i>Use Case Scenario</i> Lihat Hasil Pertemuan Konsultasi Tatap Muka ..... | 74 |
| Tabel 5.1 Entitas User .....  | 85 |





|   |     |
|---|-----|
| Tabel 5.2 Entitas cv.....   | 85  |
| Tabel 5.3 Entitas tanya_pengacara.....                              | 87  |
| Tabel 5.4 Entitas jawab_tanya.....                                  | 87  |
| Tabel 5.5 Entitas infohukum.....                                    | 88  |
| Tabel 5.6 Entitas komentar.....                                     | 88  |
| Tabel 5.7 Entitas jadwalkerja.....                                  | 89  |
| Tabel 5.8 Entitas jasa_pengacara.....                               | 89  |
| Tabel 5.9 Entitas <i>district</i> .....                             | 90  |
| Tabel 5.10 Entitas <i>provinces</i> .....                           | 90  |
| Tabel 5.11 Entitas <i>regencies</i> .....                           | 91  |
| Tabel 5.12 Entitas <i>villages</i> .....                            | 91  |
| Tabel 5.13 Entitas laporan_infohukum.....                           | 91  |
| Tabel 5.14 Entitas laporan_klien.....                               | 92  |
| Tabel 5.15 Entitas laporan_pengacara.....                           | 93  |
| Tabel 5.16 Entitas tatapmuka.....                                   | 93  |
| Tabel 5.17 Entitas ulasan.....                                      | 94  |
| Tabel 5.18 Entitas progres_kasus.....                               | 95  |
| Tabel 5.19 Algoritme Tambah Ulasan.....                             | 100 |
| Tabel 5.20 Algoritme Kirim Pertanyaan.....                          | 101 |
| Tabel 5.21 Algoritme Cari Pengacara.....                            | 101 |
| Tabel 5.22 Algoritme Tambah Progres.....                            | 102 |
| Tabel 5.23 Penjelasan Antarmuka Halaman <i>Home</i> .....           | 103 |
| Tabel 5.24 Penjelasan Antarmuka Halaman Tanya Pengacara.....        | 105 |
| Tabel 5.25 Penjelasan Antarmuka Halaman Cari Pengacara.....         | 106 |
| Tabel 5.26 Penjelasan Antarmuka Halaman Lihat Profil.....           | 107 |
| Tabel 5.27 Penjelasan Antarmuka Halaman Lihat Profil Pengacara..... | 108 |
| Tabel 5.28 Penjelasan Antarmuka Halaman Tambah Progres Kasus.....   | 109 |
| Tabel 5.29 Spesifikasi Perangkat Keras.....                         | 110 |
| Tabel 5.30 Spesifikasi Perangkat Lunak.....                         | 110 |
| Tabel 5.31 Implementasi Komponen Tambah Ulasan.....                 | 111 |
| Tabel 5.32 Implementasi Komponen Kirim Pertanyaan.....              | 112 |
| Tabel 5.33 Implementasi Komponen Cari Pengacara.....                | 113 |





|   |     |
|---|-----|
| Tabel 6.1 Pembentukan Node Algoritme fungsi tambahUlasan(id_pengacara, id_tatapmuka)..... | 119 |
| Tabel 6.2 Kasus Uji Tambah Ulasan .....   | 121 |
| Tabel 6.3 Pembentukan Node Algoritme fungsi askPengacara() .....                          | 122 |
| Tabel 6.4 Kasus Uji Kirim Pertanyaan.....   | 123 |
| Tabel 6.5 Pembentukan Node Algoritme fungsi search().....                                 | 123 |
| Tabel 6.6 Kasus Uji Cari Pengacara .....  | 125 |
| Tabel 6.7 Pembentukan Node Algoritme fungsi ambil_data () .....                           | 126 |
| Tabel 6.8 Kasus Uji Fungsi ambil_data() .....   | 127 |
| Tabel 6.9 Pengujian Validasi <i>Login</i> .....   | 129 |
| Tabel 6.10 Pengujian Validasi <i>Login</i> Alternatif Satu .....                          | 129 |
| Tabel 6.11 Pengujian Validasi <i>Login</i> Alternatif Dua .....                           | 130 |
| Tabel 6.12 Pengujian Validasi Registrasi.....   | 131 |
| Tabel 6.13 Pengujian Validasi Registrasi Alternatif Satu .....                            | 132 |
| Tabel 6.14 Pengujian Validasi Registrasi Alternatif Dua .....                             | 132 |
| Tabel 6.15 Pengujian Validasi Registrasi Alternatif Tiga .....                            | 133 |
| Tabel 6.16 Pengujian Validasi <i>Logout</i> .....   | 134 |
| Tabel 6.17 Pengujian Validasi Tambah Info Hukum .....                                     | 135 |
| Tabel 6.18 Pengujian Validasi Tambah Info Hukum Alternatif Satu.....                      | 136 |
| Tabel 6.19 Pengujian Validasi Edit Info Hukum .....                                       | 136 |
| Tabel 6.20 Pengujian Validasi Edi Info Hukum Alternatif Satu.....                         | 137 |
| Tabel 6.21 Pengujian Validasi Hapus Info Hukum .....                                      | 138 |
| Tabel 6.22 Pengujian Validasi Kirim Portofolio.....                                       | 139 |
| Tabel 6.23 Pengujian Validasi Kirim Portofolio Alternatif Satu .....                      | 140 |
| Tabel 6.24 Pengujian Validasi Edit Portofolio.....  | 141 |
| Tabel 6.25 Pengujian Validasi Edit Portofolio Alternatif Satu.....                        | 142 |
| Tabel 6.26 Pengujian Validasi Lihat Daftar Pertanyaan.....                                | 142 |
| Tabel 6.27 Pengujian Validasi Jawab Pertanyaan .....                                      | 143 |
| Tabel 6.28 Pengujian Validasi Jawab Pertanyaan Alternatif Satu.....                       | 144 |
| Tabel 6.29 Pengujian Validasi Lihat Bookingan Klien.....                                  | 144 |
| Tabel 6.30 Pengujian Validasi Terima Bookingan Klien .....                                | 145 |
| Tabel 6.31 Pengujian Validasi Tolak Bookingan Klien.....                                  | 146 |





|  |     |
|--|-----|
| Tabel 6.32 Pengujian Validasi Tolak Bookingan Klien Alternatif Satu.....       | 147 |
| Tabel 6.33 Pengujian Validasi Lihat Ulasan Klien.....                          | 147 |
| Tabel 6.34 Pengujian Validasi Lihat Riwayat Klien.....                         | 148 |
| Tabel 6.35 Pengujian Validasi Laporkan Klien.....                              | 148 |
| Tabel 6.36 Pengujian Validasi Laporkan Klien Alternatif Satu.....              | 149 |
| Tabel 6.37 Pengujian Validasi Lihat Jasa Pengacara.....                        | 150 |
| Tabel 6.38 Pengujian Validasi Lihat Jasa Pengacara.....                        | 151 |
| Tabel 6.39 Pengujian Validasi Tolak Jasa Pengacara.....                        | 151 |
| Tabel 6.40 Pengujian Validasi Tambah Jadwal Kerja.....                         | 152 |
| Tabel 6.41 Pengujian Validasi Tambah Jadwal Kerja Alternatif Satu.....         | 153 |
| Tabel 6.42 Pengujian Validasi Edit Jadwal Kerja.....                           | 154 |
| Tabel 6.43 Pengujian Validasi Edit Jadwal Kerja Alternatif Satu.....           | 155 |
| Tabel 6.44 Pengujian Validasi Edit Jadwal Kerja Alternatif Dua.....            | 155 |
| Tabel 6.45 Pengujian Validasi Hapus Jadwal Kerja.....                          | 156 |
| Tabel 6.46 Pengujian Validasi Pilih Jasa Pengacara.....                        | 157 |
| Tabel 6.47 Pengujian Validasi Pilih Jasa Pengacara Alternatif Satu.....        | 157 |
| Tabel 6.48 Pengujian Validasi Batalkan Jasa Pengacara.....                     | 158 |
| Tabel 6.49 Pengujian Validasi Batalkan Jasa Pengacara Alternatif Satu.....     | 159 |
| Tabel 6.50 Pengujian Validasi Pilih Konsultasi Tatap Muka.....                 | 160 |
| Tabel 6.51 Pengujian Validasi Pilih Konsultasi Tatap Muka Alternatif Satu..... | 161 |
| Tabel 6.52 Pengujian Validasi Lihat Bookingan Saya.....                        | 161 |
| Tabel 6.53 Pengujian Validasi Lihat Riwayat Aktivitas.....                     | 162 |
| Tabel 6.54 Pengujian Validasi Beri Ulasan.....                                 | 162 |
| Tabel 6.55 Pengujian Validasi Beri Ulasan Alternatif Satu.....                 | 163 |
| Tabel 6.56 Pengujian Validasi Beri Ulasan Alternatif Dua.....                  | 164 |
| Tabel 6.57 Pengujian Validasi Laporkan Pengacara.....                          | 164 |
| Tabel 6.58 Pengujian Validasi Laporkan Pengacara Alternatif Satu.....          | 165 |
| Tabel 6.59 Pengujian Validasi Laporkan Pengacara Alternatif Dua.....           | 166 |
| Tabel 6.60 Pengujian Validasi Kirim Pertanyaan.....                            | 167 |
| Tabel 6.61 Pengujian Validasi Kirim Pertanyaan Alternatif Satu.....            | 167 |
| Tabel 6.62 Pengujian Validasi Lihat Jawaban Pertanyaan.....                    | 168 |
| Tabel 6.63 Pengujian Validasi Validasi Pengacara.....                          | 169 |





|  |     |
|--|-----|
| Tabel 6.64 Pengujian Validasi Tolak Pengacara .....  | 169 |
| Tabel 6.65 Pengujian Validasi Lihat Portofolio Pengacara .....                                   | 170 |
| Tabel 6.66 Pengujian Validasi Lihat Laporan Info Hukum .....                                     | 171 |
| Tabel 6.67 Pengujian Validasi Hapus Laporan Info Hukum .....                                     | 171 |
| Tabel 6.68 Pengujian Validasi Hapus Laporan Info Hukum Alternatif Satu .....                     | 172 |
| Tabel 6.69 Pengujian Validasi Lihat Laporan Pengacara .....                                      | 173 |
| Tabel 6.70 Pengujian Validasi Hapus Pengacara .....  | 173 |
| Tabel 6.71 Pengujian Validasi Hapus Pengacara Alternatif Satu .....                              | 174 |
| Tabel 6.72 Pengujian Validasi Lihat Laporan Klien .....  | 174 |
| Tabel 6.73 Pengujian Validasi Hapus Klien .....  | 175 |
| Tabel 6.74 Pengujian Validasi Hapus Klien Alternatif Satu .....                                  | 176 |
| Tabel 6.75 Pengujian Validasi Beri Komentar .....  | 176 |
| Tabel 6.76 Pengujian Validasi Beri Komentar .....  | 177 |
| Tabel 6.77 Pengujian Validasi Laporkan Info Hukum .....  | 177 |
| Tabel 6.78 Pengujian Validasi Laporkan Info Hukum Alternatif Satu .....                          | 178 |
| Tabel 6.79 Pengujian Validasi Lihat Profil .....   | 179 |
| Tabel 6.80 Pengujian Validasi Edit Profil .....  | 180 |
| Tabel 6.81 Pengujian Validasi Edit Profil Alternatif Satu .....                                  | 180 |
| Tabel 6.82 Pengujian Validasi Lihat Daftar Pengacara .....                                       | 181 |
| Tabel 6.83 Pengujian Validasi Lihat Profil Pengacara .....                                       | 182 |
| Tabel 6.84 Pengujian Validasi Cari Pengacara .....   | 183 |
| Tabel 6.85 Pengujian Validasi Cari Pengacara Alternatif Satu .....                               | 184 |
| Tabel 6.86 Pengujian Validasi Lihat Info Hukum .....   | 184 |
| Tabel 6.87 Pengujian Validasi Tambah Hasil Pertemuan Konsultasi Tatap Muka .....                 | 185 |
| Tabel 6.88 Pengujian Validasi Tambah Hasil Pertemuan Konsultasi Tatap Muka Alternatif Satu ..... | 186 |
| Tabel 6.89 Pengujian Validasi Tambah Progres Kasus .....   | 187 |
| Tabel 6.90 Pengujian Validasi Tambah Progres Kasus Alternatif Satu .....                         | 188 |
| Tabel 6.91 Pengujian Validasi Edit Progres Kasus .....   | 188 |
| Tabel 6.92 Pengujian Validasi Edit Progres Kasus Alternatif Satu .....                           | 189 |
| Tabel 6.93 Pengujian Validasi Edit Progres Alternatif Dua .....                                  | 190 |





Tabel 6.94 Pengujian Validasi Hapus Progres Kasus ..... 191

Tabel 6.95 Pengujian Validasi Lihat Progres Kasus ..... 191

Tabel 6.96 Pengujian Validasi Lihat Hasil Pertemuan Konsultasi Tatap Muka .... 192





## DAFTAR GAMBAR

|   |     |
|---|-----|
| Gambar 2.1 Siklus RAD.....  | 7   |
| Gambar 2.2 Contoh <i>Use Case Diagram</i> .....                       | 9   |
| Gambar 2.3 Contoh <i>Class Diagram</i> .....                          | 11  |
| Gambar 3.1 Diagram Alir Metodologi Penelitian.....                    | 16  |
| Gambar 3.2 <i>Work Breakdown Structure</i> (WBS).....                 | 18  |
| Gambar 4.1 Identifikasi Aktor.....                                    | 21  |
| Gambar 4.2 Aturan Penomoran.....                                      | 22  |
| Gambar 4.3 <i>Use Case Diagram</i> Iterasi 1.....                     | 42  |
| Gambar 4.4 <i>Use Case Diagram</i> Iterasi 2.....                     | 43  |
| Gambar 5.1 <i>Sequence Diagram</i> Kirim Pertanyaan.....              | 76  |
| Gambar 5.2 <i>Sequence Diagram</i> Kirim Portofolio.....              | 77  |
| Gambar 5.3 <i>Sequence Diagram</i> Validasi Pengacara.....            | 78  |
| Gambar 5.4 <i>Sequence Diagram</i> Tambah Progres Kasus.....          | 79  |
| Gambar 5.5 <i>Class Diagram</i> Umum.....                             | 80  |
| Gambar 5.6 <i>Class Diagram Controller</i> Iterasi 1.....             | 81  |
| Gambar 5.7 <i>Class Diagram Controller</i> Iterasi 1.....             | 81  |
| Gambar 5.8 <i>Class Diagram Model</i> Iterasi 1.....                  | 82  |
| Gambar 5.9 <i>Class Diagram Model</i> Iterasi 2.....                  | 83  |
| Gambar 5.10 <i>Entity Relationship Diagram</i> (ERD) Iterasi 1.....   | 84  |
| Gambar 5.11 <i>Entity Relationship Diagram</i> (ERD) Iterasi 2.....   | 96  |
| Gambar 5.12 <i>Physical Data Model</i> (PDM) Iterasi 1.....           | 98  |
| Gambar 5.13 <i>Physical Data Model</i> (PDM) Iterasi 2.....           | 99  |
| Gambar 5.14 Perancangan Antarmuka Halaman <i>Home</i> .....           | 103 |
| Gambar 5.15 Perancangan Antarmuka Halaman Tanya Pengacara.....        | 104 |
| Gambar 5.16 Perancangan Antarmuka Halaman Cari Pengacara.....         | 105 |
| Gambar 5.17 Perancangan Antarmuka Halaman Lihat Profil.....           | 106 |
| Gambar 5.18 Perancangan Antarmuka Halaman Lihat Profil Pengacara..... | 107 |
| Gambar 5.19 Perancangan Antarmuka Halaman Tambah Progres Kasus.....   | 109 |
| Gambar 5.20 Implementasi Basis Data.....                              | 111 |
| Gambar 5.21 Implementasi Antarmuka Halaman <i>Home</i> .....          | 114 |





|  |     |
|--|-----|
| Gambar 5.22 Implementasi Antarmuka Halaman Tanya Pengacara.....                                | 115 |
| Gambar 5.23 Implementasi Antarmuka Halaman Cari Pengacara.....                                 | 116 |
| Gambar 5.24 Implementasi Antarmuka Halaman Lihat Profil.....                                   | 116 |
| Gambar 5.25 Implementasi Antarmuka Halaman Lihat Profil Pengacara.....                         | 117 |
| Gambar 5.26 Implementasi Antarmuka Halaman Tambah Progres Kasus.....                           | 118 |
| Gambar 6.1 <i>Flow Graph</i> Algoritme fungsi tambahUlasan(id_pengacara,<br>id_tatapmuka)..... | 120 |
| Gambar 6.2 <i>Flow Graph</i> Algoritme fungsi askPengacara().....                              | 122 |
| Gambar 6.3 <i>Flow Graph</i> Algoritme fungsi search().....                                    | 124 |
| Gambar 6.4 <i>Flow Graph</i> Algoritme fungsi ambil_data().....                                | 127 |
| Gambar 6.5 Indikator Masalah <i>Compatibility</i> SortSite.....                                | 193 |
| Gambar 6.6 Hasil Pengujian <i>Compatibility</i> .....  | 193 |





## DAFTAR LAMPIRAN

|                                 |     |
|---------------------------------|-----|
| LAMPIRAN A HASIL WAWANCARA..... | 198 |
| LAMPIRAN B ROADMAP SKRIPSI..... | 201 |





## BAB 1 PENDAHULUAN

### 1.1 Latar Belakang

Teknologi informasi saat ini sudah menjadi hal yang penting dalam kehidupan. Dimana pengaruh dari teknologi sudah sangat terasa bagi manusia baik dalam segi nilai dan fungsinya. Teknologi informasi adalah suatu teknologi yang digunakan untuk mengolah data, termasuk memproses, mendapatkan, menyusun, menyimpan, dan memanipulasi data dalam berbagai cara untuk menghasilkan suatu informasi. Dalam hal ini informasi yang berkualitas, yaitu informasi yang relevan, akurat, dan tepat waktu. Yang bisa digunakan untuk keperluan pribadi, bisnis, dan pemerintahan, juga merupakan informasi yang tepat untuk mengambil keputusan. Dimana teknologi tersebut dapat mendukung segala aktivitas dalam sistem informasi.

Hukum adalah kaidah atau aturan dalam bentuk tertulis maupun tidak tertulis, yang tersusun menjadi suatu sistem yang menentukan apa yang boleh dan tidak boleh dilakukan oleh manusia dalam kehidupan bermasyarakat (Achmad Ali, 1996:47 disitasi dalam Achmad Ali, 2012). Dalam hukum terdapat suatu sistem peraturan yang didalamnya terdapat norma-norma dan sanksi-sanksi yang bertujuan untuk mengendalikan perilaku manusia, menjaga ketertiban dan keadilan, serta mencegah terjadinya kekacauan. Keberadaan hukum bertujuan untuk melindungi setiap individu dari penyalahgunaan kekuasaan serta untuk menegakkan keadilan. Dengan adanya hukum di suatu negara, maka setiap orang di negara tersebut berhak mendapatkan keadilan dan pembelaan di depan hukum yang berlaku. Hukum ini juga bertujuan untuk menjaga dan mencegah orang agar tidak menjadi hakim atas dirinya sendiri.

Untuk melindungi dan mencegah penyalahgunaan hukum, kita memerlukan seorang yang paham tentang permasalahan yang ada pada ruang lingkup hukum yaitu pengacara. Pengacara atau advokat menurut Undang-Undang Nomor 18 Tahun 2003 adalah seseorang yang memiliki profesi untuk memberikan jasa hukum kepada orang di dalam pengadilan atau seseorang yang mempunyai izin praktik beracara di pengadilan di seluruh wilayah Negara Republik Indonesia. Dalam memilih pengacara dan menggunakan jasanya, kita harus mencari pengacara yang memiliki kredibilitas dalam menangani kasus kliennya. Memilih pengacara yang sesuai dengan kasus yang kita miliki dengan kasus-kasus yang telah ditangani oleh pengacara. Sebelum menentukan cocok atau tidaknya kasus kita dengan pengacara tersebut dibutuhkan konsultasi terlebih dahulu mengenai kasus yang kita hadapi. Pengacara akan memberikan *legal opinion* mengenai kasus kita saat berkonsultasi. Lalu jika pengacara memberikan *legal opinion* mengenai kasus yang kita konsultasikan tidak cocok, maka kita bisa mengganti pengacara yang lebih sesuai dengan kasus yang dimiliki. Saat ini banyak yang membutuhkan pengacara tapi sulit untuk mendapatkan pengacara. Banyak penyebab orang sulit menemukan pengacara, salah satunya ketidakcocokan antara pengacara dan





klien. Untuk mengetahui cocok atau tidaknya pengacara dengan klien, klien harus mendatangi tempat pengacara untuk konsultasi langsung mengenai kasus klien

Berdasarkan permasalahan yang sudah dijelaskan diatas, maka upaya untuk memudahkan klien untuk mengkonsultasikan permasalahan yang dimiliki dengan pengacara dengan mudah. Lalu klien juga dapat mencari pengacara yang sesuai dengan permasalahan yang dimiliki dan mencari pengacara yang memiliki kredibilitas tinggi sesuai dengan kasus-kasus yang telah ditangani oleh pengacara. Maka dari itu, penulis membuat penelitian dengan judul “Pengembangan Sistem Konsultasi dan Pencarian Pengacara Berbasis Website”. Dalam mengembangkan sistem ini, diharapkan hasilnya dapat memudahkan klien berkonsultasi dan menghubungi pengacara untuk kasus yang dihadapi oleh klien. Dan sistem dikembangkan dengan berbasis website diharapkan sistem mudah untuk diakses oleh pengguna.

## 1.2 Rumusan Masalah

Berdasarkan permasalahan yang diangkat pada latar belakang di atas, maka didapatkan rumusan masalah sebagai berikut:

1. Bagaimana hasil analisis dan spesifikasi kebutuhan Sistem Konsultasi dan Pencarian Pengacara?
2. Bagaimana rancangan dari Sistem Konsultasi dan Pencarian Pengacara yang sesuai dengan hasil analisis dan spesifikasi kebutuhan tersebut?
3. Bagaimana implementasi dari Sistem Konsultasi dan Pencarian Pengacara yang sesuai dengan hasil rancangan sistem tersebut?
4. Bagaimana hasil pengujian fungsionalitas dan non fungsionalitas Sistem Konsultasi dan Pencarian Pengacara?

## 1.3 Tujuan

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Menganalisis dan mencari spesifikasi kebutuhan pada Sistem Konsultasi dan Pencarian Pengacara untuk mengetahui kebutuhan apa saja yang dibutuhkan untuk sistem ini.
2. Merancang Sistem Konsultasi dan Pencarian Pengacara berdasarkan kebutuhan yang sudah didapatkan dari analisis dan spesifikasi kebutuhan dari sistem.
3. Mengimplementasikan rancangan dari Sistem Konsultasi dan Pencarian Pengacara.
4. Menguji kebutuhan fungsionalitas dan non fungsionalitas Sistem Konsultasi dan Pencarian Pengacara agar diperoleh sistem yang sesuai dengan kebutuhan.

## 1.4 Manfaat

Adapun manfaat yang diperoleh dalam penelitian ini adalah sebagai berikut:

1. Bagi pengguna





Memberikan kemudahan untuk berkonsultasi dan mencari pengacara yang cocok untuk menangani kasus pengguna.

## 2. Bagi pengacara

Memudahkan untuk mendapatkan klien dan manajemen kasus yang dimiliki oleh klien.

### 1.5 Batasan Masalah

Agar permasalahan pada penelitian ini tidak meluas, penelitian ini dibatasi pada hal-hal berikut:

1. Penelitian ini hanya difokuskan pada perancangan, implementasi, dan pengujian sistem.
2. Sistem ini dibangun dan dikembangkan menggunakan aplikasi berbasis web dengan bahasa pemrograman php menggunakan *framework* Codeigniter.

### 1.6 Sistematika Pembahasan

Sistematika pembahasan pada penelitian ini adalah sebagai berikut:

#### BAB 1 PENDAHULUAN

Memuat latar belakang yang merupakan inti dari penelitian, rumusan masalah yang berisi tentang pertanyaan-pertanyaan tentang penelitian, tujuan yang berisi tujuan yang ingin dicapai dari penelitian, manfaat yang berisi untuk siapa penelitian ini ditujukan, batasan masalah yang berisi batasan-batasan dari penelitian, dan yang terakhir adalah sistematika pembahasan yang berisi tentang struktur yang ada dalam penelitian.

#### BAB 2 LANDASAN KEPUSTAKAAN

Memuat tentang teori, konsep, model, metode, atau sistem dari literatur ilmiah, yang berhubungan dengan perancangan dan pembangunan Sistem Konsultasi dan Pencarian Pengacara Berbasis Website.

#### BAB 3 METODOLOGI PENELITIAN

Membahas metodologi yang digunakan dalam penelitian. Terdiri dari studi literatur, analisis kebutuhan, perancangan sistem, implementasi, pengujian dan analisis, dan pengambilan kesimpulan dan saran.

#### BAB 4 REKAYASA KEBUTUHAN

Membahas tentang rekayasa kebutuhan Sistem Konsultasi dan Pencarian Pengacara Berbasis Website yang terdiri dari kebutuhan fungsional dan non fungsional sistem.





## **BAB 5 PERANCANGAN DAN IMPLEMENTASI**

Terdiri dari perancangan dan implementasi aplikasi dari Sistem Konsultasi dan Pencarian Pengacara Berbasis Website. Dasar dalam pembuatan perancangan sistem adalah rekayasa kebutan, dan untuk pembuatan implementasi menggunakan dasar dari perancangan.

## **BAB 6 PENGUJIAN**

Bab ini memuat tentang evaluasi hasil pengujian yang dilakukan pada Sistem Konsultasi dan Pencarian Pengacara.

## **BAB 7 PENUTUP**

Bab ini memuat kesimpulan dan saran yang untuk penelitian ini. Kesimpulan merupakan jawaban dari rumusan masalah yang telah disebutkan sebelumnya. Sedangkan saran merupakan saran yang berguna untuk mengembangkan lebih lanjut Sistem Konsultasi dan Pencarian Pengacara.





## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

Kajian pustaka yang digunakan sebagai referensi dalam penelitian ini adalah beberapa penelitian yang membahas tentang Sistem Konsultasi dan Pencarian Pengacara Berbasis Website.

Penelitian pertama adalah penelitian yang dilakukan oleh Irfan Aswani dengan judul “Pengembangan Sistem Informasi Lembaga Pengkajian dan Konsultan Bantuan Hukum (LPKBH) Menggunakan Zachman Framework (Studi Pada LPKBH Al-Baihaqy)”. Tujuan dari penelitian ini adalah untuk memudahkan LPKBH Al-Baihaqy untuk mendapatkan klien. Cara pengumpulan data dilakukan dengan cara wawancara dan studi literatur. Pengujian *white box* adalah pengujian untuk mendeteksi kesalahan dalam penulisan kode program. Sedangkan pengujian *black box* dengan menguji 15 fitur dari sistem, lalu didapatkan hasil 93% valid, jadi dapat disimpulkan mayoritas fungsi dari sistem dapat berjalan dengan baik (Aswani, 2018).

Penelitian kedua adalah penelitian yang dilakukan oleh Maulana Malik dengan judul “Rancang Bangun Sistem Informasi Manajemen Pengelolaan dan Penjadwalan Layanan Konsultasi Bantuan Hukum Pada LBH Makassar”. Tujuan dari penelitian ini adalah untuk membantu dan mempercepat proses pengelolaan dan penyajian data layanan bantuan hukum pada LBH Makassar. Hasil pengujian *white-box* ditunjukkan dari perhitungan *flowgraph*, *predicate note*, *region*, dan *independent path* bahwa rancangan program dari penelitian ini bebas dari kesalahan logika program. Hasil pengujian *black-box* menunjukkan data masukan dengan hasil yang diharapkan sesuai. Sedangkan, hasil kuisioner mengenai aplikasi ini ke *user target* mudah dipahami, dimengerti, dan cukup membantu dalam pengelolaan data dalam hal memberikan bantuan LBH Makassar (Malik, 2017).

Penelitian ketiga adalah penelitian yang dilakukan oleh Domas Sitepu dengan judul “Sistem Konsultasi Jenis Kejahatan *Information Technology* Berbasis Web”. Tujuan dari penelitian ini adalah untuk mempermudah aparat keamanan dan masyarakat untuk mengetahui tindakan pidana orang yang melanggar penggunaan teknologi informasi. Sistem ini juga dapat membantu pengacara untuk menyelesaikan masalah, dan memberitahu kepada masyarakat dalam kasus pelanggaran pemakaian internet. Cara pengumpulan data dilakukan dengan cara observasi, wawancara, dan studi literatur. Fitur-fitur dalam sistem ini adalah untuk halaman admin bisa mengupdate, edit, hapus, dan tambah data. Kemudian fitur untuk tampilan user adalah bisa mengupload konsultasi dan dapat melihat hasil dari konsultasi tersebut. Sistem ini diuji dengan menjalankan program lalu akan diketahui kekurangan dari program tersebut. Tahap uji coba struktural dilakukan untuk mengetahui kesesuaian rancangan dengan implementasi yang sudah dilakukan, dengan menguji setiap *form* pada aplikasi yang telah dibuat. Tahap uji coba fungsional dilakukan setelah uji coba struktural. Yang terakhir uji coba





validasi yang merupakan uji coba pada hasil implementasi apakah sudah valid atau belum valid (Sitepu, 2015).

## 2.2 Work Breakdown Structure (WBS)

*Work Breakdown Structure* (WBS) merupakan proses dari proyek manajemen yang paling awal yang akan membagi ke dalam beberapa fase proyek. WBS digunakan untuk memecahkan proses dari tiap pekerjaan menjadi lebih detail. Hal ini mempunyai tujuan agar proses dari perencanaan proyek memiliki tingkat lebih baik. Dengan adanya WBS, tim proyek dapat melakukan tugas untuk menciptakan proyek yang komprehensif termasuk jadwal, sumber daya, dan anggaran (Wysocki, 2006).

## 2.3 Software Development Life Cycle (SDLC)

*Software Development Life Cycle* (SDLC) merupakan proses untuk mengembangkan atau mengubah suatu sistem dalam perangkat lunak menggunakan metodologi dan model-model yang dapat digunakan untuk mengembangkan sistem perangkat lunak (Sukanto & Shalahuddin, 2018). Pengembangan SDLC menganut fase-fase penting bagi pengembang, contohnya seperti perencanaan proyek, analisis proyek, desain, implementasi, pengujian, dan pemeliharaan.

### 2.3.1 Rapid Application Development (RAD) Model

Metode *Rapid Application Development* (RAD) adalah pendekatan berorientasi objek mengenai pengembangan sistem yang mencakup suatu metode pengembangan perangkat-perangkat lunak (Kendal, 2010). RAD terdiri dari gabungan bermacam-macam teknik terstruktur dengan teknik *prototyping* dan teknik pengembangan *joint application*. Oleh karena itu, pengembangan menggunakan RAD dapat dilakukan dalam waktu yang relatif lebih cepat.

RAD bertujuan mempersingkat waktu yang biasanya diperlukan dalam siklus hidup pengembangan sistem tradisional antara perancangan dan penerapan suatu sistem informasi. Pada akhirnya, RAD sama-sama berusaha memenuhi syarat-syarat bisnis yang berubah secara cepat.





**Gambar 2.1 Siklus RAD**

(Sumber: Kendall, 2010)

Ada tiga fase dalam RAD yang melibatkan penganalisis dan pengguna dalam tahap penilaian, perancangan, dan penerapan. Tiga fase tersebut adalah *requirements planning*, *RAD design workshop*, dan *implementation*.

## 2.4 Analisis Kebutuhan

Analisis kebutuhan merupakan langkah pertama dari siklus hidup pengembangan perangkat lunak. Analisis kebutuhan mempelajari kebutuhan dari pemakai (*end user*) untuk memperoleh definisi dari kebutuhan perangkat lunak. Analisis kebutuhan menghasilkan spesifikasi yang didapatkan dari karakteristik yang dimiliki oleh perangkat lunak yang dikembangkan, dalam hal ini dimungkinkan akan mengisyaratkan antarmuka sistem dengan elemen lainnya yang bisa mempengaruhi, dan juga menjelaskan batasan yang ada pada perangkat lunak (Pressman, 2010).

Tujuan dari analisis kebutuhan adalah untuk memahami permasalahan secara keseluruhan pada perangkat lunak yang dikembangkan seperti ruang lingkup dari produk perangkat lunak dan pemakai yang akan menggunakan perangkat lunak. Analisis kebutuhan juga mendefinisikan apa saja yang harus dikerjakan oleh pengembang perangkat lunak untuk memenuhi keinginan pemakai.

Pada tahap analisis kebutuhan dibentuk sebuah spesifikasi kebutuhan perangkat lunak, fungsi dari perangkat lunak yang dibutuhkan, performansi sistem perangkat lunak, penjadwalan proyek, identifikasi sumber daya manusia, identifikasi perangkat keras dan perangkat lunak yang dibutuhkan, dan taksiran biaya pengembangan perangkat lunak. Berikut ini adalah metode-metode dari analisis kebutuhan, antara lain:

1. Sampling
2. Wawancara
3. Kuisisioner
4. Observasi





5. Prosedur analisis
6. Pengamatan dokumen
7. Dokumen analisis kebutuhan

## 2.5 Unified Modelling Language (UML)

*Unified Modelling Language* (UML) menurut Rossa A. S dan M. Shalahuddin (2018) merupakan standarisasi bahasa pemodelan untuk membangun sebuah perangkat lunak yang dibangun menggunakan pemrograman berorientasi objek. UML berguna untuk memvisualisasikan, membangun, menspesifikasikan, dan dokumentasi dari sistem perangkat lunak. UML menjelaskan standar penulisan dari sistem *blue print*, yang mencakup konsep proses bisnis, penulisan kelas-kelas dalam bahasa pemrograman yang spesifik, skema database, dan komponen lainnya yang diperlukan dalam sistem.

UML bukan hanya sebuah bahasa pemrograman visual, tetapi juga secara langsung dapat dihubungkan ke bahasa pemrograman lainnya, seperti C++, java, *visual basic*, atau juga bisa dihubungkan langsung ke dalam *object-oriented database*.

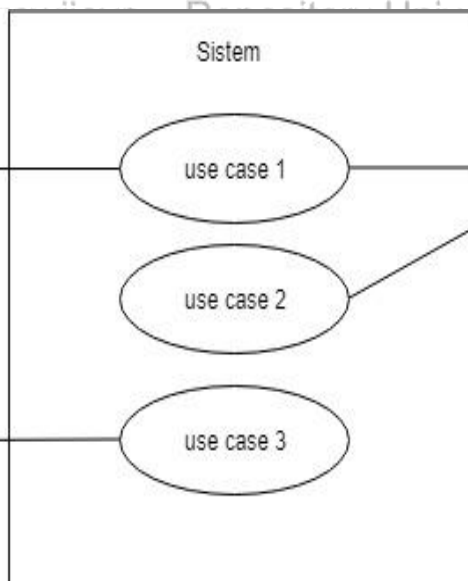
### 2.5.1 Use Case Diagram

*Use case diagram* merupakan diagram yang mendefinisikan hubungan antara sistem dengan aktor, secara grafis mendefinisikan sistem akan digunakan oleh siapa dan dengan cara apa aktor dalam berinteraksi dengan sistem (Whitten, Bentley, & Dittman, 2004). *Use case diagram* juga menggambarkan fungsionalitas yang diharapkan saat membangun sebuah sistem.

Perilaku dari sistem adalah bagaimana cara sistem beraksi dan juga bereaksi. Perilaku ini mendefinisikan aktifitas sistem yang dapat dilihat dari luar dan diuji. Perilaku sistem ini digambarkan ke dalam *use case*. *Use case* mendeskripsikan sistem, lingkungan, serta hubungan antara sistem dengan lingkungannya. *Use case* juga digunakan untuk menyusun *behavioral things* dari suatu model.

Berikut ini merupakan contoh use case diagram:





Gambar 2.2 Contoh Use Case Diagram

Simbol-simbol atau notasi yang digunakan dalam pembuatan atau penyusunan *use case diagram* dapat dilihat pada Tabel 2.1 berikut.

Tabel 2.1 Notasi pada Use Case Diagram

| NO | GAMBAR | NAMA        | KETERANGAN   |
|----|--------|-------------|--|
| 1  |        | Actor       | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.                          |
| 2  |        | Include     | Menspesifikasikan bahwa use case sumber secara eksplisit.  |
| 3  |        | Extend      | Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.    |
| 4  |        | Association | Apa yang menghubungkan antara objek satu dengan objek lainnya.   |
| 5  |        | System      | Menspesifikasikan paket yang menampilkan sistem secara terbatas.   |
| 6  |        | Use Case    | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor. |

### 2.5.2 Sequence Diagram

*Sequence diagram* adalah diagram yang menekankan interaksi yang menekankan peraturan waktu dari pesan-pesan (Rumbaugh, Jacobson, & Booch,





2005). *Sequence diagram* menggambarkan suatu skenario atau langkah-langkah yang dilakukan dari suatu kejadian untuk mendapatkan output tertentu.

Diagram ini secara khusus berasosiasi dengan *use case diagram*. *Sequence diagram* juga memperlihatkan langkah demi langkah apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam *use case*. *Sequence diagram* juga dapat merubah atribut atau *method* pada *class* yang telah dibentuk oleh *class diagram*, bahkan menciptakan sebuah *class* baru. *Sequence diagram* memodelkan aliran logika dalam sebuah sistem dalam cara yang visual.

Berikut ini merupakan notasi atau simbol-simbol yang digunakan dalam pembuatan *sequence diagram* dapat dilihat pada Tabel 2.2 berikut ini.

Tabel 2.2 Notasi pada *Sequence Diagram*

| NO | GAMBAR | NAMA                 | KETERANGAN  |
|----|--------|----------------------|---|
| 1  |        | <i>LifeLine</i>      | Objek <i>entity</i> , antarmuka yang saling berinteraksi.   |
| 2  |        | <i>Actor</i>         | Digunakan untuk menggambarkan user atau pengguna.   |
| 3  |        | <i>Message</i>       | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi. |
| 4  |        | <i>Boundary</i>      | Digunakan untuk menggambarkan sebuah form.  |
| 5  |        | <i>Control Class</i> | Digunakan untuk menghubungkan <i>boundary</i> dengan tabel.   |
| 6  |        | <i>Entity Class</i>  | Digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan.                                    |

### 2.5.3 *Class Diagram*

*Class diagram* adalah sekumpulan antarmuka, kelas, dan kerjasama serta hubungannya (Rumbaugh, Jacobson, & Booch, 2005). *Class diagram* memaparkan hubungan antar *class* dalam suatu sistem yang dibuat dan bagaimana cara agar hubungan itu bisa saling berkolaborasi untuk mencapai suatu tujuan tertentu.

*Class* terdiri dari nama *class*, atribut, dan operasi/metode. Nama *class* berfungsi untuk memberikan nama atau identitas kelas, atribut berfungsi untuk memberikan karakteristik pada data yang ada pada suatu objek dalam kelas, dan yang terakhir operasi berfungsi memberikan fungsi ke dalam objek.





Berikut ini merupakan contoh *class diagram* yang dapat dilihat pada Gambar 2.3 dan simbol dari *class diagram* yang dapat dilihat pada Tabel 2.3:



Gambar 2.3 Contoh *Class Diagram*

Tabel 2.3 Simbol *Class Diagram*

| NO | GAMBAR | NAMA                  | KETERANGAN   |
|----|--------|-----------------------|--|
| 1  |        | <i>Generalization</i> | Hubungan dimana objek anak ( <i>descendant</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).                  |
| 2  |        | <i>Class</i>          | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.  |
| 3  |        | <i>Collaboration</i>  | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi aktor.   |
| 4  |        | <i>Realization</i>    | Operasi yang benar-benar dilakukan oleh suatu objek.   |
| 5  |        | <i>Dependency</i>     | Hubungan dimana perubahan yang terjadi pada suatu element mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri. |
| 6  |        | <i>Association</i>    | Apa yang menghubungkan antara objek suatu dengan objek yang lain.  |

## 2.6 Model View Controller (MVC)

*Model View Controller* (MVC) merupakan metode yang digunakan untuk membuat aplikasi dengan memisahkan data atau *query* (*model*) dari tampilan atau *user interface* (*view*) dan cara pemrosesannya (*controller*). Pembuatan aplikasi dengan menggunakan MVC biasanya dikemas dalam *framework*, sehingga pengembang aplikasi tinggal menggunakan *framework* yang sudah disediakan (Badiyanto, 2013).

Berikut ini adalah tiga komponen utama dari suatu pola MVC dalam aplikasi yaitu:

1. *Model*, merupakan bagian yang berhubungan secara langsung dengan database untuk manipulasi data, menhandel validasi pada bagian *controller*, dan tidak bisa berhubungan secara langsung dengan *view*.





2. *View*, merupakan bagian yang menhandel *presentation logic*. *View* memiliki fungsi untuk menerima dan merepresentasikan data kepada pengguna. Pada aplikasi web, *view* biasanya berupa *template* HTML yang diatur oleh *controller*.

3. *Controller*, merupakan bagian yang menhandel bagian *model* dengan *view*. *Controller* memiliki fungsi untuk menerima *request* dan data dari pengguna, lalu menentukan apa yang akan diproses selanjutnya dari aplikasi.

## 2.7 Codeigniter

Codeigniter adalah sebuah *web application network* yang bersifat *open source* yang digunakan untuk membangun aplikasi php dinamis. Codeigniter menjadi sebuah *framework* PHP dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP yang dapat mempercepat pengembang untuk membuat sebuah aplikasi web. Selain ringan dan cepat, Codeigniter memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi kodenya. Dokumentasi yang lengkap inilah yang menjadi salah satu alasan kuat mengapa banyak orang memilih Codeigniter sebagai *framework* pilihannya. Codeigniter juga memungkinkan para pengembang untuk menggunakan *framework* secara parsial atau keseluruhan, artinya memberikan kebebasan kepada para pengembang untuk menulis bagian-bagian kode tertentu di dalam aplikasi menggunakan cara konvensional atau dengan *syntax* umum di dalam PHP, tidak harus menggunakan aturan penulisan kode di Codeigniter (Septian, 2011).

## 2.8 Bahasa Pemrograman

Bahasa pemrograman menurut Kurniadi (2011) adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas tertentu. Berikut ini adalah bahasa pemrograman yang digunakan untuk membangun sistem ini:

### 2.8.1 HTML

*Hypertext Markup Language* (HTML) adalah bahasa markup yang digunakan untuk membuat suatu halaman web. Menurut Nugroho (2006) HTML merupakan bahasa pemformatan teks untuk dokumen-dokumen pada jaringan komputer yang sering disebut *world wide web*. HTML juga termasuk bahasa pemrograman yang fleksibel karena dapat digabungkan dengan bahasa pemrograman lainnya seperti PHP, ASP, JSP, dan JavaScript.

Untuk format teks ada beberapa tag dalam dokumen-dokumen HTML. Tag yang lainnya memberitahukan bagaimana menanggapi aksi-aksi yang datang dari pengguna. Lalu tag penting lainnya adalah link yang mengandung *Uniform Resource Locator* (URL), yang merujuk pada dokumen lain di server yang sama atau komputer lain yang ada di global jaringan internet.





## 2.8.2 PHP

*Personal Home Page* (PHP) merupakan bahasa pemrograman yang tertanam pada HTML untuk membuat web menjadi lebih dinamis dan memiliki sifat server side. PHP memiliki fungsi-fungsi sebagai berikut:

### 1. Mempersingkat tatanan HTML dan CSS

Dengan PHP, dapat mempersingkat tatanan penulisan dari HTML karena tidak efektif jika tatanan suatu web terlalu panjang. Maka dari itu dengan menggunakan PHP, dapat meminimalisir jumlah baris pada HTML.

### 2. Input data

Input data maksudnya untuk tempat penyimpanan dan penginputan data pada sistem database, misalnya MySQL.

### 3. Manajemen *cookie* dan *session*

Fungsi ini untuk menyimpan *username* dan *password* pengguna pada browser.

### 4. Kompres teks

Fungsi ini untuk mengompres teks menjadi lebih singkat, padat, dan jelas.

## 2.8.3 CSS

*Cascading Style Sheet* (CSS) adalah salah satu bahasa pemrograman untuk mengontrol format tampilan sebuah halaman web yang ditulis menggunakan HTML. CSS biasanya digunakan untuk mendesain halaman HTML dan XHTML, tapi CSS dapat diaplikasikan ke segala dokumen XML, termasuk SVG dan XUL bahkan android.

CSS dibuat untuk memisahkan konten utama dengan tampilan dokumen yang meliputi layout, warna dan *font*. Pemisahan ini dapat meningkatkan daya akses konten pada web, menyediakan lebih banyak fleksibilitas dan kontrol dalam spesifikasi dari sebuah karakteristik dari sebuah tampilan, memungkinkan untuk membagi halaman untuk sebuah *formatting* dan mengurangi kerumitan dalam penulisan kode dan struktur dari konten.

## 2.8.4 JavaScript

*Javascript* dalam buku Mac Bride (2007) merupakan bahasa pemrograman berbasis browser. Kode-kodenya ditulis langsung ke dalam HTML dari halaman-halaman web dan diterjemahkan serta dieksekusi sebagai respon terhadap aktivitas-aktivitas dalam halaman web.

Berlandaskan pada definisi *JavaScript* yang disampaikan, Bride (2007) menjabarkan *JavaScript* secara lebih lanjut sebagai berikut:

1. *JavaScript* berinteraksi dengan browser. *JavaScript* dapat membaca informasi dari halaman web, seperti data yang dimasukkan melalui *field-field* dalam sebuah *form*.





2. *JavaScript* adalah bahasa *event-driven* berbasis objek. *JavaScript* memberikan respon terhadap aktivitas yang dilakukan terhadap objek-objek pada halaman web

## 2.9 MySQL

*My Structure Query Language* (MySQL) dalam buku Arief (2011) adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web ya menggunakan database sebagai sumber pengolahan datanya.

MySQL merupakan database yang pertama kali didukung oleh bahasa pemrograman *script* untuk internet (PHP dan Perl). MySQL dan PHP dianggap sebagai pasangan software pembangun aplikasi web yang ideal. MySQL lebih sering digunakan untuk membangun aplikasi berbasis web, umumnya pengembangan aplikasinya menggunakan bahasa pemrograman *script* PHP.

## 2.10 Pengujian Perangkat Lunak

Pengujian Perangkat Lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain dan pengkodean. Pengujian merepresentasikan ketidaknormalan yang terjadi pada pengembangan perangkat lunak. Selama definisi awal dari fase pembangunan, pengembangan berusaha untuk membangun perangkat lunak dari konsep yang abstrak sampai dengan implementasi.

### 2.10.1 Pengujian *White-Box*

Pengujian *white-box* adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara prosedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Pengujian *white-box* berfokus pada verifikasi dari unit terkecil yang didapatkan dari perancangan komponen atau modul dari perangkat lunak (Pressman, 2010). *Test case* dilakukan untuk memastikan bahwa semua statemen pada program telah dieksekusi paling tidak satu kali selama pengujian dan bahwa semua kondisi logis telah diuji. Pengujian *basic path*, teknik pengujian *white-box*, menggunakan grafik (matriks grafiks) untuk melakukan serangkaian pengujian yang independent secara linear yang akan memastikan cakupan. Teknik pengujian *white-box* digunakan dalam melakukan pengujian unit dan pengujian integrasi.

### 2.10.2 Pengujian *Black-Box*

Pengujian *Black-Box* atau dapat disebut pengujian validasi menurut Sommerville (2001) adalah pengujian yang mempunyai pendekatan yang pengujiannya diturunkan dari spesifikasi komponen, dan dapat juga disebut pengujian fungsional karena hanya memperhatikan fungsionalitas sistem bukan implementasi dari sistem.

Metode uji coba *black-box* memfokuskan pada keperluan fungsional dari *software*. Karena itu uji coba *black-box* memungkinkan pengembang *software*





untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Uji coba *black-box* bukan merupakan alternatif dari uji coba *white-box*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode *white-box*. Uji coba *black-box* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

- a. Fungsi-fungsi yang salah atau hilang
- b. Kesalahan *interface*
- c. Kesalahan dalam struktur data atau akses database eksternal.
- d. Kesalahan kinerja
- e. Kesalahan Inisialisasi dan terminasi

## 2.11 Aplikasi Penunjang

### 2.11.1 Sublime Text

*Sublime Text* adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai *platform operating system* dengan menggunakan teknologi Phyton API. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim, Aplikasi ini sangatlah fleksibel dan *powerfull*. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan *sublime-packages*. *Sublime Text* bukanlah aplikasi *opensource* dan juga aplikasi yang dapat digunakan dan didapatkan secara gratis, akan tetapi beberapa fitur pengembangan fungsionalitas (*packages*) dari aplikasi ini merupakan hasil dari temuan dan mendapat dukungan penuh dari komunitas serta memiliki linsensi aplikasi gratis.

*Sublime Text* mendukung berbagai bahasa pemrograman dan mampu menyajikan fitur *syntax highlight* hampir di semua bahasa pemrograman yang didukung ataupun dikembangkan oleh komunitas seperti; C, C++, C#, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL, Textile and XML.

### 2.11.2 XAMPP

XAMPP ialah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan campuran dari beberapa program. Yang mempunyai fungsi sebagai server yang berdiri sendiri (localhost), yang terdiri dari program MySQL database, Apache HTTP Server, dan penerjemah ditulis dalam bahasa pemrograman PHP dan Perl.

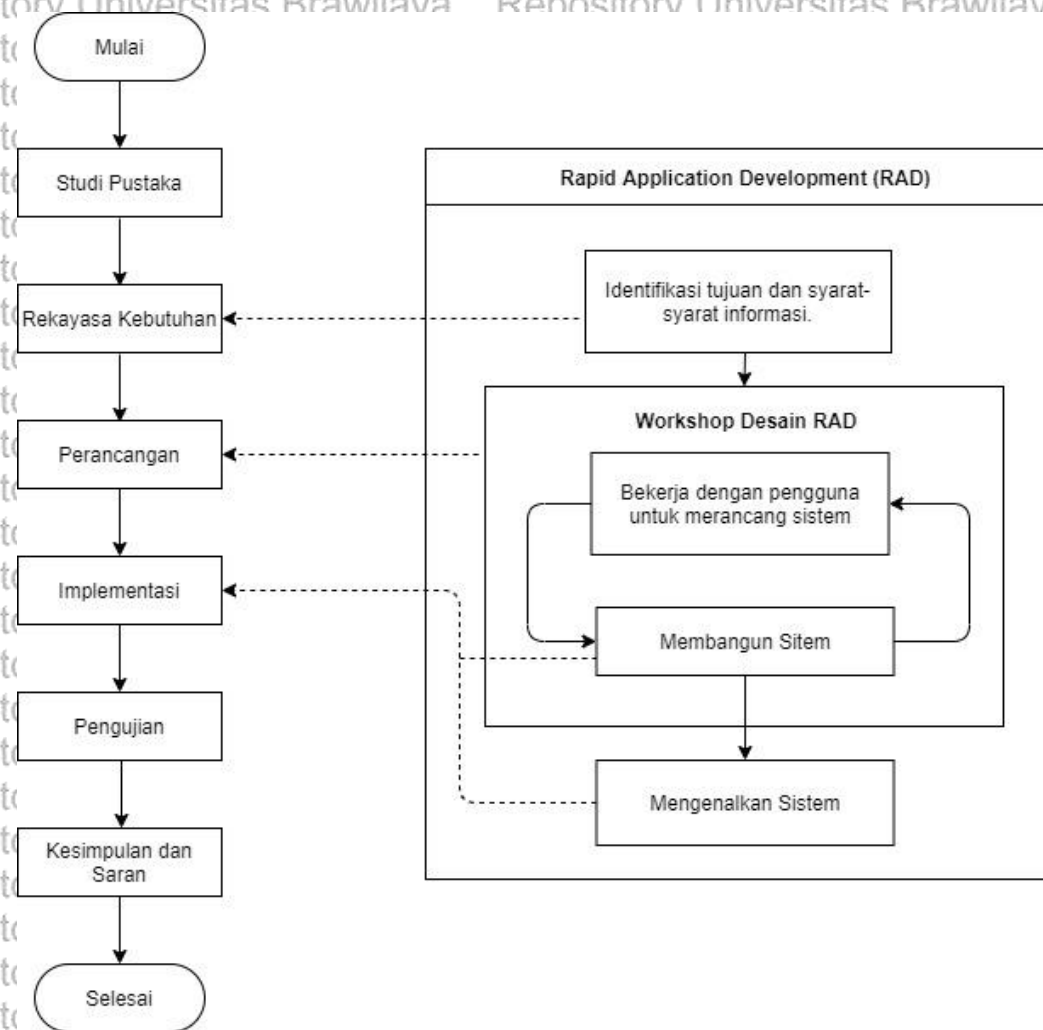
Nama XAMPP merupakan singkatan dari X (empat sistem operasi), Apache, MySQL, PHP dan Perl. Program ini tersedia di bawah GNU General Public License dan bebas, adalah mudah untuk menggunakan web server yang dapat melayani tampilan halaman web yang dinamis. Jika ingin mendapatkan xampp dapat mendownload langsung dari situs resminya.





## BAB 3 METODOLOGI PENELITIAN

Bab metodologi menjelaskan tahapan-tahapan yang akan dilakukan selama penelitian “Pengembangan Sistem Konsultasi dan Pencarian Pengacara Berbasis Website”. Pada proses untuk pengembangan sistem ini menggunakan pengembangan perangkat lunak *Rapid Application Development* (RAD) Model. Berikut ini merupakan diagram alir dari metodologi penelitian pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metodologi Penelitian

### 3.1 Studi Pustaka

Dalam penelitian ini, penulis melakukan studi pustaka terkait ilmu yang berkaitan dengan penelitian yang sedang dilakukan. Kajian pustaka dan dasar teori yang digunakan dalam penelitian ini didapatkan dari buku-buku, jurnal, karya ilmiah, paper hasil konferensi, artikel, dan dokumen-dokumen yang relevan dengan topik. Berikut ini merupakan beberapa bidang ilmu yang berhubungan dengan pengembangan Sistem Konsultasi dan Pencarian Pengacara:





1. Kajian Pustaka
2. *Work Breakdown Structure* (WBS)
3. *Software Development Life Cycle* (SDLC)
4. *Rapid Application Development* (RAD)
5. Analisis Kebutuhan
6. *Unified Modelling Language* (UML)
7. *Model View Controller* (MVC)
8. Codeigniter
9. Bahasa Pemrograman
10. MySQL
11. Pengujian Perangkat Lunak
12. Aplikasi Penunjang

### 3.2 Rekayasa Kebutuhan

Rekayasa kebutuhan adalah proses untuk menentukan kebutuhan yang digunakan untuk membuat sistem dan memastikan bahwa kebutuhan sistem yang dibangun sesuai dengan kebutuhan dari penggunanya. Sistem yang akan dibangun dapat menghasilkan suatu gambaran dengan adanya rekayasa kebutuhan sistem ini. Metode yang digunakan pada penelitian sistem konsultasi dan pencarian pengacara adalah *Object-Oriented Analysis* (OOA).

Data pada rekayasa kebutuhan didapatkan dari wawancara kepada advokat yang bernama Bapak Yayan Riyanto, S.H., M.H dan Ibu Rafiqah Aswinda Desovi, S.H. Dari wawancara tersebut didapatkan kebutuhan fungsional untuk sistem. Setelah mendapatkan kebutuhan fungsional, kemudian membuat *use case diagram* yang merupakan interaksi antara aktor dengan sistem. Selanjutnya membuat *use case scenario* untuk mendeskripsikan scenario dari tiap *use case* yang dibuat.

### 3.3 Perancangan Sistem

Perancangan sistem merupakan gambaran bagaimana sistem yang dibangun agar pemangku kepentingan mengetahui lebih detail gambaran sistem. Perancangan akan menjelaskan gambaran dari sistem yang akan dibuat dalam segi model, basis data, maupun komponen. Proses perancangan sistem terdiri dari:

1. Pemodelan *Sequence Diagram*

Pada proses perancangan ini adalah dengan membuat *sequence diagram* yang merupakan interaksi antar objek dalam melakukan suatu proses tertentu.

2. Pemodelan *Class Diagram*



Pada proses perancangan ini adalah dengan membuat *class diagram* yang merupakan gambaran dari class entitas yang terhubung dalam sistem.

### 3. Perancangan Komponen

Pada proses perancangan ini adalah dengan membuat perancangan komponen yang merupakan sampel dari penulisan algoritme dalam bentuk pseudocode.

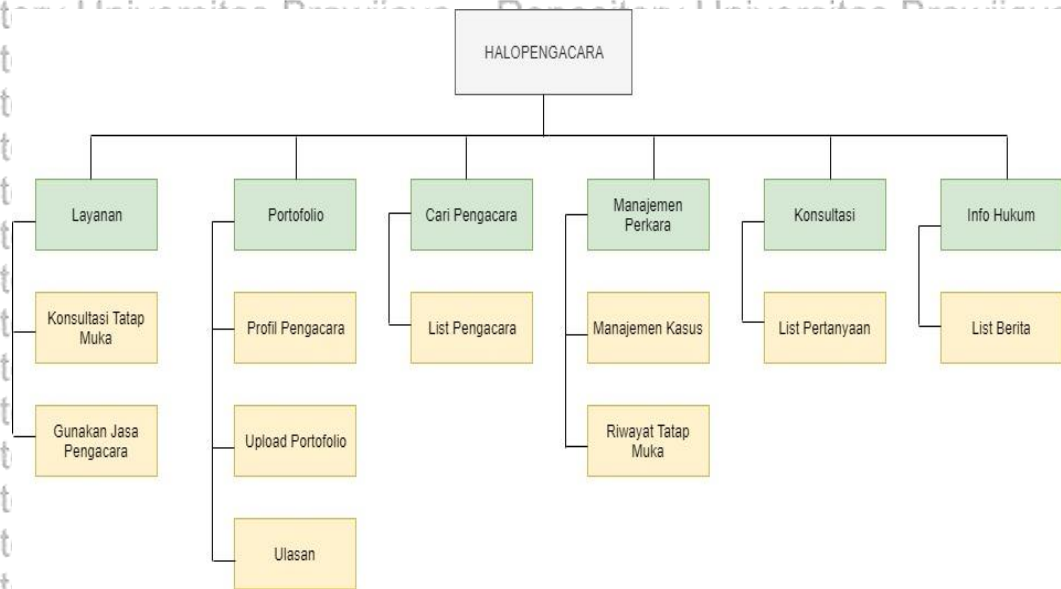
### 4. Perancangan Data

Pada proses perancangan ini adalah dengan membuat perancangan data dengan membuat ERD yang nantinya akan diimplementasikan.

### 5. Perancangan Antarmuka

Pada proses perancangan ini adalah dengan membuat perancangan antarmuka untuk menggambarkan tata letak dari sistem yang akan dibangun dan diimplementasikan.

Terdapat perancangan sistem lain yang tidak terdapat pada yang disebutkan di atas yaitu WBS (*Work Breakdown Structure*) untuk membagi perencanaan dasar sistem menjadi ke dalam beberapa fase. Berikut ini adalah WBS dari Sistem Konsultasi dan Pencarian Pengacara Berbasis Website:



**Gambar 3.2 Work Breakdown Structure (WBS)**

Gambar 3.2 merupakan WBS dari Pengembangan Sistem Konsultasi dan Pencarian Pengacara Berbasis Website. Pada WBS di atas terdapat enam modul diantaranya layanan, portofolio, cari pengacara, manajemen kasus, konsultasi, dan info hukum.

Modul pertama yaitu layanan, pada modul ini mempunyai dua *task* yaitu: konsultasi tatap muka, dan gunakan jasa pengacara. Modul kedua yaitu portofolio, pada modul ini mempunyai tiga *task* yaitu: profil pengacara, upload portofolio, dan ulasan. Modul ketiga yaitu cari pengacara, pada modul ini mempunyai satu





*task* yaitu: list pengacara. Modul keempat yaitu manajemen perkara, pada modul ini mempunyai dua *task* yaitu: manajemen kasus dan riwayat tatap muka. Modul kelima yaitu konsultasi, pada modul ini mempunyai satu *task* yaitu: list pertanyaan. Modul keenam yaitu info hukum, pada modul ini memiliki satu *task* yaitu list berita.

### 3.4 Implementasi Sistem

Pada tahap implementasi sistem adalah tahap untuk pembuatan sistem berdasarkan analisis kebutuhan dan perancangan sistem yang sudah dilakukan pada tahap sebelumnya ke dalam kode program. Pada penelitian ini, sistem diimplementasikan menggunakan *framework* codeigniter. Proses implementasi sistem pada penelitian ini terdiri dari:

#### 1. Spesifikasi Sistem

Pada spesifikasi sistem dilakukan dengan menspesifikasikan perangkat keras dan perangkat lunak yang digunakan dalam pembuatan sistem.

#### 2. Implementasi Basisdata

Pada implementasi basisdata dilakukan dengan menggunakan MySQL dimana diakses melalui PhpMyAdmin.

#### 3. Implementasi Komponen

Pada implementasi komponen merupakan implementasi algoritme yang didefinisikan pada perancangan komponen menjadi kode program.

#### 4. Implementasi Antarmuka

Pada implementasi antarmuka dilakukan dengan menggunakan bahasa pemrograman PHP, CSS, *JavaScript*, dan *Bootstrap*.

### 3.5 Pengujian Sistem

Pada tahap ini penulis melakukan pengujian terhadap program yang sudah dibuat. Pengujian yang dilakukan dalam penelitian ini adalah:

#### 1. Pengujian Unit

Pengujian unit menggunakan teknik pengujian *white-box*. Basis *path* berdasarkan algoritme pada tiap-tiap klas digunakan dalam pengujian *white-box* ini. Pada penelitian ini dilakukan dengan menjalankan klas *Driver* untuk menjalankan klas yang diuji.

#### 2. Pengujian Integrasi

Pengujian integrasi adalah pengujian yang dilakukan terhadap unit-unit program yang memiliki keterkaitan atau integrasi dengan fokus terhadap tatap muka.





### 3. Pengujian Validasi

Pengujian validasi menggunakan teknik pengujian *black-box*. Pengujian fungsionalitas dengan cara menguji apakah sistem berjalan sesuai dengan parameter yang ditentukan dalam kondisi normal maupun alternatif merupakan arti dari pengujian *black-box*.

### 4. Pengujian *Compatibility*

Pengujian *compatibility* dilakukan untuk menguji sistem yang dikembangkan dapat bekerja pada *browser* dan operating system yang berbeda.

## 3.6 Kesimpulan dan Saran

Pada tahap ini merupakan proses untuk menarik kesimpulan dan saran atas apa yang dilakukan selama pengerjaan penelitian. Dasar pengambilan kesimpulan adalah hasil analisis yang sudah dilakukan untuk dibuat kesimpulan dari keseluruhan penelitian.





## BAB 4 REKAYASA KEBUTUHAN

### 4.1 Rekayasa Kebutuhan

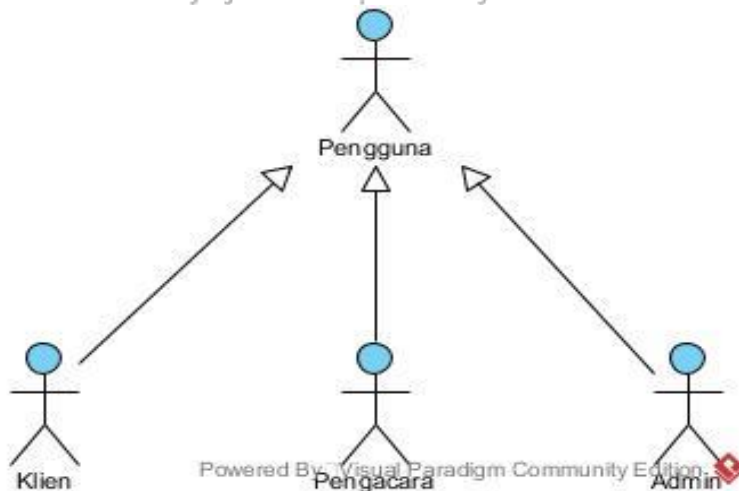
Rekayasa kebutuhan adalah tahapan yang dilakukan untuk menggambarkan dan menspesifikasikan kebutuhan pada Sistem Konsultasi dan Pencarian Pengacara.

#### 4.1.1 Gambaran Umum Sistem

Sistem Konsultasi dan Pencarian Pengacara merupakan sistem berbasis website yang dapat digunakan untuk berkonsultasi permasalahan yang dimiliki pengguna dan juga dapat mencari pengacara yang cocok untuk menangani kasus yang dimiliki oleh pengguna. Untuk pengacara dapat mendapatkan klien dengan mudah dan dapat memajemen kasus yang dimiliki oleh klien dengan mudah. Terdapat tiga pemangku kepentingan dalam sistem yaitu pengguna, pengacara, dan admin. Pengguna dapat bertanya dan mencari pengacara pada sistem. Pengacara dapat menjawab pertanyaan pengguna, menerima jadwal untuk konsultasi tatap muka kasus pengguna, dan dapat menginformasikan progress kasus yang dimiliki klien pada sistem. Admin dapat memvalidasi pengacara dari portofolio yang diajukan oleh pengacara saat mendaftar menjadi pengacara.

#### 4.1.2 Identifikasi Aktor

Identifikasi aktor merupakan penggambaran dari pengguna yang terlibat dalam sistem dan juga memberikan gambaran apa yang dikerjakan oleh sistem. Gambar 4.1 dibawah ini merupakan identifikasi aktor dari Sistem Konsultasi dan Pencarian Pengacara berbasis Web dimana terdapat empat aktor yaitu pengguna, klien, pengacara, dan admin.



Gambar 4.1 Identifikasi Aktor

Pada Tabel 4.1 merupakan Identifikasi pemangku kepentingan dimana masing-masing pemangku kepentingan dijelaskan deskripsi tugas dan peran. Pengguna



merupakan aktor umum yang merupakan pengguna yang belum terdaftar pada sistem atau belum berhasil melakukan login. Sedangkan klien, pengacara, dan admin merupakan aktor yang telah berhasil melakukan login sesuai dengan levelnya.

**Tabel 4.1 Identifikasi Pemangku Kepentingan**

| Aktor    | Spesifikasi | Deskripsi  |
|----------|-------------|--|
| Pengguna | Klien       | Klien merupakan pihak yang akan melakukan konsultasi dan pencarian pengacara pada sistem.  |
|          | Pengacara   | Pengacara merupakan pihak yang akan menerima kasus klien untuk ditangani dan menjawab konsultasi yang dilakukan oleh klien. Dan dapat memanajemen kasus klien pada sistem. |
|          | Admin       | Admin adalah pihak yang bertanggung jawab atas manajemen sistem.   |

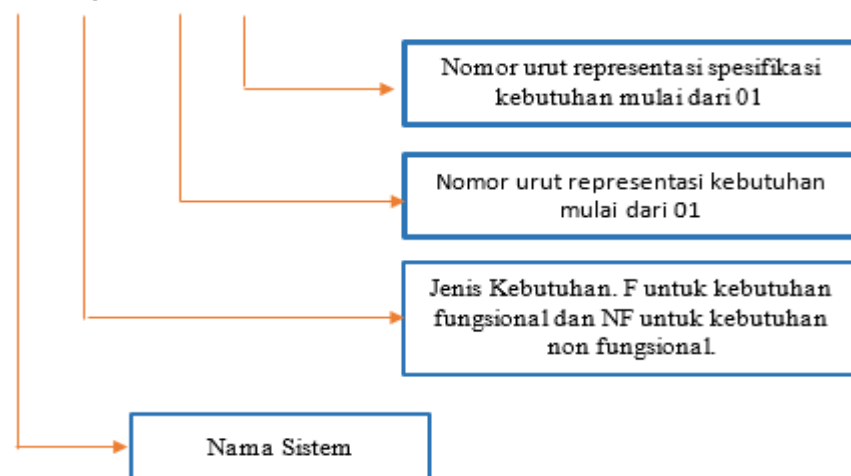
## 4.2 Spesifikasi Kebutuhan

Setelah proses gambaran umum sistem dan identifikasi aktor, selanjutnya dilakukan proses spesifikasi kebutuhan. Pada proses ini akan mendeskripsikan fungsi-fungsi yang dapat dilakukan oleh sistem. Kebutuhan dalam sistem dibagi menjadi dua yaitu kebutuhan fungsional dan kebutuhan non fungsional.

### 4.2.1 Aturan Penomoran

Kode kebutuhan : SKPP-F/NF-01-01

**SKPP – F/NF – 01 – 01**



**Gambar 4.2 Aturan Penomoran**





## 4.2.2 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang berisi proses-proses apa saja yang bisa dilakukan oleh sistem. Pada sistem konsultasi dan pencarian pengacara ini terdapat dua iterasi kebutuhan fungsional.

### 4.2.2.1 Kebutuhan Fungsional Iterasi 1

Pada Tabel 4.2 di bawah ini akan menjelaskan kebutuhan fungsional iterasi pertama dari sistem konsultasi dan pencarian pengacara.

**Tabel 4.2 Kebutuhan Fungsional Iterasi 1**

| No. | Kode         | Kebutuhan Sistem   | Use Case     | Aktor    |
|-----|--------------|--|--------------|----------|
| 1   | SKPP-F-01    | Sistem harus menyediakan fungsi <i>login</i> bagi aktor agar dapat masuk ke dalam sistem sebagai aktor yang terdaftar. |              |          |
|     | SKPP-F-01-01 | Sistem harus mempunyai halaman <i>form login</i> .   | <i>Login</i> | Pengguna |
|     | SKPP-F-01-02 | <i>Form login</i> harus menyediakan <i>field</i> untuk <i>email</i> dan <i>password</i> .                              |              |          |
|     | SKPP-F-01-03 | <i>Form login</i> harus menyediakan tombol masuk untuk dapat masuk ke dalam sistem.                                    |              |          |
| 2   | SKPP-F-02    | Sistem harus menyediakan fungsi registrasi agar aktor dapat mendaftar ke dalam sistem.                                 |              |          |
|     | SKPP-F-02-01 | Sistem harus mempunyai halaman <i>form registrasi</i> .  |              |          |
|     | SKPP-F-02-02 | <i>Form registrasi</i> harus menyediakan <i>field</i> untuk <i>username</i> , <i>email</i> , dan <i>password</i> .     | Registrasi   | Pengguna |
|     | SKPP-F-02-03 | <i>Form registrasi</i> harus menyediakan <i>select option</i> untuk aktor memilih daftar sebagai pengacara atau klien. |              |          |





| No. | Kode         | Kebutuhan Sistem   | Use Case          | Aktor     |
|-----|--------------|--|-------------------|-----------|
| 3   | SKPP-F-02-04 | Form registrasi harus menyediakan tombol daftar untuk daftar ke dalam sistem.                    |                   |           |
|     | SKPP-F-03    | Sistem harus menyediakan fungsi <i>logout</i> bagi aktor agar dapat keluar dari sistem.          | Logout            | Pengguna  |
| 4   | SKPP-F-04    | Sistem harus menyediakan fungsi tambah info hukum untuk menambahkan info hukum bagi pengacara.   |                   |           |
|     | SKPP-F-04-01 | Sistem harus mempunyai halaman <i>form</i> tambah info hukum.                                    |                   |           |
|     | SKPP-F-04-02 | Sistem harus mempunyai tombol Tambah Info Hukum pada halaman postingan saya.                     |                   |           |
|     | SKPP-F-04-03 | Form tambah info hukum harus menyediakan <i>field</i> untuk judul info hukum dan isi info hukum. | Tambah Info Hukum | Pengacara |
|     | SKPP-F-04-04 | Form tambah info hukum harus menyediakan <i>select option</i> untuk memilih kategori info hukum. |                   |           |
| 5   | SKPP-F-04-05 | Form tambah info hukum harus menyediakan <i>field</i> untuk mengupload foto info hukum.          |                   |           |
|     | SKPP-F-04-06 | Form tambah info hukum harus menyediakan tombol Simpan untuk menyimpan info hukum.               |                   |           |
| 5   | SKPP-F-05    | Sistem harus menyediakan fungsi edit info hukum untuk mengedit info hukum.                       | Edit Info Hukum   | Pengacara |





| No. | Kode         | Kebutuhan Sistem  | Use Case         | Aktor     |
|-----|--------------|---|------------------|-----------|
|     | SKPP-F-05-01 | Sistem harus mempunyai halaman <i>form</i> edit info hukum.   |                  |           |
|     | SKPP-F-05-02 | Sistem harus menyediakan tombol edit yang berada pada masing-masing judul info hukum.   |                  |           |
|     | SKPP-F-05-03 | <i>Form</i> edit info hukum harus menyediakan <i>field</i> untuk judul info hukum dan isi info hukum.                           |                  |           |
|     | SKPP-F-05-04 | <i>Form</i> edit info hukum harus menyediakan <i>select option</i> untuk memilih kategori info hukum.                           |                  |           |
|     | SKPP-F-05-05 | <i>Form</i> edit info hukum harus menyediakan <i>field</i> untuk mengupload foto info hukum.                                    |                  |           |
|     | SKPP-F-05-06 | <i>Form</i> edit info hukum harus menyediakan tombol Simpan untuk menyimpan info hukum.   |                  |           |
| 6   | SKPP-F-06    | Sistem harus menyediakan fungsi hapus info hukum untuk menghapus info hukum dari sistem.  | Hapus Info Hukum | Pengacara |
|     | SKPP-F-06-01 | Sistem harus menyediakan tombol hapus yang berada pada masing-masing judul info hukum.  |                  |           |
| 7   | SKPP-F-07    | Sistem harus menyediakan fungsi kirim portofolio untuk mengirimkan portofolio bagi pengacara yang mendaftarkan ke dalam sistem. | Kirim Portofolio | Pengacara |





| No. | Kode         | Kebutuhan Sistem   | Use Case        | Aktor     |
|-----|--------------|--|-----------------|-----------|
|     | SKPP-F-07-01 | Sistem harus mempunyai halaman <i>form</i> kirim portofolio.   |                 |           |
|     | SKPP-F-07-02 | Sistem harus menyediakan tombol Ajukan Portofolio yang berada pada halaman profil pengacara.   |                 |           |
|     | SKPP-F-07-03 | <i>Form</i> kirim portofolio harus menyediakan <i>field</i> untuk nama pengacara, tempat lahir, tanggal lahir, nomor pokok wajib pajak (npwp), alamat kantor, provinsi, kabupaten, kecamatan, kelurahan, email kantor, nomor telepon, nomor handphone, pendidikan hukum, pengalaman, dan tentang saya. |                 |           |
|     | SKPP-F-07-04 | <i>Form</i> kirim portofolio harus menyediakan <i>checkbox</i> untuk memilih bidang keahlian dan spesialis.  |                 |           |
|     | SKPP-F-07-05 | <i>Form</i> kirim portofolio harus menyediakan <i>field</i> untuk <i>upload</i> scan ktp, ijazah sarjana hukum, dan sertifikat lulus UPA.  |                 |           |
|     | SKPP-F-07-06 | <i>Form</i> kirim portofolio harus menyediakan tombol Kirim untuk menyimpan portofolio pengacara.  |                 |           |
| 8   | SKPP-F-08    | Sistem harus menyediakan fungsi edit portofolio untuk mengedit portofolio bagi pengacara.  | Edit Portofolio | Pengacara |
|     | SKPP-F-08-01 | Sistem harus mempunyai halaman <i>form</i> edit portofolio.  |                 |           |





| No. | Kode         | Kebutuhan Sistem   | Use Case                   | Aktor     |
|-----|--------------|--|----------------------------|-----------|
|     | SKPP-F-08-02 | Sistem harus menyediakan tombol Edit Portofolio yang berada pada halaman profil pengacara.   |                            |           |
|     | SKPP-F-08-03 | Form edit portofolio harus menyediakan <i>field</i> untuk nama pengacara, tempat lahir, tanggal lahir, nomor pokok wajib pajak (npwp), alamat kantor, provinsi, kabupaten, kecamatan, kelurahan, email kantor, nomor telepon, nomor handphone, pendidikan hukum, pengalaman, dan tentang saya. |                            |           |
|     | SKPP-F-08-04 | Form edit portofolio harus menyediakan <i>checkbox</i> untuk memilih bidang keahlian dan spesialis.  |                            |           |
|     | SKPP-F-08-05 | Form edit portofolio harus menyediakan <i>field</i> untuk <i>upload</i> scan ktp, ijasah sarjana hukum, dan sertifikat lulus UPA.  |                            |           |
|     | SKPP-F-08-06 | Form edit portofolio harus menyediakan tombol <i>Update</i> untuk menyimpan data portofolio pengacara yang diedit.   |                            |           |
| 9   | SKPP-F-09    | Sistem harus menyediakan fungsi lihat daftar pertanyaan untuk menampilkan daftar pertanyaan yang dikirimkan oleh klien.  | Lihat Daftar<br>Pertanyaan | Pengacara |
| 10  | SKPP-F-10    | Sistem harus menyediakan fungsi jawab pertanyaan untuk menjawab pertanyaan yang dikirimkan oleh klien.   | Jawab<br>Pertanyaan        | Pengacara |





| No. | Kode         | Kebutuhan Sistem   | Use Case               | Aktor     |
|-----|--------------|--|------------------------|-----------|
|     | SKPP-F-10-01 | Sistem harus mempunyai halaman <i>form</i> jawab pertanyaan.   |                        |           |
|     | SKPP-F-10-02 | <i>Form</i> jawab pertanyaan harus menyediakan <i>field</i> untuk jawab pertanyaan.  |                        |           |
|     | SKPP-F-10-03 | <i>Form</i> jawab pertanyaan harus menyediakan tombol Kirim untuk mengirimkan jawaban pertanyaan.                              |                        |           |
| 11  | SKPP-F-11    | Sistem harus menyediakan fungsi lihat bookingan klien untuk menampilkan daftar bookingan yang dikirimkan klien pada pengacara. | Lihat Bookingan Klien  | Pengacara |
| 12  | SKPP-F-12    | Sistem harus menyediakan fungsi terima bookingan klien untuk menerima bookingan yang dikirimkan klien pada pengacara.          | Terima Bookingan Klien | Pengacara |
|     | SKPP-F-12-01 | Sistem harus menyediakan tombol Terima yang berada pada masing-masing bookingan klien.   |                        |           |
| 13  | SKPP-F-13    | Sistem harus menyediakan fungsi tolak bookingan klien untuk menolak bookingan yang dikirimkan klien pada pengacara.            | Tolak Bookingan Klien  | Pengacara |
|     | SKPP-F-13-01 | Sistem harus menyediakan tombol Tolak yang berada pada masing-masing bookingan klien.  |                        |           |
| 14  | SKPP-F-14    | Sistem harus menyediakan fungsi lihat ulasan klien untuk menampilkan ulasan yang diberikan oleh klien.                         | Lihat Ulasan Klien     | Pengacara |
| 15  | SKPP-F-15    | Sistem harus menyediakan fungsi lihat riwayat klien untuk menampilkan daftar   | Lihat Riwayat Klien    | Pengacara |





| No. | Kode         | Kebutuhan Sistem   | Use Case              | Aktor     |
|-----|--------------|--|-----------------------|-----------|
|     |              | klien yang menggunakan pengacara.  |                       |           |
| 16  | SKPP-F-16    | Sistem harus menyediakan fungsi laporkan klien untuk melaporkan klien yang melakukan pelanggaran.  |                       |           |
|     | SKPP-F-16-01 | Sistem harus menyediakan tombol Laporkan Klien yang berada pada booking pertemuan yang sudah selesai dan manajemen kasus yang sudah disetujui. |                       |           |
|     | SKPP-F-16-02 | Sistem harus mempunyai halaman <i>form</i> laporkan klien.   | Laporkan Klien        | Pengacara |
|     | SKPP-F-16-03 | <i>Form</i> laporkan klien harus menyediakan <i>field</i> untuk menuliskan alasan melaporkan klien.  |                       |           |
|     | SKPP-F-16-04 | <i>Form</i> laporkan klien harus menyediakan tombol Simpan untuk menyimpan laporkan klien.   |                       |           |
| 17  | SKPP-F-17    | Sistem harus menyediakan fungsi lihat jasa pengacara untuk menampilkan daftar jasa pengacara yang dikirimkan klien pada pengacara.             | Lihat Jasa Pengacara  | Pengacara |
| 18  | SKPP-F-18    | Sistem harus menyediakan fungsi terima jasa pengacara untuk menerima jasa pengacara yang dikirimkan oleh klien.                                | Terima Jasa Pengacara | Pengacara |
|     | SKPP-F-18-01 | Sistem harus menyediakan tombol Terima yang berada pada masing-masing jasa pengacara yang belum disetujui.                                     |                       |           |





| No. | Kode         | Kebutuhan Sistem  | Use Case             | Aktor     |
|-----|--------------|---|----------------------|-----------|
| 19  | SKPP-F-19    | Sistem harus menyediakan fungsi tolak jasa pengacara untuk menolak jasa pengacara yang dikirimkan oleh klien. | Tolak Jasa Pengacara | Pengacara |
|     | SKPP-F-19-01 | Sistem harus menyediakan tombol Tolak yang berada pada masing-masing jasa pengacara yang belum disetujui.     |                      |           |
| 20  | SKPP-F-20    | Sistem harus menyediakan fungsi tambah jadwal kerja untuk menambahkan jadwal kerja pengacara.                 | Tambah Jadwal Kerja  | Pengacara |
|     | SKPP-F-20-01 | Sistem harus menyediakan <i>form</i> tambah jadwal kerja.   |                      |           |
|     | SKPP-F-20-02 | <i>Form</i> tambah jadwal kerja harus menyediakan <i>field</i> tanggal dan jam kerja.                         |                      |           |
|     | SKPP-F-20-03 | <i>Form</i> tambah jadwal kerja harus menyediakan tombol Simpan untuk menyimpan jadwal kerja.                 |                      |           |
| 21  | SKPP-F-21    | Sistem harus menyediakan fungsi edit jadwal kerja untuk mengedit jadwal kerja pengacara.                      | Edit Jadwal Kerja    | Pengacara |
|     | SKPP-F-21-01 | Sistem harus menyediakan <i>form</i> edit jadwal kerja.   |                      |           |
|     | SKPP-F-21-02 | <i>Form</i> edit jadwal kerja harus menyediakan <i>field</i> tanggal dan jam kerja.                           |                      |           |
|     | SKPP-F-21-03 | <i>Form</i> edit jadwal kerja harus menyediakan tombol Simpan untuk mengedit jadwal kerja.                    |                      |           |
| 22  | SKPP-F-22    | Sistem harus menyediakan fungsi hapus jadwal kerja  |                      | Pengacara |





| No. | Kode         | Kebutuhan Sistem  | Use Case                | Aktor |
|-----|--------------|---|-------------------------|-------|
|     |              | untuk menghapus jadwal kerja dari sistem.   |                         |       |
|     | SKPP-F-22-01 | Sistem harus menyediakan tombol hapus yang berada pada masing-masing jadwal kerja pengacara.                  | Hapus Jadwal Kerja      |       |
| 23  | SKPP-F-23    | Sistem harus menyediakan tombol hapus yang berada pada masing-masing judul info hukum.                        |                         |       |
|     | SKPP-F-23-01 | Sistem harus menyediakan tombol Gunakan Jasa Pengacara pada profil pengacara.                                 |                         |       |
|     | SKPP-F-23-02 | Sistem harus mempunyai halaman form pilih jasa pengacara.   | Pilih Jasa Pengacara    | Klien |
|     | SKPP-F-23-03 | Form pilih jasa pengacara harus menyediakan field untuk ceritakan masalah klien, nomor telepon, dan email.    |                         |       |
|     | SKPP-F-23-04 | Form pilih jasa pengacara harus menyediakan tombol Kirim untuk mengirimkan permintaan gunakan jasa pengacara. |                         |       |
| 24  | SKPP-F-24    | Sistem harus menyediakan fungsi batalkan jasa pengacara untuk membatalkan jasa pengacara yang dipesan.        |                         |       |
|     | SKPP-F-24-01 | Sistem harus menyediakan tombol batalkan jasa pengacara pada progres kasus.                                   | Batalkan Jasa Pengacara | Klien |
|     | SKPP-F-24-02 | Sistem harus mempunyai halaman form batalkan jasa pengacara.  |                         |       |





| No. | Kode         | Kebutuhan Sistem   | Use Case                    | Aktor |
|-----|--------------|--|-----------------------------|-------|
|     | SKPP-F-24-03 | <i>Form</i> batalkan jasa pengacara harus mempunyai <i>field</i> alasan membatalkan pengacara.   |                             |       |
|     | SKPP-F-24-04 | <i>Form</i> batalkan jasa pengacara harus menyediakan tombol Simpan untuk menyimpan alasan membatalkan jasa pengacara.   |                             |       |
| 25  | SKPP-F-25    | Sistem harus menyediakan fungsi pilih konsultasi tatap muka untuk membuat janji konsultasi tatap muka dengan pengacara.  |                             |       |
|     | SKPP-F-25-01 | Sistem harus menyediakan tombol Konsultasi Tatap Muka pada profil pengacara.   |                             |       |
|     | SKPP-F-25-02 | Sistem harus mempunyai halaman <i>form</i> pilih konsultasi tatap muka.  | Pilih Konsultasi Tatap Muka | Klien |
|     | SKPP-F-25-03 | <i>Form</i> pilih konsultasi tatap muka harus menyediakan <i>field</i> untuk bidang keahlian, tanggal konsultasi, waktu konsultasi, nomor <i>handphone</i> , <i>email</i> , dan inti permasalahan. |                             |       |
|     | SKPP-F-25-04 | <i>Form</i> pilih konsultasi tata muka harus menyediakan tombol Kirim untuk mengirimkan permintaan konsultasi tatap muka kepada pengacara.   |                             |       |
| 26  | SKPP-F-26    | Sistem harus menyediakan fungsi lihat bookingan saya untuk menampilkan daftar bookingan konsultasi tatap   | Lihat Bookingan Saya        | Klien |





| No. | Kode         | Kebutuhan Sistem  | Use Case                | Aktor |
|-----|--------------|---|-------------------------|-------|
|     |              | muka klien pada pengacara tertentu.   |                         |       |
| 27  | SKPP-F-27    | Sistem harus menyediakan fungsi lihat riwayat aktivitas menampilkan riwayat aktivitas klien.  | Lihat Riwayat Aktivitas | Klien |
| 28  | SKPP-F-28    | Sistem harus menyediakan fungsi beri ulasan untuk memberikan ulasan kepada pengacara.   |                         |       |
|     | SKPP-F-28-01 | Sistem harus menyediakan tombol Ulasan untuk memberikan ulasan.   |                         |       |
|     | SKPP-F-28-02 | Sistem harus mempunyai halaman form beri ulasan.  | Beri Ulasan             | Klien |
|     | SKPP-F-28-03 | Form beri ulasan harus menyediakan field radio button untuk memberikan rating dan field textarea untuk memberikan ulasan kepada klien |                         |       |
|     | SKPP-F-28-04 | Form beri ulasan harus menyediakan tombol Kirim untuk mengirimkan ulasan kepada pengacara.  |                         |       |
| 29  | SKPP-F-29    | Sistem harus menyediakan fungsi laporkan pengacara untuk melaporkan pengacara yang akan dilaporkan.                                   |                         |       |
|     | SKPP-F-29-01 | Sistem harus menyediakan link laporkan pengacara untuk melaporkan pengacara.  | Laporkan Pengacara      | Klien |
|     | SKPP-F-29-02 | Sistem harus mempunyai halaman form laporkan pengacara.   |                         |       |





| No. | Kode         | Kebutuhan Sistem   | Use Case                       | Aktor |
|-----|--------------|--|--------------------------------|-------|
|     | SKPP-F-29-03 | Form laporkan pengacara harus menyediakan <i>field</i> untuk alasan melaporkan pengacara.                                      |                                |       |
|     | SKPP-F-29-04 | Form laporkan pengacara harus menyediakan tombol Simpan untuk mengirimkan alasan melaporkan pengacara.                         |                                |       |
| 30  | SKPP-F-30    | Sistem harus menyediakan fungsi kirim pertanyaan untuk mengirimkan pertanyaan pada sistem untuk dijawab oleh pengacara.        |                                |       |
|     | SKPP-F-30-01 | Sistem harus menyediakan tombol tanya Pengacara pada halaman tanya pengacara.  |                                |       |
|     | SKPP-F-30-02 | Sistem harus mempunyai halaman <i>form</i> kirim pertanyaan.   | Kirim<br>Pertanyaan            | Klien |
|     | SKPP-F-30-03 | Form kirim pertanyaan harus menyediakan <i>field</i> untuk judul dan pertanyaan yang akan dikirimkan.                          |                                |       |
|     | SKPP-F-30-04 | Form kirim pertanyaan harus menyediakan tombol Kirim untuk mengirimkan pertanyaan.   |                                |       |
| 31  | SKPP-F-31    | Sistem harus menyediakan fungsi lihat jawaban pertanyaan untuk menampilkan jawaban dari pertanyaan yang dikirimkan oleh klien. | Lihat<br>Jawaban<br>Pertanyaan | Klien |
| 32  | SKPP-F-32    | Sistem harus menyediakan fungsi validasi pengacara untuk memvalidasi portofolio pengacara yang                                 | Validasi<br>Pengacara          | Admin |





| No. | Kode         | Kebutuhan Sistem   | Use Case                 | Aktor |
|-----|--------------|--|--------------------------|-------|
|     |              | mendaftar ke dalam sistem.   |                          |       |
|     | SKPP-F-32-01 | Sistem harus menyediakan tombol untuk memvalidasi pengacara pada daftar pengacara yang mengirimkan portofolio. |                          |       |
| 33  | SKPP-F-33    | Sistem harus menyediakan fungsi tolak pengacara untuk menolak pengacara yang mendaftar ke dalam sistem.        | Tolak Pengacara          | Admin |
| 34  | SKPP-F-34    | Sistem harus menyediakan fungsi lihat portofolio untuk menampilkan detail portofolio.                          | Lihat Portofolio         | Admin |
| 35  | SKPP-F-35    | Sistem harus menyediakan fungsi lihat laporan info hukum untuk menampilkan daftar info hukum yang dilaporkan.  | Lihat Laporan Info Hukum | Admin |
| 36  | SKPP-F-36    | Sistem harus menyediakan fungsi hapus lapran info hukum untuk menghapus info hukum yang dilaporkan.            | Hapus Laporan Info Hukum | Admin |
|     | SKPP-F-36-01 | Sistem harus menyediakan tombol untuk menghapus laporan info hukum pada daftar info hukum yang dilaporkan.     |                          |       |
| 37  | SKPP-F-37    | Sistem harus menyediakan fungsi lihat laporan pengacara untuk menampilkan daftar pengacara yang dilaporkan.    | Lihat Laporan Pengacara  | Admin |
| 38  | SKPP-F-38    | Sistem harus menyediakan fungsi hapus pengacara untuk menghapus pengacara yang dilaporkan.                     | Hapus Pengacara          | Admin |





| No. | Kode         | Kebutuhan Sistem  | Use Case            | Aktor               |
|-----|--------------|---|---------------------|---------------------|
|     | SKPP-F-38-01 | Sistem harus menyediakan tombol untuk menghapus pengacara pada daftar pengacara yang dilaporkan.    |                     |                     |
| 39  | SKPP-F-39    | Sistem harus menyediakan fungsi lihat laporan klien untuk menampilkan daftar klien yang dilaprkkan. | Lihat Laporan Klien | Admin               |
| 40  | SKPP-F-40    | Sistem harus menyediakan fungsi hapus klien untuk menghapus klien yang dilaporkan.                  | Hapus Klien         | Admin               |
|     | SKPP-F-40-01 | Sistem harus menyediakan tombol untuk menghapus klien pada daftar klien yang dilaporkan.            |                     |                     |
| 41  | SKPP-F-41    | Sistem harus menyediakan fungsi beri komentar untuk memberikan komentar pada info hukum.            |                     |                     |
|     | SKPP-F-41-01 | Sistem harus menyediakan <i>form</i> beri komentar.   | Beri Komentar       | Klien dan Pengacara |
|     | SKPP-F-41-02 | <i>Form</i> beri komentar harus menyediakan <i>field</i> tuliskan komentarmu.                       |                     |                     |
|     | SKPP-F-41-03 | <i>Form</i> beri komentar harus menyediakan tombol Kirim untuk mengirimkan komentar.                |                     |                     |
| 42  | SKPP-F-42    | Sistem harus menyediakan fungsi laporkan info hukum untuk melaporkan info hukum.                    | Laporkan Info Hukum | Klien dan Pengacara |
|     | SKPP-F-42-01 | Sistem harus menyediakan tombol untuk melaporkan info hukum pada info hukum yang akan dilaporkan.   |                     |                     |





| No. | Kode         | Kebutuhan Sistem   | Use Case               | Aktor                          |
|-----|--------------|--|------------------------|--------------------------------|
|     | SKPP-F-42-02 | Sistem harus mempunyai halaman <i>form</i> laporkan info hukum.  |                        |                                |
|     | SKPP-F-42-03 | <i>Form</i> laporkan info hukum harus menyediakan <i>field</i> alasan melaporkan info hukum.   |                        |                                |
|     | SKPP-F-42-04 | <i>Form</i> laporkan info hukum harus menyediakan tombol Simpan untuk menyimpan laporan info hukum.  |                        |                                |
| 43  | SKPP-F-43    | Sistem harus menyediakan fungsi lihat profil untuk melihat profil.   | Lihat Profil           | Klien dan Pengacara            |
| 44  | SKPP-F-44    | Sistem harus menyediakan fungsi edit profil untuk mengedit profil.   |                        |                                |
|     | SKPP-F-44-01 | Sistem harus menyediakan <i>form</i> edit profil.  |                        |                                |
|     | SKPP-F-44-02 | <i>Form</i> edit profil harus menyediakan <i>field</i> untuk mengedit nama lengkap, <i>email</i> , alamat, jenis kelamin, nomor telepon, dan foto. | Edit Profil            | Klien dan Pengacara            |
|     | SKPP-F-44-03 | <i>Form</i> edit profil harus menyediakan tombol <i>Update</i> untuk mengedit profil.  |                        |                                |
| 45  | SKPP-F-45    | Sistem harus menyediakan fungsi lihat daftar pengacara menampilkan pengacara yang ada pada sistem.   | Lihat Daftar Pengacara | Pengguna, Klien, dan Pengacara |
| 46  | SKPP-F-46    | Sistem harus menyediakan fungsi lihat profil pengacara untuk   | Lihat Profil Pengacara | Pengguna, Klien, dan Pengacara |





| No. | Kode         | Kebutuhan Sistem  | Use Case         | Aktor                          |
|-----|--------------|---|------------------|--------------------------------|
|     |              | menampilkan detail pengacara.   |                  |                                |
| 47  | SKPP-F-47    | Sistem harus menyediakan fungsi cari pengacara untuk mencari pengacara.                             |                  |                                |
|     | SKPP-F-47-01 | Sistem harus menyediakan <i>form</i> cari pengacara.  |                  |                                |
|     | SKPP-F-47-02 | <i>Form</i> cari pengacara harus menyediakan <i>field</i> jenis pengacara dan kabupaten.            | Cari Pengacara.  | Pengguna, Klien, dan Pengacara |
|     | SKPP-F-47-03 | <i>Form</i> cari pengacara harus menyediakan tombol Cari untuk mencari pengacara.                   |                  |                                |
| 48  | SKPP-F-48    | Sistem harus menyediakan fungsi lihat info hukum untuk menampilkan info hukum yang ada pada sistem. | Lihat Info Hukum | Pengguna, Klien, dan Pengacara |

#### 4.2.2.2 Kebutuhan Fungsional Iterasi 2

Pada Tabel 4.3 di bawah ini akan menjelaskan kebutuhan fungsional iterasi kedua dari sistem konsultasi dan pencarian pengacara.

**Tabel 4.3 Kebutuhan Fungsional Iterasi 2**

| No. | Kode         | Kebutuhan Sistem   | Use Case                                     | Aktor     |
|-----|--------------|--|--|-----------|
| 1   | SKPP-F-49    | Sistem harus menyediakan fungsi tambah hasil pertemuan konsultasi tatap muka untuk menambahkan hasil pertemuan jika konsultasi tatap muka selesai dilakukan. | Tambah Hasil Pertemuan Konsultasi Tatap Muka | Pengacara |
|     | SKPP-F-49-01 | Sistem harus menyediakan tombol Selesai yang berada pada bookingan klien yang sudah disetujui.   |  |           |
|     | SKPP-F-49-02 | Sistem harus mempunyai halaman <i>form</i> tambah hasil  |  |           |





| No. | Kode         | Kebutuhan Sistem  | Use Case       | Aktor     |
|-----|--------------|---|----------------|-----------|
|     |              | pertemuan konsultasi tatap muka.  |                |           |
|     | SKPP-F-49-03 | Form tambah hasil pertemuan harus menyediakan field untuk menuliskan hasil pertemuan tatap muka.                          |                |           |
|     | SKPP-F-49-04 | Form tambah hasil pertemuan harus menyediakan tombol Simpan untuk menyimpan berita acara.                                 |                |           |
| 2   | SKPP-F-50    | Sistem harus menyediakan fungsi tambah progres kasus untuk menambahkan progres kasus dari kasus yang ditangani pengacara. |                |           |
|     | SKPP-F-50-01 | Sistem harus menyediakan tombol Tambah Progres yang berada pada masing-masing kasus yang ditangani oleh pengacara.        | Tambah Progres | Pengacara |
|     | SKPP-F-50-02 | Sistem harus mempunyai halaman form tambah progres kasus.   | Kasus          |           |
|     | SKPP-F-50-03 | Form tambah progres kasus harus menyediakan field tanggal dan progres.  |                |           |
|     | SKPP-F-50-04 | Form tambah progres harus menyediakan tombol Simpan untuk menyimpan progres kasus.  |                |           |
| 3   | SKPP-F-51    | Sistem harus menyediakan fungsi edit progres kasus untuk mengedit progres kasus yang ditangani pengacara                  | Edit Progres   | Pengacara |
|     | SKPP-F-51-01 | Sistem harus menyediakan tombol Edit Progres yang berada pada Progres kasus   | Kasus          |           |





| No. | Kode         | Kebutuhan Sistem  | Use Case                                    | Aktor     |
|-----|--------------|---|---|-----------|
|     |              | yang sudah ditambahkan pengacara.   |   |           |
|     | SKPP-F-51-02 | Sistem harus mempunyai halaman <i>form</i> edit progres kasus.  |   |           |
|     | SKPP-F-51-03 | <i>Form</i> edit progres kasus harus menyediakan <i>field</i> tanggal dan progres.  |   |           |
|     | SKPP-F-51-04 | <i>Form</i> edit progres kasus harus menyediakan tombol Simpan untuk menyimpan edit progres kasus.  |   |           |
| 4   | SKPP-F-52    | Sistem harus menyediakan fungsi hapus progres kasus untuk menghapus progres kasus.  | Hapus Progres Kasus                         | Pengacara |
|     | SKPP-F-52-01 | Sistem harus menyediakan tombol hapus yang berada pada progres kasus yang sudah ditambahkan.  |   |           |
| 5   | SKPP-F-53    | Sistem harus menyediakan fungsi lihat progres kasus untuk menampilkan progres kasus milik klien.  | Lihat Progres Kasus                         | Klien     |
| 6   | SKPP-F-54    | Sistem harus menyediakan fungsi lihat hasil pertemuan konsultasi tatap muka untuk menampilkan hasil pertemuan setelah konsultasi tatap muka dengan pengacara selesai dilakukan. | Lihat Hasil Pertemuan Konsultasi Tatap Muka | Klien     |
|     | SKPP-F-54-01 | Sistem harus menyediakan tombol Selengkapnya pada masing-masing jadwal konsultasi tatap muka.   |   |           |





### 4.2.3 Kebutuhan Non Fungsional

Kebutuhan Non Fungsional merupakan kebutuhan yang digunakan untuk meningkatkan kepuasan pengguna dalam menggunakan Sistem Konsultasi dan Pencarian Pengacara ini. Pada table 4.4 dibawah ini merupakan daftar kebutuhan non fungsional yang ada pada sistem.

**Tabel 4.4 Daftar Kebutuhan Non Fungsional**

| No | Kode Fungsi | Nama Fungsi          | Deskripsi   |
|----|-------------|----------------------|---|
| 1  | SKPP-NF-1   | <i>Compatibility</i> | Sistem harus dapat diakses dari berbagai peramban seperti <i>Chrome</i> , <i>Firefox</i> , <i>Internet Explorer</i> , <i>Safari</i> , dan sebagainya. Sistem juga harus dapat diakses dari berbagai Operating System seperti <i>Windows</i> , <i>Android</i> , dan <i>iOS</i> . |

### 4.3 Pemodelan Kebutuhan

Pada tahap pemodelan kebutuhan terdapat *use case diagram* dan *use case scenario*. *Use case diagram* digunakan untuk menjelaskan apa saja yang dilakukan aktor pada sistem. Sedangkan untuk *use case scenario* dibuat untuk memperjelas rincian tiap *use case diagram* yang dibuat sebelumnya.

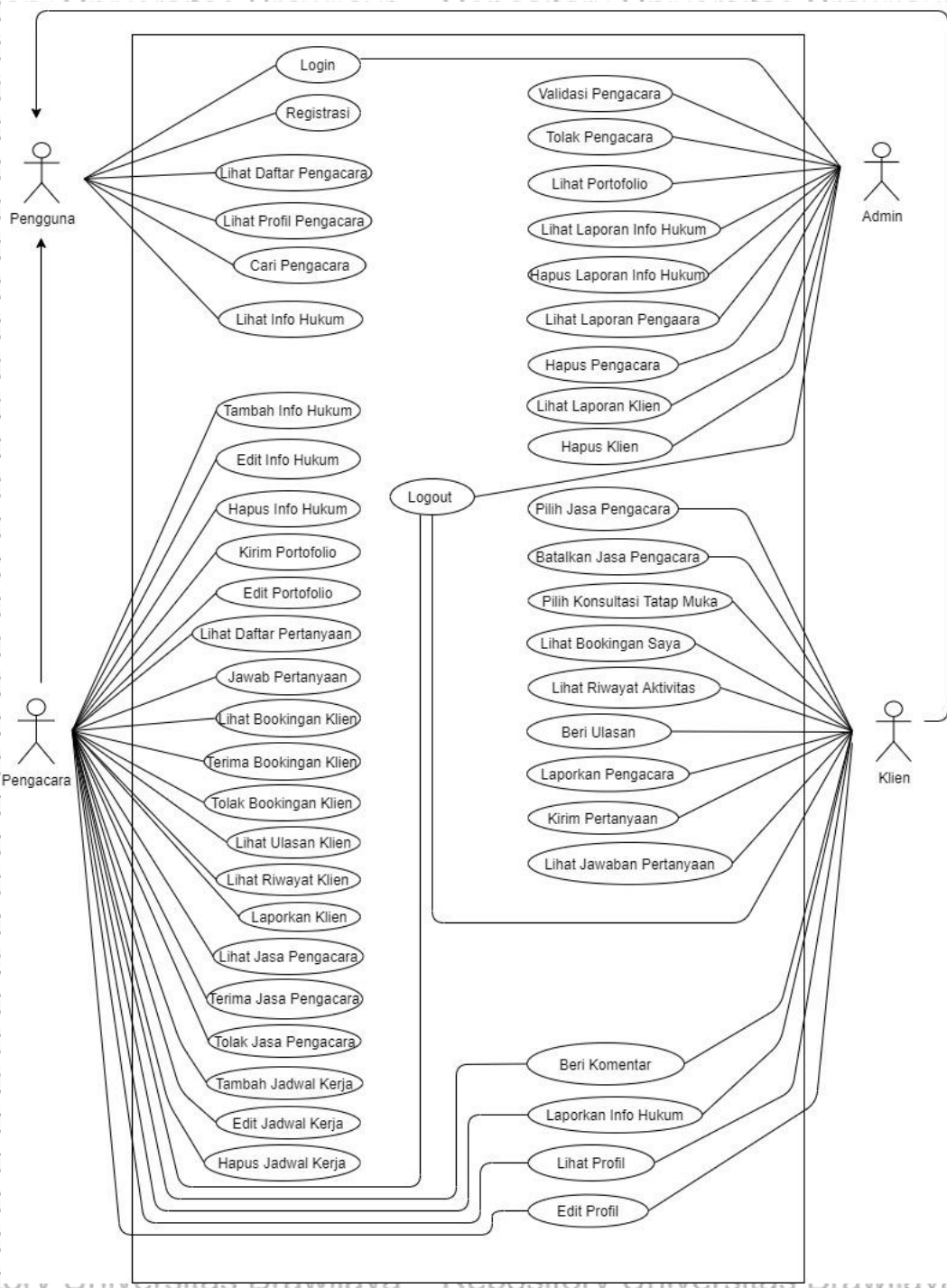
#### 4.3.1 Use Case Diagram

*Use case diagram* berfungsi untuk menunjukkan hal-hal apa saja yang bisa dilakukan oleh aktor berdasarkan dari kebutuhan fungsional yang telah didefinisikan.

##### 4.3.1.1 Use Case Diagram Iterasi 1

Dari kebutuhan fungsional yang sudah dijelaskan pada tabel 4.2 tentang kebutuhan fungsional iterasi 1, oleh karena itu dibuat *use case diagram* iterasi pertama yang memiliki 48 *use case* dan 4 aktor. Pada gambar 4.3 merupakan penjelasan dari *use case diagram* iterasi pertama dari sistem konsultasi dan pencarian pengacara.





Gambar 4.3 Use Case Diagram Iterasi 1

#### 4.3.1.2 Use Case Diagram Iterasi 2

Dari kebutuhan fungsional yang sudah dijelaskan pada tabel 4.3 tentang kebutuhan fungsional iterasi 2, oleh karena itu dibuat *use case diagram* iterasi kedua yang memiliki penambahan 6 *use case* sehingga totalnya menjadi 54 *use case* dan 4 aktor. Pada gambar 4.4 merupakan penjelasan dari *use case diagram* iterasi kedua dari sistem konsultasi dan pencarian pengacara.









### 4.3.2 Use Case Scenario

*Use case scenario* berfungsi untuk menjelaskan tentang alur kinerja yang dilakukan saat menggunakan sistem dan dapat dikategorikan menurut *use case* yang telah dibuat sebelumnya.

#### 4.3.2.1 Use Case Scenario Iterasi 1

*Use case scenario* iterasi pertama merupakan alur kinerja dari subbab 4.3.1.1 yaitu *use case diagram* iterasi pertama pada sistem pencarian dan konsultasi pengacara.

##### 1. Use Case Scenario Login

Tabel 4.5 di bawah ini merupakan *use case scenario* untuk *login*. Aktor yang dapat melakukan fungsional ini adalah Pengguna. *Use case* ini mempunyai fungsi untuk masuk ke dalam sistem.

Tabel 4.5 Use Case Scenario Login

|                         |   |
|-------------------------|---|
| <b>SKPP-F-01</b>        | <i>Login</i>  |
| <b>Objective</b>        | Aktor melakukan <i>login</i> untuk dapat masuk ke dalam sistem  |
| <b>Actor</b>            | Klien, Pengacara, dan Admin   |
| <b>Pre-Condition</b>    | Aktor berada pada halaman <i>home</i>   |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Aktor memilih menu <i>login</i> pada <i>header</i></li> <li>2. Sistem menampilkan halaman <i>login</i></li> <li>3. Aktor menginputkan <i>email</i> dan <i>password</i></li> <li>4. Aktor menekan tombol Masuk</li> <li>5. Sistem memproses <i>login</i> ke dalam sistem</li> </ol>  |
| <b>Alternative Flow</b> | <ol style="list-style-type: none"> <li>1. Jika terdapat salah satu <i>field</i> tidak terisi, maka akan menampilkan peringatan "<i>Please fill out this field</i>".</li> <li>2. Jika <i>email</i> dan <i>password</i> yang diinputkan tidak sesuai, maka aktor tidak bisa masuk ke dalam sistem dan akan menampilkan pesan peringatan "email atau password salah".</li> </ol> |
| <b>Post Condition</b>   | Aktor telah masuk ke dalam <i>home</i> sebagai aktor sesuai dengan <i>username</i> dan levelnya.  |





## 2. Use Case Scenario Registrasi

Tabel 4.6 di bawah ini merupakan *use case scenario* untuk registrasi. Aktor yang dapat melakukan fungsional ini adalah Pengguna. *Use case* ini mempunyai fungsi untuk melakukan registrasi jika belum memiliki akun.

**Tabel 4.6 Use Case Scenario Registrasi**

|                         |  |
|-------------------------|--|
| <b>SKPP-F-02</b>        | Registrasi   |
| <b>Objective</b>        | Aktor melakukan registrasi jika belum memiliki akun untuk dapat masuk pada sistem  |
| <b>Actor</b>            | Pengguna   |
| <b>Pre-Condition</b>    | Aktor berada pada halaman <i>home</i>  |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Aktor memilih menu registrasi pada menu bar</li> <li>2. Sistem menampilkan halaman registrasi</li> <li>3. Aktor menginputkan nama, <i>username</i>, <i>email</i>, dan <i>password</i></li> <li>4. Aktor menekan tombol Daftar</li> <li>5. Sistem menyimpan data registrasi ke dalam database.</li> </ol>   |
| <b>Alternative Flow</b> | <ol style="list-style-type: none"> <li>1. Jika terdapat salah satu <i>field</i> tidak terisi, maka akan menampilkan peringatan "<i>Please fill out this field</i>".</li> <li>2. Jika <i>username</i> sudah ada yang menggunakan, maka akan menampilkan peringatan "<i>The username field must contain a unique value</i>".</li> <li>3. Jika <i>email</i> sudah ada yang menggunakan, maka akan menampilkan peringatan "<i>The email field must contain a unique value</i>".</li> </ol> |
| <b>Post Condition</b>   | Sistem menampilkan halaman <i>login</i> .  |

## 3. Use Case Scenario Logout

Tabel 4.7 di bawah ini merupakan *use case scenario* untuk *Logout*. Aktor yang dapat melakukan fungsional ini adalah Pengguna. *Use case* ini mempunyai fungsi untuk keluar dari dalam sistem.

**Tabel 4.7 Use Case Scenario Logout**

|                  |                                |
|------------------|--------------------------------|
| <b>SKPP-F-03</b> | Keluar                         |
| <b>Objective</b> | Aktor dapat keluar dari sistem |
| <b>Actor</b>     | Pengguna                       |





|                         |   |
|-------------------------|---|
| <b>Pre-Condition</b>    | Aktor telah melakukan <i>login</i>                                |
| <b>Main Flow</b>        | Aktor menekan tombol keluar                                       |
| <b>Alternative Flow</b> | -   |
| <b>Post Condition</b>   | Berhasil keluar dari sistem dan menampilkan halaman <i>home</i> . |

#### 4. Use Case Scenario Tambah Info Hukum

Tabel 4.8 di bawah ini merupakan *use case scenario* untuk menambahkan info hukum. Aktor yang dapat melakukan fungsional ini adalah Pengacara. *Use case* ini mempunyai fungsi untuk menambahkan postingan info hukum.

**Tabel 4.8 Use Case Scenario Tambah Info Hukum**

|                         |   |
|-------------------------|---|
| <b>SKPP-F-04</b>        | Tambah Info Hukum   |
| <b>Objective</b>        | Aktor dapat menambahkan info hukum.   |
| <b>Actor</b>            | Pengacara.  |
| <b>Pre-Condition</b>    | Aktor berada pada halaman info hukum  |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Aktor menekan tombol Postingan Saya</li> <li>2. Sistem menampilkan daftar info hukum yang aktor punya.</li> <li>3. Aktor menekan tombol tambahkan info hukum.</li> <li>4. Sistem menampilkan <i>form</i> untuk menambahkan info hukum.</li> <li>5. Aktor mengisikan judul, isi, dan foto dari info hukum.</li> <li>6. Aktor menekan tombol simpan.</li> </ol> |
| <b>Alternative Flow</b> | Jika terdapat salah satu <i>field</i> tidak terisi, maka akan menampilkan peringatan " <i>Please fill out this field</i> ".   |
| <b>Post Condition</b>   | Sistem menyimpan data info hukum ke dalam <i>database</i> dan menampilkan halaman postingan saya.   |

#### 5. Use Case Scenario Edit Info Hukum

Tabel 4.9 di bawah ini merupakan *use case scenario* untuk mengedit info hukum. Aktor yang dapat melakukan fungsional ini adalah Pengacara. *Use case* ini mempunyai fungsi untuk mengedit postingan info hukum.





Tabel 4.9 Use Case Scenario Edit Info Hukum

|                         |  |
|-------------------------|--|
| <b>SKPP-F-05</b>        | Edit Info Hukum  |
| <b>Objective</b>        | Aktor dapat mengedit info hukum  |
| <b>Actor</b>            | Pengacara  |
| <b>Pre-Condition</b>    | Aktor berada pada halaman postingan saya   |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Aktor memilih tombol edit yang ada pada setiap info hukum yang aktor tulis.</li> <li>2. Sistem menampilkan <i>form</i> untuk mengedit info hukum.</li> <li>3. Aktor memasukkan data dari info hukum yang ingin diedit.</li> <li>4. Aktor menekan tombol Simpan.</li> </ol> |
| <b>Alternative Flow</b> | Jika terdapat salah satu <i>field</i> tidak terisi, maka akan menampilkan peringatan " <i>Please fill out this field</i> ".  |
| <b>Post Condition</b>   | Sistem menyimpan perubahan info hukum ke dalam <i>database</i> .   |

## 6. Use Case Scenario Hapus Info Hukum

Tabel 4.10 di bawah ini merupakan *use case scenario* untuk menghapus info hukum. Aktor yang dapat melakukan fungsional ini adalah Pengacara. *Use case* ini mempunyai fungsi untuk menghapus postingan info hukum.

Tabel 4.10 Use Case Scenario Hapus Info Hukum

|                         |   |
|-------------------------|---|
| <b>SKPP-F-06</b>        | Hapus Info Hukum  |
| <b>Objective</b>        | Aktor dapat menghapus info hukum yang telah dipilih.  |
| <b>Actor</b>            | Pengacara   |
| <b>Pre-Condition</b>    | Aktor berada pada halaman info hukum  |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Aktor memilih tombol Postingan Saya</li> <li>2. Sistem menampilkan daftar postingan milik aktor</li> <li>3. Aktor menekan tombol hapus pada salah satu postingan</li> </ol> |
| <b>Alternative Flow</b> | -   |
| <b>Post Condition</b>   | Sistem menyimpan perubahan ke dalam <i>database</i> .   |





### 7. Use Case Scenario Kirim Portofolio

Tabel 4.11 di bawah ini merupakan *use case scenario* untuk mengirimkan portofolio. Aktor yang dapat melakukan fungsional ini adalah Pengacara. *Use case* ini mempunyai fungsi untuk mengirimkan portofolio saat melakukan registrasi pengacara.

**Tabel 4.11 Use Case Scenario Kirim Portofolio**

|                         |   |
|-------------------------|---|
| <b>SKPP-F-07</b>        | Kirim Portofolio  |
| <b>Objective</b>        | Aktor dapat mengirimkan portofolio untuk melengkapi pendaftaran pengacara.  |
| <b>Aktor</b>            | Pengacara   |
| <b>Pre-Condition</b>    | Aktor berada pada halaman profil  |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Aktor memilih tombol Ajukan Portofolio pada halaman profil pengacara.</li> <li>2. Sistem menampilkan <i>form</i> untuk mengisi portofolio dari aktor.</li> <li>3. Aktor mengisi semua <i>field</i> yang ada pada <i>form</i>.</li> <li>4. Aktor menekan tombol Simpan.</li> </ol> |
| <b>Alternative Flow</b> | Jika terdapat salah satu <i>field</i> tidak terisi, maka akan menampilkan peringatan " <i>Please fill out this field</i> ".   |
| <b>Post Condition</b>   | Sistem menyimpan portofolio dari aktor ke dalam <i>database</i> .   |

### 8. Use Case Scenario Edit Portofolio

Tabel 4.12 di bawah ini merupakan *use case scenario* untuk mengedit portofolio. Aktor yang dapat melakukan fungsional ini adalah Pengacara. *Use case* ini mempunyai fungsi untuk mengedit portofolio jika ada yang ingin dirubah.

**Tabel 4.12 Use Case Scenario Edit Portofolio**

|                      |  |
|----------------------|--|
| <b>SKPP-F-08</b>     | Edit Portofolio  |
| <b>Objective</b>     | Aktor dapat mengedit portofolio.   |
| <b>Aktor</b>         | Pengacara  |
| <b>Pre-Condition</b> | Aktor berada pada halaman profil   |
| <b>Main Flow</b>     | <ol style="list-style-type: none"> <li>1. Aktor memilih tombol Edit Portofolio pada halaman profil pengacara.</li> </ol> |





|                         |  |
|-------------------------|--|
|                         | <p>2. Sistem menampilkan <i>form</i> untuk mengedit portofolio dari aktor.</p> <p>3. Aktor mengisikan semua <i>field</i> yang ada pada <i>form</i>.</p> <p>4. Aktor menekan tombol Simpan.</p> |
| <b>Alternative Flow</b> | Jika terdapat salah satu <i>field</i> tidak terisi, maka akan menampilkan peringatan " <i>Please fill out this field</i> ".  |
| <b>Post Condition</b>   | Sistem menyimpan portofolio dari aktor ke dalam <i>database</i> .  |

#### 9. Use Case Scenario Lihat Daftar Pertanyaan

Tabel 4.13 di bawah ini merupakan *use case scenario* untuk melihat daftar pertanyaan. Aktor yang dapat melakukan fungsional ini adalah Pengacara. *Use case* ini mempunyai fungsi untuk melihat daftar pertanyaan yang dikirimkan oleh klien.

**Tabel 4.13 Use Case Scenario Lihat Daftar Pertanyaan**

|                         |  |
|-------------------------|--|
| <b>SKPP-F-09</b>        | Lihat Daftar Pertanyaan  |
| <b>Objective</b>        | Aktor dapat melihat daftar pertanyaan yang diajukan oleh Klien |
| <b>Actor</b>            | Pengacara  |
| <b>Pre-Condition</b>    | Aktor berada pada halaman <i>home</i>                          |
| <b>Main Flow</b>        | Aktor memilih menu Tanya Pengacara pada menu bar.              |
| <b>Alternative Flow</b> | -  |
| <b>Post Condition</b>   | Sistem menampilkan daftar pertanyaan yang diajukan oleh klien. |

#### 10. Use Case Scenario Jawab Pertanyaan

Tabel 4.14 di bawah ini merupakan *use case scenario* untuk menjawab pertanyaan. Aktor yang dapat melakukan fungsional ini adalah Pengacara. *Use case* ini mempunyai fungsi untuk menjawab pertanyaan yang dikirimkan Klien.





Tabel 4.14 Use Case Scenario Jawab Pertanyaan

|                         |  |
|-------------------------|--|
| <b>SKPP-F-10</b>        | Jawab Pertanyaan   |
| <b>Objective</b>        | Aktor dapat menjawab pertanyaan yang diajukan oleh klien   |
| <b>Actor</b>            | Pengacara  |
| <b>Pre-Condition</b>    | Aktor berada pada halaman Tanya Pengacara  |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Aktor memilih pertanyaan yang ingin dijawab</li> <li>2. Sistem menampilkan pertanyaan yang dipilih aktor</li> <li>3. Aktor menekan tombol Jawab Pertanyaan</li> <li>4. Sistem menampilkan <i>form</i> untuk menginputkan jawaban</li> <li>5. Aktor menuliskan jawaban pada <i>form</i> dan menekan tombol Kirim</li> </ol> |
| <b>Alternative Flow</b> | Jika terdapat salah satu <i>field</i> tidak terisi, maka akan menampilkan peringatan " <i>Please fill out this field</i> ".  |
| <b>Post Condition</b>   | Sistem menampilkan jawaban dan menyimpannya ke dalam <i>database</i> .   |

## 11. Use Case Scenario Lihat Bookingan Klien

Tabel 4.15 di bawah ini merupakan *use case scenario* untuk melihat bookingan klien. Aktor yang dapat melakukan fungsional ini adalah Pengacara. *Use case* ini mempunyai fungsi untuk melihat daftar bookingan untuk jadwal bertemu antara klien dan pengacara.

Tabel 4.15 Use Case Scenario Lihat Bookingan Klien

|                         |   |
|-------------------------|---|
| <b>SKPP-F-11</b>        | Lihat Bookingan Klien   |
| <b>Objective</b>        | Aktor dapat melihat daftar bookingan dari klien                 |
| <b>Actor</b>            | Pengacara   |
| <b>Pre-Condition</b>    | Aktor berada pada dashboard                                     |
| <b>Main Flow</b>        | Aktor memilih menu Bookingan Pertemuan pada menu <i>sidebar</i> |
| <b>Alternative Flow</b> | -   |
| <b>Post Condition</b>   | Sistem menampilkan daftar bookingan klien.                      |





## 12. Use Case Scenario Terima Bookingan Klien

Tabel 4.16 di bawah ini merupakan *use case scenario* untuk menerima bookingan klien. Aktor yang dapat melakukan fungsional ini adalah Pengacara. *Use case* ini mempunyai fungsi untuk menerima bookingan yang diajukan oleh klien.

**Tabel 4.16 Use Case Scenario Terima Bookingan Klien**

|                         |   |
|-------------------------|---|
| <b>SKPP-F-12</b>        | Terima Bookingan Klien  |
| <b>Objective</b>        | Aktor dapat menerima bookingan dari klien.  |
| <b>Actor</b>            | Pengacara   |
| <b>Pre-Condition</b>    | Aktor berada pada halaman Bookingan Pertemuan   |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Aktor memilih tab Belum Disetujui.</li> <li>2. Sistem menampilkan bookingan klien yang belum disetujui.</li> <li>3. Aktor memilih tombol Terima.</li> </ol> |
| <b>Alternative Flow</b> | -   |
| <b>Post Condition</b>   | Sistem menyimpan bookingan yang telah diterima pada <i>database</i> .   |

## 13. Use Case Scenario Tolak Bookingan Klien

Tabel 4.17 di bawah ini merupakan *use case scenario* untuk menolak bookingan klien. Aktor yang dapat melakukan fungsional ini adalah Pengacara. *Use case* ini mempunyai fungsi untuk menolak bookingan yang diajukan oleh klien.

**Tabel 4.17 Use Case Scenario Tolak Bookingan Klien**

|                      |  |
|----------------------|--|
| <b>SKPP-F-13</b>     | Tolak Bookingan Klien  |
| <b>Objective</b>     | Aktor dapat menerima bookingan dari klien.   |
| <b>Actor</b>         | Pengacara  |
| <b>Pre-Condition</b> | Aktor berada pada halaman Bookingan Pertemuan  |
| <b>Main Flow</b>     | <ol style="list-style-type: none"> <li>1. Aktor memilih tab Belum Disetujui.</li> <li>2. Sistem menampilkan bookingan klien yang belum disetujui.</li> <li>3. Aktor memilih tombol Tolak.</li> <li>4. Sistem menampilkan <i>form</i> untuk memasukkan alasan menolak bookingan klien.</li> </ol> |





|                         |   |
|-------------------------|---|
|                         | 5. Aktor menuliskan alasan menolak bookingan klien dan menekan tombol Simpan.   |
| <b>Alternative Flow</b> | Jika terdapat salah satu <i>field</i> tidak terisi, maka akan menampilkan peringatan “ <i>Please fill out this field</i> ”. |
| <b>Post Condition</b>   | Sistem menyimpan bookingan yang telah ditolak pada <i>database</i> .  |

#### 14. Use Case Scenario Lihat Ulasan Klien

Tabel 4.18 di bawah ini merupakan *use case scenario* untuk lihat ulasan klien. Aktor yang dapat melakukan fungsional ini adalah Pengacara. *Use case* ini mempunyai fungsi untuk melihat ulasan yang diberikan oleh klien.

**Tabel 4.18 Use Case Scenario Lihat Ulasan Klien**

|                         |   |
|-------------------------|---|
| <b>SKPP-F-14</b>        | Lihat Ulasan Klien                                    |
| <b>Objective</b>        | Aktor dapat melihat ulasan yang diberikan oleh Klien. |
| <b>Actor</b>            | Pengacara   |
| <b>Pre-Condition</b>    | Aktor berada pada halaman <i>dashboard</i>            |
| <b>Main Flow</b>        | Aktor memilih menu Ulasan pada menu <i>sidebar</i> .  |
| <b>Alternative Flow</b> | -   |
| <b>Post Condition</b>   | Sistem menampilkan ulasan yang diberikan oleh Klien   |

#### 15. Use Case Scenario Lihat Riwayat Klien

Tabel 4.19 di bawah ini merupakan *use case scenario* untuk Melihat Riwayat Klien. Aktor yang dapat melakukan fungsional ini adalah Pengacara. *Use case* ini mempunyai fungsi untuk melihat riwayat klien yang ditangani oleh aktor.

**Tabel 4.19 Use Case Scenario Lihat Riwayat Klien**

|                         |  |
|-------------------------|--|
| <b>SKPP-F-15</b>        | Lihat Riwayat Klien  |
| <b>Objective</b>        | Aktor dapat melihat riwayat klien yang ditangani oleh aktor. |
| <b>Actor</b>            | Pengacara  |
| <b>Pre-Condition</b>    | Aktor berada pada halaman <i>dashboard</i>                   |
| <b>Main Flow</b>        | Aktor memilih menu Riwayat Klien pada menu <i>sidebar</i>    |
| <b>Alternative Flow</b> | -  |





|                       |                                   |
|-----------------------|-----------------------------------|
| <b>Post Condition</b> | Sistem menampilkan riwayat klien. |
|-----------------------|-----------------------------------|

#### 16. Use Case Scenario Laporkan Klien

Tabel 4.20 di bawah ini merupakan *use case scenario* untuk Melaporkan Klien. Aktor yang dapat melakukan fungsional ini adalah Pengacara. *Use case* ini mempunyai fungsi untuk melaporkan klien.

**Tabel 4.20 Use Case Scenario Laporkan Klien**

|                         |  |
|-------------------------|--|
| <b>SKPP-F-16</b>        | Laporkan Klien   |
| <b>Objective</b>        | Aktor dapat melaporkan klien   |
| <b>Actor</b>            | Pengacara  |
| <b>Pre-Condition</b>    | Aktor berada pada detail kasus yang sudah selesai  |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Aktor memilih <i>link</i> laporkan klien pada halaman detail kasus klien.</li> <li>2. Sistem menampilkan <i>form</i> untuk melaporkan klien.</li> <li>3. Aktor mengisi <i>form</i> tersebut lalu menekan tombol simpan.</li> </ol> |
| <b>Alternative Flow</b> | <ol style="list-style-type: none"> <li>1. Jika terdapat salah satu <i>field</i> tidak terisi, maka akan menampilkan peringatan "<i>Please fill out this field</i>".</li> <li>2. Jika aktor memilih tombol Batal maka klien tidak berhasil dilaporkan</li> </ol>              |
| <b>Post Condition</b>   | Sistem menyimpan laporan klien ke dalam <i>database</i> .  |

#### 17. Use Case Scenario Lihat Jasa Pengacara

Tabel 4.21 di bawah ini merupakan *use case scenario* untuk Melihat jasa pengacara. Aktor yang dapat melakukan fungsional ini adalah Pengacara. *Use case* ini mempunyai fungsi untuk melihat jasa pengacara yang ditangani oleh aktor.

**Tabel 4.21 Use Case Scenario Lihat Jasa Pengacara**

|                      |   |
|----------------------|---|
| <b>SKPP-F-17</b>     | Lihat Jasa Pengacara  |
| <b>Objective</b>     | Aktor dapat melihat jasa pengacara yang ditangani oleh aktor. |
| <b>Actor</b>         | Pengacara   |
| <b>Pre-Condition</b> | Aktor berada pada halaman <i>dashboard</i>                    |





|                         |  |
|-------------------------|--|
| <b>Main Flow</b>        | Aktor memilih menu Manajemen Kasus pada menu sidebar |
| <b>Alternative Flow</b> | -  |
| <b>Post Condition</b>   | Sistem menampilkan riwayat klien.                    |

#### 18. Use Case Scenario Terima Jasa Pengacara

Tabel 4.22 di bawah ini merupakan *use case scenario* untuk menerima jasa pengacara. Aktor yang dapat melakukan fungsional ini adalah Pengacara. *Use case* ini mempunyai fungsi untuk menerima jasa pengacara yang dipesan oleh klien.

**Tabel 4.22 Use Case Scenario Terima Jasa Pengacara**

|                         |  |
|-------------------------|--|
| <b>SKPP-F-18</b>        | Terima Jasa Pengacara  |
| <b>Objective</b>        | Aktor dapat menerima jasa pengacara yang dipesan oleh klien.   |
| <b>Actor</b>            | Pengacara  |
| <b>Pre-Condition</b>    | Aktor berada pada halaman <i>dashboard</i>   |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Aktor memilih menu Manajemen Kasus pada menu sidebar.</li> <li>2. Sistem menampilkan daftar kasus yang dipesan oleh klien.</li> <li>3. Aktor memilih salah satu kasus lalu menekan tombol Terima.</li> </ol> |
| <b>Alternative Flow</b> | -  |
| <b>Post Condition</b>   | Sistem menampilkan halaman Manajemen Kasus dan menyimpan perubahan status kasus pada <i>database</i> .   |

#### 19. Use Case Scenario Tolak Jasa Pengacara

Tabel 4.23 di bawah ini merupakan *use case scenario* untuk menolak jasa pengacara. Aktor yang dapat melakukan fungsional ini adalah Pengacara. *Use case* ini mempunyai fungsi untuk menolak jasa pengacara yang dipesan oleh klien.





Tabel 4.23 Use Case Scenario Tolak Jasa Pengacara

|                         |  |
|-------------------------|--|
| <b>SKPP-F-19</b>        | Tolak Jasa Pengacara   |
| <b>Objective</b>        | Aktor dapat menolak jasa pengacara yang dipesan oleh klien.  |
| <b>Actor</b>            | Pengacara  |
| <b>Pre-Condition</b>    | Aktor berada pada halaman <i>dashboard</i>   |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Aktor memilih menu Manajemen Kasus pada menu <i>sidebar</i>.</li> <li>2. Sistem menampilkan daftar kasus yang dipesan oleh klien.</li> <li>3. Aktor memilih salah satu kasus lalu menekan tombol Tolak.</li> </ol> |
| <b>Alternative Flow</b> | -  |
| <b>Post Condition</b>   | Sistem menampilkan halaman Manajemen Kasus dan menyimpan perubahan status kasus pada <i>database</i> .   |

## 20. Use Case Scenario Tambah Jadwal Kerja

Tabel 4.24 di bawah ini merupakan *use case scenario* untuk Menambahkan Jadwal Kerja. Aktor yang dapat melakukan fungsional ini adalah Pengacara. *Use case* ini mempunyai fungsi untuk menambahkan jadwal kerja pengacara.

Tabel 4.24 Use Case Scenario Tambah Jadwal Kerja

|                         |   |
|-------------------------|---|
| <b>SKPP-F-20</b>        | Tambah Jadwal Kerja   |
| <b>Objective</b>        | Aktor dapat menambahkan jadwal kerja  |
| <b>Actor</b>            | Pengacara   |
| <b>Pre-Condition</b>    | Aktor berada pada <i>dashboard</i>  |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Aktor memilih menu Jadwal Kerja pada menu <i>sidebar</i>.</li> <li>2. Sistem menampilkan <i>form</i> untuk menginputkan jadwal kerja dan tabel jadwal kerja pengacara.</li> <li>3. Aktor menginputkan tanggal kerja dan jam kerja pada <i>field</i> yang tersedia pada <i>form</i>.</li> <li>4. Aktor menekan tombol Simpan.</li> </ol> |
| <b>Alternative Flow</b> | Jika terdapat salah satu <i>field</i> tidak terisi, maka akan menampilkan peringatan " <i>Please fill out this field</i> ".   |





|                       |  |
|-----------------------|--|
| <b>Post Condition</b> | Sistem menyimpan jadwal kerja ke dalam <i>database</i> dan menampilkannya pada tabel jadwal kerja <i>available</i> . |
|-----------------------|--|

#### 21. Use Case Scenario Edit Jadwal Kerja

Tabel 4.25 di bawah ini merupakan *use case scenario* untuk Mengedit Jadwal Kerja. Aktor yang dapat melakukan fungsional ini adalah Pengacara. *Use case* ini mempunyai fungsi untuk mengedit jadwal kerja pengacara.

**Tabel 4.25 Use Case Scenario Edit Jadwal Kerja**

|                         |   |
|-------------------------|---|
| <b>SKPP-F-21</b>        | Edit Jadwal Kerja   |
| <b>Objective</b>        | Aktor dapat mengedit Jadwal Kerja   |
| <b>Actor</b>            | Pengacara   |
| <b>Pre-Condition</b>    | Aktor berada pada halaman Jadwal Kerja  |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Aktor memilih link Edit tabel Jadwal Kerja <i>Available</i>.</li> <li>2. Sistem akan menampilkan modal berbentuk <i>form</i> untuk mengedit jadwal kerja.</li> <li>3. Aktor memasukkan jadwal kerja dan jam kerja yang akan diganti pada <i>form</i>.</li> <li>4. Aktor menekan tombol Simpan.</li> </ol> |
| <b>Alternative Flow</b> | <ol style="list-style-type: none"> <li>1. Jika terdapat salah satu <i>field</i> tidak terisi, maka akan menampilkan peringatan "<i>Please fill out this field</i>".</li> <li>2. Jika aktor memilih tombol Batal maka jadwal kerja tidak jadi diubah.</li> </ol>   |
| <b>Post Condition</b>   | Sistem menyimpan jadwal kerja ke dalam <i>database</i> dan menampilkannya pada tabel jadwal kerja <i>available</i> .  |

#### 22. Use Case Scenario Hapus Jadwal Kerja

Tabel 4.26 di bawah ini merupakan *use case scenario* untuk Menghapus Jadwal Kerja. Aktor yang dapat melakukan fungsional ini adalah Pengacara. *Use case* ini mempunyai fungsi untuk menghapus jadwal kerja pengacara.

**Tabel 4.26 Use Case Scenario Hapus Jadwal Kerja**

|                  |                                    |
|------------------|------------------------------------|
| <b>SKPP-F-22</b> | Hapus Jadwal Kerja                 |
| <b>Objective</b> | Aktor dapat menghapus jadwal kerja |





|                         |  |
|-------------------------|--|
| <b>Actor</b>            | Pengacara  |
| <b>Pre-Condition</b>    | Aktor berada pada halaman Jadwal Kerja   |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Aktor memilih link Hapus tabel Jadwal Kerja Available.</li> <li>2. Sistem menampilkan <i>alert</i> apakah yakin untuk menghapus jadwal kerja.</li> <li>3. Aktor menekan tombol Ya</li> </ol> |
| <b>Alternative Flow</b> |  |
| <b>Post Condition</b>   | Sistem menyimpan perubahan ke dalam <i>database</i> dan menghapus jadwal kerja pada tabel jadwal kerja <i>available</i> .  |

### 23. Use Case Scenario Pilih Jasa Pengacara

Tabel 4.27 di bawah ini merupakan *use case scenario* untuk memilih jasa pengacara. Aktor yang dapat melakukan fungsional ini adalah Klien. *Use case* ini mempunyai fungsi untuk memilih pengacara untuk menggunakan jasanya.

**Tabel 4.27 Use Case Scenario Pilih Jasa Pengacara**

|                         |   |
|-------------------------|---|
| <b>SKPP-F-23</b>        | Pilih Jasa Pengacara  |
| <b>Objective</b>        | Aktor dapat memilih jasa pengacara yang diinginkan  |
| <b>Actor</b>            | Klien   |
| <b>Pre-Condition</b>    | Aktor berada pada halaman profil pengacara  |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Aktor menekan tombol Gunakan Jasa Pengacara</li> <li>2. Sistem akan menampilkan <i>form</i> untuk membuat janji bertemu dengan pengacara</li> <li>3. Aktor memasukkan masalah, nomer telepon, dan email.</li> <li>4. Aktor menekan tombol kirim.</li> </ol> |
| <b>Alternative Flow</b> | Jika terdapat salah satu <i>field</i> tidak terisi, maka akan menampilkan peringatan " <i>Please fill out this field</i> ".   |
| <b>Post Condition</b>   | Sistem menyimpan permintaan gunakan jasa pengacara dan jadwal bertemu ke dalam <i>database</i> .  |

### 24. Use Case Scenario Batalkan Jasa Pengacara

Tabel 4.28 di bawah ini merupakan *use case scenario* untuk membatalkan jasa pengacara. Aktor yang dapat melakukan fungsional ini adalah Klien. *Use*





case ini mempunyai fungsi untuk membatalkan menggunakan jasa pengacara.

**Tabel 4.28 Use Case Scenario Batalkan Jasa Pengacara**

|                         |  |
|-------------------------|--|
| <b>SKPP-F-24</b>        | Batalkan Jasa Pengacara  |
| <b>Objective</b>        | Aktor dapat membatalkan penggunaan jasa pengacara.   |
| <b>Actor</b>            | Klien  |
| <b>Pre-Condition</b>    | Aktor berada pada halaman Progres Kasus.   |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Aktor memilih kasus yang akan dibatalkan.</li> <li>2. Sistem menampilkan detail kasus.</li> <li>3. Aktor memilih tombol Batalkan Jasa Pengacara.</li> <li>4. Aktor memasukkan alasan membatalkan pengacara dan menekan tombol Simpan.</li> </ol> |
| <b>Alternative Flow</b> | Jika terdapat salah satu <i>field</i> tidak terisi, maka akan menampilkan peringatan " <i>Please fill out this field</i> ".  |
| <b>Post Condition</b>   | Sistem menyimpan pembatalan jasa pengacara ke dalam <i>database</i> .  |

#### 25. Use Case Scenario Pilih Konsultasi Tatap Muka

Tabel 4.29 di bawah ini merupakan *use case scenario* untuk memilih konsultasi tatap muka. Aktor yang dapat melakukan fungsional ini adalah Klien. *Use case* ini mempunyai fungsi untuk memilih jadwal untuk konsultasi tatap muka dengan pengacara.

**Tabel 4.29 Use Case Scenario Pilih Konsultasi Tatap Muka**

|                      |   |
|----------------------|---|
| <b>SKPP-F-25</b>     | Pilih Konsultasi Tatap Muka   |
| <b>Objective</b>     | Aktor dapat memilih konsultasi tatap muka   |
| <b>Actor</b>         | Klien   |
| <b>Pre-Condition</b> | Aktor berada pada halaman profil pengacara  |
| <b>Main Flow</b>     | <ol style="list-style-type: none"> <li>1. Aktor memilih tombol Konsultasi Tatap Muka.</li> <li>2. Sistem menampilkan <i>form</i> untuk membuat janji bertemu dengan pengacara.</li> <li>3. Aktor memilih bidang keahlian, tanggal konsultasi, waktu konsultasi, nomor <i>handphone</i>, <i>email</i>, dan inti permasalahan.</li> <li>4. Aktor menekan tombol Kirim.</li> </ol> |





|                         |   |
|-------------------------|---|
| <b>Alternative Flow</b> | Jika terdapat salah satu <i>field</i> tidak terisi, maka akan menampilkan peringatan “ <i>Please fill out this field</i> ”. |
| <b>Post Condition</b>   | Sistem menyimpan permintaan konsultasi tatap muka pada <i>database</i> .  |

#### 26. Use Case Scenario Lihat Bookingan Saya

Tabel 4.30 di bawah ini merupakan *use case scenario* untuk melihat bookingan saya. Aktor yang dapat melakukan fungsional ini adalah Klien. *Use case* ini mempunyai fungsi untuk melihat jadwal booking untuk melakukan konsultasi ke pengacara.

**Tabel 4.30 Use Case Scenario Lihat Bookingan Saya**

|                         |  |
|-------------------------|--|
| <b>SKPP-F-26</b>        | Lihat Bookingan Saya   |
| <b>Objective</b>        | Aktor dapat melihat jadwal booking pengacara                 |
| <b>Actor</b>            | Klien  |
| <b>Pre-Condition</b>    | Aktor berada pada halaman <i>dashboard</i>                   |
| <b>Main Flow</b>        | Aktor memilih menu Bookingan Saya dari menu <i>sidebar</i> . |
| <b>Alternative Flow</b> | -  |
| <b>Post Condition</b>   | Sistem menampilkan daftar bookingan aktor.                   |

#### 27. Use Case Scenario Lihat Riwayat Aktivitas

Tabel 4.31 di bawah ini merupakan *use case scenario* untuk melihat riwayat aktivitas. Aktor yang dapat melakukan fungsional ini adalah Klien. *Use case* ini mempunyai fungsi untuk melihat *history* dari aktivitas yang dilakukan oleh aktor.

**Tabel 4.31 Use Case Scenario Lihat Riwayat Aktivitas**

|                         |  |
|-------------------------|--|
| <b>SKPP-F-27</b>        | Lihat Riwayat Aktivitas                                    |
| <b>Objective</b>        | Aktor dapat melihat aktivitas apa saja yang dilakukan.     |
| <b>Actor</b>            | Klien  |
| <b>Pre-Condition</b>    | Aktor berada pada halaman <i>dashboard</i>                 |
| <b>Main Flow</b>        | Aktor memilih menu Riwayat Aktivitas pada <i>sidebar</i> . |
| <b>Alternative Flow</b> | -  |
| <b>Post Condition</b>   | Sistem menampilkan riwayat aktivitas dari aktor.           |





## 28. Use Case Scenario Beri Ulasan

Tabel 4.32 di bawah ini merupakan *use case scenario* untuk memberikan ulasan. Aktor yang dapat melakukan fungsional ini adalah Klien. *Use case* ini mempunyai fungsi untuk memberikan ulasan kepada pengacara yang telah digunakan jasanya.

**Tabel 4.32 Use Case Scenario Beri Ulasan**

|                         |  |
|-------------------------|--|
| <b>SKPP-F-28</b>        | Beri Ulasan  |
| <b>Objective</b>        | Aktor dapat memberikan ulasan  |
| <b>Actor</b>            | Klien  |
| <b>Pre-Condition</b>    | Aktor berada pada halaman Bookingan Saya atau Progres Kasus  |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Aktor memilih kasus yang sudah selesai ditangani pengacara.</li> <li>2. Sistem menampilkan detail kasus.</li> <li>3. Aktor memilih tombol Ulasan.</li> <li>4. Sistem menampilkan <i>form</i> untuk mengisi ulasan.</li> <li>5. Aktor memilih rating dan menuliskan ulasan pada <i>form</i>.</li> <li>6. Aktor memilih tombol Kirim.</li> </ol> |
| <b>Alternative Flow</b> | <ol style="list-style-type: none"> <li>1. Jika terdapat salah satu <i>field</i> tidak terisi, maka akan menampilkan peringatan "<i>Please fill out this field</i>".</li> <li>2. Jika aktor sudah memberikan ulasan, maka akan menampilkan peringatan "Anda Sudah Mengirimkan Ulasan".</li> </ol>   |
| <b>Post Condition</b>   | Sistem menyimpan ulasan pada <i>database</i> .   |

## 29. Use Case Scenario Laporkan Pengacara

Tabel 4.33 di bawah ini merupakan *use case scenario* untuk melaporkan pengacara. Aktor yang dapat melakukan fungsional ini adalah Klien. *Use case* ini mempunyai fungsi untuk melaporkan pengacara.

**Tabel 4.33 Use Case Scenario Laporkan Pengacara**

|                  |                                  |
|------------------|----------------------------------|
| <b>SKPP-F-29</b> | Laporkan Pengacara               |
| <b>Objective</b> | Aktor dapat melaporkan pengacara |
| <b>Actor</b>     | Klien                            |





|                         |   |
|-------------------------|---|
| <b>Pre-Condition</b>    | Aktor berada pada halaman detail progres kasus atau detail bookingan saya.  |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Aktor memilih link Laporkan Pengacara pada halaman detail kasus atau detail bookingan saya yang dipilih.</li> <li>2. Sistem menampilkan <i>form</i> untuk melaporkan pengacara.</li> <li>3. Aktor mengisi alasan melaporkan pengacara pada <i>form</i> lalu menekan tombol Simpan.</li> </ol> |
| <b>Alternative Flow</b> | <ol style="list-style-type: none"> <li>1. Jika terdapat salah satu <i>field</i> tidak terisi, maka akan menampilkan peringatan "<i>Please fill out this field</i>".</li> <li>2. Jika aktor memilih tombol Batal maka pengacara tidak berhasil dilaporkan.</li> </ol>  |
| <b>Post Condition</b>   | Sistem menyimpan laporan pengacara ke dalam <i>database</i> .   |

### 30. Use Case Scenario Kirim Pertanyaan

Tabel 4.34 di bawah ini merupakan *use case scenario* untuk mengirimkan pertanyaan. Aktor yang dapat melakukan fungsional ini adalah Klien. *Use case* ini mempunyai fungsi untuk mengirimkan pertanyaan kepada pengacara.

**Tabel 4.34 Use Case Scenario Kirim Pertanyaan**

|                         |   |
|-------------------------|---|
| <b>SKPP-F-30</b>        | Kirim Pertanyaan  |
| <b>Objective</b>        | Aktor dapat mengirimkan pertanyaan umum ke dalam sistem   |
| <b>Actor</b>            | Klien   |
| <b>Pre-Condition</b>    | Aktor berada pada halaman Tanya pengacara   |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Aktor memilih tombol Tanya Pengacara.</li> <li>2. Sistem akan menampilkan <i>form</i> untuk mengisi pertanyaan dari aktor.</li> <li>3. Aktor menginputkan pertanyaan dengan mengisi <i>field</i> judul topik dan pertanyaan.</li> <li>4. Aktor menekan tombol Kirim.</li> </ol> |
| <b>Alternative Flow</b> | Jika terdapat salah satu <i>field</i> tidak terisi, maka akan menampilkan peringatan " <i>Please fill out this field</i> ".   |





|                       |   |
|-----------------------|---|
| <b>Post Condition</b> | Sistem menyimpan pertanyaan pada <i>database</i> dan menampilkannya ke halaman Tanya Pengacara. |
|-----------------------|---|

### 31. Use Case Scenario Lihat Jawaban Pertanyaan

Tabel 4.35 di bawah ini merupakan *use case scenario* untuk melihat jawaban pertanyaan. Aktor yang dapat melakukan fungsional ini adalah Klien. *Use case* ini mempunyai fungsi untuk melihat jawaban dari pertanyaan yang sudah klien kirimkan.

**Tabel 4.35 Use Case Scenario Lihat Jawaban Pertanyaan**

|                         |  |
|-------------------------|--|
| <b>SKPP-F-31</b>        | Lihat Jawaban Pertanyaan   |
| <b>Objective</b>        | Aktor melihat jawaban pertanyaan yang diajukan pada halaman Tanya Pengacara  |
| <b>Actor</b>            | Klien  |
| <b>Pre-Condition</b>    | Aktor berada pada halaman <i>dashboard</i> .   |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Aktor memilih menu Diskusi pada <i>sidebar</i>.</li> <li>2. Sistem menampilkan daftar pertanyaan yang dikirimkan oleh aktor.</li> <li>3. Aktor memilih salah satu pertanyaan.</li> </ol> |
| <b>Alternative Flow</b> | -  |
| <b>Post Condition</b>   | Sistem menampilkan jawaban pada pertanyaan.  |

### 32. Use Case Scenario Validasi Pengacara

Tabel 4.36 di bawah ini merupakan *use case scenario* untuk mevalidasi pengacara. Aktor yang dapat melakukan fungsional ini adalah Admin. *Use case* ini mempunyai fungsi untuk memvalidasi pengacara yang mendaftarkan diri ke dalam sistem.

**Tabel 4.36 Use Case Scenario Validasi Pengacara**

|                      |   |
|----------------------|---|
| <b>SKPP-F-32</b>     | Validasi Pengacara  |
| <b>Objective</b>     | Aktor dapat memvalidasi pengacara yang mendaftar ke dalam sistem  |
| <b>Actor</b>         | Admin   |
| <b>Pre-Condition</b> | Aktor berada pada halaman <i>dashboard</i> .  |
| <b>Main Flow</b>     | <ol style="list-style-type: none"> <li>1. Aktor memilih menu validasi pengacara pada <i>sidebar</i>.</li> </ol> |





|                         |   |
|-------------------------|---|
|                         | 2. Sistem menampilkan daftar pengacara yang mendaftarkan ke sistem. |
|                         | 3. Aktor menekan tombol Validasi untuk memvalidasi pengacara.       |
| <b>Alternative Flow</b> | -   |
| <b>Post Condition</b>   | Sistem menyimpan validasi pengacara ke dalam <i>database</i> .      |

### 33. Use Case Scenario Tolak Pengacara

Tabel 4.37 di bawah ini merupakan *use case scenario* untuk menolak pengacara. Aktor yang dapat melakukan fungsional ini adalah Admin. *Use case* ini mempunyai fungsi untuk menolak pengacara yang mendaftarkan diri ke dalam sistem.

**Tabel 4.37 Use Case Scenario Tolak Pengacara**

|                         |   |
|-------------------------|---|
| <b>SKPP-F-33</b>        | Tolak Pengacara   |
| <b>Objective</b>        | Aktor dapat menolak pengacara yang mendaftar ke dalam sistem  |
| <b>Actor</b>            | Admin   |
| <b>Pre-Condition</b>    | Aktor berada pada halaman <i>dashboard</i> .  |
| <b>Main Flow</b>        | 1. Aktor memilih menu validasi pengacara pada <i>sidebar</i> .<br>2. Sistem menampilkan daftar pengacara yang mendaftarkan ke sistem.<br>3. Aktor menekan tombol Tolak untuk menolak pengacara. |
| <b>Alternative Flow</b> | -   |
| <b>Post Condition</b>   | Sistem menyimpan tolak pengacara ke dalam <i>database</i> .   |

### 34. Use Case Scenario Lihat Portofolio Pengacara

Table 4.38 di bawah ini merupakan *use case scenario* untuk melihat portofolio pengacara. Aktor yang dapat melakukan fungsional ini adalah Admin. *Use case* ini mempunyai fungsi untuk melihat portofolio yang dikirimkan oleh pengacara.

**Tabel 4.38 Use Case Scenario Lihat Portofolio Pengacara**

|                  |                            |
|------------------|----------------------------|
| <b>SKPP-F-34</b> | Lihat Portofolio Pengacara |
|------------------|----------------------------|





|                         |   |
|-------------------------|---|
| <b>Objective</b>        | Aktor melihat portofolio yang dikirimkan pengacara.   |
| <b>Actor</b>            | Admin   |
| <b>Pre-Condition</b>    | Aktor berada pada halaman Validasi Pengacara  |
| <b>Main Flow</b>        | Aktor memilih salah satu tombol Lihat yang ada pada daftar pengacara di halaman Validasi Pengacara. |
| <b>Alternative Flow</b> | -   |
| <b>Post Condition</b>   | Sistem menampilkan portofolio yang dikirimkan oleh pengacara.                                       |

### 35. Use Case Scenario Lihat Laporan Info Hukum

Table 4.39 di bawah ini merupakan *use case scenario* untuk Melihat Laporan Info Hukum. Aktor yang dapat melakukan fungsional ini adalah Admin. *Use case* ini mempunyai fungsi untuk melihat info hukum yang dilaporkan.

**Tabel 4.39 Use Case Scenario Lihat Laporan Info Hukum**

|                         |  |
|-------------------------|--|
| <b>SKPP-F-35</b>        | Lihat Laporan Info Hukum   |
| <b>Objective</b>        | Aktor dapat mengetahui daftar info hukum yang dilaporkan.        |
| <b>Actor</b>            | Admin  |
| <b>Pre-Condition</b>    | Aktor berada pada halaman <i>dashboard</i> .                     |
| <b>Main Flow</b>        | Aktor memilih menu Laporan Info Hukum pada menu <i>sidebar</i> . |
| <b>Alternative Flow</b> | -  |
| <b>Post Condition</b>   | Sistem menampilkan daftar info hukum yang dilaporkan             |

### 36. Use Case Scenario Hapus Laporan Info Hukum

Table 4.40 di bawah ini merupakan *use case scenario* untuk Menghapus Laporan Info Hukum. Aktor yang dapat melakukan fungsional ini adalah Admin. *Use case* ini mempunyai fungsi untuk menghapus info hukum yang dilaporkan.

**Tabel 4.40 Use Case Scenario Hapus Laporan Info Hukum**

|                  |  |
|------------------|--|
| <b>SKPP-F-36</b> | Hapus Laporan Info Hukum                         |
| <b>Objective</b> | Aktor dapat menghapus info hukum yang dilaporkan |
| <b>Actor</b>     | Admin  |





|                         |   |
|-------------------------|---|
| <b>Pre-Condition</b>    | Aktor berada pada halaman Laporan Info Hukum  |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Sistem menampilkan alasan mengapa info hukum tersebut dilaporkan</li> <li>2. Aktor menekan tombol Validasi pada aksi jika alasan pelaporan sesuai.</li> </ol> |
| <b>Alternative Flow</b> | Jika alasan pelaporan info hukum tidak sesuai, aktor menekan tombol Tolak untuk membatalkan pelaporan.  |
| <b>Post Condition</b>   | Sistem menyimpan perubahan pada <i>database</i> .   |

### 37. Use Case Scenario Lihat Laporan Pengacara

Table 4.41 di bawah ini merupakan *use case scenario* untuk Melihat Laporan Pengacara. Aktor yang dapat melakukan fungsional ini adalah Admin. *Use case* ini mempunyai fungsi untuk melihat pengacara yang dilaporkan oleh klien.

**Tabel 4.41 Use Case Scenario Lihat Laporan Pengacara**

|                         |   |
|-------------------------|---|
| <b>SKPP-F-37</b>        | Lihat Laporan Pengacara   |
| <b>Objective</b>        | Aktor dapat mengetahui daftar pengacara yang dilaporkan.        |
| <b>Actor</b>            | Admin   |
| <b>Pre-Condition</b>    | Aktor berada pada halaman <i>dashboard</i> .                    |
| <b>Main Flow</b>        | Aktor memilih menu Laporan Pengacara pada menu <i>sidebar</i> . |
| <b>Alternative Flow</b> | -   |
| <b>Post Condition</b>   | Sistem menampilkan daftar pengacara yang dilaporkan             |

### 38. Use Case Scenario Hapus Pengacara

Table 4.42 di bawah ini merupakan *use case scenario* untuk Menghapus Pengacara. Aktor yang dapat melakukan fungsional ini adalah Admin. *Use case* ini mempunyai fungsi untuk menghapus pengacara yang dilaporkan.

**Tabel 4.42 Use Case Scenario Hapus Pengacara**

|                  |   |
|------------------|---|
| <b>SKPP-F-38</b> | Hapus Pengacara                                 |
| <b>Objective</b> | Aktor dapat menghapus pengacara yang dilaporkan |
| <b>Actor</b>     | Admin   |





|                         |   |
|-------------------------|---|
| <b>Pre-Condition</b>    | Aktor berada pada halaman Laporan Pengacara   |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Sistem menampilkan alasan mengapa pengacara tersebut dilaporkan.</li> <li>2. Aktor menekan tombol Validasi jika alasan pelaporan sesuai.</li> </ol> |
| <b>Alternative Flow</b> | Jika alasan pelaporan pengacara tidak sesuai, aktor menekan tombol Tolak untuk membatalkan pelaporan.   |
| <b>Post Condition</b>   | Sistem menyimpan perubahan pada <i>database</i> .   |

### 39. Use Case Scenario Lihat Laporan Klien

Table 4.43 di bawah ini merupakan *use case scenario* untuk Melihat Laporan Klien. Aktor yang dapat melakukan fungsional ini adalah Admin. *Use case* ini mempunyai fungsi untuk melihat klien yang dilaporkan.

**Tabel 4.43 Use Case Scenario Lihat Laporan Klien**

|                         |   |
|-------------------------|---|
| <b>SKPP-F-39</b>        | Lihat Laporan Klien   |
| <b>Objective</b>        | Aktor dapat mengetahui daftar klien yang dilaporkan.        |
| <b>Actor</b>            | Admin   |
| <b>Pre-Condition</b>    | Aktor berada pada halaman <i>dashboard</i> .                |
| <b>Main Flow</b>        | Aktor memilih menu Laporan Klien pada menu <i>sidebar</i> . |
| <b>Alternative Flow</b> |   |
| <b>Post Condition</b>   | Sistem menampilkan daftar Klien yang dilaporkan.            |

### 40. Use Case Scenario Hapus Klien

Table 4.44 di bawah ini merupakan *use case scenario* untuk Menghapus Klien. Aktor yang dapat melakukan fungsional ini adalah Admin. *Use case* ini mempunyai fungsi untuk menghapus klien yang dilaporkan.

**Tabel 4.44 Use Case Scenario Hapus Klien**

|                      |   |
|----------------------|---|
| <b>SKPP-F-40</b>     | Hapus Klien                                 |
| <b>Objective</b>     | Aktor dapat menghapus klien yang dilaporkan |
| <b>Actor</b>         | Admin                                       |
| <b>Pre-Condition</b> | Aktor berada pada halaman Laporan Klien     |





|                         |   |
|-------------------------|---|
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Sistem menampilkan alasan mengapa klien tersebut dilaporkan.</li> <li>2. Aktor menekan tombol Validasi jika alasan pelaporan sesuai.</li> </ol> |
| <b>Alternative Flow</b> | Jika alasan pelaporan klien tidak sesuai, aktor menekan tombol Tolak untuk membatalkan pelaporan.   |
| <b>Post Condition</b>   | Sistem menyimpan perubahan pada <i>database</i> .   |

#### 41. Use Case Scenario Beri Komentar

Tabel 4.45 di bawah ini merupakan *use case scenario* untuk memberi komentar. Aktor yang dapat melakukan fungsional ini adalah Klien dan Pengacara. *Use case* ini mempunyai fungsi untuk memberikan komentar pada info hukum.

**Tabel 4.45 Use Case Scenario Beri Komentar**

|                         |  |
|-------------------------|--|
| <b>SKPP-F-41</b>        | Beri Komentar  |
| <b>Objective</b>        | Aktor dapat memberikan komentar pada info hukum.   |
| <b>Actor</b>            | Klien dan Pengacara.   |
| <b>Pre-Condition</b>    | Aktor berada pada halaman detail info hukum  |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Aktor mengisikan <i>field</i> tinggalkan komentar dengan mengisikan komen yang ingin disampaikan.</li> <li>2. Aktor menekan tombol Kirim.</li> </ol> |
| <b>Alternative Flow</b> | Jika terdapat salah satu <i>field</i> tidak terisi, maka akan menampilkan peringatan " <i>Please fill out this field</i> ".  |
| <b>Post Condition</b>   | Sistem menyimpan komentar pada <i>database</i> lalu menampilkan pada halaman info hukum.   |

#### 42. Use Case Scenario Laporkan Info Hukum

Tabel 4.46 di bawah ini merupakan *use case scenario* untuk melaporkan info hukum. Aktor yang dapat melakukan fungsional ini adalah Klien dan Pengacara. *Use case* ini mempunyai fungsi untuk melaporkan info hukum yang menyimpang.

**Tabel 4.46 Use Case Diagram Laporkan Info Hukum**

|                  |                                   |
|------------------|-----------------------------------|
| <b>SKPP-F-46</b> | Laporkan Info Hukum               |
| <b>Objective</b> | Aktor dapat melaporkan info hukum |





|                         |  |
|-------------------------|--|
| <b>Actor</b>            | Klien dan pengacara  |
| <b>Pre-Condition</b>    | Aktor berada pada halaman detail info hukum  |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Aktor menekan <i>link</i> Laporkan yang terdapat pada info hukum.</li> <li>2. Sistem akan menampilkan <i>form</i> untuk melaporkan info hukum.</li> <li>3. Aktor mengisi alasan melaporkan info hukum.</li> <li>4. Aktor menekan tombol simpan.</li> </ol> |
| <b>Alternative Flow</b> | <ol style="list-style-type: none"> <li>1. Jika terdapat salah satu <i>field</i> tidak terisi, maka akan menampilkan peringatan "<i>Please fill out this field</i>".</li> <li>2. Jika memilih batalkan maka info hukum tidak berhasil dilaporkan.</li> </ol>  |
| <b>Post Condition</b>   | Sistem menyimpan laporan info hukum pada <i>database</i> .   |

#### 43. Use Case Scenario Lihat Profil

Tabel 4.47 di bawah ini merupakan *use case scenario* untuk melihat profil. Aktor yang dapat melakukan fungsional ini adalah Klien dan Pengacara. *Use case* ini mempunyai fungsi untuk melihat data diri aktor.

**Tabel 4.47 Use Case Scenario Lihat Melihat Profil**

|                         |  |
|-------------------------|--|
| <b>SKPP-F-43</b>        | Lihat Profil   |
| <b>Objective</b>        | Aktor dapat melihat profil yang berisikan data diri dari aktor   |
| <b>Actor</b>            | Klien dan Pengacara  |
| <b>Pre-Condition</b>    | Aktor berhasil <i>login</i> dan ada pada halaman <i>home</i> sistem.   |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Aktor memilih menu profil pada menu bar.</li> <li>2. Aktor memilih menu profil pada <i>sidebar</i>.</li> </ol> |
| <b>Alternative Flow</b> |  |
| <b>Post Condition</b>   | Sistem menampilkan halaman profil.   |





#### 44. Use Case Scenario Edit Profil

Tabel 4.48 di bawah ini merupakan *use case scenario* untuk mengedit profil. Aktor yang dapat melakukan fungsional ini adalah Klien dan Pengacara. *Use case* ini mempunyai fungsi untuk mengedit profil atau data diri.

**Tabel 4.48 Use Case Scenario Edit Profil**

|                         |  |
|-------------------------|--|
| <b>SKPP-F-44</b>        | Mengedit Profil  |
| <b>Objective</b>        | Aktor dapat mengedit profil  |
| <b>Actor</b>            | Klien dan Pengacara  |
| <b>Pre-Condition</b>    | Aktor berada pada halaman profil   |
| <b>Main Flow</b>        | 1. Aktor mengisikan perubahan pada <i>field</i> yang diinginkan datanya untuk diubah.<br>2. Aktor menekan tombol <i>Update</i> . |
| <b>Alternative Flow</b> | Jika terdapat salah satu <i>field</i> tidak terisi, maka akan menampilkan peringatan " <i>Please fill out this field</i> ".      |
| <b>Post Condition</b>   | Sistem menyimpan perubahan profil ke dalam <i>database</i> .   |

#### 45. Use Case Scenario Lihat Daftar Pengacara

Tabel 4.49 di bawah ini merupakan *use case scenario* untuk Melihat Daftar Pengacara. Aktor yang dapat melakukan fungsional ini adalah Pengguna, Klien, dan Pengacara. *Use case* ini mempunyai fungsi untuk melihat halaman daftar pengacara.

**Tabel 4.49 Use Case Scenario Lihat Daftar Pengacara**

|                         |   |
|-------------------------|---|
| <b>SKPP-F-45</b>        | Lihat Daftar Pengacara  |
| <b>Objective</b>        | Aktor dapat melihat halaman daftar pengacara  |
| <b>Actor</b>            | Pengguna, Klien, dan Pengacara.   |
| <b>Pre-Condition</b>    | Aktor berada pada halaman <i>home</i> .   |
| <b>Main Flow</b>        | Aktor memilih daftar pengacara dan menekan tombol Lihat Semua yang ada di <i>home</i> . |
| <b>Alternative Flow</b> | -   |
| <b>Post Condition</b>   | Sistem menampilkan halaman daftar pengacara.  |

#### 46. Use Case Scenario Lihat Profil Pengacara

Tabel 4.50 di bawah ini merupakan *use case scenario* untuk melihat profil pengacara. Aktor yang dapat melakukan fungsional ini adalah Pengguna,





Klien, dan Pengacara. *Use case* ini mempunyai fungsi untuk melihat halaman profil pengacara.

**Tabel 4.50 Use Case Scenario Lihat Profil Pengacara**

|                         |  |
|-------------------------|--|
| <b>SKPP-F-46</b>        | Lihat Profil Pengacara   |
| <b>Objective</b>        | Aktor dapat melihat halaman profil pengacara                               |
| <b>Actor</b>            | Pengguna, Klien, dan Pengacara.  |
| <b>Pre-Condition</b>    | Aktor berada pada halaman daftar pengacara                                 |
| <b>Main Flow</b>        | Aktor memilih salah satu pengacara yang ada pada halaman daftar pengacara. |
| <b>Alternative Flow</b> | -  |
| <b>Post Condition</b>   | Sistem menampilkan halaman profil pengacara yang dipilih.                  |

#### 47. Use Case Scenario Cari Pengacara

Tabel 4.51 di bawah ini merupakan *use case scenario* untuk mencari pengacara. Aktor yang dapat melakukan fungsional ini adalah Pengguna, Klien, dan Pengacara. *Use case* ini mempunyai fungsi untuk mencari pengacara.

**Tabel 4.51 Use Case Scenario Mencari Pengacara**

|                         |   |
|-------------------------|---|
| <b>SKPP-F-47</b>        | Cari Pengacara  |
| <b>Objective</b>        | Aktor dapat mencari pengacara.  |
| <b>Actor</b>            | Pengguna, Klien, dan Pengacara.   |
| <b>Pre-Condition</b>    | Aktor berada pada <i>home</i> .   |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Aktor memilih menu cari pengacara yang ada pada menu bar.</li> <li>2. Aktor memilih jenis pengacara dan dari daerah mana pengacara yang ingin dicari.</li> <li>3. Aktor menekan tombol Cari Pengacara.</li> </ol> |
| <b>Alternative Flow</b> | Jika jenis pengacara dan kabupaten tidak diisi, maka akan menampilkan "Pengacara Tidak Ditemukan".  |
| <b>Post Condition</b>   | Sistem menampilkan daftar pengacara sesuai dengan kategori yang sudah ditentukan.   |





#### 48. Use Case Scenario Lihat Info Hukum

Tabel 4.52 di bawah ini merupakan *use case scenario* untuk melihat info hukum. Aktor yang dapat melakukan fungsional ini adalah Pengguna, Klien, dan Pengacara. *Use case* ini mempunyai fungsi untuk melihat detail info hukum.

**Tabel 4.52 Use Case Scenario Lihat Info Hukum**

|                         |  |
|-------------------------|--|
| <b>SKPP-F-48</b>        | Lihat Info Hukum   |
| <b>Objective</b>        | Aktor dapat melihat info hukum yang telah dipilih.       |
| <b>Actor</b>            | Pengguna, Klien, dan Pengacara.                          |
| <b>Pre-Condition</b>    | Aktor berada pada halaman info hukum.                    |
| <b>Main Flow</b>        | Aktor memilih info hukum yang ingin dibaca selengkapnya. |
| <b>Alternative Flow</b> | -  |
| <b>Post Condition</b>   | Sistem menampilkan info hukum yang telah dipilih.        |

#### 4.3.2.2 Use Case Scenario Iterasi 2

*Use case scenario* iterasi kedua merupakan alur kinerja dari subbab 4.3.1.2 yaitu *use case diagram* iterasi kedua pada sistem pencarian dan konsultasi pengacara.

##### 1. Use Case Scenario Tambah Hasil Pertemuan Konsultasi Tatap Muka

Tabel 4.53 di bawah ini merupakan *use case scenario* untuk Menambahkan hasil pertemuan konsultasi tatap muka. Aktor yang dapat melakukan fungsional ini adalah Pengacara. *Use case* ini mempunyai fungsi untuk menambahkan hasil pertemuan dari hasil konsultasi tatap muka.

**Tabel 4.53 Use Case Scenario Hasil Pertemuan Konsultasi Tatap Muka**

|                      |  |
|----------------------|--|
| <b>SKPP-F-49</b>     | Tambah Hasil Pertemuan Konsultasi Tatap Muka   |
| <b>Objective</b>     | Aktor dapat menambahkan hasil pertemuan dari konsultasi tatap muka.  |
| <b>Actor</b>         | Pengacara  |
| <b>Pre-Condition</b> | Aktor berada pada halaman dashboard  |
| <b>Main Flow</b>     | <ol style="list-style-type: none"> <li>1. Aktor memilih menu Bookingan Pertemuan pada menu <i>sidebar</i>.</li> <li>2. Sistem menampilkan daftar pertemuan dari aktor.</li> <li>3. Aktor memilih tab Disetujui.</li> </ol> |





|                         |   |
|-------------------------|---|
|                         | <ol style="list-style-type: none"> <li>4. Sistem menampilkan kasus yang telah disetujui.</li> <li>5. Aktor memilih tombol Selesai</li> <li>6. Sistem menampilkan <i>form</i> untuk memasukkan hasil pertemuan.</li> <li>7. Aktor menuliskan hasil pertemuan dan memilih tombol Simpan.</li> </ol> |
| <b>Alternative Flow</b> | Jika terdapat salah satu <i>field</i> tidak terisi, maka akan menampilkan peringatan " <i>Please fill out this field</i> ".   |
| <b>Post Condition</b>   | Sistem menyimpan berita acara pada <i>database</i> .  |

## 2. Use Case Scenario Tambah Progres Kasus

Tabel 4.54 di bawah ini merupakan *use case scenario* untuk Menambahkan Progres Kasus. Aktor yang dapat melakukan fungsional ini adalah Pengacara. *Use case* ini mempunyai fungsi untuk menambahkan progres kasus.

**Tabel 4.54 Use Case Scenario Tambah Progres Kasus**

|                         |  |
|-------------------------|--|
| <b>SKPP-F-50</b>        | Tambah Progres Kasus   |
| <b>Objective</b>        | Aktor dapat menambahkan progres kasus milik Klien.   |
| <b>Actor</b>            | Pengacara  |
| <b>Pre-Condition</b>    | Aktor berada pada halaman Manajemen Kasus  |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Sistem menampilkan daftar kasus yang ditangani aktor.</li> <li>2. Aktor memilih salah satu kasus.</li> <li>3. Sistem menampilkan detail kasus.</li> <li>4. Aktor menekan tombol Tambahkan Progres Kasus.</li> <li>5. Sistem menampilkan <i>form</i> untuk menuliskan progres kasus.</li> <li>6. Aktor menuliskan progress dari kasus dan memilih tombol Simpan.</li> </ol> |
| <b>Alternative Flow</b> | Jika terdapat salah satu <i>field</i> tidak terisi, maka akan menampilkan peringatan " <i>Please fill out this field</i> ".  |
| <b>Post Condition</b>   | Sistem menyimpan progres kasus ke dalam <i>database</i> .  |





### 3. Use Case Scenario Edit Progres Kasus

Tabel 4.55 di bawah ini merupakan *use case scenario* untuk Mengedit Progres Kasus. Aktor yang dapat melakukan fungsional ini adalah Pengacara. *Use case* ini mempunyai fungsi untuk mengedit progres kasus.

**Tabel 4.55 Use Case Scenario Edit Progres Kasus**

|                         |   |
|-------------------------|---|
| <b>SKPP-F-51</b>        | Edit Progres Kasus  |
| <b>Objective</b>        | Aktor dapat mengedit progres kasus.   |
| <b>Actor</b>            | Pengacara   |
| <b>Pre-Condition</b>    | Aktor berada pada halaman Manajemen Kasus   |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Sistem menampilkan daftar kasus yang ditangani pengacara.</li> <li>2. Aktor memilih salah satu kasus.</li> <li>3. Sistem menampilkan detail kasus.</li> <li>4. Aktor menekan tombol Edit pada progres kasus yang dipilih.</li> <li>5. Sistem menampilkan <i>form</i> untuk mengedit progres kasus.</li> <li>6. Aktor menuliskan progres dari kasus yang ingin diganti dan memilih tombol Simpan.</li> </ol> |
| <b>Alternative Flow</b> | Jika terdapat salah satu <i>field</i> tidak terisi, maka akan menampilkan peringatan " <i>Please fill out this field</i> ".   |
| <b>Post Condition</b>   | Sistem menyimpan edit progres kasus ke dalam <i>database</i> .  |

### 4. Use Case Scenario Hapus Progres Kasus

Tabel 4.56 di bawah ini merupakan *use case scenario* untuk Menghapus Progres Kasus. Aktor yang dapat melakukan fungsional ini adalah Pengacara. *Use case* ini mempunyai fungsi untuk menghapus progres kasus.

**Tabel 4.56 Use Case Scenario Hapus Progres Kasus**

|                      |  |
|----------------------|--|
| <b>SKPP-F-52</b>     | Hapus Progres Kasus  |
| <b>Objective</b>     | Aktor dapat menghapus progres kasus.   |
| <b>Actor</b>         | Pengacara  |
| <b>Pre-Condition</b> | Aktor berada pada halaman Manajemen Kasus  |
| <b>Main Flow</b>     | <ol style="list-style-type: none"> <li>1. Sistem menampilkan daftar kasus yang ditangani aktor.</li> </ol> |





|                         |  |
|-------------------------|--|
|                         | <ol style="list-style-type: none"> <li>Aktor memilih salah satu kasus.</li> <li>Sistem menampilkan detail kasus.</li> <li>Aktor menekan tombol Hapus pada progres kasus yang pilih.</li> </ol> |
| <b>Alternative Flow</b> | -  |
| <b>Post Condition</b>   | Sistem menghapus progres kasus pada <i>database</i> .  |

#### 5. Use Case Scenario Lihat Progres Kasus

Tabel 4.57 di bawah ini merupakan *use case scenario* untuk melihat progres kasus. Aktor yang dapat melakukan fungsional ini adalah Klien. *Use case* ini mempunyai fungsi untuk melihat progres kasus yang dimiliki.

**Tabel 4.57 Use Case Scenario Lihat Progres Kasus**

|                         |   |
|-------------------------|---|
| <b>SKPP-F-53</b>        | Lihat Progress Kasus  |
| <b>Objective</b>        | Aktor dapat melihat progres kasus dari kasus yang dimiliki.   |
| <b>Actor</b>            | Klien   |
| <b>Pre-Condition</b>    | Aktor berada pada halaman <i>dashboard</i>  |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>Aktor memilih menu progres kasus dari menu <i>sidebar</i>.</li> <li>Sistem menampilkan halaman progres kasus.</li> <li>Aktor memilih kasus yang ingin dilihat progresnya dan menekan tombol Selengkapnya.</li> </ol> |
| <b>Alternative Flow</b> | -   |
| <b>Post Condition</b>   | Sistem menampilkan progres dari kasus yang dipilih.   |

#### 6. Use Case Scenario Lihat Hasil Pertemuan Konsultasi Tatap Muka

Tabel 4.58 di bawah ini merupakan *use case scenario* untuk melihat hasil pertemuan konsultasi tatap muka. Aktor yang dapat melakukan fungsional ini adalah Klien. *Use case* ini mempunyai fungsi untuk melihat hasil pertemuan setelah konsultasi tatap muka selesai dilakukan.

**Tabel 4.58 Use Scenario Lihat Hasil Pertemuan Konsultasi Tatap Muka**

|                  |  |
|------------------|--|
| <b>SKPP-F-54</b> | Lihat Hasil Pertemuan Konsultasi Tatap Muka                |
| <b>Objective</b> | Aktor dapat melihat hasil pertemuan konsultasi tatap muka. |
| <b>Actor</b>     | Klien  |





|                         |  |
|-------------------------|--|
| <b>Pre-Condition</b>    | Aktor berada pada halaman Bookingan Saya   |
| <b>Main Flow</b>        | <ol style="list-style-type: none"> <li>1. Sistem menampilkan daftar bookingan aktor.</li> <li>2. Aktor memilih bookingan yang ingin dilihat hasil pertemuannya dan menekan tombol Selengkapnya.</li> </ol> |
| <b>Alternative Flow</b> |  |
| <b>Post Condition</b>   | Sistem menampilkan hasil pertemuan konsultasi tatap muka.  |



## BAB 5 PERANCANGAN DAN IMPLEMENTASI

### 5.1 Perancangan

Perancangan merupakan proses yang dilakukan setelah proses analisis kebutuhan. Proses perancangan pada Sistem Konsultasi dan Pencarian Pengacara dibagi menjadi beberapa tahap yakni perancangan *sequence diagram*, perancangan *class diagram*, perancangan basis data, dan perancangan antarmuka. Setelah proses perancangan sistem, kemudian dilakukan implementasi sistem sesuai dengan hasil dari rancangan yang telah dibuat.

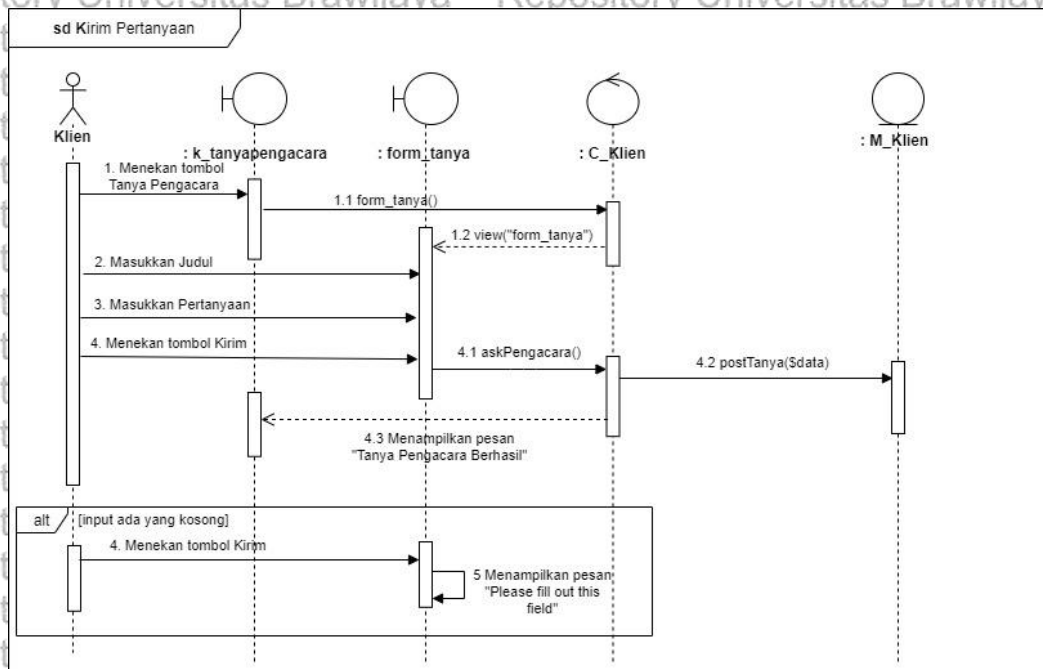
#### 5.1.1 Perancangan *Sequence Diagram*

*Sequence diagram* adalah diagram yang menggambarkan bagaimana objek saling berinteraksi dengan sistem melalui pesan dari *usecase* atau suatu operasi tertentu. Pada subbab ini akan dijelaskan *sequence diagram* dari Sistem Konsultasi dan Pencarian Pengacara yang dibagi menjadi dua iterasi yaitu iterasi pertama dan iterasi kedua.

##### 5.1.1.1 *Sequence Diagram* Iterasi 1

Berikut ini merupakan *sequence diagram* iterasi pertama yang akan dijelaskan diantaranya *sequence diagram* kirim pertanyaan, *sequence diagram* kirim portofolio, dan *sequence diagram* validasi pengacara.

#### 1. *Sequence Diagram* Kirim Pertanyaan



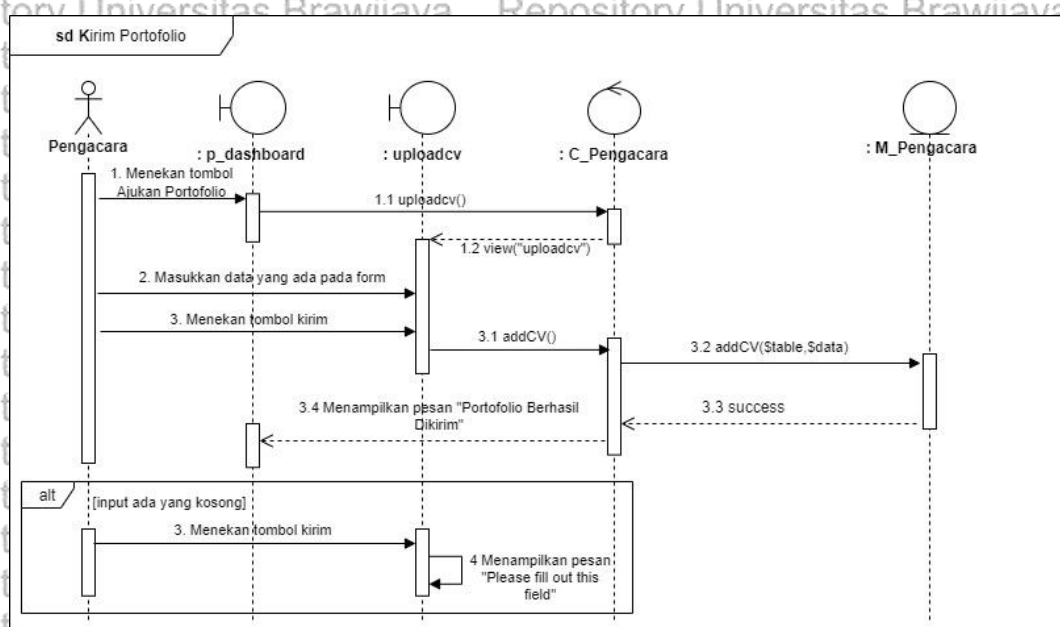
Gambar 5.1 *Sequence Diagram* Kirim Pertanyaan





Pada gambar 5.1 merupakan *sequence diagram* kirim pertanyaan. Dalam interaksi tersebut menjelaskan bagaimana alur klien saat mengirimkan pertanyaan pada sistem. Pertama klien menekan tombol Tanya Pengacara pada *view* `k_tanyapengacara`. Lalu sistem akan memanggil *method* `form_tanya()` pada *controller* `C_Klien`. Kemudian sistem menampilkan *form* untuk memasukkan pertanyaan. Selanjutnya klien memasukkan judul dan pertanyaan pada *form* lalu menekan tombol kirim. Lalu sistem memanggil *method* `askPengacara()` yang ada pada *controller* `C_Klien`. Kemudian sistem akan memanggil *method* `postTanya(data)` pada model `M_Klien`. Lalu setelah proses penambahan tanya pengacara sukses, sistem akan menampilkan notifikasi Tanya Pengacara Berhasil dan kembali ke *view* `k_tanyapengacara`. Pada *sequence diagram* terdapat satu *alternatif flow* yaitu jika terdapat inputan yang tidak diisi maka sistem akan menampilkan *alert* "Please fill out this field".

## 2. Sequence Diagram Kirim Portofolio



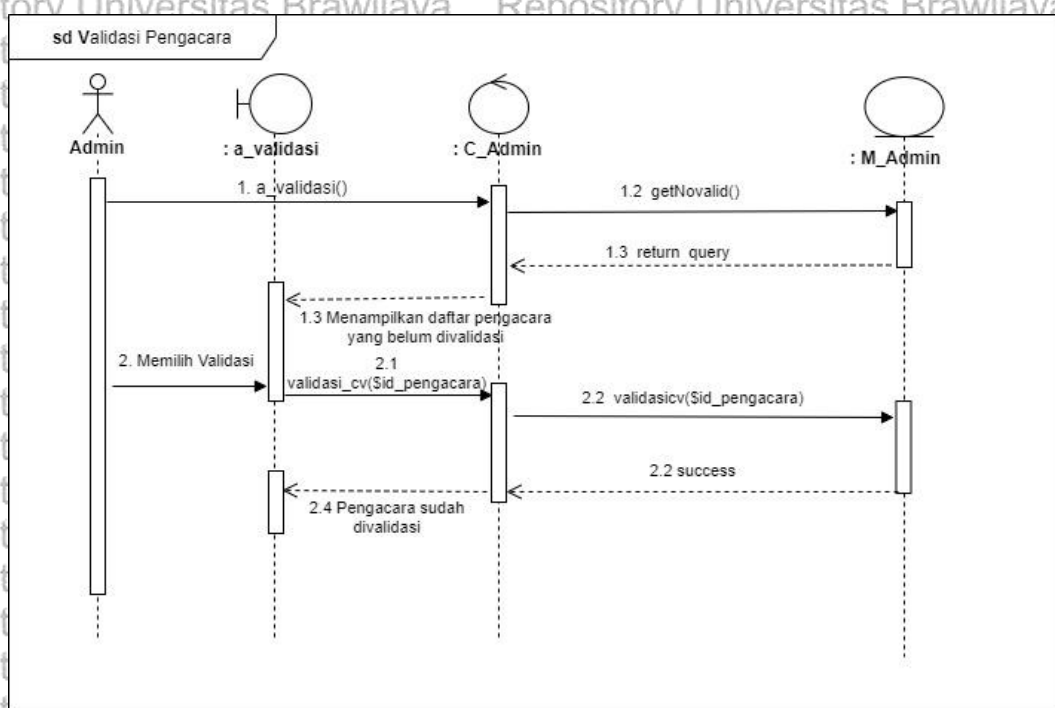
**Gambar 5.2 Sequence Diagram Kirim Portofolio**

Pada gambar 5.2 merupakan *sequence diagram* Kirim Portofolio. Dalam interaksi tersebut menjelaskan bagaimana alur pengacara dalam mengirimkan portofolio. Pertama pengacara menekan tombol Ajukan Portofolio pada *view* `p_dashboard`. Lalu sistem akan memanggil *method* `uploadcv()` pada *controller* `C_Pengacara` dan sistem akan menampilkan *form* untuk mengisi data portofolio. Selanjutnya pengacara memasukkan semua inputan data yang ada pada *form* lalu menekan tombol kirim. Kemudian sistem memanggil *method* `addCV()` pada *controller* `C_Pengacara`. Lalu sistem akan memanggil *method* `addCV($stable, $data)` pada model `M_Pengacara`. Kemudian jika data yang dikirimkan sukses masuk kedalam *database*, maka sistem akan menampilkan notifikasi "Portofolio Berhasil Dikirim". Pada *sequence diagram* terdapat satu *alternatif flow* yaitu jika terdapat inputan yang tidak diisi maka sistem akan menampilkan *alert* "Please fill out this field".





### 3. Sequence Diagram Validasi Pengacara



Gambar 5.3 Sequence Diagram Validasi Pengacara

Pada gambar 5.3 merupakan *sequence diagram* validasi pengacara. Dalam interaksi tersebut menjelaskan bagaimana admin memvalidasi pengacara yang mendaftarkan dirinya pada sistem. Pertama sistem akan memanggil *method* `a_validasi()` pada *controller* `C_Admin`. Lalu sistem akan memanggil *method* `getNovalid()` pada *model* `M_Admin`. Kemudian sistem menampilkan daftar pengacara yang belum divalidasi pada *view* `a_validasi`. Selanjutnya admin memilih *link* validasi pada pengacara yang akan divalidasi. Kemudian sistem memanggil *method* `validasi_cv($id_pengacara)`. Lalu sistem memanggil *method* `validasicv($id_pengacara)` pada *model* `M_Admin`. Kemudian jika data sudah terbaharui pada *database* maka statusnya *success* dan pada *view* `a_validasi` pengacara yang sudah divalidasi tidak ada lagi di tabel daftar pengacara.

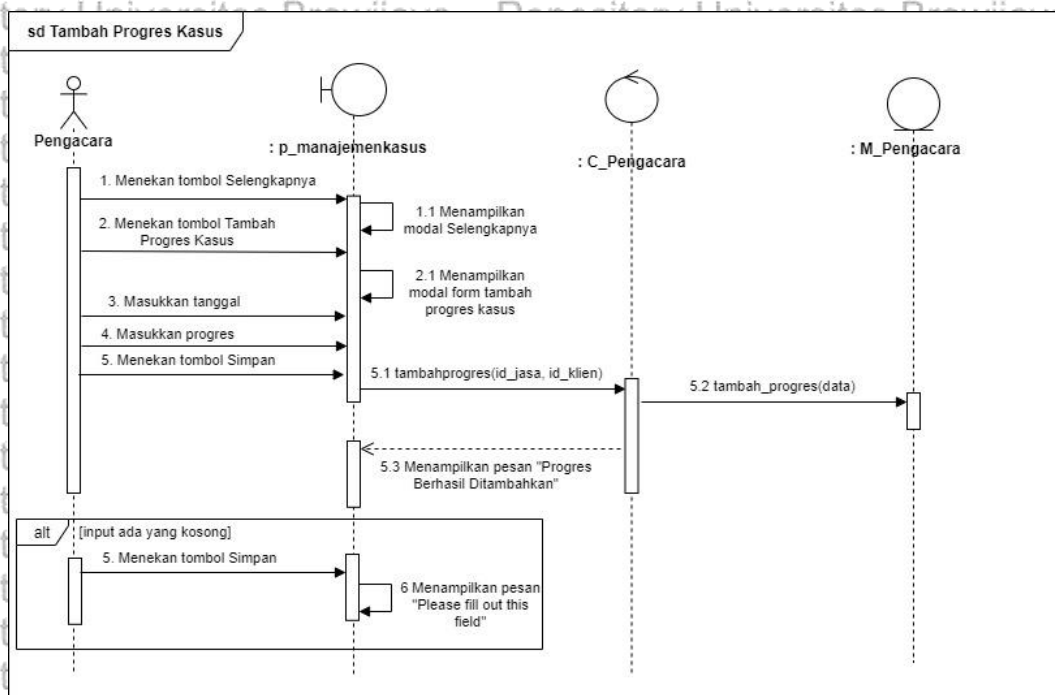
#### 5.1.1.2 Sequence Diagram Iterasi 2

Berikut ini merupakan *sequence diagram* iterasi kedua yang akan dijelaskan adalah *sequence diagram* tambah progres kasus. Pada gambar 5.4 dibawah ini merupakan *sequence diagram* tambah progres kasus. Dalam interaksi tersebut menjelaskan bagaimana pengacara menambahkan progres kasus pada kasus yang ditangani oleh pengacara. Pertama pengacara menekan tombol Selengkapnya, lalu sistem akan menampilkan modal selengkapnya. Kemudian pengacara menekan tombol Tambah Progres Kasus, lalu sistem akan menampilkan modal *form* tambah progres kasus. Pengacara memasukkan tanggal dan progres, kemudian menekan tombol Simpan. Sistem akan memanggil *method* `tambahprogres(id_jasa, id_klien)` pada *controller* `C_Pengacara`. Lalu memanggil *method* `tambah_progres(data)` pada *model* `M_Pengacara`. Dan sistem akan





menampilkan pesan “Progres Berhasil Ditambahkan”. Pada *sequence diagram* terdapat satu *alternatif flow* yaitu jika terdapat inputan yang tidak diisi maka sistem akan menampilkan alert “Please fill out this field”.



Gambar 5.4 Sequence Diagram Tambah Progres Kasus

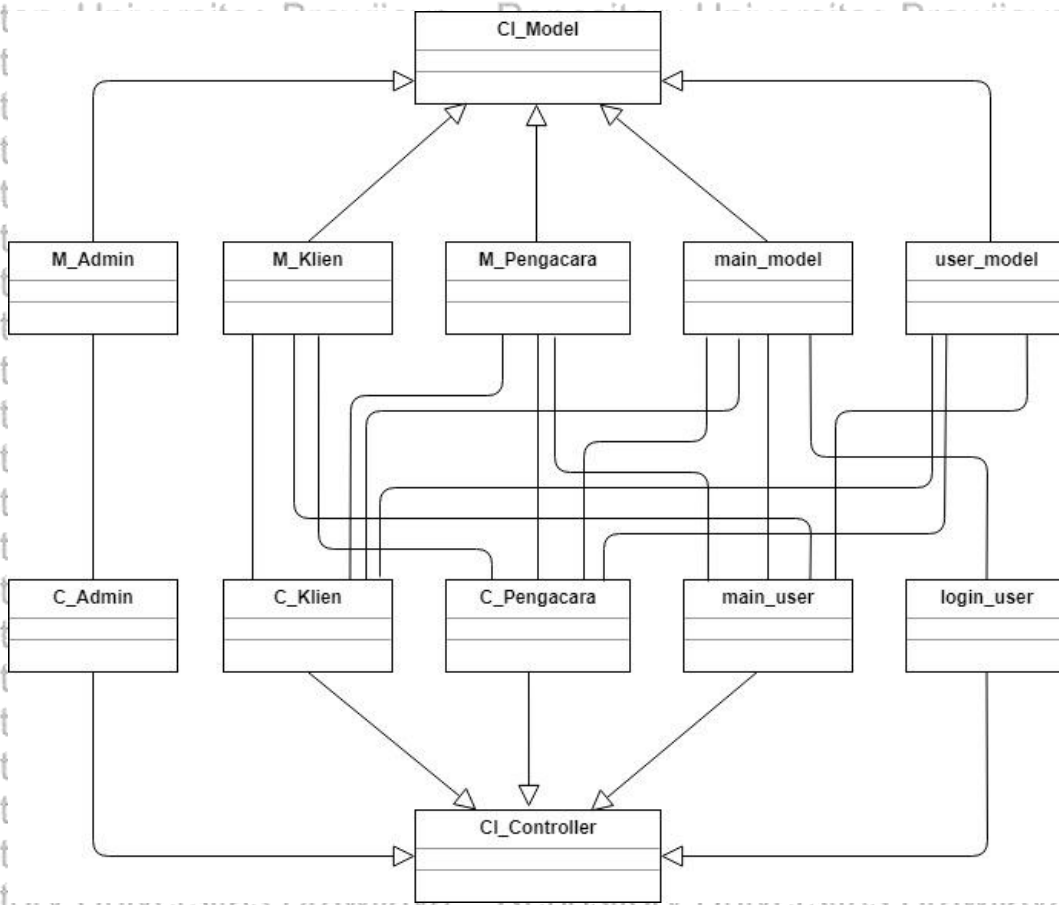
### 5.1.2 Perancangan Class Diagram

Perancangan *Class Diagram* merupakan tahap untuk mendeskripsikan struktur *class* yang ada pada sistem. Pada perancangan *class diagram* dari Sistem Konsultasi dan Pencarian Pengacara dibagi menjadi 2, yaitu perancangan *class diagram* umum dan perancangan *class diagram* detail.

#### 1. Perancangan Class Diagram Umum

Perancangan *class diagram* umum pada Sistem Konsultasi dan Pencarian Pengacara ini menggunakan *framework codeigniter* sehingga mengimplementasikan dua *class* utama yaitu *CI\_Controller* dan *CI\_Model*. *CI\_controller* merupakan *parent class* dari masing-masing *class controller*, sedangkan *CI\_Model* merupakan *parent class* dari masing-masing *class model*. Terdapat lima *class controller* yaitu *main\_user*, *login\_user*, *C\_Pengacara*, *C\_Klien* dan *C\_Admin*. Sedangkan pada *class model* terdiri dari lima *class* yaitu *user\_model*, *main\_model*, *M\_Pengacara*, *M\_Klien*, dan *M\_admin*. Pada gambar 5.5 merupakan gambar dari perancangan *class diagram* umum pada sistem ini.





**Gambar 5.5 Class Diagram Umum**

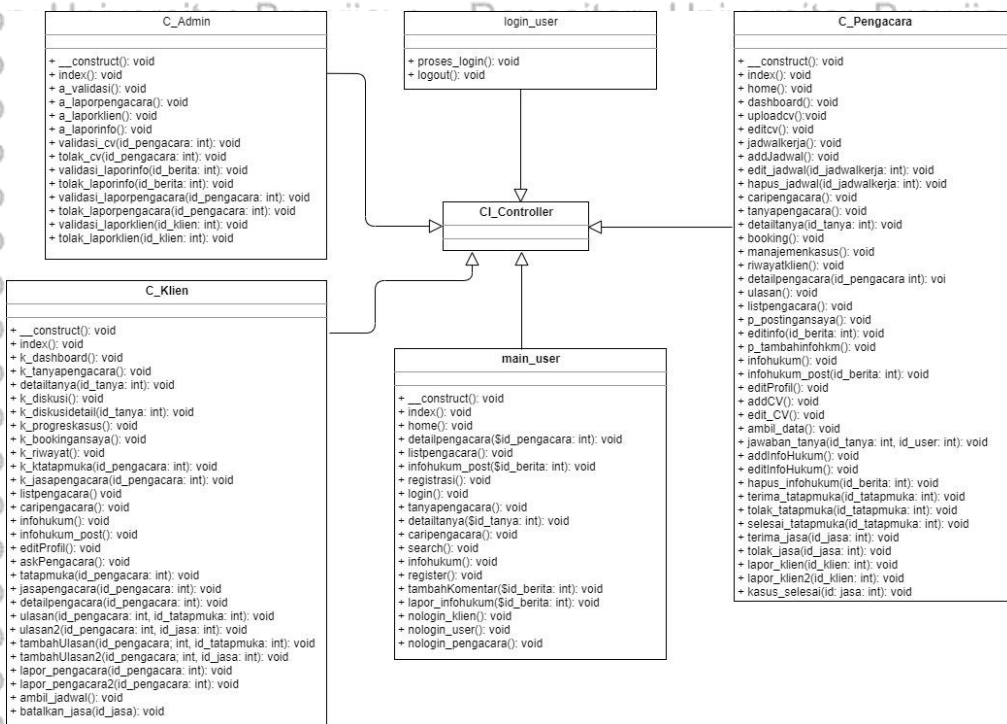
## 2. Perancangan *Class Diagram* Detail

Perancangan *class diagram* detail bertujuan untuk menggambarkan struktur dari *class* seperti atribut dan operasi yang ada dalam suatu *class*. Perancangan *class diagram* pada Sistem Konsultasi dan Pencarian Pengacara akan dijelaskan menjadi dua kelas, yaitu kelas *controller* dan kelas *model*.

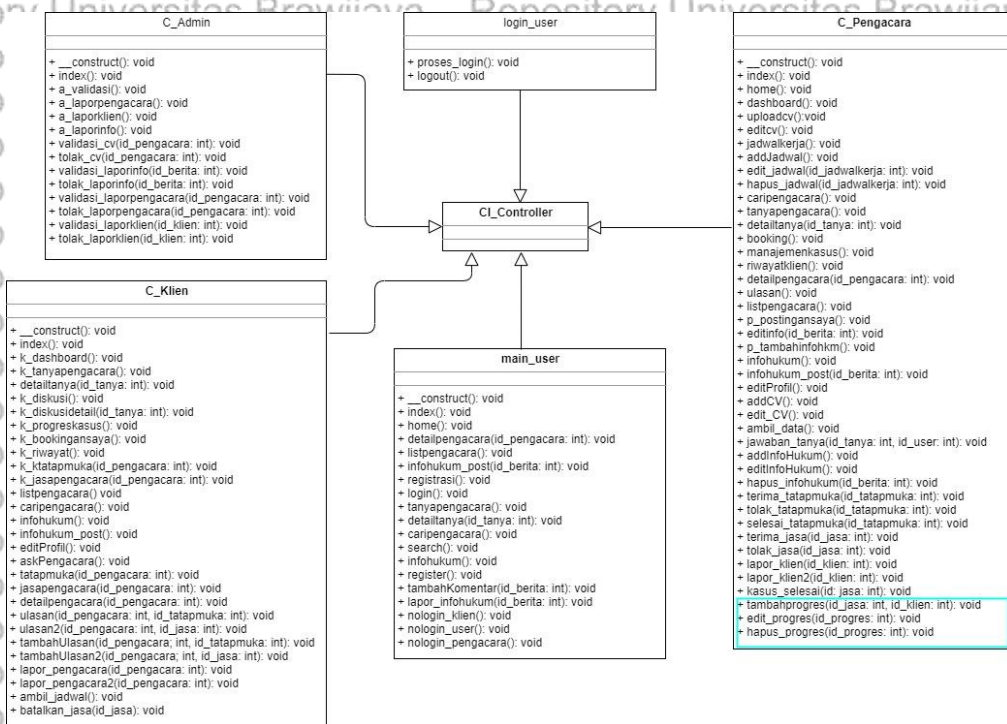
### a) *Class Diagram Controller*

Pada perancangan *class diagram controller* pada Sistem Konsultasi dan Pencarian Pengacara terdiri dari *class diagram controller* iterasi pertama dan iterasi kedua. *Class diagram controller* iterasi pertama digambarkan pada Gambar 5.6 terdapat lima kelas *controller* antara lain *login\_user*, *main\_user*, *C\_Admin*, *C\_Pengacara*, dan *C\_Klien*.





Gambar 5.6 Class Diagram Controller Iterasi 1



Gambar 5.7 Class Diagram Controller Iterasi 2

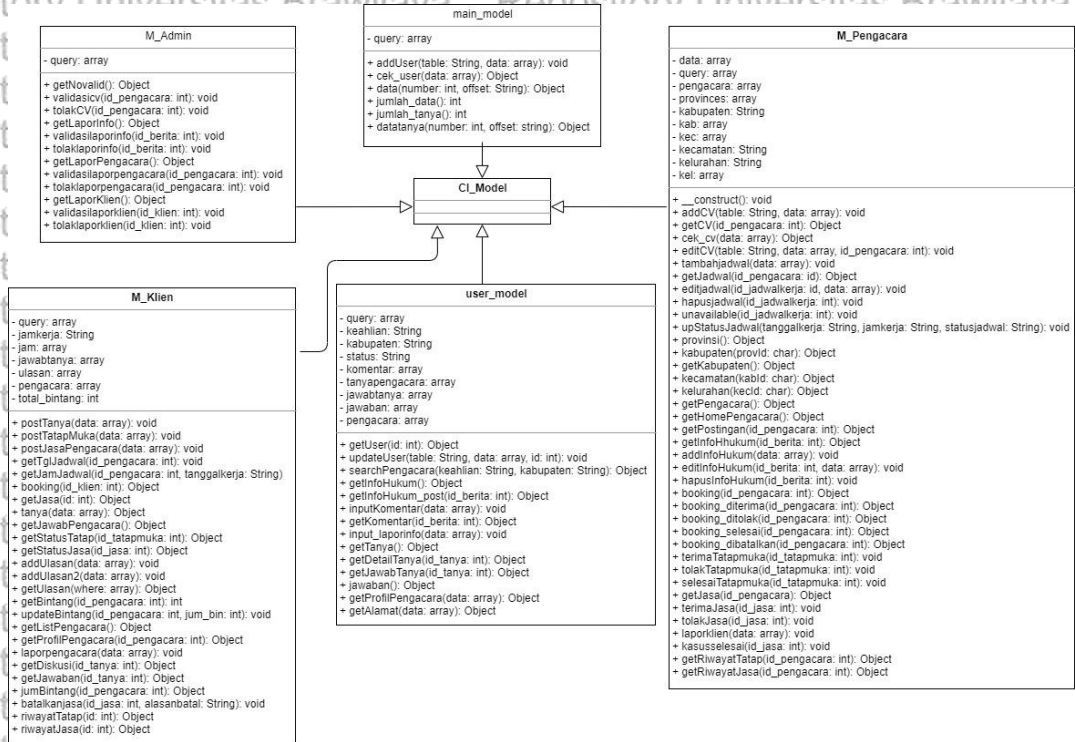
Class diagram controller iterasi kedua digambarkan pada Gambar 5.7 terdiri dari kelas yang sama seperti class diagram controller iterasi pertama. Tapi, mempunyai tambahan method pada kelas C\_Pengacara yaitu method tambahprogres(id\_jasa, id\_klien) untuk menambahkan progres kasus,



method `edit_progres(id_progres)` untuk mengedit progres kasus, dan method `hapus_progres($id_progres)` untuk menghapus progres kasus.

**b) Class Diagram Model**

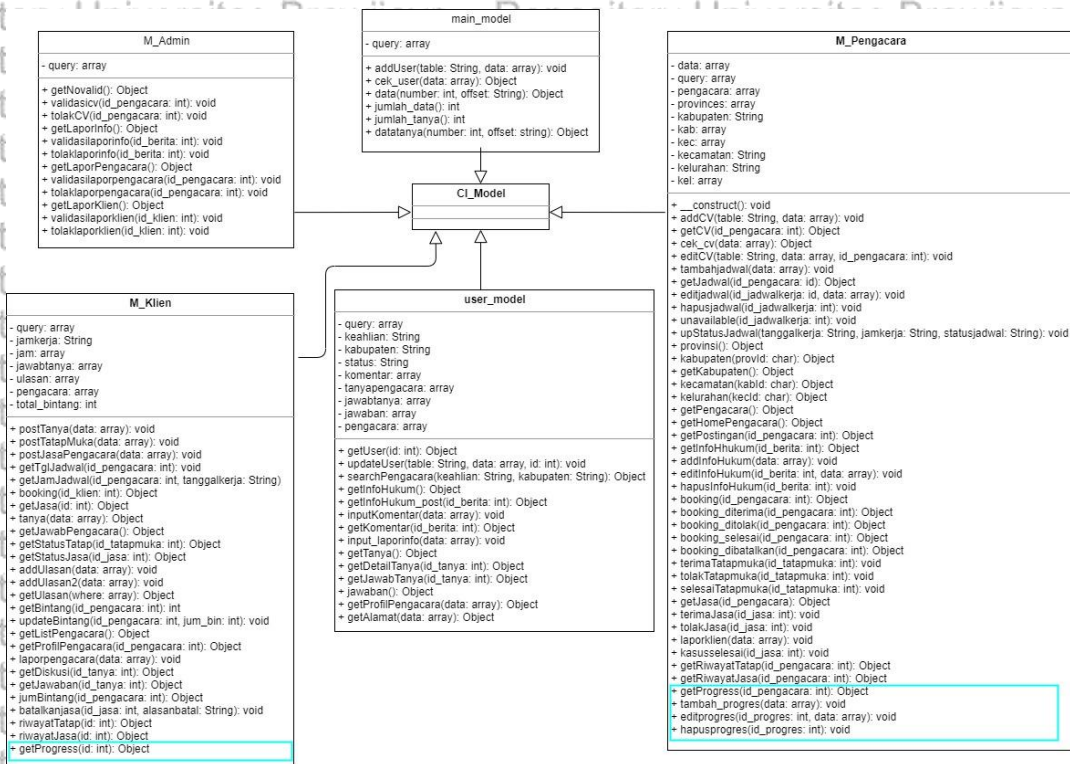
Pada perancangan *class diagram* model pada Sistem Konsultasi dan Pencarian Pengacara terdiri dari *class diagram model* iterasi pertama dan *class diagram model* iterasi kedua. *Class diagram model* iterasi pertama digambarkan pada Gambar 5.8 terdapat lima kelas model antara lain `main_model`, `user_model`, `M_Admin`, `M_Pengacara`, dan `M_Klien`.



**Gambar 5.8 Class Diagram Model Iterasi 1**

*Class diagram model* iterasi kedua digambarkan pada Gambar 5.9 terdiri dari kelas yang sama seperti *class diagram model* iterasi pertama. Tapi, mempunyai tambahan method pada kelas `M_Klien` yaitu method `getProgress(id)` untuk menampilkan progres kasus klien dan pada kelas `M_Pengacara` yaitu method `getProgress(id pengacara)` untuk menampilkan progres kasus klien, method `tambah_progres(data)` untuk menambahkan progres kasus klien, method `editprogres(id_progres)` untuk mengedit progres kasus klien, dan method `hapusprogres(id_progres)` untuk menghapus progres kasus klien.





Gambar 5.9 Class Diagram Model Iterasi 2

### 5.1.3 Perancangan Basis Data

Perancangan basis data merupakan proses untuk menggambarkan basis data yang digunakan untuk mendukung operasi dari sistem. Perancangan basis data pada Sistem Konsultasi dan Pencarian Pengacara terdapat dua pemodelan yaitu *Entity Relationship Diagram* (ERD) dan *Physical Data Model* (PDM).

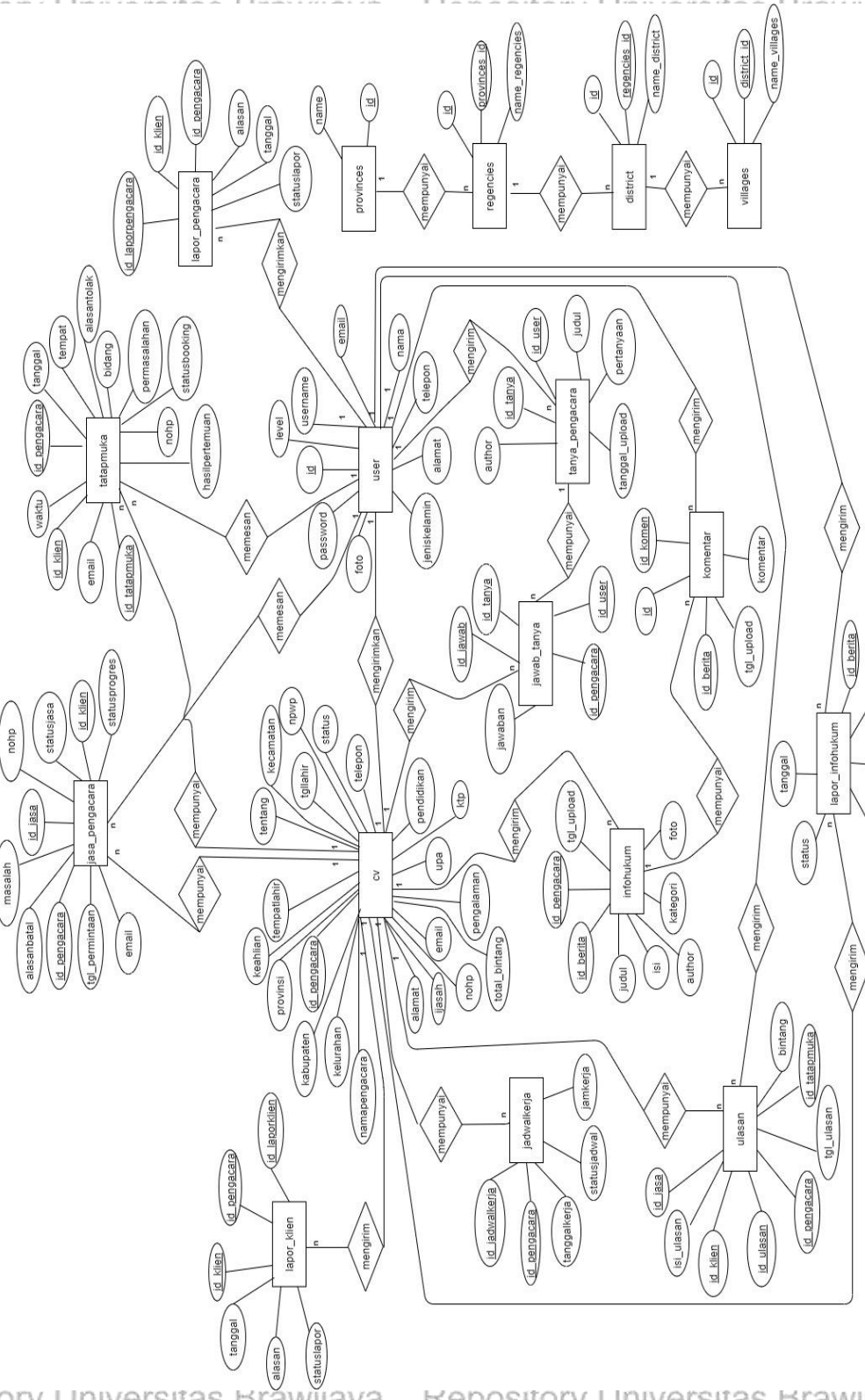
#### 5.1.3.1 Entity Relationship Diagram (ERD)

*Entity Relationship Diagram* (ERD) merupakan diagram yang digunakan untuk menggambarkan hubungan dari tiap entitas beserta relasinya yang ada pada sistem. Pada ERD sistem pencarian dan konsultasi pengacara terbagi menjadi ERD iterasi pertama dan ERD iterasi kedua.

##### 1. Entity Relationship Diagram (ERD) Iterasi 1

Perancangan ERD iterasi 1 dibagi menjadi 17 entitas yaitu entitas user, cv, tanya\_pengacara, jawab\_tanya, infohukum, komentar, jadwalkerja, jasa\_pengacara, district, provinces, regencies, villages, lapor\_infohukum, lapor\_klien, lapor\_pengacara, tatapmuka, dan ulasan. Pada Gambar 5.10 merupakan gambar *Entity Relationship Diagram* (ERD) dari iterasi pertama.





Gambar 5.10 Entity Relationship Diagram (ERD) Iterasi 1





#### a. Entitas User

Entitas user berfungsi untuk menyimpan data dari semua aktor yaitu klien, pengacara, dan admin. Data ini berguna untuk aktor melakukan login pada sistem. Berikut ini penjelasan atribut dari entitas user pada tabel 5.1.

**Tabel 5.1 Entitas User**

| No | Nama Atribut | Tipe    | Panjang Nilai | Keterangan  |
|----|--------------|---------|---------------|---|
| 1  | id           | Int     | 11            | Merupakan id dari user yang juga merupakan <i>primary key</i> . |
| 2  | username     | Varchar | 50            | Berisi username aktor.  |
| 3  | email        | Varchar | 255           | Berisi email aktor.   |
| 4  | password     | Varchar | 255           | Berisi password aktor.  |
| 5  | nama         | Varchar | 255           | Berisi nama aktor.  |
| 6  | telepon      | Varchar | 255           | Berisi telepon aktor.   |
| 7  | alamat       | Varchar | 255           | Berisi alamat aktor.  |
| 8  | foto         | Varchar | 255           | Berisi foto aktor.  |
| 9  | level        | Varchar | 225           | Berisi level aktor.   |
| 10 | jeniskelamin | Varchar | 255           | Berisi jenis kelamin aktor.                                     |

#### b. Entitas cv

Entitas cv berfungsi untuk menyimpan data dari cv pengacara yang mendaftarkan ke dalam sistem. Data ini berguna untuk memvalidasi apakah aktor yang mendaftarkan diri ke sistem benar-benar berprofesi sebagai pengacara. Berikut ini penjelasan atribut dari entitas cv pada tabel 5.2.

**Tabel 5.2 Entitas cv**

| No | Nama Atribut  | Tipe    | Panjang Nilai | Keterangan   |
|----|---------------|---------|---------------|--|
| 1  | id_pengacara  | Int     | 11            | Merupakan id dari pengacara yang juga merupakan <i>primary key</i> . |
| 2  | namapengacara | Varchar | 50            | Berisi nama panjang pengacara.                                       |
| 3  | tempatlahir   | Varchar | 255           | Berisi tempat lahir pengacara.                                       |
| 4  | tglahir       | Varchar | 255           | Berisi tanggal lahir pengacara.                                      |





| No | Nama Atribut  | Tipe    | Panjang Nilai | Keterangan  |
|----|---------------|---------|---------------|---|
| 5  | npwp          | Varchar | 255           | Berisi npwp pengacara.                                    |
| 6  | alamat        | Varchar | 255           | Berisi alamat lengkap pengacara.                          |
| 7  | provinsi      | Varchar | 255           | Berisi provinsi pengacara.                                |
| 8  | kabupaten     | Varchar | 255           | Berisi kabupaten pengacara.                               |
| 9  | kecamatan     | Varchar | 255           | Berisi kecamatan pengacara.                               |
| 10 | kelurahan     | Varchar | 255           | Berisi kelurahan pengacara.                               |
| 11 | email         | Varchar | 255           | Berisi email pengacara.                                   |
| 12 | telepon       | Int     | 11            | Berisi telepon pengacara.                                 |
| 13 | nohp          | Int     | 11            | Berisi nohp pengacara.                                    |
| 14 | pendidikan    | Varchar | 225           | Berisi pendidikan aktor.                                  |
| 15 | pengalaman    | Varchar | 255           | Berisi pengalaman pengacara.                              |
| 16 | tentang       | Varchar | 255           | Berisi tentang pengacara.                                 |
| 17 | keahlian      | Varchar | 255           | Berisi keahlian pengacara.                                |
| 18 | ktp           | Varchar | 255           | Berisi <i>file</i> ktp pengacara.                         |
| 19 | ijasah        | Varchar | 255           | Berisi <i>file</i> ijasah pengacara.                      |
| 20 | upa           | Varchar | 255           | Berisi <i>file</i> upa (ujian profesi advokat) pengacara. |
| 21 | total_bintang | Int     | 11            | Berisi jumlah bintang yang didapatkan pengacara.          |
| 22 | status        | Varchar | 255           | Berisi status dari pengacara.                             |

c. Entitas tanya\_pengacara

Entitas tanya\_pengacara berfungsi untuk menyimpan data dari Tanya pengacara. Berikut ini penjelasan atribut dari entitas tanya\_pengacara pada tabel 5.3.





Tabel 5.3 Entitas tanya\_pengacara

| No | Nama Atribut   | Tipe     | Panjang Nilai | Keterangan  |
|----|----------------|----------|---------------|---|
| 1  | id_tanya       | Int      | 11            | Merupakan id dari pertanyaan yang juga merupakan <i>primary key</i> . |
| 2  | id_user        | Int      | 11            | Merupakan id dari user yang bertanya.                                 |
| 3  | judul          | Varchar  | 255           | Berisi judul pertanyaan.  |
| 4  | pertanyaan     | Varchar  | 255           | Berisi pertanyaan.  |
| 5  | author         | Varchar  | 255           | Berisi author dari pertanyaan.  |
| 6  | tanggal_upload | Datetime | -             | Berisi tanggal upload pertanyaan.                                     |

## d. Entitas jawab\_tanya

Entitas jawab\_tanya berfungsi untuk menyimpan data dari jawaban pertanyaan dari Tanya pengacara. Berikut ini penjelasan atribut dari entitas jawab\_tanya pada tabel 5.4.

Tabel 5.4 Entitas jawab\_tanya

| No | Nama Atribut | Tipe    | Panjang Nilai | Keterangan   |
|----|--------------|---------|---------------|--|
| 1  | id_jawab     | Int     | 11            | Merupakan id dari jawaban Tanya pengacara yang juga merupakan <i>primary key</i> . |
| 2  | id_tanya     | Int     | 11            | Berisi id dari tanya pengacara.  |
| 3  | id_user      | Int     | 11            | Berisi id dari user.   |
| 4  | id_pengacara | Int     | 11            | Berisi id dari pengacara.  |
| 5  | jawaban      | Varchar | 255           | Berisi jawaban tanya pengacara.  |

## e. Entitas infohukum

Entitas infohukum berfungsi untuk menyimpan data dari info hukum yang diinputkan oleh pengacara. Berikut ini penjelasan atribut dari entitas infohukum pada tabel 5.5.





Tabel 5.5 Entitas infohukum

| No | Nama Atribut | Tipe     | Panjang Nilai | Keterangan  |
|----|--------------|----------|---------------|---|
| 1  | id_berita    | Int      | 11            | Merupakan id dari berita yang juga merupakan <i>primary key</i> . |
| 2  | id_pengacara | Varchar  | 50            | Merupakan id dari pengacara yang telah membuat info hukum.        |
| 3  | judul        | Varchar  | 255           | Berisi judul dari info hukum.                                     |
| 4  | isi          | Longtext | -             | Berisi isi info hukum.  |
| 5  | kategori     | Varchar  | 255           | Berisi kategori info hukum.                                       |
| 6  | foto         | Varchar  | 255           | Berisi foto infohukum.  |
| 7  | author       | Varchar  | 255           | Berisi nama author infohukum.                                     |
| 8  | tgl_upload   | datetime | -             | Berisi tanggal upload infohukum.                                  |

## f. Entitas komentar

Entitas komentar berfungsi untuk menyimpan data dari komentar pada info hukum. Berikut ini penjelasan atribut dari entitas komentar pada tabel 5.6.

Tabel 5.6 Entitas komentar

| No | Nama Atribut | Tipe      | Panjang Nilai | Keterangan   |
|----|--------------|-----------|---------------|--|
| 1  | id_komen     | Int       | 11            | Merupakan id dari komen yang juga merupakan <i>primary key</i> . |
| 2  | id           | Int       | 11            | Berisi id dari <i>user</i> yang melakukan komen.                 |
| 3  | id_berita    | Int       | 11            | Berisi id dari berita yang diberikan komentar.                   |
| 4  | komentar     | Varchar   | 255           | Berisi komentar.   |
| 5  | tgl_upload   | Timestamp | -             | Berisi tanggal upload komentar.                                  |





g. Entitas jadwalkerja

Entitas jadwalkerja berfungsi untuk menyimpan data dari jadwal kerja pengacara. Berikut ini penjelasan atribut dari entitas jadwalkerja pada tabel 5.7.

**Tabel 5.7 Entitas jadwalkerja**

| No | Nama Atribut       | Tipe    | Panjang Nilai | Keterangan  |
|----|--------------------|---------|---------------|---|
| 1  | id_jadwalkerj<br>a | Int     | 11            | Merupakan id dari jadwal kerja yang juga merupakan <i>primary key</i> . |
| 2  | Id_pengacara       | Int     | 11            | Berisi id dari pengacara yang menambahkan jadwal kerja.                 |
| 3  | tanggalkerja       | Varchar | 50            | Berisi tanggal kerja pengacara.   |
| 4  | jamkerja           | Varchar | 100           | Berisi jam kerja sesuai tanggal kerja pengacara.                        |
| 5  | statusjadwal       | Varchar | 100           | Berisi status jadwal pengacara.   |

h. Entitas jasa\_pengacara

Entitas jasa\_pengacara berfungsi untuk menyimpan data dari layanan gunakan jasa pengacara. Berikut ini penjelasan atribut dari entitas jasa\_pengacara pada tabel 5.8.

**Tabel 5.8 Entitas jasa\_pengacara**

| No | Nama Atribut | Tipe    | Panjang Nilai | Keterangan  |
|----|--------------|---------|---------------|---|
| 1  | id_jasa      | Int     | 11            | Merupakan id dari jasa pengacara yang juga merupakan <i>primary key</i> . |
| 2  | id_pengacara | Int     | 11            | Berisi id dari pengacara yang jasanya digunakan oleh klien.               |
| 3  | id_klien     | Int     | 11            | Berisi id dari klien yang memesan layanan jasa pengacara.                 |
| 4  | masalah      | Varchar | 255           | Berisi masalah yang akan diajukan oleh klien ke pengacara.                |





| No | Nama Atribut   | Tipe      | Panjang Nilai | Keterangan   |
|----|----------------|-----------|---------------|--|
| 5  | nohp           | Varchar   | 50            | Berisi nomer <i>handphone</i> yang dimiliki klien. |
| 6  | email          | Varchar   | 50            | Berisi email dari klien.                           |
| 7  | tgl_permintaan | timestamp | -             | Berisi tanggal dari gunakan jasa pengacara.        |
| 8  | statusjasa     | Varchar   | 50            | Berisi status dari jasa pengacara.                 |
| 9  | statusprogres  | Varchar   | 50            | Berisi status progres dari jasa pengacara.         |
| 10 | alasanbatal    | Varchar   | 255           | Berisi alasan membatalkan jasa pengacara.          |

#### i. Entitas *district*

Entitas *district* berfungsi untuk menampilkan daftar-daftar kecamatan yang ada di Indonesia. Berikut ini penjelasan atribut dari entitas *district* pada tabel 5.9.

**Tabel 5.9 Entitas *district***

| No | Nama Atribut  | Tipe    | Panjang Nilai | Keterangan   |
|----|---------------|---------|---------------|--|
| 1  | id            | Char    | 7             | Merupakan id dari kecamatan yang juga merupakan <i>primary key</i> . |
| 2  | regency_id    | Char    | 4             | Berisi id dari kabupaten.  |
| 3  | name_district | Varchar | 255           | Berisi nama dari kecamatan.  |

#### j. Entitas *provinces*

Entitas *provinces* berfungsi untuk menampilkan daftar-daftar provinsi yang ada di Indonesia. Berikut ini penjelasan atribut dari entitas *provinces* pada tabel 5.10.

**Tabel 5.10 Entitas *provinces***

| No | Nama Atribut | Tipe    | Panjang Nilai | Keterangan  |
|----|--------------|---------|---------------|---|
| 1  | id           | Char    | 2             | Merupakan id dari provinsi yang juga merupakan <i>primary key</i> . |
| 2  | name         | Varchar | 255           | Berisi nama dari provinsi.  |





k. Entitas *regencies*

Entitas *regencies* berfungsi untuk menampilkan daftar-daftar kabupaten yang ada di Indonesia. Berikut ini penjelasan atribut dari entitas *regencies* pada tabel 5.11

**Tabel 5.11 Entitas *regencies***

| No | Nama Atribut   | Tipe    | Panjang Nilai | Keterangan   |
|----|----------------|---------|---------------|--|
| 1  | id             | Char    | 4             | Merupakan id dari kabupaten yang juga merupakan <i>primary key</i> . |
| 2  | provinces_id   | Char    | 2             | Berisi id dari provinsi.   |
| 3  | name_regencies | Varchar | 255           | Berisi nama dari provinsi.   |

l. Entitas *villages*

Entitas *villages* berfungsi untuk menampilkan daftar-daftar desa yang ada di Indonesia. Berikut ini penjelasan atribut dari entitas *villages* pada tabel 5.12.

**Tabel 5.12 Entitas *villages***

| No | Nama Atribut | Tipe    | Panjang Nilai | Keterangan  |
|----|--------------|---------|---------------|---|
| 1  | id           | Char    | 10            | Merupakan id dari desa yang juga merupakan <i>primary key</i> . |
| 2  | district_id  | Char    | 7             | Berisi id dari kecamatan.                                       |
| 3  | name_village | Varchar | 255           | Berisi nama dari desa.  |

m. Entitas lapor\_infohukum

Entitas lapor\_infohukum berfungsi untuk menyimpan data dari infohukum yang dilaporkan oleh klien atau pengacara. Berikut ini penjelasan atribut dari entitas lapor\_infohukum pada tabel 5.13.

**Tabel 5.13 Entitas lapor\_infohukum**

| No | Nama Atribut      | Tipe | Panjang Nilai | Keterangan  |
|----|-------------------|------|---------------|---|
| 1  | id_laporinfohukum | Int  | 11            | Merupakan id dari laporan info hukum yang juga merupakan <i>primary key</i> . |
| 2  | id_pengguna       | Int  | 11            | Berisi id dari pengguna yang melaporkan info hukum.                           |





| No | Nama Atribut | Tipe      | Panjang Nilai | Keterangan                                  |
|----|--------------|-----------|---------------|---|
| 3  | id_berita    | Int       | 11            | Berisi id dari info hukum yang dilaporkan   |
| 4  | alasan       | Varchar   | 255           | Berisi alasan mengapa info hukum dilaporkan |
| 5  | tanggal      | timestamp | -             | Berisi tanggal dari pelaporan info hukum.   |
| 6  | status       | Varchar   | 100           | Berisi status info hukum yang dilaporkan.   |

#### n. Entitas lapor\_klien

Entitas lapor\_klien berfungsi untuk menyimpan data dari klien yang dilaporkan oleh pengacara. Berikut ini penjelasan atribut dari entitas lapor\_klien pada tabel 5.14.

**Tabel 5.14 Entitas lapor\_klien**

| No | Nama Atribut  | Tipe      | Panjang Nilai | Keterangan   |
|----|---------------|-----------|---------------|--|
| 1  | id_laporklien | Int       | 11            | Merupakan id dari lapor klien yang juga merupakan <i>primary key</i> . |
| 2  | id_pengacara  | Int       | 11            | Berisi id dari pengacara yang melaporan klien.                         |
| 3  | id_klien      | Int       | 11            | Berisi id dari klien yang dilaporkan oleh pengacara.                   |
| 4  | alasan        | Varchar   | 255           | Berisi alasan mengapa klien dilaporkan.                                |
| 5  | tanggal       | timestamp | -             | Berisi tanggal dari pelaporan klien.                                   |
| 6  | statuslapor   | Varchar   | 100           | Berisi status klien yang dilaporkan.                                   |

#### o. Entitas lapor\_pengacara

Entitas lapor\_pengacara berfungsi untuk menyimpan data dari pengacara yang dilaporkan oleh klien. Berikut ini penjelasan atribut dari entitas lapor\_pengacara pada tabel 5.15.





Tabel 5.15 Entitas lapor\_pengacara

| No | Nama Atribut      | Tipe      | Panjang Nilai | Keterangan   |
|----|-------------------|-----------|---------------|--|
| 1  | id_laporpengacara | Int       | 11            | Merupakan id dari lapor pengacara yang juga merupakan <i>primary key</i> . |
| 2  | id_pengacara      | Int       | 11            | Berisi id dari pengacara yang dilaporkan klien.                            |
| 3  | id_klien          | Int       | 11            | Berisi id dari klien yang melaporkan pengacara.                            |
| 4  | alasan            | Varchar   | 255           | Berisi alasan mengapa pengacara dilaporkan.                                |
| 5  | tanggal           | timestamp | -             | Berisi tanggal dari pelaporan pengacara.                                   |
| 6  | statuslapor       | Varchar   | 100           | Berisi status pengacara yang dilaporkan.                                   |

## p. Entitas tatapmuka

Entitas tatapmuka berfungsi untuk menyimpan data dari konsultasi tatap muka yang dipesan klien. Berikut ini penjelasan atribut dari entitas tatapmuka pada tabel 5.16.

Tabel 5.16 Entitas tatapmuka

| No | Nama Atribut | Tipe    | Panjang Nilai | Keterangan   |
|----|--------------|---------|---------------|--|
| 1  | id_tatapmuka | Int     | 11            | Merupakan id dari lapor pengacara yang juga merupakan <i>primary key</i> . |
| 2  | id_pengacara | Int     | 11            | Berisi id dari pengacara yang dipesan oleh klien.                          |
| 3  | id_klien     | Int     | 11            | Berisi id dari klien yang memesan tatapmuka.                               |
| 4  | Bidang       | Varchar | 255           | Berisi bidang yang menjadi topic permasalahan klien.                       |
| 5  | tanggal      | Varchar | 255           | Berisi tanggal dari pertemuan tatap muka.                                  |
| 6  | Waktu        | Varchar | 255           | Berisi waktu dari pertemuan tatap muka.                                    |
| 7  | nohp         | Varchar | 50            | Berisi nohp dari klien.  |





| No | Nama Atribut   | Tipe     | Panjang Nilai | Keterangan                                     |
|----|----------------|----------|---------------|--|
| 8  | email          | Varchar  | 255           | Berisi email dari klien.                       |
| 9  | tempat         | Varchar  | 255           | Berisi tempat untuk bertemu.                   |
| 10 | permasalahan   | Longtext | -             | Berisi permasalahan klien.                     |
| 11 | hasilpertemuan | Varchar  | 255           | Berisi berita acara dari pertemuan tatap muka. |
| 12 | alasantolak    | Varchar  | 255           | Berisi alasan menolak konsultasi tatap muka.   |
| 13 | statusbookin   | Varchar  | 255           | Berisi status dari tatap muka.                 |

#### q. Entitas ulasan

Entitas ulasan berfungsi untuk menyimpan data dari ulasan yang diberikan klien untuk pengacara. Berikut ini penjelasan atribut dari entitas ulasan pada tabel 5.17.

**Tabel 5.17 Entitas ulasan**

| No | Nama Atribut | Tipe      | Panjang Nilai | Keterangan  |
|----|--------------|-----------|---------------|---|
| 1  | id_ulasan    | Int       | 11            | Merupakan id dari ulasan yang juga merupakan <i>primary key</i> . |
| 2  | id_pengacara | Int       | 11            | Berisi id dari pengacara yang diulas oleh klien.                  |
| 3  | id_klien     | Int       | 11            | Berisi id dari klien yang mengulas pengacara.                     |
| 4  | id_tatapmuka | Int       | 11            | Berisi id dari layanan tatap muka.                                |
| 5  | id_jasa      | Int       | 11            | Berisi id dari layanan jasa pengacara.                            |
| 6  | tgl_ulasan   | timestamp | -             | Berisi tanggal dimana ulasan dikirimkan.                          |
| 7  | bintang      | Int       | 5             | Berisi bintang yang diberikan kepada pengacara.                   |
| 8  | isi_ulasan   | Varchar   | 255           | Berisi ulasan dari klien.   |





## 2. Entity Relationship Diagram (ERD) Iterasi 2

Perancangan ERD iterasi 2 terdapat penambahan entitas yaitu entitas *progres\_kasus*. Pada gambar 5.11 merupakan gambar *Entity Relationship Diagram* (ERD) dari iterasi kedua.

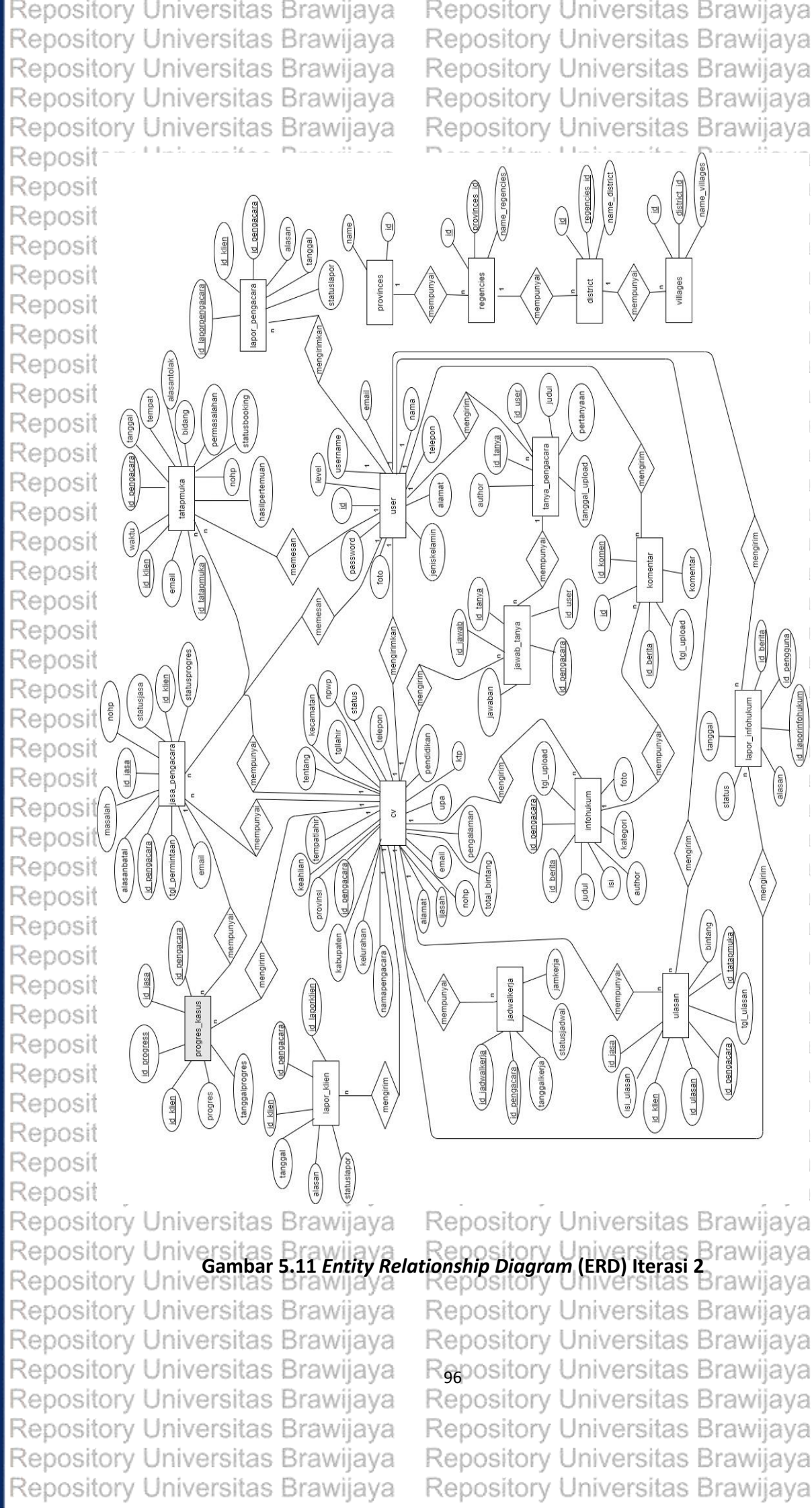
### a. Entitas *progres\_kasus*

Entitas *progres\_kasus* berfungsi untuk menyimpan data dari progres kasus klien. Berikut ini penjelasan atribut dari entitas *progres\_kasus* pada tabel 5.18.

**Tabel 5.18 Entitas *progres\_kasus***

| No | Nama Atribut               | Tipe    | Panjang Nilai | Keterangan   |
|----|----------------------------|---------|---------------|--|
| 1  | <i>id_progres</i>          | Int     | 11            | Merupakan id dari progres yang juga merupakan <i>primary key</i> . |
| 2  | <i>id_jasa</i>             | Int     | 11            | Berisi id dari jasa pengacara.                                     |
| 3  | <i>id_pengacara</i>        | Int     | 11            | Berisi id dari pengacara.  |
| 4  | <i>id_klien</i>            | Int     | 11            | Berisi id dari klien.  |
| 5  | <i>tanggalprogr<br/>es</i> | Varchar | 100           | Berisi tanggal dari progres jasa pengacara.                        |
| 6  | <i>progres</i>             | Varchar | 255           | Berisi progres dari jasa pengacara.                                |





Gambar 5.11 Entity Relationship Diagram (ERD) Iterasi 2



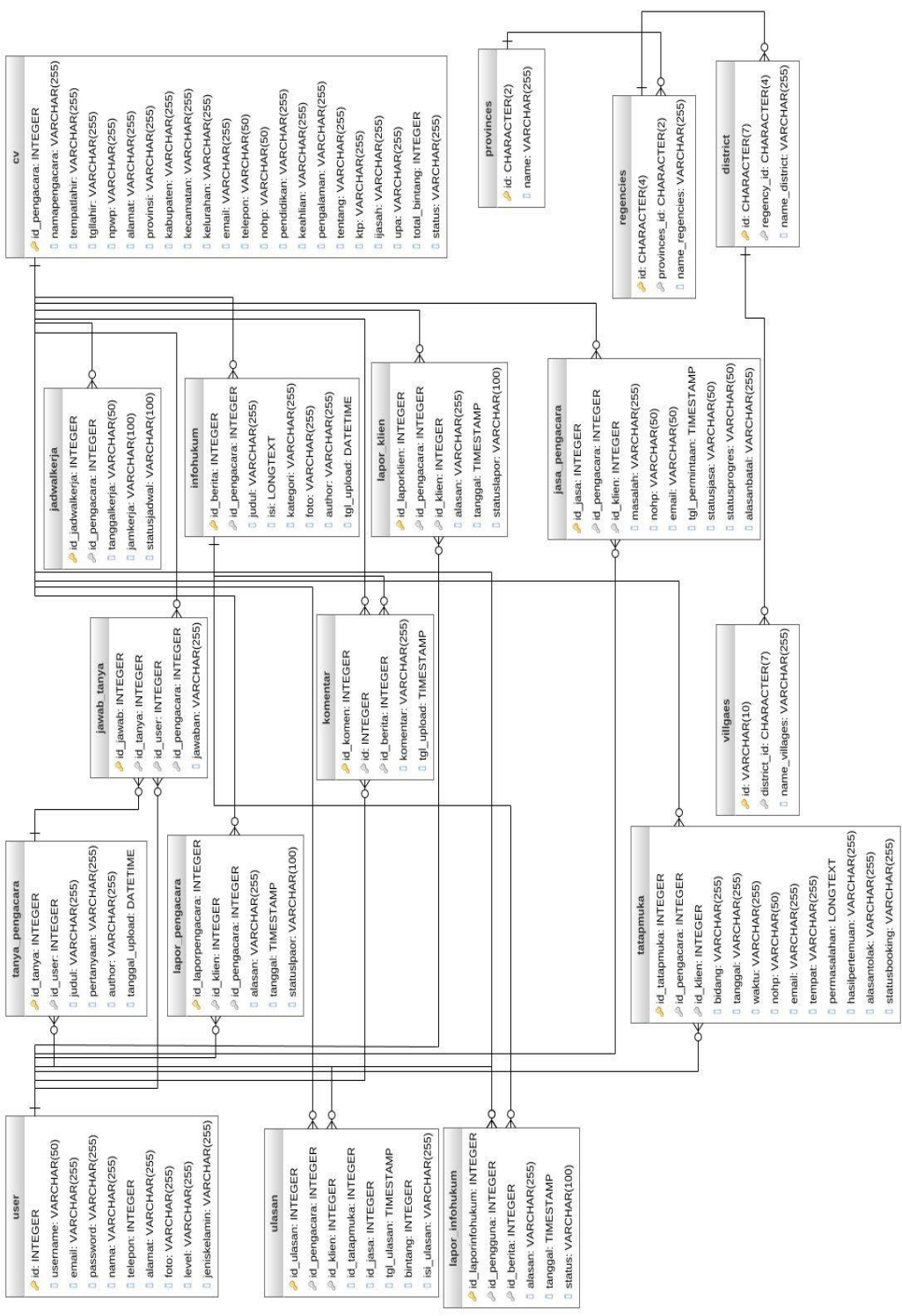


### 5.1.3.2 Physical Data Model (PDM)

*Physical Data Model* (PDM) merupakan tahap yang digunakan untuk menggambarkan rancangan basis data dengan tabel yang berisikan atribut seperti nama kolom, tipe data kolom, *primary key* (PK), *foreign key* (FK), dan relasi antar tabel. PDM pada sistem konsultasi dan pencarian pengacara dibagi menjadi dua yaitu PDM iterasi pertama dan PDM iterasi kedua berdasarkan dengan ERD yang sudah dibuat sebelumnya. PDM iterasi pertama terdapat 17 tabel sama seperti ERD iterasi pertama. Tabel-tabel tersebut antara lain tabel cv untuk menyimpan data dari pengacara, tabel user untuk menyimpan data dari pengguna sistem, tabel jasa\_pengacara untuk menyimpan data dari layanan gunakan jasa pengacara, tabel tatapmuka untuk menyimpan data dari layanan konsultasi tatap muka, tabel lapor\_pengacara untuk menyimpan pengacara yang dilaporkan oleh klien, tabel tanya\_pengacara untuk menyimpan pertanyaan yang dikirimkan oleh klien, tabel jawab\_tanya untuk menyimpan jawaban dari pengacara untuk pertanyaan yang dikirimkan klien, tabel komentar untuk menyimpan komentar dari info hukum, tabel infohukum untuk menyimpan info hukum yang ditulis pengacara, tabel lapor\_infohukum untuk menyimpan info hukum yang dilaporkan oleh klien atau pengacara, tabel ulasan untuk menyimpan ulasan dari klien untuk pengacara, tabel jadwalkerja untuk menyimpan jadwal kerja pengacara, tabel lapor\_klien untuk menyimpan klien yang dilaporkan oleh pengacara, tabel provinces untuk menyimpan data semua provinsi di Indonesia, tabel regencies untuk menyimpan data semua kabupaten di Indonesia, tabel district untuk menyimpan data semua kecamatan di Indonesia, dan tabel villages menyimpan data semua desa di Indonesia. PDM iterasi pertama digambarkan pada gambar 5.12.

PDM iterasi kedua sesuai dari ERD iterasi kedua mendapatkan 1 penambahan tabel yaitu tabel progres\_kasus yang digunakan untuk menyimpan progres dari kasus yang ditangani pengacara. PDM iterasi kedua digambarkan pada gambar 5.13.





Gambar 5.12 Physical Data Model (PDM) Iterasi 1









#### 5.1.4 Perancangan Komponen

Perancangan komponen merupakan tahapan untuk menjelaskan mengenai method pada *class diagram* yang sudah dimodelkan pada perancangan *class diagram*. Perancangan komponen ini memiliki tujuan untuk menjabarkan algoritme method pada *class*. Pada perancangan sistem konsultasi dan pencarian pengacara terdapat dua iterasi yaitu perancangan komponen iterasi pertama dan perancangan komponen iterasi 2.

##### 5.1.4.1 Perancangan Komponen Iterasi 1

Pada perancangan komponen iterasi pertama digunakan tiga algoritme *method* yaitu *method* `TambahUlasan(id_pengacara, id_tatapmuka)` untuk menambahkan ulasan, *method* `askpengacara()` untuk mengirimkan pertanyaan, dan *method* `search()` untuk mencari pengacara.

##### 1. Perancangan Komponen Tambah Ulasan

Nama Kelas : C\_Klien  
 Nama Method : TambahUlasan(id\_pengacara, id\_tatapmuka)  
 Deskripsi : Metode ini merupakan *method* yang dijalankan ketika klien memilih Ulasan yang ada pada halaman Bookingan Saya tepatnya pada bookingan yang statusnya sudah selesai.  
 Algoritme :

Tabel 5.19 Algoritme Tambah Ulasan

| No | Algoritme Tambah Ulasan  |
|----|--|
| 1  | MULAI  |
| 2  | Inisialisasi variable <code>id_klien</code> dengan <code>id</code> dari data yang                |
| 3  | sudah disimpan sebelumnya  |
| 4  | Menginputkan bintang dan ulasan untuk pengacara yang   |
| 5  | dipilih  |
| 6  | IF ( <code>bintang == 0</code> )   |
| 7  | Mengirimkan pesan melalui alert "Rating tidak boleh  |
| 8  | kosong"  |
| 9  | ELSE   |
| 10 | Menginisialisasi data dengan array <code>id_pengacara</code> dengan                              |
| 11 | inputan <code>id_pengacara</code> , <code>id_klien</code> dengan inputan <code>id_klien</code> , |
| 12 | <code>id_tatapmuka</code> dengan inputan <code>id_Tatapmuka</code> , bintang                     |
| 13 | dengan inputan bintang, dan isi ulasan dengan inputan  |
| 14 | ulasan   |
| 15 | Memasukkan data ulasan pada database dengan data yang  |
| 16 | ada pada variable array data   |
| 17 | Menginisialisasi variable <code>jum</code> dengan memanggil method                               |
| 18 | <code>getBintang(id_pengacara)</code> pada database  |
| 19 |  |
| 20 |  |
| 21 |  |
| 22 |  |
| 23 |  |
| 24 |  |





|    |   |
|----|---|
| 25 | Memasukkan data update bintang pada database dengan     |
| 26 | data id_pengacara, bintang, dan jum_bin                 |
| 27 |   |
| 28 | Redirect ke halaman detailpengacara dengan id_pengacara |
| 29 | END IF  |
|    | SELESAI   |

## 2. Perancangan Komponen Kirim Pertanyaan

Nama Kelas : C\_Klien

Nama *Method* : askPengacara()

Deskripsi : Metode ini merupakan *method* yang dijalankan ketika klien memilih tombol Tanya Pengacara yang ada pada halaman Tanya Pengacara dan melakukan penambahan data pada *form* tanya

Algoritme :

Tabel 5.20 Algoritme Kirim Pertanyaan

| No | Algoritme Kirim Pertanyaan                                |
|----|---|
| 1  | MULAI   |
| 2  |   |
| 3  | Inisialisasi variable id dengan id dari data yang sudah   |
| 4  | disimpan sebelumnya                                       |
| 5  | Inisialisasi variable username dengan username dari data  |
| 6  | yang sudah disimpan sebelumnya                            |
| 7  |   |
| 8  | Menginputkan field judul dan pertanyaan pada form tanya   |
| 9  | Inisialisasi variable data dengan array yang berisikan    |
| 10 | variable yang ada pada form tanya                         |
| 11 |   |
| 12 | Memasukkan data tanya pengacara pada database dengan data |
| 13 | yang ada pada variable array data                         |
| 14 | Redirect ke halaman k_tanyapengacara                      |
| 15 |   |
| 16 | SELESAI   |

## 3. Perancangan Komponen Cari Pengacara

Nama Kelas : main\_user

Nama *Method* : search()

Deskripsi : Metode ini merupakan *method* yang dijalankan ketika klien mencari pengacara pada halaman cari pengacara.

Algoritme :

Tabel 5.21 Algoritme Cari Pengacara

| No | Algoritme Cari Pengacara |
|----|--------------------------|
|----|--------------------------|





|    |   |
|----|---|
| 1  | MULAI   |
| 2  | Menginputkan field keahlian dan kabupaten           |
| 3  |   |
| 4  | Memasukkan data cari pengacara pada database dengan |
| 5  | parameter keahlian dan kabupaten                    |
| 6  |   |
| 7  | IF (level == klien)                                 |
| 8  | Menampilkan halaman hasilcari pada klien            |
| 9  |   |
| 10 | ELSE IF (level == pegacara)                         |
| 11 | Menampilkan halaman hasilcari pada pengacara        |
| 12 |   |
| 13 | ELSE  |
| 14 | Menampilkan halaman hasilcari                       |
| 15 |   |
| 16 | END IF  |
| 17 | SELESAI   |
| 18 |   |

#### 5.1.4.2 Perancangan Komponen Iterasi 2

Pada perancangan komponen iterasi kedua digunakan algoritme *method* yaitu *method* tambahprogres(id\_jasa, id\_klien) untuk menambahkan progres kasus klien.

Nama Kelas : C\_Pengacara

Nama Method : Tambahprogres(id\_jasa, id\_klien)

Deskripsi : Metode ini merupakan *method* yang dijalankan pengacara untuk menambahkan progres kasus klien.

Algoritme :

**Tabel 5.22 Algoritme Tambah Progres**

| No | Algoritme Tambah Progres   |
|----|--|
| 1  | MULAI  |
| 2  | Menginputkan field tanggal dan progres   |
| 3  |  |
| 4  | Inisialisasi variable id_pengacara dengan id_pengacara yang disimpan pada saat login |
| 5  |  |
| 6  | Inisialisasi variabel data berbentuk array dengan id_jasa                            |
| 7  | inputan id_jasa, id_pengacara inputan id_pengacara,                                  |
| 8  | id_klien inputan id_klien, tanggalprogres inputan tanggal,                           |
| 9  | dan progres inputan progres  |
| 10 |  |
| 11 | Memasukkan data tambah progres pada database dengan data pada variable array data    |
| 12 |  |
| 13 | Menampilkan pesan dalam bentuk flashdata "Progres Berhasil Ditambahkan"              |
| 14 |  |
| 15 |  |
| 16 | Redirect ke halamana manajemenkasus  |
| 17 |  |
| 18 | SELESAI  |





### 5.1.5 Perancangan Antarmuka

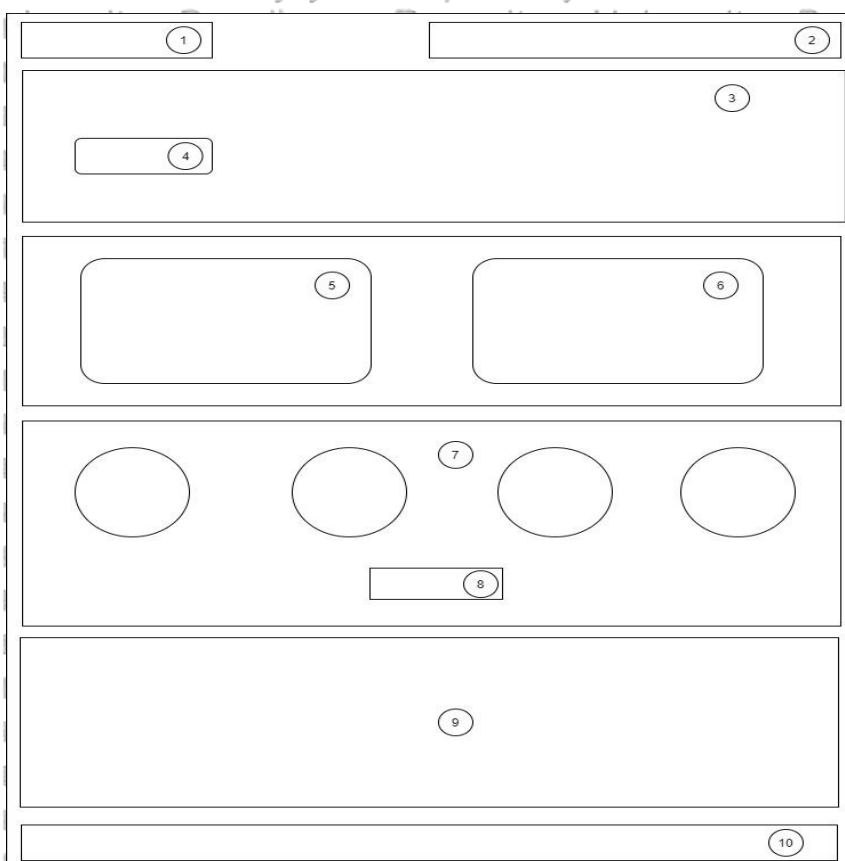
Perancangan antarmuka merupakan perancangan yang menghasilkan tampilan dari sistem yang dibuat untuk mempermudah komunikasi antara sistem dan pengguna. Dalam perancangan ini akan ditunjukkan beberapa rancangan antarmuka sistem diantaranya antarmuka halaman *home*, antarmuka halaman tanya pengacara, antarmuka halaman cari pengacara, antarmuka halaman *dashboard* pengacara, dan antarmuka halaman profil pengacara.

#### 5.1.5.1 Perancangan Antarmuka Iterasi 1

Perancangan antarmuka iterasi pertama pada sistem konsultasi dan pencarian pengacara menggambarkan tampilan dari sistem.

##### 1. Perancangan Antarmuka Halaman *Home*

Perancangan antarmuka halaman *home* digambarkan pada gambar 5.32 dan penjelasan dari perancangan antarmuka halaman *home* dijelaskan pada tabel 5.14.



**Gambar 5.14 Perancangan Antarmuka Halaman *Home***

**Tabel 5.23 Penjelasan Antarmuka Halaman *Home***

| No | Nama Objek | Tipe        | Keterangan                    |
|----|------------|-------------|-------------------------------|
| 1  | Logo       | <i>Link</i> | Menampilkan logo dari sistem. |

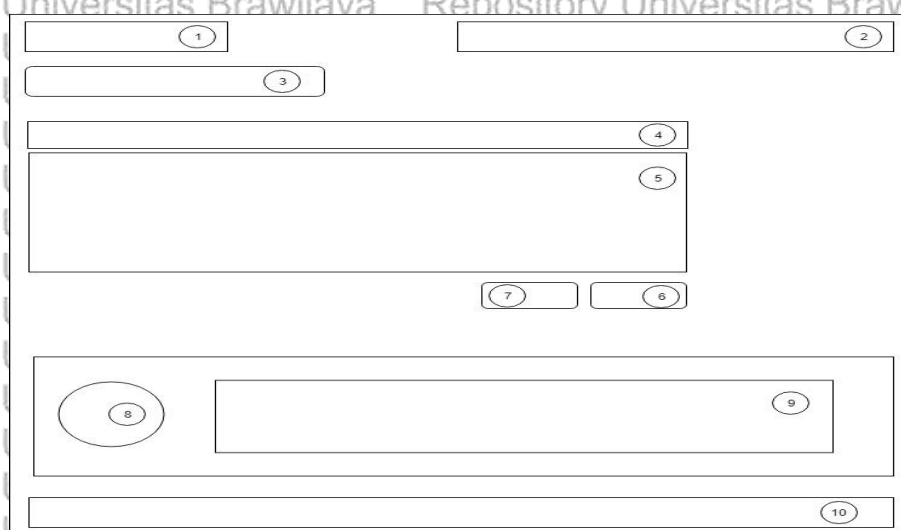




| No | Nama Objek                    | Tipe          | Keterangan   |
|----|-------------------------------|---------------|--|
| 2  | Menu bar                      | <i>Link</i>   | Menampilkan menu utama yang ada pada sistem.                 |
| 3  | Gambar                        | <i>Image</i>  | Menampilkan gambar utama dari sistem.                        |
| 4  | Konsultasi Sekarang!          | <i>Button</i> | Menampilkan tombol untuk mengkonsultasikan permasalahan.     |
| 5  | Layanan Konsultasi Tatap Muka | <i>Link</i>   | Menampilkan layanan konsultasi tatap muka.                   |
| 6  | Layanan pendampingan hukum    | <i>Link</i>   | Menampilkan layanan pendampingan hukum.                      |
| 7  | Daftar Pengacara              | <i>Link</i>   | Menampilkan 4 pengacara dari pengacara yang ada pada sistem. |
| 8  | Lihat Semua                   | <i>Button</i> | Menampilkan tombol untuk melihat daftar pengacara.           |
| 9  | Tentang halopengacara         | <i>Text</i>   | Menampilkan informasi tentang halopengacara                  |
| 10 | Footer                        | <i>Text</i>   | Menampilkan <i>copyright</i> dari sistem.                    |

## 2. Perancangan Antarmuka Halaman Tanya Pengacara

Perancangan antarmuka halaman tanya pengacara digambarkan pada gambar 5.15 dan penjelasan dari perancangan antarmuka halaman tanya pengacara dijelaskan pada tabel 5.24.



**Gambar 5.15 Perancangan Antarmuka Halaman Tanya Pengacara**



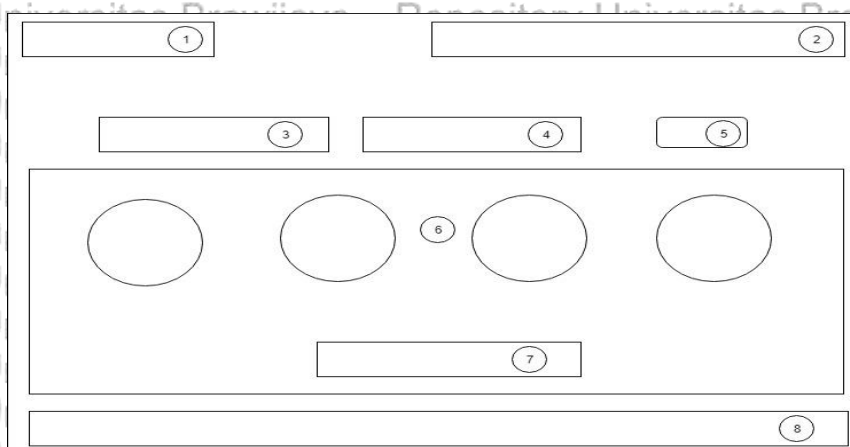


Tabel 5.24 Penjelasan Antarmuka Halaman Tanya Pengacara

| No | Nama Objek        | Tipe           | Keterangan   |
|----|-------------------|----------------|--|
| 1  | Logo              | <i>Link</i>    | Menampilkan logo dari sistem.  |
| 2  | Menu bar          | <i>Link</i>    | Menampilkan menu utama yang ada pada sistem.                         |
| 3  | Tanya Pengacara   | <i>Button</i>  | Menampilkan tombol untuk bertanya kepada pengacara.                  |
| 4  | Judul Topik       | <i>Textbox</i> | Menampilkan <i>textbox</i> untuk mengisi judul topic pertanyaan.     |
| 5  | Pertanyaan        | <i>Textbox</i> | Menampilkan <i>textbox</i> untuk mengisi pertanyaan untuk pengacara. |
| 6  | Batal             | <i>Button</i>  | Menampilkan tombol untuk membatalkan pertanyaan.                     |
| 7  | Kirim             | <i>Button</i>  | Menampilkan tombol untuk mengirimkan pertanyaan.                     |
| 8  | Foto              | <i>Image</i>   | Menampilkan foto dari pengirim pertanyaan terbaru.                   |
| 9  | Detail Pertanyaan | <i>Text</i>    | Menampilkan pertanyaan dari pengirim pertanyaan terbaru.             |
| 10 | <i>Footer</i>     | <i>Text</i>    | Menampilkan <i>copyright</i> dari sistem.                            |

### 3. Perancangan Antarmuka Halaman Cari Pengacara

Perancangan antarmuka halaman cari pengacara digambarkan pada gambar 5.16 dan penjelasan dari perancangan antarmuka halaman cari pengacara dijelaskan pada tabel 5.25.



Gambar 5.16 Perancangan Antarmuka Halaman Cari Pengacara



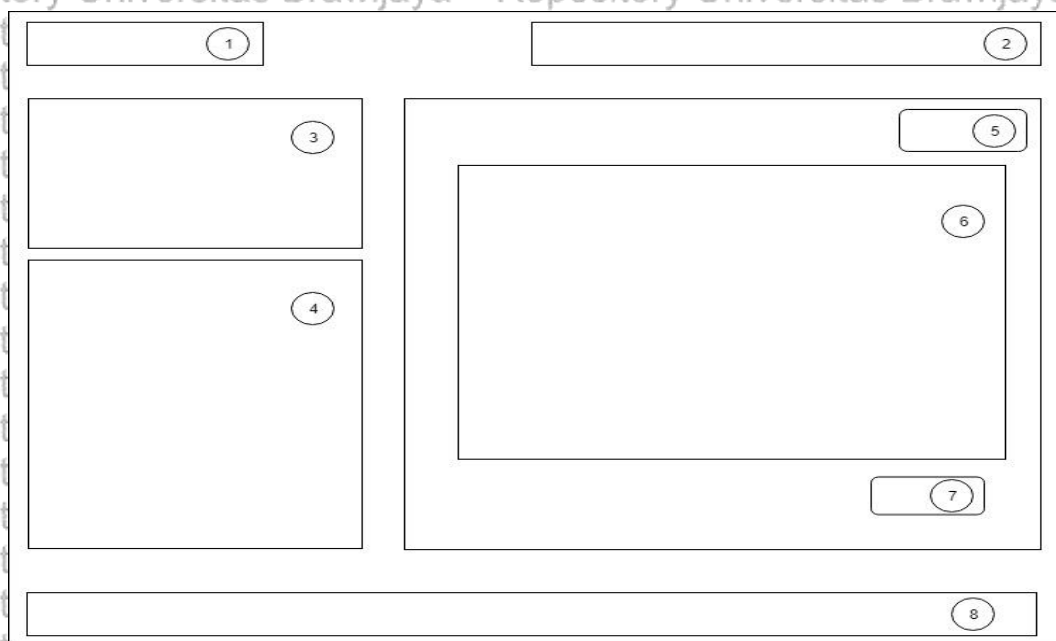


Tabel 5.25 Penjelasan Antarmuka Halaman Cari Pengacara

| No | Nama Objek         | Tipe          | Keterangan  |
|----|--------------------|---------------|---|
| 1  | Logo               | <i>Link</i>   | Menampilkan logo dari sistem.                                       |
| 2  | Menu bar           | <i>Link</i>   | Menampilkan menu utama yang ada pada sistem.                        |
| 3  | Jenis Pengacara    | <i>Select</i> | Menampilkan pilihan untuk memilih jenis pengacara yang akan dicari. |
| 4  | <i>Star Rating</i> | <i>Select</i> | Menampilkan pilihan untuk memilih berapa rating dari pengacara.     |
| 5  | Cari               | <i>Button</i> | Menampilkan tombol untuk mencari pengacara.                         |
| 6  | Daftar Pengacara   | <i>Link</i>   | Menampilkan daftar pengacara yang ada pada sistem.                  |
| 7  | <i>Pagination</i>  | <i>Button</i> | Menampilkan paginasi untuk ke halaman selanjutnya.                  |
| 8  | <i>Footer</i>      | <i>Text</i>   | Menampilkan <i>copyright</i> dari sistem.                           |

#### 4. Perancangan Antarmuka Halaman Lihat Profil

Perancangan antarmuka halaman lihat profil digambarkan pada gambar 5.17 dan penjelasan dari perancangan antarmuka halaman melihat profil dijelaskan pada tabel 5.26.



Gambar 5.17 Perancangan Antarmuka Halaman Lihat Profil



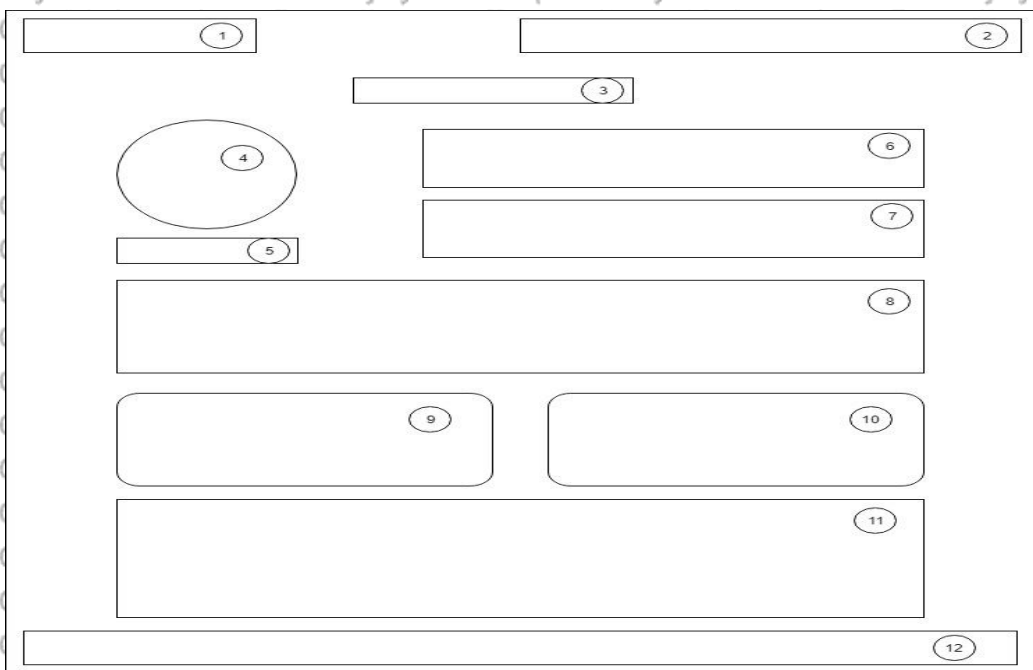


Tabel 5.26 Penejelasan Antarmuka Halaman Lihat Profil

| No | Nama Objek    | Tipe          | Keterangan   |
|----|---------------|---------------|--|
| 1  | Logo          | <i>Link</i>   | Menampilkan logo dari sistem.                                  |
| 2  | Menu bar      | <i>Link</i>   | Menampilkan menu utama yang ada pada sistem.                   |
| 3  | Foto profil   | <i>Image</i>  | Menampilkan gambar profil dari pengguna.                       |
| 4  | Menu vertikal | <i>Link</i>   | Menampilkan menu dari <i>dashboard</i> masing-masing pengguna. |
| 5  | Upload CV     | <i>Button</i> | Menampilkan tombol untuk mengupload CV untuk pengacara.        |
| 6  | Profil        | <i>Form</i>   | Menampilkan profil dari pengguna.                              |
| 7  | <i>Update</i> | <i>Button</i> | Menampilkan tombol untuk memperbaharui profil.                 |
| 8  | <i>Footer</i> | <i>Text</i>   | Menampilkan <i>copyright</i> dari sistem.                      |

### 5. Perancangan Antarmuka Halaman Lihat Profil Pengacara

Perancangan antarmuka halaman lihat profil pengacara digambarkan pada gambar 5.18 dan penjelasan dari perancangan antarmuka halaman melihat profil pengacara dijelaskan pada tabel 5.27.



Gambar 5.18 Perancangan Antarmuka Halaman Lihat Profil Pengacara





Tabel 5.27 Penjelasan Antarmuka Halaman Lihat Profil Pengacara

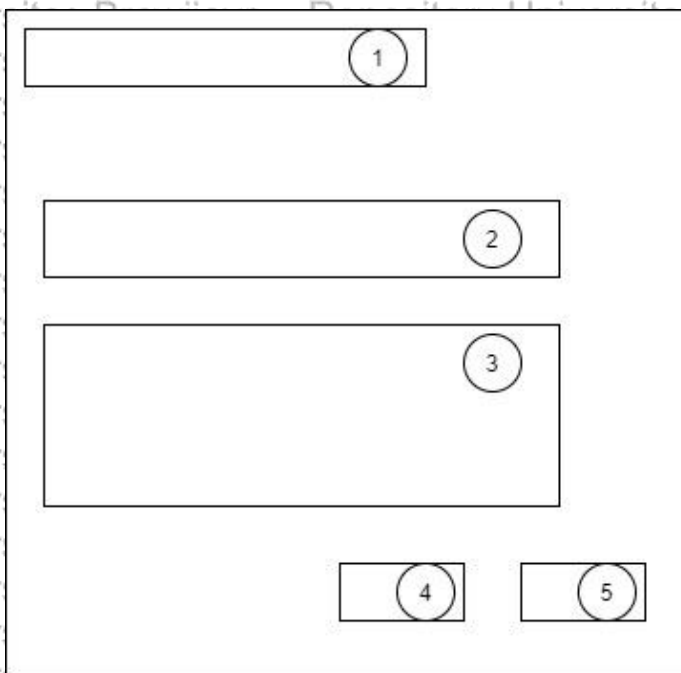
| No | Nama Objek                    | Tipe          | Keterangan   |
|----|-------------------------------|---------------|--|
| 1  | Logo                          | <i>Link</i>   | Menampilkan logo dari sistem.  |
| 2  | Menu bar                      | <i>Link</i>   | Menampilkan menu utama yang ada pada sistem.                           |
| 3  | Nama pengacara                | <i>Text</i>   | Menampilkan nama dari pengacara.                                       |
| 4  | Foto                          | <i>Image</i>  | Menampilkan foto dari pengacara.                                       |
| 5  | <i>Rating</i>                 | <i>Text</i>   | Menampilkan <i>rating</i> yang didapatkan oleh pengacara.              |
| 6  | Tentang Pengacara             | <i>Text</i>   | Menampilkan informasi tentang pengacara.                               |
| 7  | Pendidikan dan Pengalaman     | <i>Text</i>   | Menampilkan Informasi tentang pendidikan dan pengalaman pengacara.     |
| 8  | Bidang Keahlian dan Spesialis | <i>Text</i>   | Menampilkan informasi tentang bidang keahlian dan spesialis pengacara. |
| 9  | Konsultasi Tatap Muka         | <i>Button</i> | Menampilkan tombol untuk berkonsultasi tatap muka dengan pengacara.    |
| 10 | Gunakan Jasa Pengacara        | <i>Button</i> | Menampilkan tombol untuk menggunakan jasa pengacara.                   |
| 11 | Ulasan                        | <i>Text</i>   | Menampilkan ulasan yang didapatkan oleh pengacara.                     |
| 12 | <i>Footer</i>                 | <i>Text</i>   | Menampilkan <i>copyright</i> dari sistem.                              |

### 5.1.5.2 Perancangan Antarmuka Iterasi 2

Perancangan antarmuka iterasi kedua pada sistem konsultasi dan pencarian pengacara menggambarkan penambahan tampilan dikarenakan adanya penambahan kebutuhan fungsional yang dijelaskan pada subbab 4.2.2.2 kebutuhan fungsional iterasi 2.

Perancangan antarmuka yang digunakan sebagai sampel adalah halaman tambah progres kasus digambarkan pada gambar 5.19 dan penjelasan dari perancangan antarmuka halaman tambah progres kasus dijelaskan pada tabel 5.28.





**Gambar 5.19** Perancangan Antarmuka Halaman Tambah Progres Kasus

**Tabel 5.28** Penjelasan Antarmuka Halaman Tambah Progres Kasus

| No | Nama Objek              | Tipe                   | Keterangan  |
|----|-------------------------|------------------------|---|
| 1  | Tambahkan Progres Kasus | <i>Label</i>           | Menampilkan judul dari halaman.                                       |
| 2  | Tanggal                 | <i>Input text</i>      | Input <i>text</i> untuk memberikan tanggal progres kasus              |
| 3  | Progres                 | <i>Input text area</i> | Input <i>textarea</i> untuk memberikan penjelasan dari progres kasus. |
| 4  | Batal                   | <i>Button</i>          | Tombol untuk membatalkan penambahan progres kasus.                    |
| 5  | Simpan                  | <i>Button</i>          | Tombol untuk menyimpan progres kasus.                                 |

## 5.2 Implementasi

Pada bab ini menjelaskan tentang implementasi yang dilakukan saat pembuatan sistem. Sebelum menjelaskan implementasi, spesifikasi sistem yang digunakan dalam implementasi sistem akan dijelaskan terlebih dahulu. Spesifikasi tersebut terbagi menjadi spesifikasi perangkat keras dan spesifikasi perangkat lunak. Setelah spesifikasi sistem, akan dijelaskan implementasi yang merupakan implementasi basis data, implementasi kode program, dan implementasi antarmuka.





### 5.2.1 Spesifikasi Sistem

Spesifikasi sistem menjelaskan mengenai spesifikasi sistem yang digunakan dalam membangun sistem. Pada spesifikasi sistem dibedakan menjadi dua yaitu spesifikasi perangkat keras dan spesifikasi perangkat lunak.

#### 5.2.1.1 Spesifikasi Perangkat Keras

Perangkat keras yang digunakan untuk membangun sistem dijelaskan pada Tabel 5.29.

**Tabel 5.29 Spesifikasi Perangkat Keras**

| No | Nama Komponen    | Spesifikasi                                    |
|----|------------------|--|
| 1  | Model            | ASUS A407M                                     |
| 2  | <i>Processor</i> | Intel(R) Celeron(R) N4000 CPU @ 1.10GHz 1.10GH |
| 3  | <i>Memory</i>    | 4GB RAM  |

#### 5.2.1.2 Spesifikasi Perangkat Lunak

Perangkat lunak yang digunakan untuk membangun sistem dijelaskan pada Tabel 5.30.

**Tabel 5.30 Spesifikasi Perangkat Lunak**

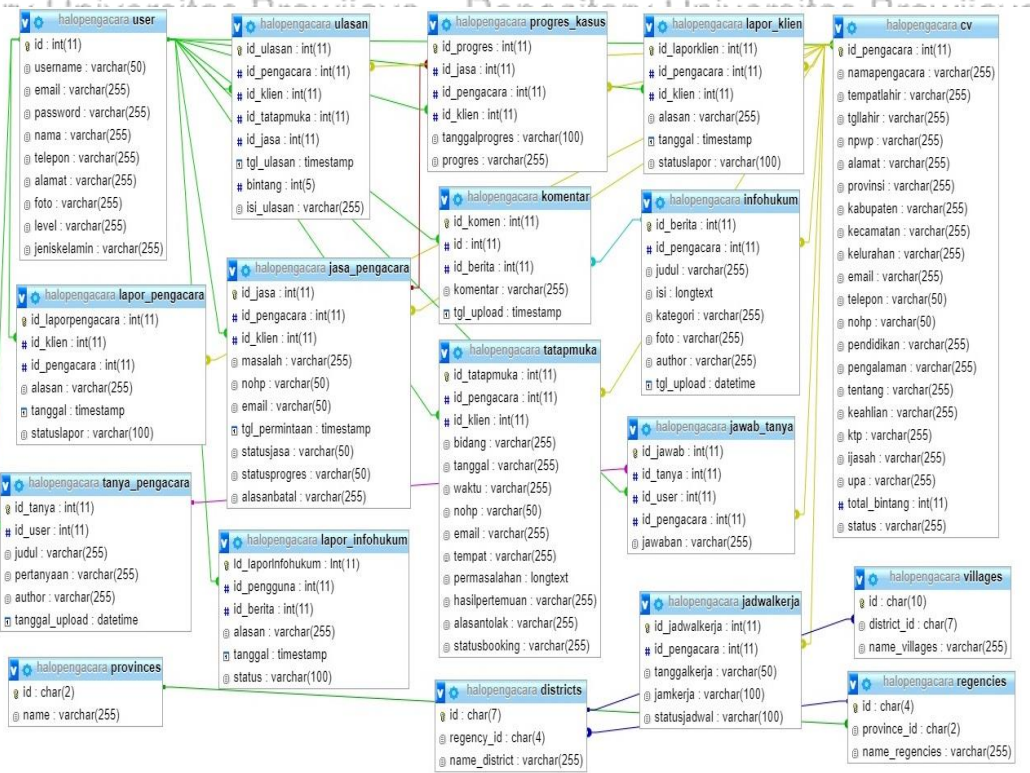
| No | Nama Komponen          | Spesifikasi                            |
|----|------------------------|--|
| 1  | Sistem Operasi         | Windows 10 Home Single Language 64 bit |
| 2  | Bahasa Pemrograman     | PHP                                    |
| 3  | Editor Pemrograman     | Sublime Text 3                         |
| 4  | <i>Web Server</i>      | Apache 2.0                             |
| 5  | <i>Framework</i>       | CodeIgniter                            |
| 6  | <i>Markup Language</i> | HTML, CSS, JavaScript                  |
| 7  | Database               | MySQL                                  |
| 8  | <i>Browser</i>         | Google Chrome, Firefox                 |
| 9  | Editor Dokumentasi     | Microsoft Office Word 2013             |

### 5.2.2 Implementasi Basis Data

Implementasi basis data merupakan tahap untuk mengimplementasikan basis data yang sebelumnya sudah didefinisikan di tahap perancangan basis data. Sesuai dengan hasil perancangan basis data terdapat 18 tabel yaitu tabel user, cv, tanya\_pengacara, jawab\_tanya, infohukum, komentar, jadwalkerja, *district*,



provinces, regencies, villages, jasa\_pengacara, tatapmuka, progress\_kasus, lapor\_infohukum, lapor\_klien, lapor\_pengacara, dan ulasan. Pada gambar 5.20 adalah gambar dari implementasi basis data.



Gambar 5.20 Implementasi Basis Data

### 5.2.3 Implementasi Komponen

Implementasi komponen merupakan tahapan untuk mengimplementasikan algoritme yang telah didefinisikan pada tahapan perancangan komponen menjadi kode program. Pada implementasi komponen, sistem ini menggunakan bahasa pemrograman PHP.

#### 5.2.3.1 Implementasi Komponen Tambah Ulasan

Pada tabel 5.31 merupakan implementasi komponen tambah ulasan sesuai dengan perancangan komponen tambah ulasan iterasi 1 yang berada pada sub bab 5.1.4.1. Implementasi komponen tambah ulasan didapatkan dari algoritme fungsi tambahUlasan(id\_pengacara, id\_tatapmuka) pada controller bernama C\_Klien.

Tabel 5.31 Implementasi Komponen Tambah Ulasan

| Source code Tambah Ulasan |  |
|---------------------------|--|
| 1                         | public function tambahUlasan(\$id_pengacara,\$id_tatapmuka){ |
| 2                         | \$id_klien = \$this->session->userdata('id');                |
| 3                         | \$bintang = \$this->input->post('bintang');                  |
| 4                         | \$ulasan = \$this->input->post('ulasan');                    |





### Source code Tambah Ulasan

```

5         if($bintang == 0){
6             echo "<script
7 language='javascript'>alert('rating tidak boleh
8 kosong.');
```

```

9             javascript:history.go(-1);</script>";
10        }else{
11            $data = array(
12                'id_pengacara' => $id_pengacara,
13                'id_klien' => $id_klien,
14                'id_tatapmuka' => $id_tatapmuka,
15                'bintang' => $bintang,
16                "isi_ulasan" => $ulasan
17            );
18            $this->M_Klien->addUlasan($data);
19            $jum = $this->M_Klien-
20            >getBintang($id_pengacara);
21            $jum_bin = $jum;
22            $this->M_Klien-
23            >updateBintang($id_pengacara,$bintang,$jum_bin);
24            redirect('C_Klien/detailpengacara/'.$id_pengacara);
25        }
26    }
27 }
28 }

```

### 5.2.3.2 Implementasi Komponen Kirim Pertanyaan

Pada tabel 5.32 merupakan implementasi komponen kirim pertanyaan sesuai dengan perancangan komponen kirim pertanyaan iterasi 1 yang berada pada sub bab 5.1.4.1. Implementasi komponen kirim pertanyaan didapatkan dari algoritme fungsi askPengacara() pada *controller* bernama C\_Klien.

**Tabel 5.32 Implementasi Komponen Kirim Pertanyaan**

### Source Code Kirim Pertanyaan

```

1 public function askPengacara(){
2     $id = $this->session->userdata('id');
3     $username = $this->session->userdata('username');
4     $judul = $this->input->post('judul');
5     $pertanyaan = $this->input->post('pertanyaan');

```





```

6
7     $data = array(
8         'id_user' => $id,
9         'judul' => $judul,
10        'pertanyaan' => $pertanyaan,
11        'author' => $username
12    );
13
14    $insert = $this->M_Klien->postTanya($data);
15    redirect('C_Klien/k_tanyapengacara');
16 }

```

### 5.2.3.3 Implementasi Komponen Cari Pengacara

Pada tabel 5.33 merupakan implementasi komponen cari pengacara sesuai dengan perancangan komponen cari pengacara iterasi 1 yang berada pada sub bab 5.1.4.1. Implementasi komponen cari pengacara didapatkan dari algoritme fungsi `search()` pada *controller* bernama `main_user`.

**Tabel 5.33 Implementasi Komponen Cari Pengacara**

**Source Code Cari Pengacara**

```

1
2 public function search() {
3
4     $keahlian = $this->input->post('keahlian');
5     $kabupaten = $this->input->post('kabupaten');
6
7     $data['list'] = $this->user_model->
8     >searchpengacara($keahlian, $kabupaten);
9
10
11     if($this->session->userdata('level') ==
12     'klien'){
13         $this->load->view("view_header");
14         $this->load->view("view_menu_klien");
15         $this->load->view("hasilcari", $data);
16         $this->load->view("view_footer");
17     }else if($this->session->userdata('level') ==
18     "pengacara"){
19         $this->load->view("view_header");
20         $this->load->view("view_menu_pengacara");
21         $this->load->view("hasilcari", $data);
22         $this->load->view("view_footer");
23     }else{
24         $this->load->view("view_header");
25     }
26 }
27
28
29
30

```





### Source Code Cari Pengacara

```

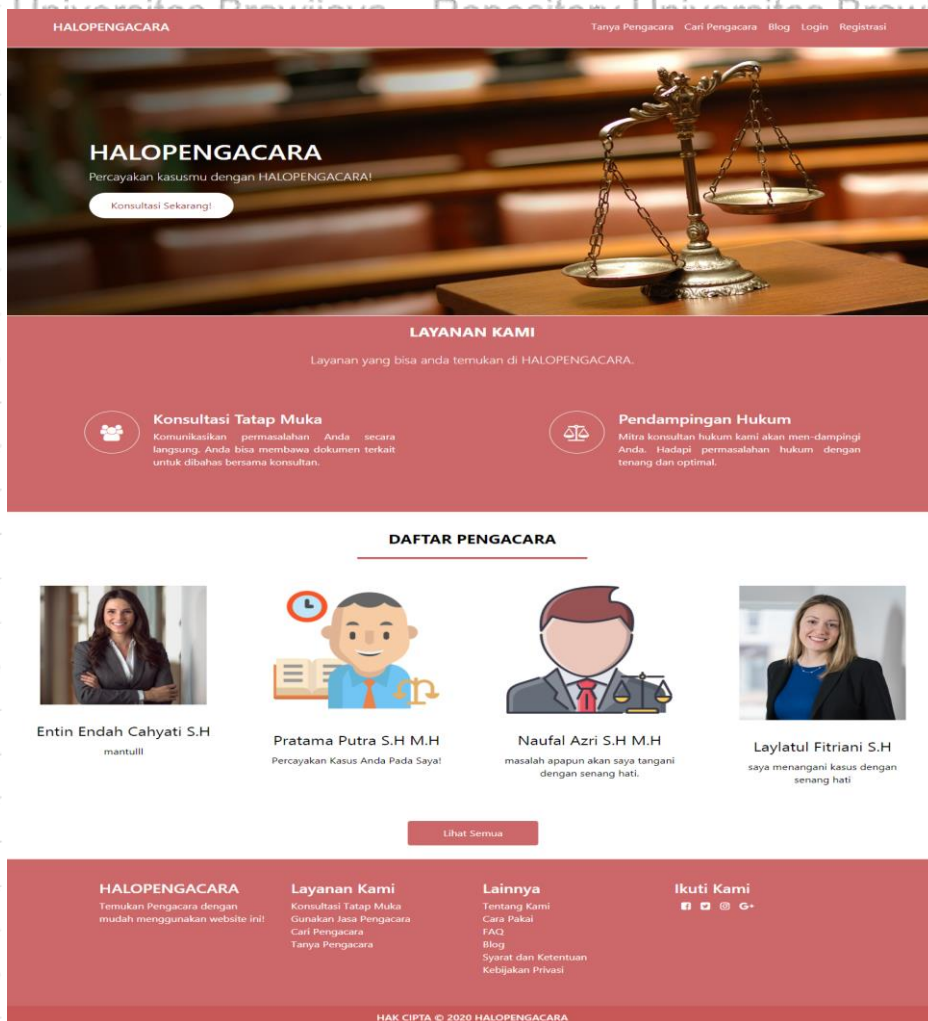
31     $this->load->view("view_menu");
32
33     $this->load->view("hasilcari", $data);
34
35     $this->load->view("view_footer");
36
37 }
  
```

## 5.2.4 Implementasi Antarmuka

Implementasi antarmuka merupakan penerapan perancangan antarmuka yang sudah diimplementasikan ke dalam sistem.

### 5.2.4.1 Implementasi Antarmuka Halaman *Home*

Gambar 5.21 merupakan hasil implementasi antarmuka halaman *home* yang didasarkan pada subbab perancangan yaitu 5.1.5.1 Perancangan Antarmuka Iterasi 1 Halaman *Home*.



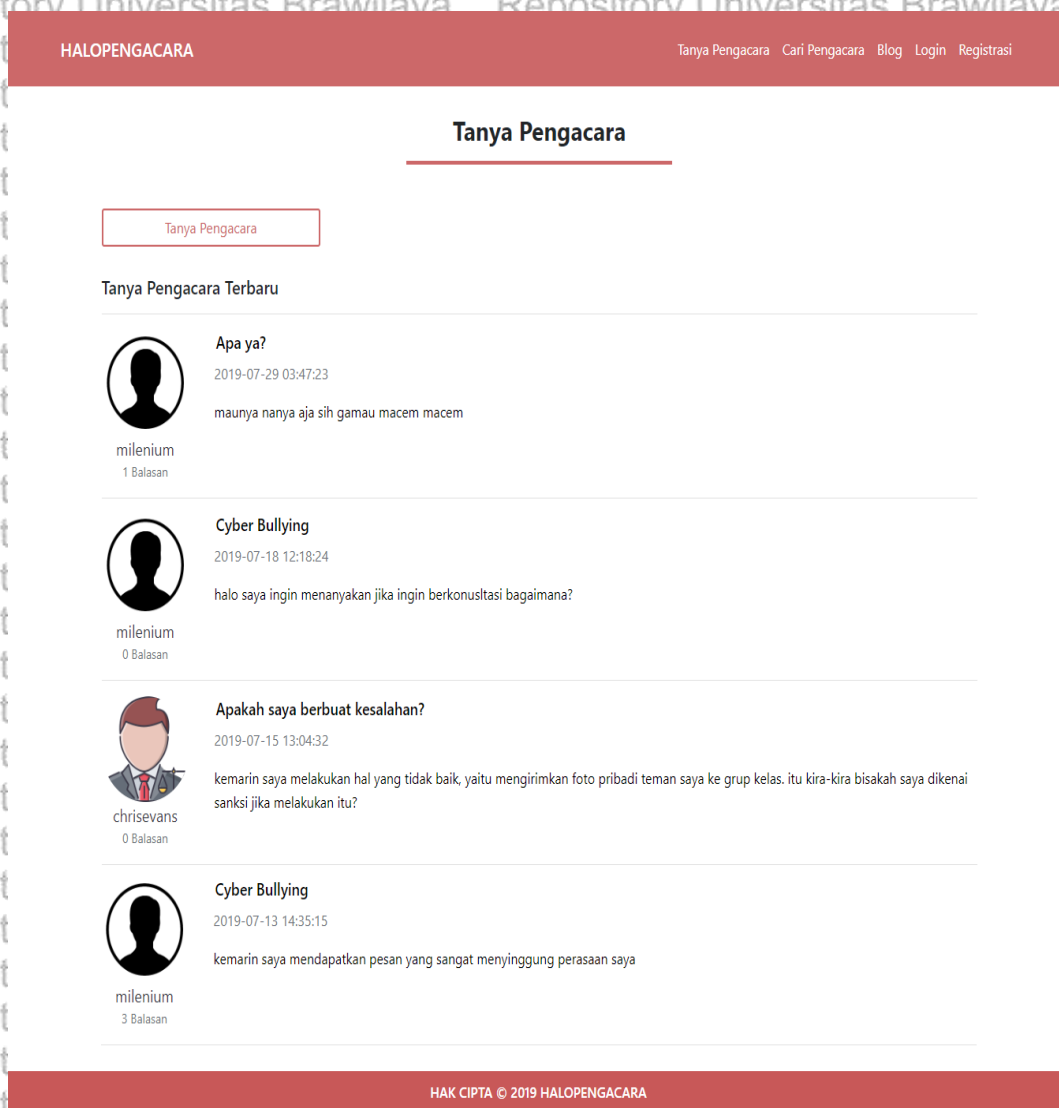
Gambar 5.21 Implementasi Antarmuka Halaman *Home*





### 5.2.4.2 Implementasi Antarmuka Halaman Tanya Pengacara

Gambar 5.22 merupakan hasil implementasi antarmuka halaman tanya pengacara yang didasarkan pada subbab perancangan yaitu 5.1.5.1 Perancangan Antarmuka Iterasi 1 Halaman Tanya Pengacara.

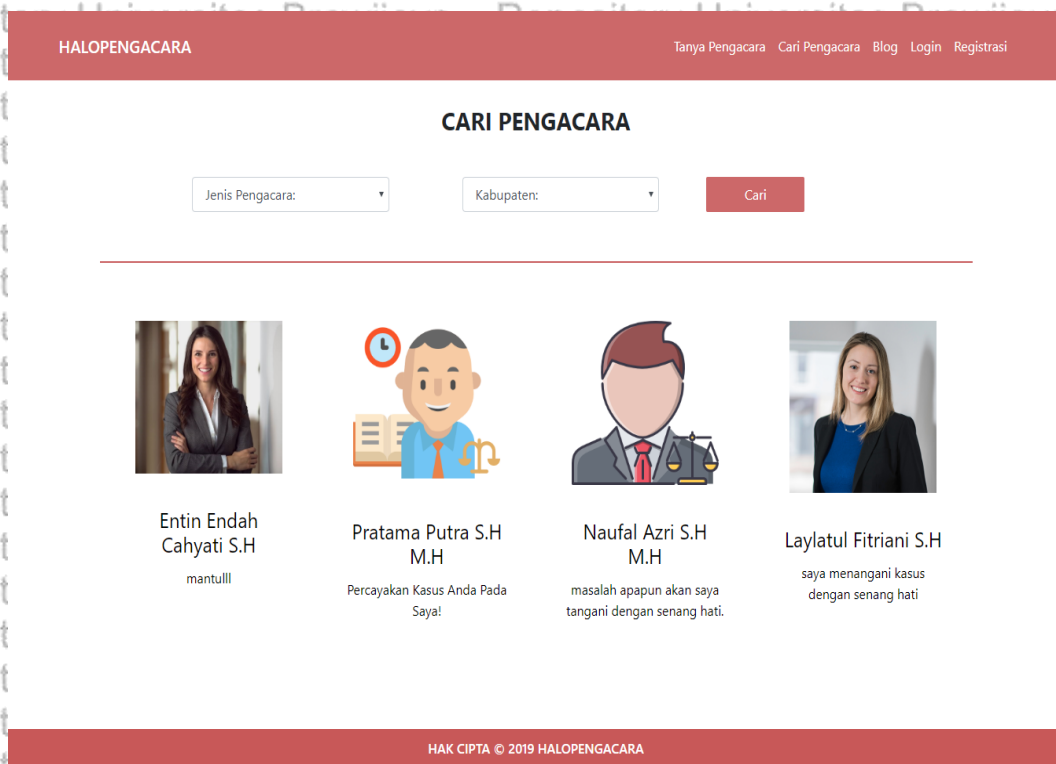


**Gambar 5.22 Implementasi Antarmuka Halaman Tanya Pengacara**

### 5.2.4.3 Implementasi Antarmuka Halaman Cari Pengacara

Gambar 5.23 merupakan hasil implementasi antarmuka halaman cari pengacara yang didasarkan pada subbab perancangan yaitu 5.1.5.1 Perancangan Antarmuka Iterasi 1 Halaman Cari Pengacara

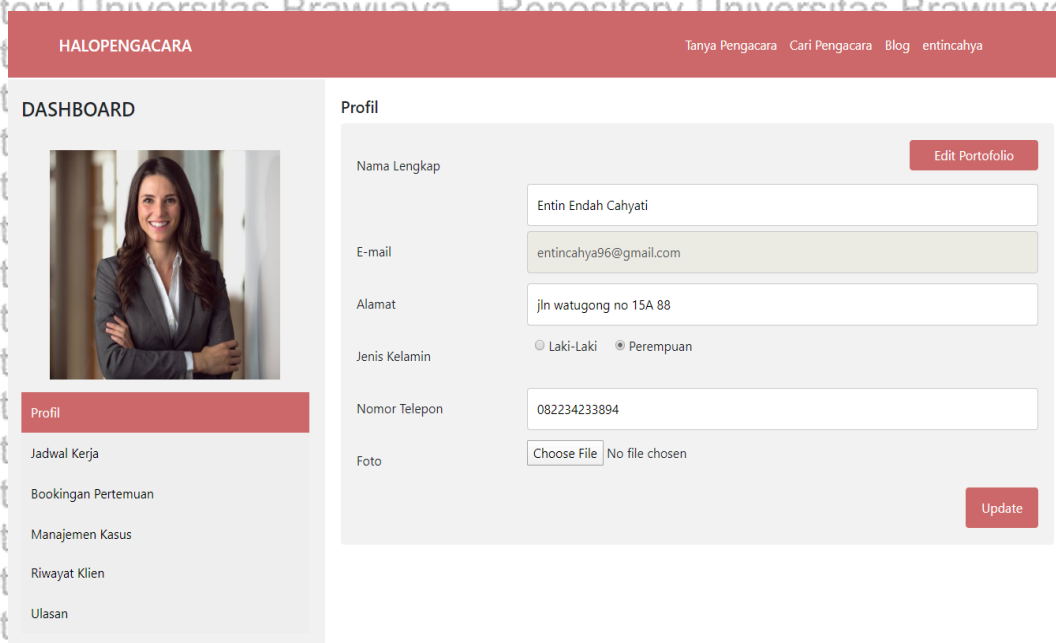




**Gambar 5.23 Implementasi Antarmuka Halaman Cari Pengacara**

#### 5.2.4.4 Implementasi Antarmuka Halaman Lihat Profil

Gambar 5.24 merupakan hasil implementasi antarmuka halaman lihat profil yang didasarkan pada subbab perancangan yaitu 5.1.5.2 Perancangan Antarmuka Iterasi 1 Halaman lihat profil.



**Gambar 5.24 Implementasi Antarmuka Halaman Lihat Profil**





### 5.2.4.5 Implementasi Antarmuka Halaman Lihat profil Pengacara

Gambar 5.25 merupakan hasil implementasi antarmuka halaman lihat profil pengacara yang didasarkan pada subbab perancangan yaitu 5.1.5.1 Perancangan Antarmuka Iterasi 1 Halaman Lihat Profil Pengacara.

The screenshot shows a user profile page for a lawyer named Pratama Putra S.H M.H. The page is titled 'HALOPENGACARA' and includes navigation links for 'Tanya Pengacara', 'Cari Pengacara', 'Blog', 'Login', and 'Registrasi'. The profile information includes a name, a location (Jin teuku umar no 150, Kelurahan kebayoran lama selatan, kecamatan kebayoran lama, kabupaten kota jakarta selatan, DKI Jakarta), and a rating of 4 stars from 1 review. The 'Tentang Pengacara' section states 'Percayakan Kasus Anda Pada Saya!'. The 'Pendidikan dan Pengalaman' section lists 'S.H dari Universitas Brawijaya M.H dari Universitas Indonesia' and 'Menjabat sebagai wakil ketua Peradi lowokwaru'. The 'Bidang Keahlian dan Spesialisasi' section is a table with two rows: 'Keluarga' (6 Terselesaikan, 5 stars) and 'Ketenagakerjaan' (2 Terselesaikan, 3 stars). The 'Layanan' section offers two options: 'Konsultasi Tatap Muka' and 'Gunakan Jasa Pengacara', each with a 'Pilih' button. The 'Ulasan' section shows a review from Chris Evans with a 5-star rating and the text 'good' dated 2020-06-25 00:24:02. The footer contains 'HAK CIPTA © 2020 HALOPENGACARA'.

**Gambar 5.25 Implementasi Antarmuka Halaman Lihat Profil Pengacara**

### 5.2.4.6 Implementasi Antarmuka Halaman Tambah Progres Kasus

Gambar 5.26 merupakan hasil implementasi antarmuka halaman tambah progres kasus yang didasarkan pada subbab perancangan yaitu 5.1.5.2 Perancangan Antarmuka Iterasi 2 Halaman Tambah Progres Kasus.





**Tambahkan Progres Kasus** X

Tanggal

dd-mm-yyyy

Progres

Batal Simpan

Gambar 5.26 Implementasi Antarmuka Halaman Tambah Progres Kasus



## BAB 6 PENGUJIAN

Pengujian merupakan tahapan yang dilakukan setelah tahap implementasi sistem. Pengujian bertujuan untuk mengecek apakah implementasi yang dilakukan apakah sudah sesuai dengan analisis kebutuhan dan perancangan sistem, dan juga untuk menemukan jika ada kesalahan yang ada pada sistem. Pengujian yang dilakukan adalah pengujian unit, pengujian integrasi, pengujian validasi, dan pengujian *compatibility*.

### 6.1 Pengujian Unit

Pengujian unit adalah pengujian yang dilakukan untuk memastikan bahwa implementasi sudah sesuai dengan yang diinginkan melalui pengujian algoritme atau *method* yang sudah dibangun. Pengujian unit menggunakan basis *path*. *Method* yang akan diuji dalam pengujian basis *path* adalah *method* edit portofolio, mengirimkan pertanyaan, dan menambahkan info hukum.

#### 6.1.1 Pengujian Unit Tambah Ulasan

Pada tabel 6.1 merupakan pembentukan node *flowgraph* untuk algoritme dari *method* `tambahUlasan(id_pengacara, id_tatapmuka)` yang terdapat pada klas C\_Klien berdasarkan rancangan algoritme tambah ulasan pada sub bab 5.1.4.1 Perancangan Komponen tambah ulasan iterasi 1.

Tabel 6.1 Pembentukan Node Algoritme fungsi `tambahUlasan(id_pengacara, id_tatapmuka)`





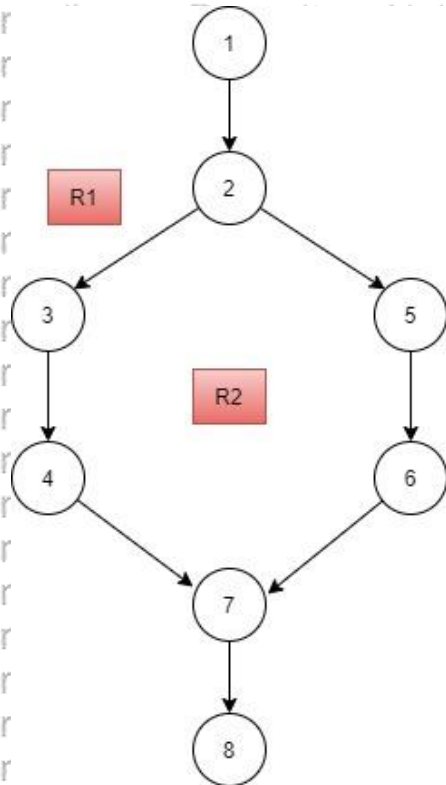


```

tambahUlasan(id_pengacara, id_tatapmuka)
{
    Menginisialisasi variable jum dengan memanggil
    method getBintang(id_pengacara) pada database
    Memasukkan data update bintang pada database dengan
    data id_pengacara, bintang, dan jum_bin
    Redirect ke halaman detailpengacara dengan
    id_pengacara
    END IF
    SELESAI
}
    
```

1. Flow Graph

Gambar 6.1 merupakan flow graph yang didapatkan dari algoritme yang ditunjukkan pada Tabel 6.1 Pembentukan Node Algoritma fungsi tambahUlasan(id\_pengacara, id\_tatapmuka).



Gambar 6.1 Flow Graph Algoritme fungsi tambahUlasan(id\_pengacara, id\_tatapmuka)

2. Cyclomatic Complexity

$V(G) = \text{jumlah region}$   
 $= 2$ , yaitu R1 dan R2





$$V(G) = \text{edge} - \text{node} + 2$$

$$= (8 - 8) + 2$$

$$= 2$$

$$V(G) = \text{predicate node} + 1$$

$$= 1 + 1$$

$$= 2$$

### 3. *Independent Path*

$$\text{Jalur 1} = 1 - 2 - 3 - 4 - 7 - 8$$

$$\text{Jalur 2} = 1 - 2 - 5 - 6 - 7 - 8$$

Berdasarkan dari hasil *independent path* sebelumnya yang digunakan sebagai kasus uji. Tabel 6.2 merupakan penjelasan dari kasus uji dari algoritme fungsi `tambahUlasan(id_pengacara, id_tatapmuka)`.

**Tabel 6.2 Kasus Uji Tambah Ulasan**

| Jalur | Prosedur Uji   | Expected Result                                       | Result  | status |
|-------|--|---|---|--------|
| 1     | Klas Driver memanggil <i>method</i> <code>tambahUlasan(id_pengacara, id_tatapmuka)</code> dari klas <code>C_Klien</code> dengan inisialisasi bintang = 0.                                  | Menunjukkan pesan berisi "Rating tidak boleh kosong". | Menunjukkan pesan berisi "Rating tidak boleh kosong". | Valid  |
| 2     | Klas Driver memanggil <i>method</i> <code>tambahUlasan(id_pengacara, id_tatapmuka)</code> dari klas <code>C_Klien</code> dengan inisialisasi bintang = 4, <code>isi_ulasan = good</code> . | Pengembalian bernilai <i>true</i>                     | Pengembalian bernilai <i>true</i>                     | Valid  |

### 6.1.2 Pengujian Unit Kirim Pertanyaan

Pada tabel 6.3 merupakan pembentukan node path untuk algoritme dari `method askPengacara()` yang terdapat pada klas `C_Klien` berdasarkan rancangan algoritme kirim pertanyaan pada sub bab 5.1.4.1 Perancangan Komponen kirim pertanyaan iterasi 1.



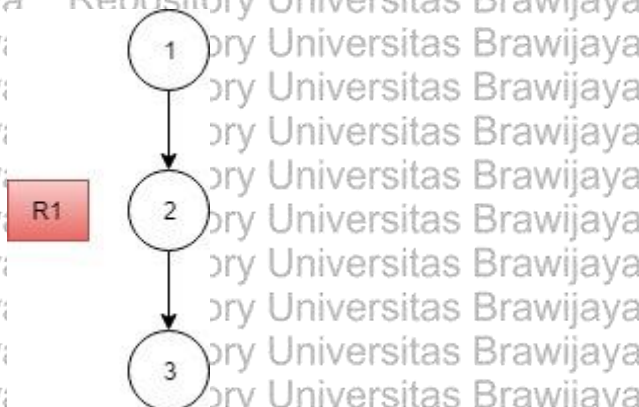


**Tabel 6.3 Pembentukan Node Algoritme fungsi askPengacara()**

|   |
|---|
| askPengacara()  |
| MULAI <b>1</b>  |
| Inisialisasi variable id dengan id dari data yang sudah disimpan sebelumnya                 |
| Inisialisasi variable username dengan username dari data yang sudah disimpan sebelumnya     |
| Menginputkan field judul dan pertanyaan pada form tanya                                     |
| Inisialisasi variable data dengan array yang berisikan variable yang ada pada form tanya    |
| Memasukkan data tanya pengacara pada database dengan data yang ada pada variable array data |
| Redirect ke halaman k_tanyapengacara  |
| SELESAI <b>3</b>  |

**1. Flow Graph**

Gambar 6.2 merupakan *flow graph* yang didapatkan dari algoritme yang ditunjukkan pada Tabel 6.3 Pembentukan Node Algoritma fungsi askPengacara().



**Gambar 6.2 Flow Graph Algoritme fungsi askPengacara()**

**2. Cyclomatic Complexity**

$V(G)$  = jumlah region

= 1, yaitu R1

$$V(G) = \text{edge} - \text{node} + 2$$

$$= (2 - 3) + 2$$

$$= 1$$

$$V(G) = \text{predicate node} + 1$$

$$= 0 + 1$$





3. *Independent Path*

Jalur 1 = 1 - 2 - 3

Berdasarkan dari hasil *independent path* sebelumnya yang digunakan sebagai kasus uji. Tabel 6.4 merupakan penjelasan dari kasus uji dari algoritme fungsi askPengacara ().

**Tabel 6.4 Kasus Uji Kirim Pertanyaan**

| Jalur | Prosedur Uji   | Expected Result                   | Result                           | status |
|-------|--|-----------------------------------|----------------------------------|--------|
| 1     | Klas Driver memanggil <i>method</i> askPengacara () dari klas C_Klien dengan inialisasi judul = pertanyaan, pertanyaan = haloo | Pengembalian bernilai <i>true</i> | Penembalian bernilai <i>true</i> | Valid  |

6.1.3 Pengujian Unit Cari Pengacara

Pada tabel 6.5 merupakan pembentukan node path untuk algoritme dari *method* search () yang terdapat pada klas main\_user berdasarkan rancangan algoritme cari pengacara pada sub bab 5.1.4.1 Perancangan Komponen cari pengacara iterasi 1.

**Tabel 6.5 Pembentukan Node Algoritme fungsi search()**

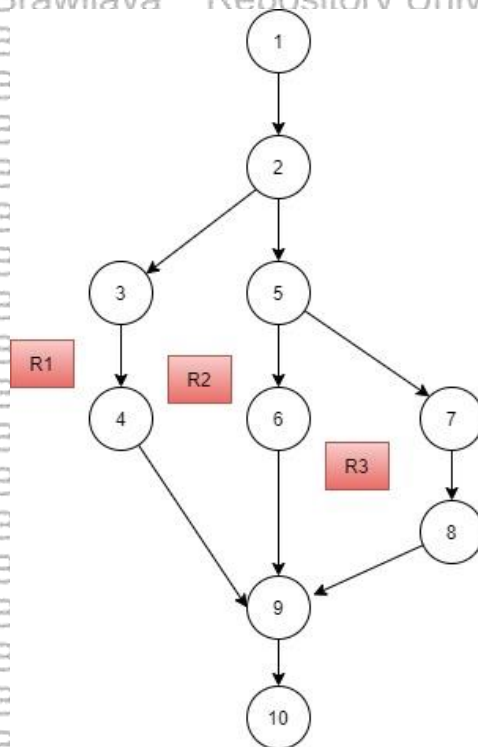
|  |
|--|
| search ()  |
| MULAI 1  |
| Menginputkan field keahlian dan kabupaten  |
| Memasukkan data cari pengacara pada database dengan parameter keahlian dan kabupaten 2 |
| IF (level == klien) 3  |
| Menampilkan halaman hasilcari pada klien 4   |
| ELSE IF (level == pegacara) 5  |
| Menampilkan halaman hasilcari pada pengacara 6   |
| ELSE 7   |
| Menampilkan halaman hasilcari 8  |
| END IF 9   |
| SELESAI 10   |





### 1. Flow Graph

Gambar 6.3 merupakan *flow graph* yang didapatkan dari algoritme yang ditunjukkan pada Tabel 6.5 Pembentukan Node Algoritma fungsi `search()`.



**Gambar 6.3 Flow Graph Algoritme fungsi `search()`**

### 2. Cyclomatic Complexity

$V(G)$  = jumlah region  
= 3, yaitu R1, R2, dan R3

$$V(G) = \text{edge} - \text{node} + 2$$

$$= (11 - 10) + 2$$

$$= 3$$

$$V(G) = \text{predicate node} + 1$$

$$= 2 + 1$$

$$= 3$$

### 3. Independent Path

Jalur 1 = 1 – 2 – 3 – 4 – 9 – 10

Jalur 2 = 1 – 2 – 5 – 6 – 9 – 10

Jalur 3 = 1 – 2 – 5 – 7 – 8 – 9 – 10





Berdasarkan dari hasil *independent path* sebelumnya yang digunakan sebagai kasus uji. Tabel 6.6 merupakan penjelasan dari kasus uji dari algoritme fungsi *search()*.

Tabel 6.6 Kasus Uji Cari Pengacara

| Jalur | Prosedur Uji   | Expected Result                   | Result                            | status |
|-------|--|-----------------------------------|-----------------------------------|--------|
| 1     | Klas Driver memanggil <i>method search()</i> dari kelas <code>main_user</code> dengan inisialisasi keahlian = bisnis, kabupaten = malang kota, dan login dalam level Klien     | Pengembalian bernilai <i>true</i> | Pengembalian bernilai <i>true</i> | Valid  |
| 2     | Klas Driver memanggil <i>method search()</i> dari kelas <code>main_user</code> dengan inisialisasi keahlian = bisnis, kabupaten = malang kota, dan login dalam level pengacara | Pengembalian bernilai <i>true</i> | Pengembalian bernilai <i>true</i> | Valid  |
| 3     | Klas Driver memanggil <i>method search()</i> dari kelas <code>main_user</code> dengan inisialisasi keahlian = bisnis, kabupaten = malang kota,                                 | Pengembalian bernilai <i>true</i> | Pengembalian bernilai <i>true</i> | Valid  |





| Jalur | Prosedur Uji              | Expected Result | Result | status |
|-------|---------------------------|-----------------|--------|--------|
|       | dan tidak melakukan login |                 |        |        |

## 6.2 Pengujian Integrasi

Pengujian integrasi merupakan tahap untuk menguji relasi antar unit-unit yang ada pada sistem. Pengujian ini menggunakan metode *basis path* dalam *white-box testing* berdasarkan algoritme yang telah diimplementasikan pada sistem. Klas Driver pada pengujian ini dibutuhkan untuk menjalankan *class* yang akan diuji. Berikut ini *method* yang digunakan sebagai sampel pengujian integrasi adalah *method* `ambil_data()`.

### 6.2.1 Pengujian Integrasi *Method* `ambil_data()` kelas *C\_Pengacara*

Pada tabel 6.7 merupakan pembentukan node path untuk algoritme dari *method* `ambil_data()` yang terdapat pada kelas *C\_Pengacara* dimana memiliki fungsi untuk mengambil data kabupaten, kecamatan, dan kelurahan.

**Tabel 6.7 Pembentukan Node Algoritme fungsi `ambil_data()`**

```

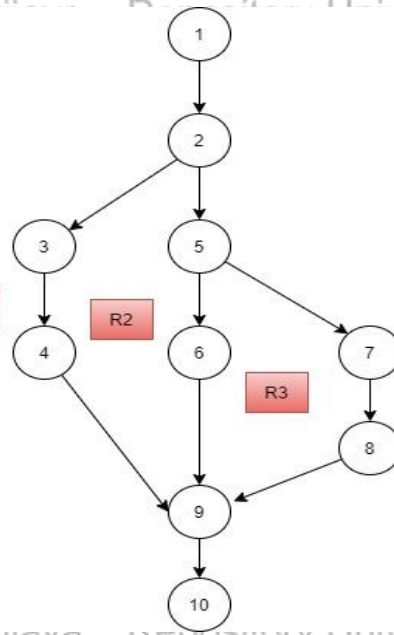
jadwalkerja()
MULAI 1
Inisialisasi variable modul dengan inputan dari modul
Inisialisasi variable id dengan inputan dari id 2
Inisialisasi variable date_now dengan tanggal hari ini
IF (modul == kabupaten) 3
    Menampilkan pengembalian dari pemanggilan method 4
    kabupaten(id) pada kelas M_Pengacara
ELSE IF (modul == kecamatan) 5
    Menampilkan pengembalian dari pemanggilan method 6
    kecamatan(id) pada kelas M_Pengacara
ELSE IF (modul == kelurahan) 7
    Menampilkan pengembalian dari pemanggilan method 8
    kelurahan(id) pada kelas M_Pengacara
END IF 9
SELESAI 10

```

#### 1. Flow Graph

Gambar 6.4 merupakan *flow graph* yang didapatkan dari algoritme yang ditunjukkan pada Tabel 6.7 Pembentukan Node Algoritma fungsi `ambil_data()`.





Gambar 6.4 Flow Graph Algoritme fungsi ambil\_data()

2. Cyclomatic Complexity

$V(G)$  = jumlah region  
 = 3, yaitu R1, R2, dan R3

$V(G)$  = edge - node + 2  
 = (11 - 10) + 2  
 = 3

$V(G)$  = predicate node + 1  
 = 2 + 1  
 = 3

3. Independent Path

Jalur 1 = 1 - 2 - 3 - 4 - 9 - 10

Jalur 2 = 1 - 2 - 5 - 6 - 9 - 10

Jalur 3 = 1 - 2 - 5 - 7 - 8 - 9 - 10

Berdasarkan dari hasil *independent path* sebelumnya yang digunakan sebagai kasus uji. Tabel 6.8 merupakan penjelasan dari kasus uji dari algoritme fungsi `ambil_data()`.

Tabel 6.8 Kasus Uji Fungsi `ambil_data()`

| Jalur | Prosedur Uji                        | Expected Result                           | Result                                    | status |
|-------|-------------------------------------|---|---|--------|
| 1     | Klas Driver memanggil <i>method</i> | Menampilkan nama kabupaten "Kota Malang". | Menampilkan nama kabupaten "Kota Malang". | Valid  |





| Jalur | Prosedur Uji   | Expected Result                            | Result                                     | status |
|-------|--|--|--|--------|
|       | ambil data () yang memanggil method kabupaten(id) pada klas M_Pengacara dengan id = 3573 dan modul = kabupaten.                                    |  |  |        |
| 2     | Klas Driver memanggil method ambil_data () yang memanggil method kecamatan(id) pada klas M_Pengacara dengan id = 3573050 dan modul = kecamatan.    | Menampilkan nama kecamatan "Lowokwaru".    | Menampilkan nama kecamatan "Lowokwaru".    | Valid  |
| 3     | Klas Driver memanggil method ambil_data () yang memanggil method kelurahan(id) pada klas M_Pengacara dengan id = 3573050004 dan modul = kelurahan. | Menampilkan nama kelurahan "Ketawanggede". | Menampilkan nama kelurahan "Ketawanggede". | Valid  |

### 6.3 Pengujian Validasi

Pengujian validasi merupakan tahap untuk menguji sistem yang dibangun apakah sesuai dengan kebutuhan fungsional yang telah ditetapkan pada tahap rekayasa kebutuhan. Teknik dari pengujian validasi menggunakan teknik *black-box testing* untuk mengetahui sistem sudah dibangun sesuai dengan kondisi normal maupun kondisi alternatif.





### 6.3.1 Pengujian Validasi Login

#### 1. Pengujian validasi login

Tabel 6.9 merupakan hasil dari pengujian validasi *use case login* yang berhasil.

**Tabel 6.9 Pengujian Validasi Login**

|                 |   |
|-----------------|---|
| Kode Kebutuhan  | SKPP-F-01   |
| Kasus Uji       | Login   |
| Prosedur        | <ol style="list-style-type: none"> <li>Mengakses halaman <i>login</i>.</li> <li>Memasukkan email = "entincahnya96@gmail.com" dan password "114k412s"</li> <li>Menekan tombol <b>LOGIN</b>.</li> </ol> |
| Expected Result | Aktor berhasil masuk ke dalam sistem sesuai dengan level aktor yaitu pengacara, klien, dan admin.   |
| Result          | Aktor berhasil masuk ke dalam sistem sesuai dengan level aktor yaitu pengacara.   |
|                 |   |
| Status          | Valid   |

#### 2. Pengujian validasi login alternatif satu

Tabel 6.10 merupakan hasil dari pengujian validasi *use case login* yang tidak berhasil karena kolom yang ada pada *form login* dibiarkan kosong.

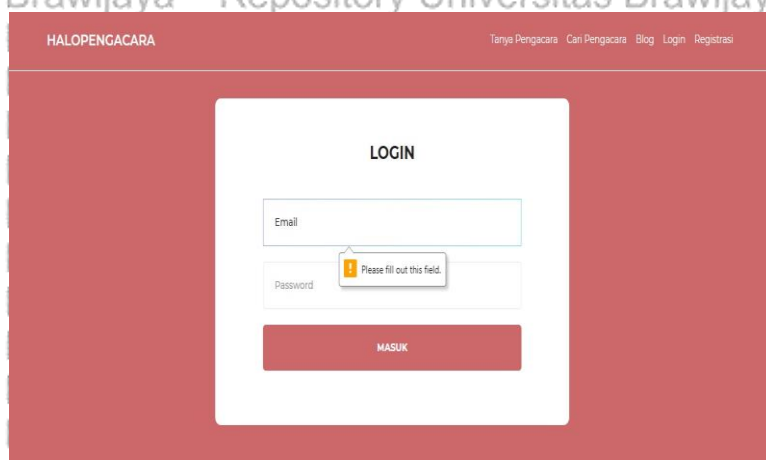
**Tabel 6.10 Pengujian Validasi Login Alternatif Satu**

|                |  |
|----------------|--|
| Kode Kebutuhan | SKPP-F-01  |
| Kasus Uji      | Login dengan kondisi alternatif satu   |
| Prosedur       | <ol style="list-style-type: none"> <li>Mengakses halaman <i>login</i>.</li> <li>Menekan tombol <b>LOGIN</b> tanpa mengisi <i>email</i> dan <i>password</i> yang ada pada <i>form login</i>.</li> </ol> |





|                 |   |
|-----------------|---|
| Expected Result | Sistem menampilkan pesan "Please fill out this field" |
| Result          | Sistem menampilkan pesan "Please fill out this field" |
| Status          | Valid   |

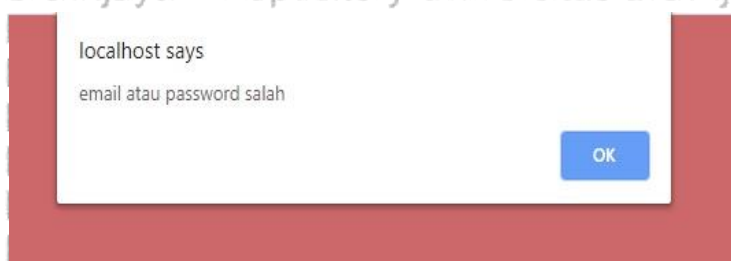


### 3. Pengujian validasi *login* alternatif dua

Tabel 6.11 merupakan hasil dari pengujian validasi *use case login* yang tidak berhasil karena memasukkan *email* atau *password* tidak benar.

**Tabel 6.11 Pengujian Validasi *Login* Alternatif Dua**

|                 |   |
|-----------------|---|
| Kode Kebutuhan  | SKPP-F-01   |
| Kasus Uji       | <i>Login</i> dengan kondisi alternatif dua  |
| Prosedur        | <ol style="list-style-type: none"> <li>Mengakses halaman <i>login</i>.</li> <li>Memasukkan <i>email</i> "entincahya96@gmail.com" dan <i>password</i> "123456".</li> </ol> |
| Expected Result | Aktor tidak bisa masuk ke dalam sistem dan menampilkan peringatan "email atau password salah".  |
| Result          | Aktor tidak bisa masuk ke dalam sistem dan menampilkan peringatan "email atau password salah".  |
| Status          | Valid   |





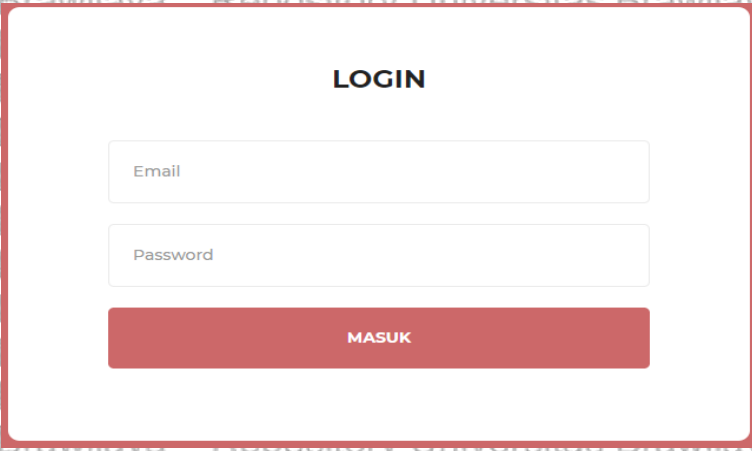


### 6.3.2 Pengujian Validasi Registrasi

#### 1. Pengujian validasi registrasi

Tabel 6.12 merupakan hasil dari pengujian validasi *use case* registrasi yang berhasil.

**Tabel 6.12 Pengujian Validasi Registrasi**

|                        |  |
|------------------------|--|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-02</b>   |
| Kasus Uji              | Registrasi   |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakses halaman Registrasi.</li> <li>2. Memasukkan <i>username</i> "entinendah"</li> <li>3. Memasukkan <i>email</i> "entinendah@gmail.com"</li> <li>4. Memasukkan <i>password</i> "123456"</li> <li>5. Memilih Daftar Menjadi "Klien"</li> <li>6. Menekan tombol DAFTAR</li> </ol> |
| <i>Expected Result</i> | Sistem berhasil menyimpan data registrasi yang telah dimasukkan oleh aktor, lalu sistem menampilkan halaman <i>login</i> .   |
| <i>Result</i>          | <p>Sistem berhasil menyimpan data registrasi yang telah dimasukkan oleh aktor, lalu sistem menampilkan halaman <i>login</i>.</p>   |
| Status                 | Valid  |

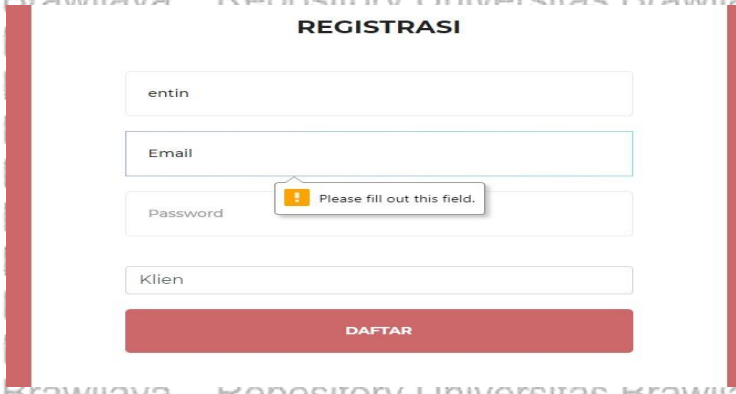
#### 2. Pengujian validasi registrasi alternatif satu

Tabel 6.13 merupakan hasil dari pengujian validasi *use case* registrasi yang tidak berhasil karena kolom pada *form* registrasi ada yang kosong.





Tabel 6.13 Pengujian Validasi Registrasi Alternatif Satu

|                 |  |
|-----------------|--|
| Kode Kebutuhan  | SKPP-F-02  |
| Kasus Uji       | Registrasi dengan kondisi alternatif satu  |
| Prosedur        | <ol style="list-style-type: none"> <li>1. Mengakses halaman Registrasi.</li> <li>2. Memasukkan <i>username</i> "entin"</li> <li>3. Memilih Daftar Menjadi "Klien"</li> <li>4. Menekan tombol DAFTAR tanpa mengisi email dan password.</li> </ol> |
| Expected Result | Sistem menampilkan pesan "Please fill out this field"  |
| Result          | <p>Sistem menampilkan pesan "Please fill out this field"</p>    |
| Status          | Valid  |

## 3. Pengujian validasi registrasi alternatif dua


Tabel 6.14 merupakan hasil dari pengujian validasi *use case* registrasi yang tidak berhasil karena menggunakan *username* yang sudah terdaftar pada sistem.

Tabel 6.14 Pengujian Validasi Registrasi Alternatif Dua

|                 |   |
|-----------------|---|
| Kode Kebutuhan  | SKPP-F-02   |
| Kasus Uji       | Registrasi dengan kondisi alternatif dua  |
| Prosedur        | <ol style="list-style-type: none"> <li>1. Mengakses halaman Registrasi</li> <li>2. Memasukkan <i>username</i> "entincahya"</li> <li>3. Memasukkan email "entinentin@gmail.com"</li> <li>4. Memilih Daftar Menjadi "Pengacara"</li> <li>5. Menekan tombol DAFTAR.</li> </ol> |
| Expected Result | Sistem menampilkan pesan "The Username field must contain a unique value"   |





|        |  |
|--------|--|
| Result | Sistem menampilkan pesan <i>"The Username field must contain a unique value"</i><br> |
| Status | Valid  |

#### 4. Pengujian validasi registrasi alternatif tiga

Tabel 6.15 merupakan hasil dari pengujian validasi *use case* registrasi yang tidak berhasil karena menggunakan *email* yang sudah terdaftar pada sistem.

**Tabel 6.15 Pengujian Validasi Registrasi Alternatif Tiga**

|                 |   |
|-----------------|---|
| Kode Kebutuhan  | SKPP-F-02   |
| Kasus Uji       | Registrasi dengan kondisi alternatif tiga   |
| Prosedur        | <ol style="list-style-type: none"> <li>1. Mengakses halaman Registrasi.</li> <li>2. Memasukkan <i>username</i> "entınca"</li> <li>3. Memasukkan email "entincanya96@gmail.com"</li> <li>4. Memilih Daftar Menjadi "Pengacara"</li> <li>5. Menekan tombol DAFTAR.</li> </ol> |
| Expected Result | Sistem menampilkan pesan <i>"The Email field must contain a unique value"</i>   |
| Result          | Sistem menampilkan pesan <i>"The Email field must contain a unique value"</i>   |





| REGISTRASI    |   |
|---------------|---|
| Username      | <input type="text"/>  |
| Email         | <input type="text"/>  |
|               | <small>The Email field must contain a unique value.</small> |
| Password      | <input type="password"/>                                    |
| Pengacara     | <input type="text"/>  |
| <b>DAFTAR</b> |   |
| Status        | Valid   |

### 6.3.3 Pengujian Validasi Logout

Tabel 6.16 merupakan hasil dari pengujian validasi *use case logout* yang berhasil.

**Tabel 6.16 Pengujian Validasi Logout**

|  |  |
|--|--|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-03</b>   |
| <b>Kasus Uji</b>   | <i>Logout</i>  |
| <b>Prosedur</b>  | <ol style="list-style-type: none"> <li>Memilih <i>dropdown username</i> yang terdapat pada <i>menu bar</i>.</li> <li>Memilih <i>Keluar</i> pada <i>dropdown username</i>.</li> </ol> |
| <b>Expected Result</b>   | Sistem memproses akun aktor untuk keluar dan menampilkan halaman <i>home</i> .   |
| <b>Result</b>  | Sistem memproses akun aktor untuk keluar dan menampilkan halaman <i>home</i> .   |
|  |  |
| Status   | Valid  |





### 6.3.4 Pengujian Validasi Tambah Info Hukum

#### 1. Pengujian validasi tambah info hukum

Tabel 6.17 merupakan hasil dari pengujian validasi *use case* tambah info hukum yang berhasil.

**Tabel 6.17 Pengujian Validasi Tambah Info Hukum**

| Kode Kebutuhan         | SKPP-F-04  |
|------------------------|--|
| Kasus Uji              | Tambah Info Hukum  |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Aktor mengakses halaman Postingan Saya.</li> <li>2. Menekan tombol Tambah Info Hukum.</li> <li>3. Memasukkan judul info hukum "Pengujian Info Hukum".</li> <li>4. Memasukkan isi info hukum "postingan ini dimaksudkan untuk menguji info hukum.".</li> <li>5. Memasukkan kategori info hukum "Kejahatan".</li> <li>6. Memasukkan foto info hukum.</li> <li>7. Menekan tombol Simpan.</li> </ol> |
| <i>Expected Result</i> | Sistem menyimpan data info hukum ke dalam <i>database</i> dan menampilkan halaman Postingan Saya.  |
| <i>Result</i>          | Sistem menyimpan data info hukum ke dalam <i>database</i> dan menampilkan halaman Postingan Saya.  |
| Status                 | Valid  |



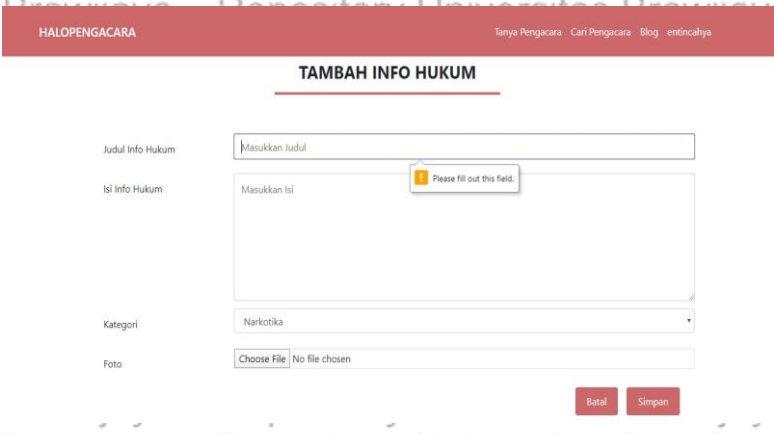
#### 2. Pengujian validasi tambah info hukum alternatif satu

Tabel 6.18 merupakan hasil dari pengujian validasi *use case* tambah info hukum yang tidak berhasil karena ada kolom pada *form* tambah info hukum yang kosong.





Tabel 6.18 Pengujian Validasi Tambah Info Hukum Alternatif Satu

|                 |   |
|-----------------|---|
| Kode Kebutuhan  | SKPP-F-04   |
| Kasus Uji       | Tambah Info Hukum dengan kondisi alternatif satu  |
| Prosedur        | <ol style="list-style-type: none"> <li>1. Aktor mengakses halaman Postingan Saya.</li> <li>2. Menekan tombol Tambah Info Hukum.</li> <li>3. Menekan tombol Simpan tanpa mengisi judul, isi, kategori, dan foto info hukum.</li> </ol> |
| Expected Result | Sistem menampilkan pesan <i>"Please fill out this field"</i> .  |
| Result          | <p>Sistem menampilkan pesan <i>"Please fill out this field"</i>.</p>   |
| Status          | Valid   |

### 6.3.5 Pengujian Validasi Edit Info Hukum

1. Pengujian validasi edit info hukum


Tabel 6.19 merupakan hasil dari pengujian validasi *use case* edit info hukum yang berhasil.

Tabel 6.19 Pengujian Validasi Edit Info Hukum

|                |   |
|----------------|---|
| Kode Kebutuhan | SKPP-F-05   |
| Kasus Uji      | Edit Info Hukum   |
| Prosedur       | <ol style="list-style-type: none"> <li>1. Mengakses halaman Postingan Saya.</li> <li>2. Menekan tombol Edit pada info hukum berjudul "Pengujian Info Hukum"</li> <li>3. Mengubah judul info hukum menjadi "Pengujian Info Hukum Dua"</li> <li>4. Mengubah Isi info hukum menjadi "postingan ini dimaksudkan untuk menguji info hukum dua."</li> </ol> |





|                        |   |
|------------------------|---|
|                        | 5. Menekan tombol Simpan.   |
| <i>Expected Result</i> | Sistem menyimpan perubahan info hukum ke dalam database dan menampilkan halaman Postingan Saya.   |
| <i>Result</i>          | Sistem mentimpan perubahan info hukum ke dalam database dan menampilkan halaman Postingan Saya.<br> |
| <i>Status</i>          | Valid   |

## 2. Pengujian validasi edit info hukum alternatif satu

Tabel 6.20 merupakan hasil dari pengujian validasi *use case* edit info hukum yang tidak berhasil karena ada kolom pada *form* edit info hukum yang kosong.

**Tabel 6.20 Pengujian Validasi Edi Info Hukum Alternatif Satu**

|                        |   |
|------------------------|---|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-05</b>  |
| Kasus Uji              | Edit Info Hukum dengan kondisi alternatif satu  |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakses halaman Postingan Saya.</li> <li>2. Menekan tombol Edit pada info hukum berjudul "Pengujian Info Hukum"</li> <li>3. Mengubah judul info hukum menjadi "Pengujian Info Hukum Dua"</li> <li>4. Menekan tombol Simpan tanpa mengisi kolom isi info hukum.</li> </ol> |
| <i>Expected Result</i> | Sistem menampilkan pesan " <i>Please fill out this field</i> ".   |
| <i>Result</i>          | Sistem menampilkan pesan " <i>Please fill out this field</i> ".   |





| EDIT INFO HUKUM  |  |
|------------------|--|
| Judul Info Hukum | <input type="text" value="Pengujian Info Hukum Dua"/>  |
| Isi Info Hukum   | <div style="border: 1px solid #ccc; height: 50px;"></div>  |
| Kategori         | Kejahatan <span style="border: 1px solid #ccc; padding: 2px;">Please fill out this field.</span> |
| Foto             | <input type="button" value="Choose File"/> No file chosen  |
| Status           | Valid  |

### 6.3.6 Pengujian Validasi Hapus Info Hukum

Tabel 6.21 merupakan hasil dari pengujian validasi *use case* hapus info hukum yang berhasil.

Tabel 6.21 Pengujian Validasi Hapus Info Hukum

|                        |  |
|------------------------|--|
| <b>Kode Kebutuhan</b>  | SKPP-F-06  |
| <b>Kasus Uji</b>       | Hapus Info Hukum   |
| <b>Prosedur</b>        | <ol style="list-style-type: none"> <li>1. Mengakses halaman Postingan Saya.</li> <li>2. Menekan tombol Hapus pada info hukum berjudul "Pengujian Info Hukum Dua".</li> </ol> |
| <b>Expected Result</b> | Sistem menghapus info hukum berjudul "Pengujian Info Hukum Dua" dan akan menampilkan halaman Postingan Saya.   |
| <b>Result</b>          | Sistem menghapus info hukum berjudul "Pengujian Info Hukum Dua" dan akan menampilkan halaman Postingan Saya.   |
| Status                 | Valid  |







### 6.3.7 Pengujian Validasi Kirim Portofolio

#### 1. Pengujian validasi kirim portofolio

Tabel 6.22 merupakan hasil dari pengujian validasi *use case* Kirim Portofolio yang berhasil.

**Tabel 6.22 Pengujian Validasi Kirim Portofolio**

| Kode Kebutuhan | SKPP-F-07  |
|----------------|--|
| Kasus Uji      | Kirim Portofolio   |
| Prosedur       | <ol style="list-style-type: none"> <li>1. Mengakses halaman Profil.</li> <li>2. Menekan tombol "Ajukan Portofolio".</li> <li>3. Memasukkan Nama Pengacara "Emma Watson".</li> <li>4. Memasukkan Tempat Lahir "Jember".</li> <li>5. Memasukkan Nomor Pokok Wajib Pajak (NPWP) "97.988.577.8-123.000".</li> <li>6. Memasukkan Alamat Kantor "Jln sumpersari no167".</li> <li>7. Memasukkan Provinsi dengan memilih <i>select option</i> "JAWA TIMUR".</li> <li>8. Memasukkan Kabupaten dengan memilih <i>select option</i> "KABUPATEN JEMBER".</li> <li>9. Memasukkan Kecamatan dengan memilih <i>select option</i> "SUMBERSARI".</li> <li>10. Memasukkan Kelurahan dengan memilih <i>select option</i> "SUMBERSARI".</li> <li>11. Memasukkan <i>email</i> Kantor "emmawatsOn@gmail.com"</li> <li>12. Memasukkan Nomor Telepon "865745".</li> <li>13. Memasukkan Nomor <i>Handphone</i> "085678934000".</li> <li>14. Memasukkan Pendidikan Hukum "S.H dari Universitas Jember".</li> <li>15. Memasukkan Pengalaman "Menjabat sebagai pengacara berprestasi 2019".</li> <li>16. Memasukkan Tentang Saya "Percayakan kasus anda dengan saya."</li> <li>17. Mencentang <i>checkbox</i> Bidang Keahlian dan Spesialis "Keluarga", "Bisnis", dan "Kekayaan Intelektual".</li> <li>18. Memasukkan file Scan KTP dengan tipe file pdf.</li> </ol> |





|                        |  |
|------------------------|--|
|                        | <p>19. Memasukkan file Ijasah Sarjana Hukum dengan tipe file pdf.</p> <p>20. Memasukkan file Sertifikat Lulus UPA dengan tipe file pdf.</p> <p>21. Menekan tombol Kirim.</p> |
| <i>Expected Result</i> | Sistem menyimpan portofolio yang dikirimkan aktor ke dalam database dan menampilkan halaman Profil dengan pesan "Ajukan Portofolio Berhasil"                                 |
| <i>Result</i>          | <p>Sistem menyimpan portofolio yang dikirimkan aktor ke dalam database dan menampilkan halaman Profil dengan pesan "Ajukan Portofolio Berhasil".</p>                         |
| <i>Status</i>          | Valid  |

## 2. Pengujian validasi kirim portofolio alternatif satu

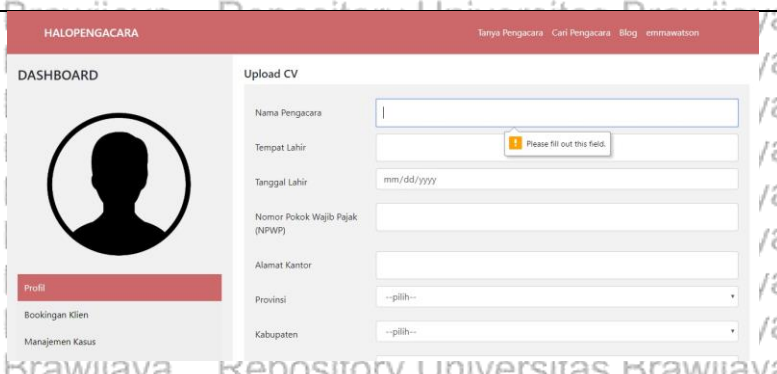
Tabel 6.23 merupakan hasil dari pengujian validasi *use case* kirim portofolio yang tidak berhasil karena ada kolom pada *form* ajukan portofolio yang kosong.

**Tabel 6.23 Pengujian Validasi Kirim Portofolio Alternatif Satu**

|                        |  |
|------------------------|--|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-07</b>   |
| Kasus Uji              | Kirim Portofolio dengan kondisi alternatif satu  |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakses halaman Profil.</li> <li>2. Menekan tombol "Ajukan Portofolio"</li> <li>3. Menekan tombol Kirim tanpa mengisi email dan password.</li> </ol> |
| <i>Expected Result</i> | Sistem menampilkan pesan " <i>Please fill out this field</i> ".  |
| <i>Result</i>          | Sistem menampilkan pesan " <i>Please fill out this field</i> ".  |





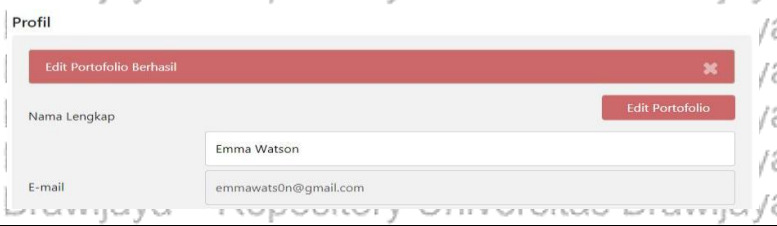
|  |       |
|--|-------|
|  |       |
| Status   | Valid |

### 6.3.8 Pengujian Validasi Edit Portofolio

#### 1. Pengujian validasi edit portofolio

Tabel 6.24 merupakan hasil dari pengujian validasi *use case* edit portofolio yang berhasil.

**Tabel 6.24 Pengujian Validasi Edit Portofolio**

|                        |  |
|------------------------|--|
| Kode Kebutuhan         | SKPP-F-08  |
| Kasus Uji              | Edit Portofolio  |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakses halaman Profil.</li> <li>2. Menekan tombol "Edit Portofolio"</li> <li>3. Mengubah Tempat Lahir menjadi "Banyuwangi"</li> <li>4. Menekan tombol <i>Update</i>.</li> </ol> |
| <i>Expected Result</i> | Sistem menyimpan perubahan portofolio ke dalam database dan menampilkan halaman Profil dengan pesan "Edit Portofolio Berhasil".  |
| <i>Result</i>          | <p>Sistem menyimpan perubahan portofolio ke dalam database dan menampilkan halaman Profil dengan pesan "Edit Portofolio Berhasil".</p>   |
| Status                 | Valid  |

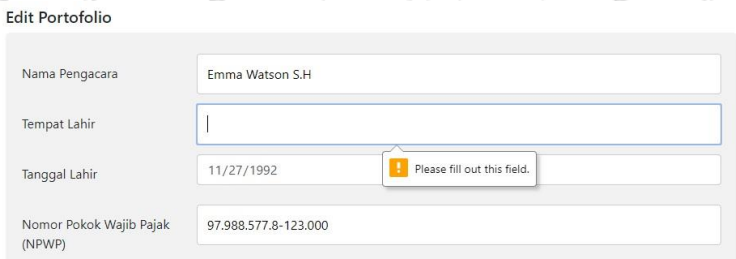




## 2. Pengujian validasi edit portofolio alternatif satu

Tabel 6.25 merupakan hasil dari pengujian validasi *use case* edit portofolio yang tidak berhasil karena ada kolom pada *form* edit portofolio yang kosong.

**Tabel 6.25 Pengujian Validasi Edit Portofolio Alternatif Satu**

|                        |  |
|------------------------|--|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-08</b>   |
| Kasus Uji              | Edit Portofolio dengan kondisi alternatif satu   |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakses halaman Profil.</li> <li>2. Menekan tombol "Edit Portofolio".</li> <li>3. Mengkosongkan kolom Tempat Lahir.</li> <li>4. Menekan tombol <i>Update</i>.</li> </ol> |
| <i>Expected Result</i> | Sistem menampilkan pesan " <i>Please fill out this field</i> ".  |
| <i>Result</i>          | <p>Sistem menampilkan pesan "<i>Please fill out this field</i>".</p>    |
| Status                 | Valid  |

## 6.3.9 Pengujian Validasi Lihat Daftar Pertanyaan

Tabel 6.26 merupakan hasil dari pengujian validasi *use case* lihat daftar pengacara yang berhasil.

**Tabel 6.26 Pengujian Validasi Lihat Daftar Pertanyaan**

|                        |  |
|------------------------|--|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-09</b>   |
| Kasus Uji              | Lihat Daftar Pertanyaan  |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakses halaman <i>home</i>.</li> <li>2. Memilih menu "Tanya Pengacara" pada <i>menu bar</i>.</li> </ol> |
| <i>Expected Result</i> | Sistem menampilkan daftar pertanyaan yang diajukan oleh Klien.   |
| <i>Result</i>          | Sistem menampilkan daftar pertanyaan yang diajukan oleh Klien.   |





|        |       |
|--------|-------|
|        |       |
| Status | Valid |

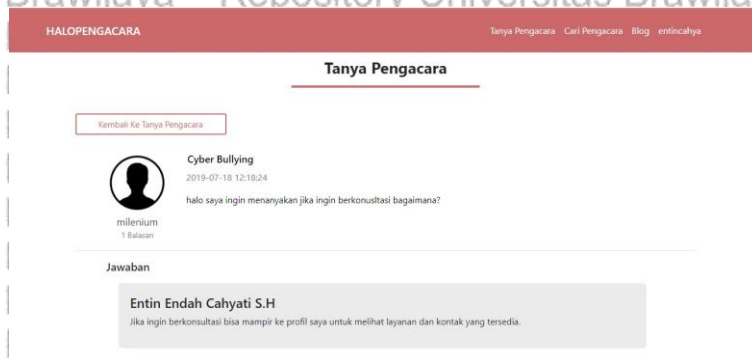
### 6.3.10 Pengujian Validasi Jawab Pertanyaan

#### 1. Pengujian validasi jawab pertanyaan

Tabel 6.27 merupakan hasil dari pengujian validasi *use case* jawab pertanyaan yang berhasil.

**Tabel 6.27 Pengujian Validasi Jawab Pertanyaan**

|                        |   |
|------------------------|---|
| Kode Kebutuhan         | SKPP-F-10   |
| Kasus Uji              | Jawab Pertanyaan  |
| Prosedur               | <ol style="list-style-type: none"> <li>Mengakses halaman Tanya Pengacara.</li> <li>Memilih pertanyaan dengan judul "<i>Cyber Bullying</i>".</li> <li>Mengisikan jawaban pada kolom Jawaban Pertanyaan "Jika ingin berkonsultasi bisa mampir ke profil saya untuk melihat layanan dan kontak yang tersedia."</li> <li>Menekan tombol Kirim.</li> </ol> |
| <i>Expected Result</i> | Sistem menyimpan jawaban pertanyaan ke dalam <i>database</i> dan menampilkan jawaban dari pertanyaan.   |
| <i>Result</i>          | Sistem menyimpan jawaban pertanyaan ke dalam <i>database</i> dan menampilkan jawaban dari pertanyaan.   |
| Status                 | Valid   |








## 2. Pengujian validasi jawab pertanyaan alternatif satu

Tabel 6.28 merupakan hasil dari pengujian validasi *use case* jawab pertanyaan yang tidak berhasil karena ada kolom pada *form* jawaban pertanyaan yang kosong.

**Tabel 6.28 Pengujian Validasi Jawab Pertanyaan Alternatif Satu**

|                        |  |
|------------------------|--|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-10</b>   |
| Kasus Uji              | Jawab Pertanyaan dengan kondisi alternatif satu  |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakses halaman Tanya Pengacara.</li> <li>2. Memilih pertanyaan dengan judul "<i>Cyber Bullying</i>".</li> <li>3. Menekan tombol Kirim tanpa mengisikan kolom Jawab Pertanyaan.</li> </ol> |
| <i>Expected Result</i> | Sistem menampilkan pesan " <i>Please fill out this field</i> ".  |
| <i>Result</i>          | Sistem menampilkan pesan " <i>Please fill out this field</i> ".<br>   |
| Status                 | Valid  |

## 6.3.11 Pengujian Validasi Lihat Bookingan Klien

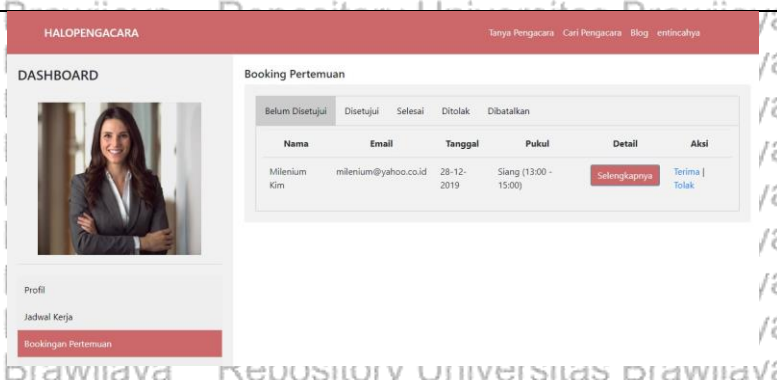
Tabel 6.29 merupakan hasil dari pengujian validasi *use case* lihat bookingan klien yang berhasil.

**Tabel 6.29 Pengujian Validasi Lihat Bookingan Klien**

|                        |  |
|------------------------|--|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-11</b>   |
| Kasus Uji              | Lihat Bookingan Klien  |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakses halaman <i>Dashboard</i>.</li> <li>2. Memilih menu Booking Pertemuan pada <i>sidebar</i>.</li> <li>3. Memilih <i>tablink</i> Belum Disetujui.</li> </ol> |
| <i>Expected Result</i> | Sistem menampilkan daftar bookingan klien yang belum disetujui.  |
| <i>Result</i>          | Sistem menampilkan daftar bookingan klien yang belum disetujui.  |



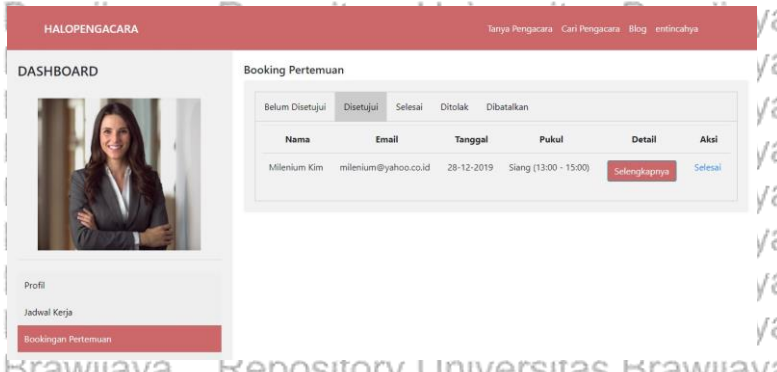


|  |              |
|--|--------------|
|  |              |
| <p>Status</p>  | <p>Valid</p> |

### 6.3.12 Pengujian Validasi Terima Bookingan Klien

Tabel 6.30 merupakan hasil dari pengujian validasi *use case* terima bookingan klien yang berhasil.

**Tabel 6.30 Pengujian Validasi Terima Bookingan Klien**

|  |   |
|--|---|
| <p>Kode Kebutuhan</p>  | <p>SKPP-F-12</p>  |
| <p>Kasus Uji</p>   | <p>Terima Bookingan Klien</p>   |
| <p>Prosedur</p>  | <ol style="list-style-type: none"> <li>1. Mengakses halaman Bookingan Pertemuan.</li> <li>2. Memilih <i>tablink</i> Belum Disetujui.</li> <li>3. Memilih bookingan klien yang akan disetujui.</li> <li>4. Menekan <i>link</i> Terima pada bookingan klien.</li> </ol> |
| <p><i>Expected Result</i></p>  | <p>Sistem menyimpan data bookingan diterima pada <i>database</i> dan menampilkan halaman Bookingan Pertemuan.</p>   |
| <p><i>Result</i></p>   | <p>Sistem menyimpan data bookingan diterima pada <i>database</i> dan menampilkan halaman Bookingan Pertemuan.</p>   |
|  |   |
| <p>Status</p>  | <p>Valid</p>  |





### 6.3.13 Pengujian Validasi Tolak Bookingan Klien

#### 1. Pengujian validasi tolak bookingan klien

Tabel 6.31 merupakan hasil dari pengujian validasi *use case* tolak bookingan klien yang berhasil.

**Tabel 6.31 Pengujian Validasi Tolak Bookingan Klien**

|                        |   |
|------------------------|---|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-13</b>  |
| Kasus Uji              | Tolak Bookingan Klien   |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakses halaman Booking Pertemuan.</li> <li>2. Memilih <i>tablink</i> Belum Disetujui.</li> <li>3. Memilih bookingan klien yang akan ditolak.</li> <li>4. Menekan <i>link</i> Tolak pada bookingan klien.</li> <li>5. Memasukkan alasan menolak bookingan klien “Maaf, saya tidak bisa menerima konsultasi tatap muka anda.”</li> <li>6. Menekan tombol Simpan.</li> </ol>  |
| <i>Expected Result</i> | Sistem menyimpan data tolak bookingan klien pada <i>database</i> dan menampilkan halaman Bookingan Pertemuan.   |
| <i>Result</i>          | Sistem menyimpan data tolak bookingan klien pada <i>database</i> dan menampilkan halaman Bookingan Pertemuan.   |
|                        | <p>The screenshot shows a web dashboard for 'HALOPENGACARA'. On the left is a user profile for 'Milenium Kim' with a photo and contact information. On the right, under 'Booking Pertemuan', there is a table with columns: 'Belum Disetujui', 'Disetujui', 'Selesai', 'Ditolak', and 'Dibatalkan'. The 'Ditolak' column is active. A table entry shows 'Milenium Kim' with email 'milenium@yahoo.co.id', date '27-12-2019', and time 'Sore (15:00 - 17:00)'. A red button labeled 'Selengkapnya' is next to the entry.</p> |
| Status                 | Valid   |

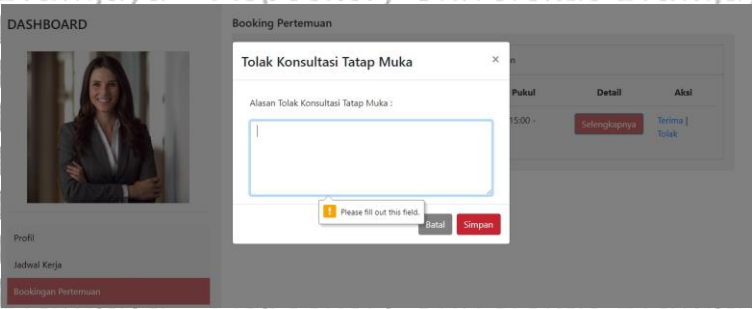
#### 2. Pengujian validasi tolak bookingan klien alternatif satu

Tabel 6.32 merupakan hasil dari pengujian validasi *use case* tolak bookingan klien yang tidak berhasil karena ada kolom pada *form* alasan tolak bookingan klien yang kosong.





Tabel 6.32 Pengujian Validasi Tolak Bookingan Klien Alternatif Satu

|                 |   |
|-----------------|---|
| Kode Kebutuhan  | SKPP-F-13   |
| Kasus Uji       | Tolak Bookingan Klien dengan kondisi alternatif satu  |
| Prosedur        | <ol style="list-style-type: none"> <li>1. Mengakses halaman Booking Pertemuan.</li> <li>2. Memilih <i>tablink</i> Belum Disetujui.</li> <li>3. Memilih bookingan klien yang akan ditolak.</li> <li>4. Menekan <i>link</i> Tolak pada bookingan klien.</li> <li>5. Menekan tombol Simpan tanpa memasukkan alasan tolak bookingan klien.</li> </ol> |
| Expected Result | Sistem menampilkan pesan “Please fill out this field”.  |
| Result          | <p>Sistem menampilkan pesan “Please fill out this field”.</p>    |
| Status          | Valid   |

### 6.3.14 Pengujian Validasi Lihat Ulasan Klien

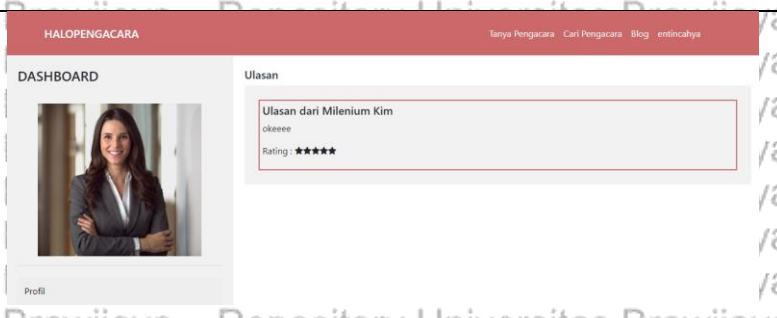
Tabel 6.33 merupakan hasil dari pengujian validasi *use case* lihat ulasan klien yang berhasil.

Tabel 6.33 Pengujian Validasi Lihat Ulasan Klien

|                 |  |
|-----------------|--|
| Kode Kebutuhan  | SKPP-F-14  |
| Kasus Uji       | Lihat Ulasan Klien   |
| Prosedur        | <ol style="list-style-type: none"> <li>1. Mengakses halaman <i>dashboard</i>.</li> <li>2. Memilih menu Ulasan pada menu <i>sidebar</i>.</li> </ol> |
| Expected Result | Sistem menampilkan ulasan yang diberikan oleh klien.   |
| Result          | Sistem menampilkan ulasan yang diberikan oleh klien.   |





|  |              |
|--|--------------|
|  |              |
| <p>Status</p>  | <p>Valid</p> |

### 6.3.15 Pengujian Validasi Lihat Riwayat Klien

Tabel 6.34 merupakan hasil dari pengujian validasi *use case* lihat riwayat klien yang berhasil.

Tabel 6.34 Pengujian Validasi Lihat Riwayat Klien

| <p>Kode Kebutuhan</p>         | <p>SKPP-F-15</p>  |               |       |               |           |            |               |           |            |               |           |            |               |
|-------------------------------|---|---------------|-------|---------------|-----------|------------|---------------|-----------|------------|---------------|-----------|------------|---------------|
| <p>Kasus Uji</p>              | <p>Lihat Riwayat Klien</p>  |               |       |               |           |            |               |           |            |               |           |            |               |
| <p>Prosedur</p>               | <ol style="list-style-type: none"> <li>1. Mengakases halaman <i>dashboard</i>.</li> <li>2. Memilih menu Riwayat Klien pada menu <i>sidebar</i>.</li> </ol>  |               |       |               |           |            |               |           |            |               |           |            |               |
| <p><i>Expected Result</i></p> | <p>Sistem menampilkan riwayat klien yang pernah ditangani pengacara.</p>  |               |       |               |           |            |               |           |            |               |           |            |               |
| <p><i>Result</i></p>          | <p>Sistem menampilkan riwayat klien yang pernah ditangani pengacara.</p> <div data-bbox="576 1279 1334 1514"> <p>Riwayat Klien</p> <table border="1"> <thead> <tr> <th>Nama</th> <th>Kasus</th> <th>Selesai Kasus</th> </tr> </thead> <tbody> <tr> <td>Esmeralda</td> <td>Perceraian</td> <td>22 April 2019</td> </tr> <tr> <td>Esmeralda</td> <td>Perceraian</td> <td>22 April 2019</td> </tr> <tr> <td>Esmeralda</td> <td>Perceraian</td> <td>22 April 2019</td> </tr> </tbody> </table> </div> | Nama          | Kasus | Selesai Kasus | Esmeralda | Perceraian | 22 April 2019 | Esmeralda | Perceraian | 22 April 2019 | Esmeralda | Perceraian | 22 April 2019 |
| Nama                          | Kasus   | Selesai Kasus |       |               |           |            |               |           |            |               |           |            |               |
| Esmeralda                     | Perceraian  | 22 April 2019 |       |               |           |            |               |           |            |               |           |            |               |
| Esmeralda                     | Perceraian  | 22 April 2019 |       |               |           |            |               |           |            |               |           |            |               |
| Esmeralda                     | Perceraian  | 22 April 2019 |       |               |           |            |               |           |            |               |           |            |               |
| <p>Status</p>                 | <p>Valid</p>  |               |       |               |           |            |               |           |            |               |           |            |               |

### 6.3.16 Pengujian Validasi Laporkan Klien

1. Pengujian validasi laporkan klien

Tabel 6.35 merupakan hasil dari pengujian validasi *use case* laporkan klien yang berhasil.

Tabel 6.35 Pengujian Validasi Laporkan Klien

|                       |                  |
|-----------------------|------------------|
| <p>Kode Kebutuhan</p> | <p>SKPP-F-16</p> |
|-----------------------|------------------|





|                 |  |
|-----------------|--|
| Kasus Uji       | Laporkan Klien   |
| Prosedur        | <ol style="list-style-type: none"> <li>1. Mengakses halaman Booking Pertemuan atau Manajemen Kasus.</li> <li>2. Memilih <i>tablink</i> Selesai pada halaman Booking Pertemuan.</li> <li>3. Menekan tombol Selengkapnya pada klien bernama "Molaaayo".</li> <li>4. Menekan <i>link</i> "Laporkan Klien disini jika ada masalah dengan klien!"</li> <li>5. Memasukkan alasan melaporkan klien "masalah klien tidak benar".</li> <li>6. Menekan tombol Simpan.</li> </ol> |
| Expected Result | Sistem menyimpan data laporan klien pada <i>database</i> dan menampilkan halaman Bookingan Pertemuan.  |
| Result          | <p>Sistem menyimpan data laporan klien pada <i>database</i> dan menampilkan halaman Bookingan Pertemuan.</p>   |
| Status          | Valid  |

## 2. Pengujian validasi laporkan klien alternatif satu

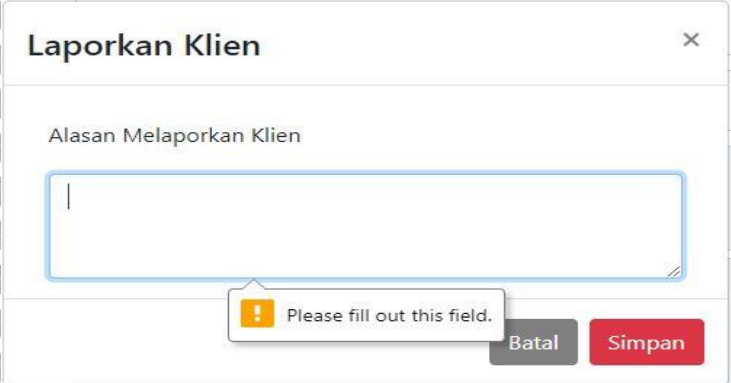
Tabel 6.36 merupakan hasil dari pengujian validasi *use case* laporkan klien yang tidak berhasil karena ada kolom pada *form* laporkan klien yang kosong.

**Tabel 6.36 Pengujian Validasi Laporkan Klien Alternatif Satu**

| Kode Kebutuhan | SKPP-F-16   |
|----------------|---|
| Kasus Uji      | Laporkan Klien dengan kondisi alternatif satu   |
| Prosedur       | <ol style="list-style-type: none"> <li>1. Mengakses halaman Booking Pertemuan atau Manajemen Kasus.</li> <li>2. Memilih <i>tablink</i> Selesai pada halaman Booking Pertemuan.</li> <li>3. Menekan tombol Selengkapnya pada klien bernama "Molaaayo".</li> <li>4. Menekan <i>link</i> "Laporkan Klien disini jika ada masalah dengan klien!"</li> </ol> |



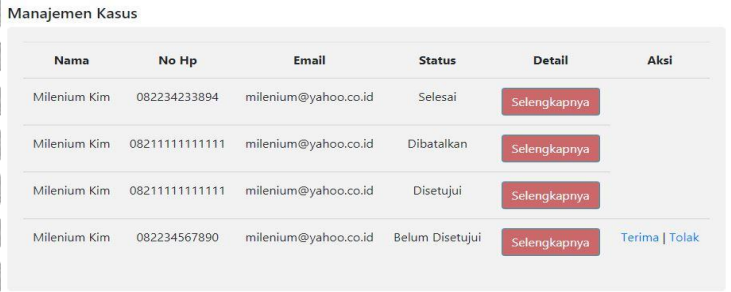


|                        |  |
|------------------------|--|
|                        | 5. Menekan tombol Simpan tanpa mengisi alasan melaporkan klien.                    |
| <i>Expected Result</i> | Sistem menampilkan pesan “ <i>Please fill out this field</i> ”.                    |
| <i>Result</i>          | Sistem menampilkan pesan “ <i>Please fill out this field</i> ”.                    |
|                        |  |
| <i>Status</i>          | Valid  |

### 6.3.17 Pengujian Validasi Lihat Jasa Pengacara

Tabel 6.37 merupakan hasil dari pengujian validasi *use case* lihat jasa pengacara yang dipesan oleh klien.

**Tabel 6.37 Pengujian Validasi Lihat Jasa Pengacara**

|                        |  |
|------------------------|--|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-17</b>   |
| <b>Kasus Uji</b>       | Lihat Jasa Pengacara   |
| <b>Prosedur</b>        | <ol style="list-style-type: none"> <li>Mengakases halaman <i>dashboard</i>.</li> <li>Memilih menu Manajemen Kasus pada menu <i>sidebar</i>.</li> </ol> |
| <i>Expected Result</i> | Sistem menampilkan daftar klien yang menggunakan layanan jasa pengacara.   |
| <i>Result</i>          | Sistem menampilkan daftar klien yang menggunakan layanan jasa pengacara.   |
|                        |    |
| <i>Status</i>          | Valid  |

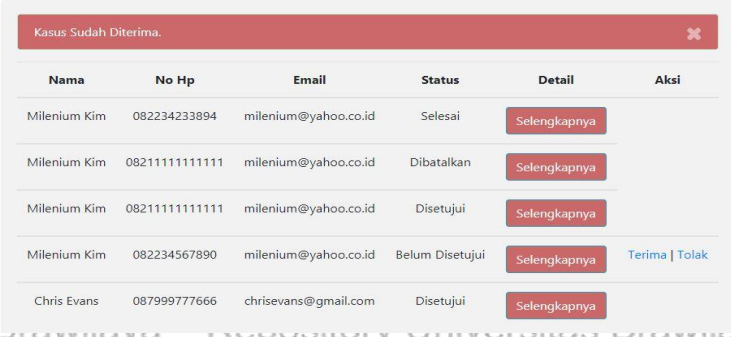




### 6.3.18 Pengujian Validasi Terima Jasa Pengacara

Tabel 6.38 merupakan hasil dari pengujian validasi *use case* terima jasa pengacara untuk menerima jasa pengacara yang dikirimkan klien.

**Tabel 6.38 Pengujian Validasi Lihat Jasa Pengacara**

|                        |  |
|------------------------|--|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-18</b>   |
| Kasus Uji              | Terima Jasa Pengacara  |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakases halaman Manajemen Kasus.</li> <li>2. Memilih kasus dengan nama "Chris Evans".</li> <li>3. Menekan <i>link</i> Terima.</li> </ol>  |
| <i>Expected Result</i> | Sistem menyimpan data jasa pengacara diterima pada <i>database</i> dan menampilkan halaman Manajemen Kasus dengan pesan "Kasus Sudah Diterima".  |
| <i>Result</i>          | <p>Sistem menyimpan data jasa pengacara diterima pada <i>database</i> dan menampilkan halaman Manajemen Kasus dengan pesan "Kasus Sudah Diterima".</p> <p>Manajemen Kasus</p>  |
| Status                 | Valid  |

### 6.3.19 Pengujian Validasi Tolak Jasa Pengacara

Tabel 6.39 merupakan hasil dari pengujian validasi *use case* tolak jasa pengacara untuk menolak jasa pengacara yang dikirimkan klien.

**Tabel 6.39 Pengujian Validasi Tolak Jasa Pengacara**

|                       |  |
|-----------------------|--|
| <b>Kode Kebutuhan</b> | <b>SKPP-F-19</b>   |
| Kasus Uji             | Tolak Jasa Pengacara   |
| Prosedur              | <ol style="list-style-type: none"> <li>1. Mengakases halaman Manajemen Kasus.</li> <li>2. Memilih kasus dengan nama "Chris Evans".</li> <li>3. Menekan <i>link</i> Tolak.</li> </ol> |





|                        |   |
|------------------------|---|
| <i>Expected Result</i> | Sistem menyimpan data jasa pengacara ditolak pada <i>database</i> dan menampilkan halaman Manajemen Kasus dengan pesan “Kasus Sudah Ditolak”. |
| <i>Result</i>          | Sistem menyimpan data jasa pengacara ditolak pada <i>database</i> dan menampilkan halaman Manajemen Kasus dengan pesan “Kasus Sudah Ditolak”. |
| <i>Status</i>          | Valid   |

Manajemen Kasus

Kasus Sudah Ditolak.

| Nama         | No Hp          | Email                | Status          | Detail | Aksi                        |
|--------------|----------------|----------------------|-----------------|--------|-----------------------------|
| Milenium Kim | 082234233894   | milenium@yahoo.co.id | Selesai         |        | Selengkapnya                |
| Milenium Kim | 08211111111111 | milenium@yahoo.co.id | Dibatalkan      |        | Selengkapnya                |
| Milenium Kim | 08211111111111 | milenium@yahoo.co.id | Disetujui       |        | Selengkapnya                |
| Milenium Kim | 082234567890   | milenium@yahoo.co.id | Belum Disetujui |        | Selengkapnya Terima   Tolak |
| Chris Evans  | 087999777666   | chrisevans@gmail.com | Ditolak         |        | Selengkapnya                |

### 6.3.20 Pengujian Validasi Tambah Jadwal Kerja

#### 1. Pengujian validasi tambah jadwal kerja

Tabel 6.40 merupakan hasil dari pengujian validasi *use case* tambah jadwal kerja yang berhasil.

**Tabel 6.40 Pengujian Validasi Tambah Jadwal Kerja**

|                        |  |
|------------------------|--|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-20</b>   |
| Kasus Uji              | Tambah Jadwal Kerja  |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakses halaman Jadwal Kerja.</li> <li>2. Memasukkan tanggal “09-01-2020”.</li> <li>3. Memilih <i>select option</i> Jam Kerja “Siang (13:00 – 15:00)”.</li> <li>4. Menekan tombol Simpan.</li> </ol> |
| <i>Expected Result</i> | Sistem menyimpan jadwal kerja ke dalam <i>database</i> dan menampilkannya pada tabel jadwal kerja <i>available</i> serta menampilkan pesan “Jadwal Kerja Berhasil Ditambahkan.”  |
| <i>Result</i>          | Sistem menyimpan jadwal kerja ke dalam <i>database</i> dan menampilkannya pada tabel jadwal kerja <i>available</i> serta menampilkan pesan “Jadwal Kerja Berhasil Ditambahkan.”  |





|               | <p>HALOPENGACARA</p> <p>Tanya Pengacara Cari Pengacara Blog nabilafauziah</p> <p>DASHBOARD</p> <p>Jadwal Kerja</p> <p>Jadwal Kerja Berhasil Ditambahkan.</p> <p>Input Jadwal Kerja</p> <p>Tanggal</p> <p>Jam Kerja</p> <ul style="list-style-type: none"> <li>Pagi (09:00 - 12:00)</li> <li>Siang (13:00 - 15:00)</li> <li>Sore (15:00 - 17:00)</li> <li>Malam (18:00 - 20:00)</li> </ul> <p>Simpan</p> <p>Jadwal Kerja Available</p> <table border="1"> <thead> <tr> <th>Tanggal</th> <th>Jam</th> <th>Status</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>09-01-2020</td> <td>Siang (13:00 - 15:00)</td> <td>available</td> <td>Edit   Hapus</td> </tr> </tbody> </table> | Tanggal   | Jam          | Status | Aksi | 09-01-2020 | Siang (13:00 - 15:00) | available | Edit   Hapus |
|---------------|---|-----------|--------------|--------|------|------------|-----------------------|-----------|--------------|
| Tanggal       | Jam   | Status    | Aksi         |        |      |            |                       |           |              |
| 09-01-2020    | Siang (13:00 - 15:00)   | available | Edit   Hapus |        |      |            |                       |           |              |
| <p>Status</p> | <p>Valid</p>  |           |              |        |      |            |                       |           |              |

2. Pengujian validasi tambah jadwal kerja alternatif satu

Tabel 6.41 merupakan hasil dari pengujian validasi *use case* tambah jadwal kerja yang tidak berhasil karena ada kolom pada *form* tambah jadwal kerja yang kosong.

Tabel 6.41 Pengujian Validasi Tambah Jadwal Kerja Alternatif Satu

|                        |   |
|------------------------|---|
| <p>Kode Kebutuhan</p>  | <p>SKPP-F-20</p>  |
| <p>Kasus Uji</p>       | <p>Tambah Jadwal Kerja dengan kondisi alternatif satu</p>   |
| <p>Prosedur</p>        | <ol style="list-style-type: none"> <li>Mengakses halaman Jadwal Kerja.</li> <li>Memasukkan tanggal "09-01-2020".</li> <li>Menekan tombol Simpan tanpa mengisi tanggal dan jam kerja.</li> </ol> |
| <p>Expected Result</p> | <p>Sistem menampilkan pesan "Please fill out this field".</p>   |
| <p>Result</p>          | <p>Sistem menampilkan pesan "Please fill out this field".</p>   |
| <p>Status</p>          | <p>Valid</p>  |



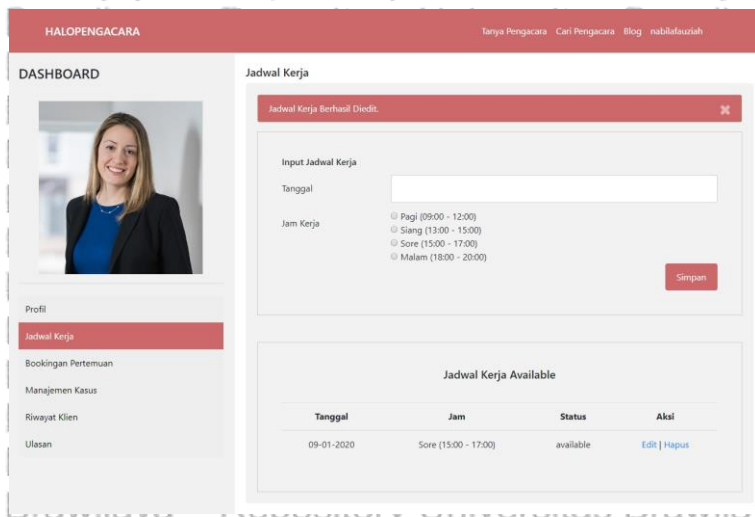


### 6.3.21 Pengujian Validasi Edit Jadwal Kerja

#### 1. Pengujian validasi edit jadwal kerja

Tabel 6.42 merupakan hasil dari pengujian validasi *use case* edit jadwal kerja yang berhasil.

**Tabel 6.42 Pengujian Validasi Edit Jadwal Kerja**

| Kode Kebutuhan         | SKPP-F-21   |
|------------------------|---|
| Kasus Uji              | Edit Jadwal Kerja.  |
| Prosedur               | <ol style="list-style-type: none"> <li>Mengakses halaman Jadwal Kerja.</li> <li>Memilih jadwal kerja pada tanggal "09-01-2010".</li> <li>Menekan <i>link</i> Edit.</li> <li>Mengganti Jam Kerja menjadi "Sore (15:00 – 17:00)" pada modal Edit Jadwal Kerja.</li> <li>Menekan tombol Simpan.</li> </ol> |
| <i>Expected Result</i> | Sistem menyimpan jadwal kerja ke dalam <i>database</i> dan menampilkannya pada tabel jadwal kerja <i>available</i> serta menampilkan pesan "Jadwal Kerja Berhasil Diedit."  |
| <i>Result</i>          | Sistem menyimpan jadwal kerja ke dalam <i>database</i> dan menampilkannya pada tabel jadwal kerja <i>available</i> serta menampilkan pesan "Jadwal Kerja Berhasil Diedit."  |
|                        |   |
| Status                 | Valid   |

#### 2. Pengujian validasi edit jadwal kerja alternatif satu

Tabel 6.43 merupakan hasil dari pengujian validasi *use case* edit jadwal kerja yang tidak berhasil karena ada kolom pada *form* edit jadwal kerja yang kosong.





Tabel 6.43 Pengujian Validasi Edit Jadwal Kerja Alternatif Satu

|                        |   |
|------------------------|---|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-21</b>  |
| Kasus Uji              | Edit Jadwal Kerja dengan kondisi alternatif satu  |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakses halaman Jadwal Kerja.</li> <li>2. Memilih jadwal kerja pada tanggal "09-01-2010".</li> <li>3. Menekan <i>link</i> Edit.</li> <li>4. Menekan tombol Simpan tanpa mengisi tanggal.</li> </ol> |
| <i>Expected Result</i> | Sistem menampilkan pesan " <i>Please fill out this field</i> ".   |
| <i>Result</i>          | <p>Sistem menampilkan pesan "<i>Please fill out this field</i>".</p>   |
| Status                 | Valid   |

## 3. Pengujian validasi edit jadwal kerja alternatif dua

Tabel 6.44 merupakan hasil dari pengujian validasi *use case* edit jadwal kerja yang tidak berhasil karena membatalkan edit jadwal kerja.

Tabel 6.44 Pengujian Validasi Edit Jadwal Kerja Alternatif Dua

|                        |  |
|------------------------|--|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-21</b>   |
| Kasus Uji              | Edit Jadwal Kerja dengan kondisi alternatif dua  |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakses halaman Jadwal Kerja.</li> <li>2. Memilih jadwal kerja pada tanggal "24-06-2020".</li> <li>3. Menekan <i>link</i> Edit.</li> <li>4. Menekan tombol Batal.</li> </ol> |
| <i>Expected Result</i> | Sistem menampilkan halaman Jadwal Kerja  |
| <i>Result</i>          | Sistem menampilkan halaman Jadwal Kerja  |





| Jadwal Kerja Available |                       |           |  |
|------------------------|-----------------------|-----------|--|
| Tanggal                | Jam                   | Status    | Aksi   |
| 24-06-2020             | Siang (13:00 - 15:00) | available | <a href="#">Edit</a>   <a href="#">Hapus</a> |
| 24-06-2020             | Pagi (09:00 - 12:00)  | available | <a href="#">Edit</a>   <a href="#">Hapus</a> |

|        |       |
|--------|-------|
| Status | Valid |
|--------|-------|

### 6.3.22 Pengujian Validasi Hapus Jadwal Kerja

Tabel 6.45 merupakan hasil dari pengujian validasi *use case* hapus jadwal kerja yang berhasil.

**Tabel 6.45 Pengujian Validasi Hapus Jadwal Kerja**

|                 |   |
|-----------------|---|
| Kode Kebutuhan  | SKPP-F-22   |
| Kasus Uji       | Hapus Jadwal Kerja.   |
| Prosedur        | <ol style="list-style-type: none"> <li>1. Mengakses halaman Jadwal Kerja.</li> <li>2. Memilih jadwal kerja pada tanggal "10-01-2020".</li> <li>3. Menekan <i>link</i> Hapus.</li> </ol> |
| Expected Result | Sistem menyimpan perubahan ke dalam <i>database</i> dan menghapus jadwal kerja pada tabel jadwal kerja <i>available</i> serta menampilkan pesan "Jadwal Kerja Berhasil Dihapus".        |
| Result          | Sistem menyimpan perubahan ke dalam <i>database</i> dan menghapus jadwal kerja pada tabel jadwal kerja <i>available</i> serta menampilkan pesan "Jadwal Kerja Berhasil Dihapus".        |
| Status          | Valid   |

The screenshot shows a user dashboard for 'HALOPENGACARA'. On the right, a modal window displays a success message: 'Jadwal Kerja Berhasil Dihapus.' Below this is a form for 'Input Jadwal Kerja' with fields for 'Tanggal' and 'Jam Kerja'. The 'Jam Kerja' field has radio button options: 'Pagi (09:00 - 12:00)', 'Siang (13:00 - 15:00)', 'Sore (15:00 - 17:00)', and 'Malam (18:00 - 20:00)'. A 'Simpan' button is at the bottom right. Below the modal, a table titled 'Jadwal Kerja Available' shows a row for '09-01-2020' with 'Sore (15:00 - 17:00)', 'available' status, and 'Edit | Hapus' actions.



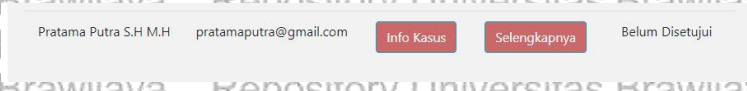


### 6.3.23 Pengujian Validasi Pilih Jasa Pengacara

#### 1. Pengujian validasi pilih jasa pengacara

Tabel 6.46 merupakan hasil dari pengujian validasi *use case* pilih jasa pengacara yang berhasil.

**Tabel 6.46 Pengujian Validasi Pilih Jasa Pengacara**

| Kode Kebutuhan         | SKPP-F-23  |
|------------------------|--|
| Kasus Uji              | Pilih Jasa Pengacara   |
| Prosedur               | <ol style="list-style-type: none"> <li>Mengakses halaman profil pengacara Pratama Putra S.H M.H.</li> <li>Memilih tombol Pilih pada gunakan jasa pengacara.</li> <li>Memasukkan masalah "Saya mempunyai masalah dengan partner bisnis saya".</li> <li>Memasukkan nomor <i>handphone</i> "081234567890".</li> <li>Memasukkan <i>email</i> "chrisevans@gmail.com".</li> <li>Menekan tombol Kirim.</li> </ol> |
| <i>Expected Result</i> | Sistem menyimpan data jasa pengacara pada <i>database</i> dan menampilkan halaman Progres Kasus.   |
| <i>Result</i>          | <p>Sistem menyimpan data jasa pengacara pada <i>database</i> dan menampilkan halaman Progres Kasus.</p>    |
| Status                 | Valid  |

#### 2. Pengujian validasi pilih jasa pengacara alternatif satu

Tabel 6.47 merupakan hasil dari pengujian validasi *use case* pilih jasa pengacara yang tidak berhasil karena ada kolom pada *form* pilih jasa pengacara yang kosong.

**Tabel 6.47 Pengujian Validasi Pilih Jasa Pengacara Alternatif Satu**

| Kode Kebutuhan | SKPP-F-23   |
|----------------|---|
| Kasus Uji      | Pilih Jasa Pengacara dengan kondisi alternatif satu   |
| Prosedur       | <ol style="list-style-type: none"> <li>Mengakses halaman profil pengacara Pratama Putra S.H M.H.</li> <li>Memilih tombol Pilih pada gunakan jasa pengacara.</li> <li>Menekan tombol Kirim tanpa mengisi masalah, nomor <i>handphone</i>, dan <i>email</i>.</li> </ol> |





|                        |   |
|------------------------|---|
| <i>Expected Result</i> | Sistem menampilkan pesan “Please fill out this field”.  |
| <i>Result</i>          | <p>Sistem menampilkan pesan “Please fill out this field”.</p> <p>Gunakan Jasa Pengacara<br/>Bantu Kami Memahami Masalah Anda</p> <p>Ceritakan Masalah Anda</p> <p>Masukkan Nomor Handphone</p> <p>Masukkan Email</p> <p>Batal Kirim</p> |
| <i>Status</i>          | Valid   |

### 6.3.24 Pengujian Validasi Batalkan Jasa Pengacara

#### 1. Pengujian validasi batalkan jasa pengacara

Tabel 6.48 merupakan hasil dari pengujian validasi *use case* batalkan jasa pengacara yang berhasil.

**Tabel 6.48 Pengujian Validasi Batalkan Jasa Pengacara**

|                        |   |
|------------------------|---|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-24</b>  |
| <b>Kasus Uji</b>       | Batalkan Jasa Pengacara   |
| <b>Prosedur</b>        | <ol style="list-style-type: none"> <li>1. Mengakses halaman Progres Kasus.</li> <li>2. Memilih tombol Selengkapnya pada pengacara Laylatul Fitriani S.H.</li> <li>3. Memilih tombol Batalkan Jasa Pengacara.</li> <li>4. Memasukkan alasan membatalkan pengacara “Saya tidak yakin dengan pengacara”.</li> <li>5. Menekan tombol Simpan.</li> </ol> |
| <i>Expected Result</i> | Sistem menyimpan data batalkan jasa pengacara pada <i>database</i> dan menampilkan halaman Progres Kasus.   |
| <i>Result</i>          | Sistem menyimpan data batalkan jasa pengacara pada <i>database</i> dan menampilkan halaman Progres Kasus.   |






| Status | Progres Kasus                       |                          |                |              |
|--------|-------------------------------------|--------------------------|----------------|--------------|
|        | Nama Pengacara                      | Email                    | Progress Kasus | Status       |
| Valid  | Jasa Pengacara Berhasil Dibatalkan. |                          |                |              |
|        | Entin Endah Cahyati S.H             | entincahya96@gmail.com   | Info Kasus     | Selengkapnya |
|        | Pratama Putra S.H M.H               | pratamaputra@gmail.com   | Info Kasus     | Selengkapnya |
|        | Naufal Azri S.H M.H                 | naufalazri10@gmail.com   | Info Kasus     | Selengkapnya |
|        | Laylatul Fitriani S.H               | laylatulfitria@gmail.com | Info Kasus     | Selengkapnya |

2. Pengujian validasi batalkan jasa pengacara alternatif satu

Tabel 6.49 merupakan hasil dari pengujian validasi *use case* batalkan jasa pengacara yang tidak berhasil karena ada kolom pada *form* batalkan jasa pengacara yang kosong.

**Tabel 6.49 Pengujian Validasi Batalkan Jasa Pengacara Alternatif Satu**

| Kode Kebutuhan         | SKPP-F-24   |
|------------------------|---|
| Kasus Uji              | Batalkan Jasa Pengacara dengan kondisi alternatif satu  |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakses halaman Progres Kasus.</li> <li>2. Memilih tombol Selengkapnya pada pengacara Laylatul Fitriani S.H.</li> <li>3. Memilih tombol Batalkan Jasa Pengacara.</li> <li>4. Menekan tombol Simpan tanpa mengisi alasan membatalkan pengacara.</li> </ol> |
| <i>Expected Result</i> | Sistem menampilkan pesan " <i>Please fill out this field</i> ".   |
| <i>Result</i>          | <p>Sistem menampilkan pesan "<i>Please fill out this field</i>".</p>    |
| Status                 | Valid   |





### 6.3.25 Pengujian Validasi Pilih Konsultasi Tatap Muka

#### 1. Pengujian validasi pilih konsultasi tatap muka

Tabel 6.50 merupakan hasil dari pengujian validasi *use case* Pilih Konsultasi Tatap Muka yang berhasil.

**Tabel 6.50 Pengujian Validasi Pilih Konsultasi Tatap Muka**

| Kode Kebutuhan          | SKPP-F-25   |                       |                              |                 |        |        |                         |            |                       |                              |         |                         |            |                      |                              |         |                         |            |                      |                              |         |                         |            |                       |                              |           |                         |            |                      |                              |                 |                       |            |                      |                              |                 |
|-------------------------|---|-----------------------|------------------------------|-----------------|--------|--------|-------------------------|------------|-----------------------|------------------------------|---------|-------------------------|------------|----------------------|------------------------------|---------|-------------------------|------------|----------------------|------------------------------|---------|-------------------------|------------|-----------------------|------------------------------|-----------|-------------------------|------------|----------------------|------------------------------|-----------------|-----------------------|------------|----------------------|------------------------------|-----------------|
| Kasus Uji               | Pilih Konsultasi Tatap Muka   |                       |                              |                 |        |        |                         |            |                       |                              |         |                         |            |                      |                              |         |                         |            |                      |                              |         |                         |            |                       |                              |           |                         |            |                      |                              |                 |                       |            |                      |                              |                 |
| Prosedur                | <ol style="list-style-type: none"> <li>Mengakses halaman Profil Pengacara.</li> <li>Memilih tombol Pilih pada layanan Konsultasi Tatap Muka.</li> <li>Memasukkan Bidang Keahlian dengan memilih <i>select option</i> "Teknologi Informasi".</li> <li>Memasukkan Tanggal Konsultasi dengan memilih <i>select option</i> "27-12-2019".</li> <li>Memasukkan Waktu Konsultasi dengan memilih <i>select option</i> "Sore (15:00-17:00)".</li> <li>Memasukkan Nomor Handphone "082234233894".</li> <li>Memasukkan <i>Email</i> "milenium@yahoo.co.id".</li> <li>Memasukkan Inti Permasalahan "penipuan sms mama minta pulsa."</li> <li>Menekan tombol Kirim.</li> </ol>   |                       |                              |                 |        |        |                         |            |                       |                              |         |                         |            |                      |                              |         |                         |            |                      |                              |         |                         |            |                       |                              |           |                         |            |                      |                              |                 |                       |            |                      |                              |                 |
| <i>Expected Result</i>  | Sistem menyimpan permintaan konsultasi tatap muka pada <i>database</i> dan menampilkan halaman Bookingan Saya.  |                       |                              |                 |        |        |                         |            |                       |                              |         |                         |            |                      |                              |         |                         |            |                      |                              |         |                         |            |                       |                              |           |                         |            |                      |                              |                 |                       |            |                      |                              |                 |
| <i>Result</i>           | <p>Sistem menyimpan permintaan konsultasi tatap muka pada <i>database</i> dan menampilkan halaman Bookingan Saya.</p>  <table border="1"> <thead> <tr> <th>Nama Pengacara</th> <th>Tanggal</th> <th>Pukul</th> <th>Detail</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>Entin Endah Cahyati S.H</td> <td>30-08-2019</td> <td>Siang (12:00 - 15:00)</td> <td><a href="#">Selengkapnya</a></td> <td>Selesai</td> </tr> <tr> <td>Entin Endah Cahyati S.H</td> <td>29-08-2019</td> <td>Sore (16:00 - 19:00)</td> <td><a href="#">Selengkapnya</a></td> <td>Selesai</td> </tr> <tr> <td>Entin Endah Cahyati S.H</td> <td>19-12-2019</td> <td>Sore (15:00 - 17:00)</td> <td><a href="#">Selengkapnya</a></td> <td>Selesai</td> </tr> <tr> <td>Entin Endah Cahyati S.H</td> <td>28-12-2019</td> <td>Siang (13:00 - 15:00)</td> <td><a href="#">Selengkapnya</a></td> <td>Disetujui</td> </tr> <tr> <td>Entin Endah Cahyati S.H</td> <td>27-12-2019</td> <td>Sore (15:00 - 17:00)</td> <td><a href="#">Selengkapnya</a></td> <td>Belum Disetujui</td> </tr> <tr> <td>Pratama Putra S.H M.H</td> <td>30-08-2019</td> <td>Sore (16:00 - 19:00)</td> <td><a href="#">Selengkapnya</a></td> <td>Belum Disetujui</td> </tr> </tbody> </table> | Nama Pengacara        | Tanggal                      | Pukul           | Detail | Status | Entin Endah Cahyati S.H | 30-08-2019 | Siang (12:00 - 15:00) | <a href="#">Selengkapnya</a> | Selesai | Entin Endah Cahyati S.H | 29-08-2019 | Sore (16:00 - 19:00) | <a href="#">Selengkapnya</a> | Selesai | Entin Endah Cahyati S.H | 19-12-2019 | Sore (15:00 - 17:00) | <a href="#">Selengkapnya</a> | Selesai | Entin Endah Cahyati S.H | 28-12-2019 | Siang (13:00 - 15:00) | <a href="#">Selengkapnya</a> | Disetujui | Entin Endah Cahyati S.H | 27-12-2019 | Sore (15:00 - 17:00) | <a href="#">Selengkapnya</a> | Belum Disetujui | Pratama Putra S.H M.H | 30-08-2019 | Sore (16:00 - 19:00) | <a href="#">Selengkapnya</a> | Belum Disetujui |
| Nama Pengacara          | Tanggal   | Pukul                 | Detail                       | Status          |        |        |                         |            |                       |                              |         |                         |            |                      |                              |         |                         |            |                      |                              |         |                         |            |                       |                              |           |                         |            |                      |                              |                 |                       |            |                      |                              |                 |
| Entin Endah Cahyati S.H | 30-08-2019  | Siang (12:00 - 15:00) | <a href="#">Selengkapnya</a> | Selesai         |        |        |                         |            |                       |                              |         |                         |            |                      |                              |         |                         |            |                      |                              |         |                         |            |                       |                              |           |                         |            |                      |                              |                 |                       |            |                      |                              |                 |
| Entin Endah Cahyati S.H | 29-08-2019  | Sore (16:00 - 19:00)  | <a href="#">Selengkapnya</a> | Selesai         |        |        |                         |            |                       |                              |         |                         |            |                      |                              |         |                         |            |                      |                              |         |                         |            |                       |                              |           |                         |            |                      |                              |                 |                       |            |                      |                              |                 |
| Entin Endah Cahyati S.H | 19-12-2019  | Sore (15:00 - 17:00)  | <a href="#">Selengkapnya</a> | Selesai         |        |        |                         |            |                       |                              |         |                         |            |                      |                              |         |                         |            |                      |                              |         |                         |            |                       |                              |           |                         |            |                      |                              |                 |                       |            |                      |                              |                 |
| Entin Endah Cahyati S.H | 28-12-2019  | Siang (13:00 - 15:00) | <a href="#">Selengkapnya</a> | Disetujui       |        |        |                         |            |                       |                              |         |                         |            |                      |                              |         |                         |            |                      |                              |         |                         |            |                       |                              |           |                         |            |                      |                              |                 |                       |            |                      |                              |                 |
| Entin Endah Cahyati S.H | 27-12-2019  | Sore (15:00 - 17:00)  | <a href="#">Selengkapnya</a> | Belum Disetujui |        |        |                         |            |                       |                              |         |                         |            |                      |                              |         |                         |            |                      |                              |         |                         |            |                       |                              |           |                         |            |                      |                              |                 |                       |            |                      |                              |                 |
| Pratama Putra S.H M.H   | 30-08-2019  | Sore (16:00 - 19:00)  | <a href="#">Selengkapnya</a> | Belum Disetujui |        |        |                         |            |                       |                              |         |                         |            |                      |                              |         |                         |            |                      |                              |         |                         |            |                       |                              |           |                         |            |                      |                              |                 |                       |            |                      |                              |                 |
| Status                  | Valid   |                       |                              |                 |        |        |                         |            |                       |                              |         |                         |            |                      |                              |         |                         |            |                      |                              |         |                         |            |                       |                              |           |                         |            |                      |                              |                 |                       |            |                      |                              |                 |

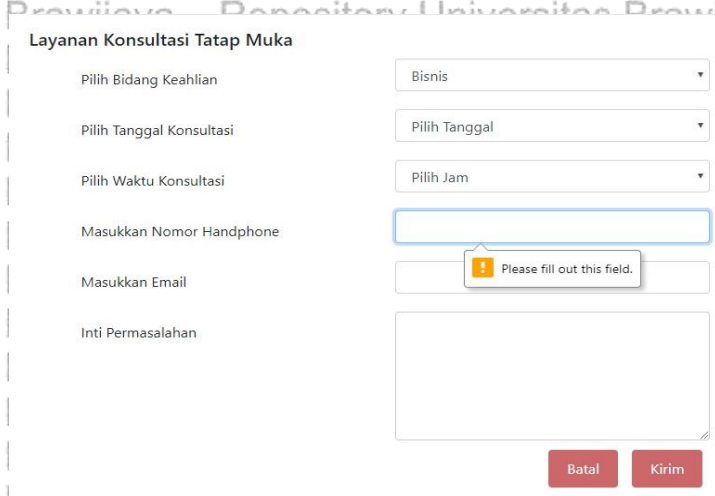




## 2. Pengujian validasi pilih konsultasi tatap muka alternatif satu

Tabel 6.51 merupakan hasil dari pengujian validasi *use case* pilih konsultasi tatap muka yang tidak berhasil karena ada kolom pada *form* pilih konsultasi tatap muka yang kosong.

**Tabel 6.51 Pengujian Validasi Pilih Konsultasi Tatap Muka Alternatif Satu**

| Kode Kebutuhan         | SKPP-F-25  |
|------------------------|--|
| Kasus Uji              | Pilih Konsultasi Tatap Muka dengan kondisi alternatif satu   |
| Prosedur               | <ol style="list-style-type: none"> <li>Mengakses halaman Profil Pengacara.</li> <li>Memilih tombol Pilih pada layanan Konsultasi Tatap Muka.</li> <li>Menekan tombol Kirim tanpa mengisi Bidang Keahlian, Tanggal Konsultasi, Waktu Konsultasi, Nomor Handphone, <i>Email</i>, dan Inti Permasalahan.</li> </ol> |
| <i>Expected Result</i> | Sistem menampilkan pesan " <i>Please fill out this field</i> ".  |
| <i>Result</i>          | <p>Sistem menampilkan pesan "<i>Please fill out this field</i>".</p>    |
| Status                 | Valid  |

## 6.3.26 Pengujian Validasi Lihat Bookingan Saya


Tabel 6.52 merupakan hasil dari pengujian validasi *use case* lihat bookingan saya yang berhasil.

**Tabel 6.52 Pengujian Validasi Lihat Bookingan Saya**

| Kode Kebutuhan | SKPP-F-26                         |
|----------------|-----------------------------------|
| Kasus Uji      | Lihat Bookingan Saya              |
| Prosedur       | Mengakses halaman Bookingan Saya. |






|                        |   |
|------------------------|---|
| <i>Expected Result</i> | Sistem menampilkan daftar bookingan tatap muka.   |
| <i>Result</i>          | Sistem menampilkan daftar bookingan tatap muka.<br> |
| <i>Status</i>          | Valid   |

### 6.3.27 Pengujian Validasi Lihat Riwayat Aktivitas

Tabel 6.53 merupakan hasil dari pengujian validasi *use case* lihat riwayat aktivitas yang berhasil.

**Tabel 6.53 Pengujian Validasi Lihat Riwayat Aktivitas**

|                        |  |
|------------------------|--|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-27</b>   |
| Kasus Uji              | Lihat Riwayat Aktivitas  |
| Prosedur               | Mengakses halaman Riwayat Aktivitas.   |
| <i>Expected Result</i> | Sistem menampilkan daftar riwayat aktivitas.   |
| <i>Result</i>          | Sistem menampilkan daftar riwayat aktivitas.<br> |
| <i>Status</i>          | Valid  |

### 6.3.28 Pengujian Validasi Beri Ulasan

#### 1. Pengujian Validasi Beri Ulasan

Tabel 6.54 merupakan hasil dari pengujian validasi *use case* beri ulasan yang berhasil.

**Tabel 6.54 Pengujian Validasi Beri Ulasan**

|                       |                                     |
|-----------------------|-------------------------------------|
| <b>Kode Kebutuhan</b> | <b>SKPP-F-28</b>                    |
| Kasus Uji             | Beri Ulasan                         |
| Prosedur              | 1. Mengakses halaman Bookingan Saya |






|                        |   |
|------------------------|---|
|                        | <ol style="list-style-type: none"> <li>Menekan tombol Selengkapnya pada pengacara Entin Endah Cahyati S.H.</li> <li>Menekan tombol Ulasan.</li> <li>Menekan bintang ke 5</li> <li>Memasukkan ulasan "Sangat puas dengan pengacara ini."</li> <li>Menekan tombol Kirim.</li> </ol> |
| <i>Expected Result</i> | Sistem menyimpan data ulasan pada <i>database</i> dan menampilkan halaman profil pengacara.   |
| <i>Result</i>          | <p>Sistem menyimpan data ulasan pada <i>database</i> dan menampilkan halaman profil pengacara.</p>    |
| Status                 | Valid   |

## 2. Pengujian Validasi Beri Ulasan Alternatif Satu

Tabel 6.55 merupakan hasil dari pengujian validasi *use case* beri ulasan yang tidak berhasil karena ada kolom pada *form* beri ulasan yang kosong.

**Tabel 6.55 Pengujian Validasi Beri Ulasan Alternatif Satu**

|                        |   |
|------------------------|---|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-28</b>  |
| Kasus Uji              | Beri Ulasan Alternatif Satu   |
| Prosedur               | <ol style="list-style-type: none"> <li>Mengakses halaman Bookingan Saya</li> <li>Menekan tombol Selengkapnya pada pengacara Entin Endah Cahyati S.H.</li> <li>Menekan tombol Ulasan.</li> <li>Menekan tombol Kirim tanpa mengisi bintang dan ulasan.</li> </ol> |
| <i>Expected Result</i> | Sistem menampilkan pesan " <i>Please fill out this field</i> ".   |
| <i>Result</i>          | <p>Sistem menampilkan pesan "<i>Please fill out this field</i>".</p>    |





|        |       |
|--------|-------|
| Status | Valid |
|--------|-------|

### 3. Pengujian Validasi Berikan Ulasan Alternatif Dua

Tabel 6.56 merupakan hasil dari pengujian validasi *use case* beri ulasan yang tidak berhasil karena klien sudah mengirimkan ulasan.

**Tabel 6.56 Pengujian Validasi Beri Ulasan Alternatif Dua**

|                        |  |
|------------------------|--|
| Kode Kebutuhan         | SKPP-F-28  |
| Kasus Uji              | Beri Ulasan Alternatif Dua   |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakses halaman Bookingan Saya</li> <li>2. Menekan tombol Selengkapnya pada pengacara Entin Endah Cahyati S.H.</li> <li>3. Menekan tombol Ulasan.</li> </ol> |
| <i>Expected Result</i> | Sistem menampilkan pesan “Anda sudah mengirimkan ulasan!”  |
| <i>Result</i>          | <p>Sistem menampilkan pesan “Anda sudah mengirimkan ulasan!”</p>   |
| Status                 | Valid  |

### 6.3.29 Pengujian Validasi Laporkan Pengacara

#### 1. Pengujian validasi laporkan pengacara

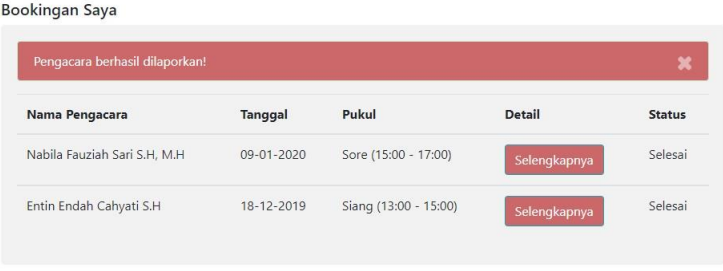
Tabel 6.57 merupakan hasil dari pengujian validasi *use case* laporkan pengacara yang berhasil.

**Tabel 6.57 Pengujian Validasi Laporkan Pengacara**

|                |   |
|----------------|---|
| Kode Kebutuhan | SKPP-F-29   |
| Kasus Uji      | Laporkan Pengacara.   |
| Prosedur       | <ol style="list-style-type: none"> <li>1. Mengakses halaman Bookingan Saya.</li> <li>2. Memilih nama pengacara “Nabilah Fauziah Sari S.H, M.H”</li> <li>3. Menekan tombol Selengkapnya.</li> <li>4. Menekan <i>link</i> Laporkan Pengacara.</li> <li>5. Memasukkan Alasan melaporkan Pengacara “pengacara tidak serius melayani kasus saya.”</li> </ol> |





|                        |  |
|------------------------|--|
|                        | 6. Menekan tombol Simpan.  |
| <i>Expected Result</i> | Sistem menyimpan laporan pengacara ke dalam <i>database</i> dan menampilkan pesan "Pengacara Berhasil Dilaporkan!"   |
| <i>Result</i>          | Sistem menyimpan laporan pengacara ke dalam <i>database</i> dan menampilkan pesan "Pengacara Berhasil Dilaporkan!"<br><br> |
| <i>Status</i>          | Valid  |

## 2. Pengujian validasi laporkan pengacara alternatif satu

Tabel 6.58 merupakan hasil dari pengujian validasi *use case* konsultasi tatap muka yang tidak berhasil karena ada kolom pada *form* konsultasi tatap muka yang kosong.

**Tabel 6.58 Pengujian Validasi Laporkan Pengacara Alternatif Satu**

|                        |   |
|------------------------|---|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-29</b>  |
| Kasus Uji              | Laporkan Pengacara dengan kondisi alternatif satu   |
| Prosedur               | <ol style="list-style-type: none"> <li>Mengakses halaman Bookingan Saya.</li> <li>Memilih nama pengacara "Nabilah Fauziah Sari S.H, M.H".</li> <li>Menekan tombol Selengkapnya.</li> <li>Menekan <i>link</i> Laporkan Pengacara.</li> <li>Menekan tombol Simpan tanpa mengisi alasan melaporkan pengacara.</li> </ol> |
| <i>Expected Result</i> | Sistem menampilkan pesan " <i>Please fill out this field</i> ".   |
| <i>Result</i>          | Sistem menampilkan pesan " <i>Please fill out this field</i> ".   |





|              |   |
|--------------|---|
| Permasalahan | saya mendapatkan email scam berupa ancaman untuk membayarkan uang |
| Berita Acara |   |
| Status       | Valid   |

### 3. Pengujian validasi laporkan pengacara alternatif dua

Tabel 6.59 merupakan hasil dari pengujian validasi *use case* konsultasi tatap muka yang tidak berhasil karena membatalkan laporkan pengacara.

**Tabel 6.59 Pengujian Validasi Laporkan Pengacara Alternatif Dua**

| Kode Kebutuhan         | SKPP-F-29  |
|------------------------|--|
| Kasus Uji              | Laporkan Pengacara dengan kondisi alternatif dua   |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakses halaman Bookingan Saya.</li> <li>2. Memilih nama pengacara "Nabilah Fauziah Sari S.H, M.H".</li> <li>3. Menekan tombol Selengkapnya.</li> <li>4. Menekan <i>link</i> Laporkan Pengacara.</li> <li>5. Menekan tombol Simpan tanpa mengisi alasan melaporkan pengacara.</li> </ol> |
| <i>Expected Result</i> | Sistem menampilkan <i>modal</i> Selengkapnya dari bookingan saya.  |
| <i>Result</i>          | <p>Sistem menampilkan <i>modal</i> Selengkapnya dari bookingan saya.</p>   |
| Status                 | Valid  |






### 6.3.30 Pengujian Validasi Kirim Pertanyaan

#### 1. Pengujian validasi kirim pertanyaan

Tabel 6.60 merupakan hasil dari pengujian validasi *use case* kirim pertanyaan yang berhasil.

**Tabel 6.60 Pengujian Validasi Kirim Pertanyaan**

| Kode Kebutuhan         | SKPP-F-30  |
|------------------------|--|
| Kasus Uji              | Kirim Pertanyaan   |
| Prosedur               | <ol style="list-style-type: none"> <li>Mengakses halaman Tanya Pengacara</li> <li>Menekan tombol Tanya Pengacara</li> <li>Mengisikan Judul "contoh pertanyaan"</li> <li>Mengisikan Pertanyaan "contoh pertanyaan untuk validasi"</li> <li>Menekan tombol Kirim.</li> </ol> |
| <i>Expected Result</i> | Sistem menyimpan pertanyaan pada <i>database</i> dan menampilkannya ke halaman Tanya Pengacara dengan notifikasi "Tanya Pengacara Berhasil Dikirim".   |
| <i>Result</i>          | <p>Sistem menyimpan pertanyaan pada <i>database</i> dan menampilkannya ke halaman Tanya Pengacara dengan notifikasi "Tanya Pengacara Berhasil Dikirim".</p>                            |
| Status                 | Valid  |

#### 2. Pengujian validasi kirim pertanyaan alternatif satu

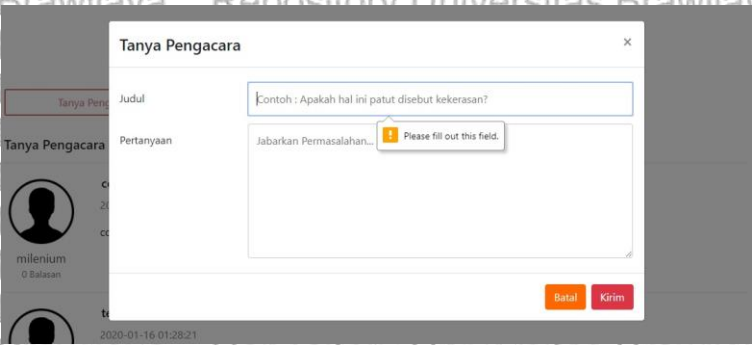
Tabel 6.61 merupakan hasil dari pengujian validasi *use case* kirim pertanyaan yang tidak berhasil karena ada kolom pada *form* kirim pertanyaan yang kosong.

**Tabel 6.61 Pengujian Validasi Kirim Pertanyaan Alternatif Satu**

| Kode Kebutuhan | SKPP-F-30                                       |
|----------------|---|
| Kasus Uji      | Kirim Pertanyaan dengan kondisi alternatif satu |






|                 |  |
|-----------------|--|
| Prosedur        | <ol style="list-style-type: none"> <li>1. Mengakses halaman Tanya Pengacara</li> <li>2. Menekan tombol Tanya Pengacara</li> <li>3. Menekan tombol Kirim tanpa mengisi judul dan pertanyaan.</li> </ol> |
| Expected Result | Sistem menampilkan pesan "Please fill out this field".   |
| Result          | <p>Sistem menampilkan pesan "Please fill out this field".</p>    |
| Status          | Valid  |

### 6.3.31 Pengujian Validasi Lihat Jawaban Pertanyaan

Tabel 6.62 merupakan hasil dari pengujian validasi *use case* kirim pertanyaan yang berhasil.

Tabel 6.62 Pengujian Validasi Lihat Jawaban Pertanyaan

|                 |  |
|-----------------|--|
| Kode Kebutuhan  | SKPP-F-31  |
| Kasus Uji       | Lihat Jawaban Pertanyaan   |
| Prosedur        | <ol style="list-style-type: none"> <li>1. Mengakses halaman Diskusi.</li> <li>2. Memilih pertanyaan berjudul "Cyber Bullying"</li> </ol> |
| Expected Result | Sistem menampilkan jawaban pertanyaan.   |
| Result          | <p>Sistem menampilkan jawaban pertanyaan.</p>        |
| Status          | Valid  |

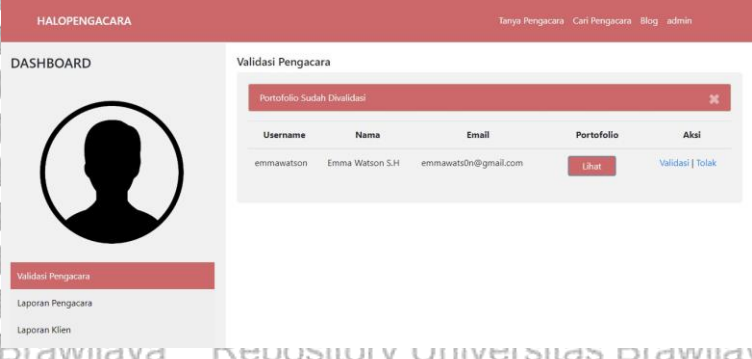




### 6.3.32 Pengujian Validasi Validasi Pengacara

Tabel 6.63 merupakan hasil dari pengujian validasi *use case* validasi pengacara yang berhasil.

**Tabel 6.63 Pengujian Validasi Validasi Pengacara**

|                        |   |
|------------------------|---|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-32</b>  |
| Kasus Uji              | Validasi Pengacara  |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakses halaman Validasi Pengacara pada <i>dashboard</i> admin.</li> <li>2. Memilih pengacara bernama "Nabila Fauziah Sari S.H, M.H"</li> <li>3. Menekan <i>link</i> Validasi.</li> </ol> |
| <i>Expected Result</i> | Sistem menyimpan validasi pengacara ke dalam <i>database</i> dan menampilkan pesan "Pengacara Sudah Divalidasi".  |
| <i>Result</i>          | <p>Sistem menyimpan validasi pengacara ke dalam <i>database</i> dan menampilkan pesan "Pengacara Sudah Divalidasi".</p>                            |
| Status                 | Valid   |

### 6.3.33 Pengujian Validasi Tolak Pengacara

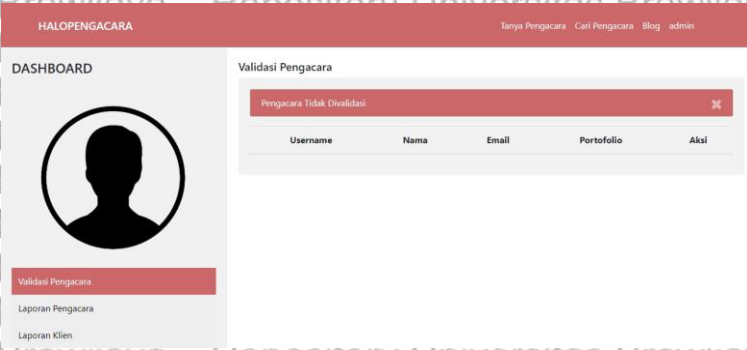
Tabel 6.64 merupakan hasil dari pengujian validasi *use case* tolak pengacara yang berhasil.

**Tabel 6.64 Pengujian Validasi Tolak Pengacara**

|                       |  |
|-----------------------|--|
| <b>Kode Kebutuhan</b> | <b>SKPP-F-33</b>   |
| Kasus Uji             | Tolak Pengacara  |
| Prosedur              | <ol style="list-style-type: none"> <li>1. Mengakses halaman Validasi Pengacara pada <i>dashboard</i> admin.</li> <li>2. Memilih pengacara bernama "Emma Watson S.H".</li> <li>3. Menekan <i>link</i> Tolak.</li> </ol> |





|                        |   |
|------------------------|---|
| <i>Expected Result</i> | Sistem menghapus data pengacara yang ditolak pada <i>database</i> dan menampilkan pesan "Pengacara Tidak Divalidasi".   |
| <i>Result</i>          | Sistem menghapus data pengacara yang ditolak pada <i>database</i> dan menampilkan pesan "Pengacara Tidak Divalidasi".<br> |
| Status                 | Valid   |

### 6.3.34 Pengujian Validasi Lihat Portofolio Pengacara

Tabel 6.65 merupakan hasil dari pengujian validasi *use case* lihat portofolio pengacara yang berhasil.

**Tabel 6.65 Pengujian Validasi Lihat Portofolio Pengacara**

|                        |   |
|------------------------|---|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-34</b>  |
| Kasus Uji              | Lihat Portofolio Pengacara  |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakses halaman <i>dashboard</i> admin.</li> <li>2. Memilih pengacara yang akan dilihat portofolionya.</li> <li>3. Menekan tombol Lihat.</li> </ol> |
| <i>Expected Result</i> | Sistem menampilkan modal yang menampilkan portofolio yang dimiliki oleh pengacara.  |
| <i>Result</i>          | Sistem menampilkan modal yang menampilkan portofolio yang dimiliki oleh pengacara.  |





|  |                            |
|--|----------------------------|
|  | <p><b>Status</b> Valid</p> |
|--|----------------------------|

### 6.3.35 Pengujian Validasi Lihat Laporan Info Hukum

Tabel 6.66 merupakan hasil dari pengujian validasi *use case* lihat laporan info hukum yang berhasil.

Tabel 6.66 Pengujian Validasi Lihat Laporan Info Hukum

| <b>Kode Kebutuhan</b>   | SKPP-F-35   |  |              |                  |                  |        |      |   |              |                |              |                  |  |                      |              |  |              |                  |  |                      |              |   |              |                  |                  |
|---|---|--|--------------|------------------|------------------|--------|------|---|--------------|----------------|--------------|------------------|--|----------------------|--------------|--|--------------|------------------|--|----------------------|--------------|---|--------------|------------------|------------------|
| <b>Kasus Uji</b>  | Lihat Laporan Info Hukum  |  |              |                  |                  |        |      |   |              |                |              |                  |  |                      |              |  |              |                  |  |                      |              |   |              |                  |                  |
| <b>Prosedur</b>   | Mengakses halaman Laporan Info Hukum  |  |              |                  |                  |        |      |   |              |                |              |                  |  |                      |              |  |              |                  |  |                      |              |   |              |                  |                  |
| <b>Expected Result</b>  | Sistem menampilkan daftar info hukum yang dilaporkan.   |  |              |                  |                  |        |      |   |              |                |              |                  |  |                      |              |  |              |                  |  |                      |              |   |              |                  |                  |
| <b>Result</b>   | <p>Sistem menampilkan daftar info hukum yang dilaporkan.</p> <p>Laporan Info Hukum</p> <table border="1"> <thead> <tr> <th>Judul Info Hukum</th> <th>Author</th> <th>Alasan Pelaporan</th> <th>Pelapor</th> <th>Status</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>RUU Hukum Acara Perdata Bukan Prioritas dalam Prolegnas, Mengapa?</td> <td>pratamaputra</td> <td>iseng hehehehe</td> <td>Milenium Kim</td> <td>tidak divalidasi</td> <td></td> </tr> <tr> <td>coba coba info hukum</td> <td>pratamaputra</td> <td>berita yang dituliskan penulis tidak jelas</td> <td>Milenium Kim</td> <td>sudah divalidasi</td> <td></td> </tr> <tr> <td>coba coba info hukum</td> <td>pratamaputra</td> <td>info hukum memuat info yang tidak jelas</td> <td>Milenium Kim</td> <td>belum divalidasi</td> <td>Validasi   Tolak</td> </tr> </tbody> </table> | Judul Info Hukum                           | Author       | Alasan Pelaporan | Pelapor          | Status | Aksi | RUU Hukum Acara Perdata Bukan Prioritas dalam Prolegnas, Mengapa? | pratamaputra | iseng hehehehe | Milenium Kim | tidak divalidasi |  | coba coba info hukum | pratamaputra | berita yang dituliskan penulis tidak jelas | Milenium Kim | sudah divalidasi |  | coba coba info hukum | pratamaputra | info hukum memuat info yang tidak jelas | Milenium Kim | belum divalidasi | Validasi   Tolak |
| Judul Info Hukum  | Author  | Alasan Pelaporan                           | Pelapor      | Status           | Aksi             |        |      |   |              |                |              |                  |  |                      |              |  |              |                  |  |                      |              |   |              |                  |                  |
| RUU Hukum Acara Perdata Bukan Prioritas dalam Prolegnas, Mengapa? | pratamaputra  | iseng hehehehe                             | Milenium Kim | tidak divalidasi |                  |        |      |   |              |                |              |                  |  |                      |              |  |              |                  |  |                      |              |   |              |                  |                  |
| coba coba info hukum  | pratamaputra  | berita yang dituliskan penulis tidak jelas | Milenium Kim | sudah divalidasi |                  |        |      |   |              |                |              |                  |  |                      |              |  |              |                  |  |                      |              |   |              |                  |                  |
| coba coba info hukum  | pratamaputra  | info hukum memuat info yang tidak jelas    | Milenium Kim | belum divalidasi | Validasi   Tolak |        |      |   |              |                |              |                  |  |                      |              |  |              |                  |  |                      |              |   |              |                  |                  |
| <b>Status</b>   | Valid   |  |              |                  |                  |        |      |   |              |                |              |                  |  |                      |              |  |              |                  |  |                      |              |   |              |                  |                  |

### 6.3.36 Pengujian Validasi Hapus Laporan Info Hukum

1. Pengujian validasi hapus laporan info hukum


Tabel 6.67 merupakan hasil dari pengujian validasi *use case* hapus laporan info hukum yang berhasil.

Tabel 6.67 Pengujian Validasi Hapus Laporan Info Hukum

|                       |           |
|-----------------------|-----------|
| <b>Kode Kebutuhan</b> | SKPP-F-36 |
|-----------------------|-----------|






| Kasus Uji   | Hapus Laporan Info Hukum  |                  |              |                  |         |        |      |   |              |                |              |                  |  |
|---|---|------------------|--------------|------------------|---------|--------|------|---|--------------|----------------|--------------|------------------|--|
| Prosedur  | <ol style="list-style-type: none"> <li>Mengakses halaman Laporan Info Hukum.</li> <li>Menekan tombol Validasi pada judul info hukum “coba-coba info hukum”.</li> </ol>  |                  |              |                  |         |        |      |   |              |                |              |                  |  |
| <i>Expected Result</i>  | Sistem menghapus info hukum pada <i>database</i> dan menampilkan halaman Laporan Info Hukum dengan notifikasi “Info Hukum Sudah Dihapus”.   |                  |              |                  |         |        |      |   |              |                |              |                  |  |
| <i>Result</i>   | <p>Sistem menghapus info hukum pada <i>database</i> dan menampilkan halaman Laporan Info Hukum dengan notifikasi “Info Hukum Sudah Dihapus”.</p>  <p>Laporan Info Hukum</p> <table border="1"> <thead> <tr> <th>Judul Info Hukum</th> <th>Author</th> <th>Alasan Pelaporan</th> <th>Pelapor</th> <th>Status</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>RUU Hukum Acara Perdata Bukan Prioritas dalam Prolegnas, Mengapa?</td> <td>pratamaputra</td> <td>iseng hehehehe</td> <td>Milenium Kim</td> <td>tidak divalidasi</td> <td></td> </tr> </tbody> </table> | Judul Info Hukum | Author       | Alasan Pelaporan | Pelapor | Status | Aksi | RUU Hukum Acara Perdata Bukan Prioritas dalam Prolegnas, Mengapa? | pratamaputra | iseng hehehehe | Milenium Kim | tidak divalidasi |  |
| Judul Info Hukum  | Author  | Alasan Pelaporan | Pelapor      | Status           | Aksi    |        |      |   |              |                |              |                  |  |
| RUU Hukum Acara Perdata Bukan Prioritas dalam Prolegnas, Mengapa? | pratamaputra  | iseng hehehehe   | Milenium Kim | tidak divalidasi |         |        |      |   |              |                |              |                  |  |
| Status  | Valid   |                  |              |                  |         |        |      |   |              |                |              |                  |  |

## 2. Pengujian validasi hapus laporan info hukum alternatif satu

Tabel 6.68 merupakan hasil dari pengujian validasi *use case* hapus laporan info hukum yang tidak berhasil karena aktor menolak laporan info hukum.

**Tabel 6.68 Pengujian Validasi Hapus Laporan Info Hukum Alternatif Satu**

| Kode Kebutuhan  | SKPP-F-36   |                  |              |                  |         |        |      |   |              |                |              |                  |  |                      |             |             |              |                  |  |
|---|---|------------------|--------------|------------------|---------|--------|------|---|--------------|----------------|--------------|------------------|--|----------------------|-------------|-------------|--------------|------------------|--|
| Kasus Uji   | Hapus Laporan Info Hukum dengan kondisi alternatif satu   |                  |              |                  |         |        |      |   |              |                |              |                  |  |                      |             |             |              |                  |  |
| Prosedur  | <ol style="list-style-type: none"> <li>Mengakses halaman Laporan Info Hukum.</li> <li>Menekan tombol Tolak pada judul info hukum “coba-coba info hukum”.</li> </ol>   |                  |              |                  |         |        |      |   |              |                |              |                  |  |                      |             |             |              |                  |  |
| <i>Expected Result</i>  | Sistem menampilkan pesan “Info Hukum Tidak Dihapus”.  |                  |              |                  |         |        |      |   |              |                |              |                  |  |                      |             |             |              |                  |  |
| <i>Result</i>   | <p>Sistem menampilkan pesan “Info Hukum Tidak Dihapus”.</p>  <p>Laporan Info Hukum</p> <table border="1"> <thead> <tr> <th>Judul Info Hukum</th> <th>Author</th> <th>Alasan Pelaporan</th> <th>Pelapor</th> <th>Status</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>RUU Hukum Acara Perdata Bukan Prioritas dalam Prolegnas, Mengapa?</td> <td>pratamaputra</td> <td>iseng hehehehe</td> <td>Milenium Kim</td> <td>tidak divalidasi</td> <td></td> </tr> <tr> <td>coba coba info hukum</td> <td>entincahnya</td> <td>tidak jelas</td> <td>Milenium Kim</td> <td>tidak divalidasi</td> <td></td> </tr> </tbody> </table> | Judul Info Hukum | Author       | Alasan Pelaporan | Pelapor | Status | Aksi | RUU Hukum Acara Perdata Bukan Prioritas dalam Prolegnas, Mengapa? | pratamaputra | iseng hehehehe | Milenium Kim | tidak divalidasi |  | coba coba info hukum | entincahnya | tidak jelas | Milenium Kim | tidak divalidasi |  |
| Judul Info Hukum  | Author  | Alasan Pelaporan | Pelapor      | Status           | Aksi    |        |      |   |              |                |              |                  |  |                      |             |             |              |                  |  |
| RUU Hukum Acara Perdata Bukan Prioritas dalam Prolegnas, Mengapa? | pratamaputra  | iseng hehehehe   | Milenium Kim | tidak divalidasi |         |        |      |   |              |                |              |                  |  |                      |             |             |              |                  |  |
| coba coba info hukum  | entincahnya   | tidak jelas      | Milenium Kim | tidak divalidasi |         |        |      |   |              |                |              |                  |  |                      |             |             |              |                  |  |





|        |       |
|--------|-------|
| Status | Valid |
|--------|-------|

### 6.3.37 Pengujian Validasi Lihat Laporan Pengacara

Tabel 6.69 merupakan hasil dari pengujian validasi *use case* lihat laporan pengacara yang berhasil.

Tabel 6.69 Pengujian Validasi Lihat Laporan Pengacara

| Kode Kebutuhan               | SKPP-F-37  |                |                  |  |        |      |                         |                              |              |                  |  |                              |   |         |                  |  |
|------------------------------|--|----------------|------------------|--|--------|------|-------------------------|------------------------------|--------------|------------------|--|------------------------------|---|---------|------------------|--|
| Kasus Uji                    | Lihat Laporan Pengacara  |                |                  |  |        |      |                         |                              |              |                  |  |                              |   |         |                  |  |
| Prosedur                     | Mengakses halaman Laporan Pengacara.   |                |                  |  |        |      |                         |                              |              |                  |  |                              |   |         |                  |  |
| Expected Result              | Sistem menampilkan daftar pengacara yang dilaporkan.   |                |                  |  |        |      |                         |                              |              |                  |  |                              |   |         |                  |  |
| Result                       | Sistem menampilkan daftar pengacara yang dilaporkan.<br><div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Nama Pengacara</th> <th style="text-align: left;">Alasan Pelaporan</th> <th style="text-align: left;">Pelapor</th> <th style="text-align: left;">Status</th> <th style="text-align: left;">Aksi</th> </tr> </thead> <tbody> <tr> <td>Entin Endah Cahyati S.H</td> <td>pengacara ini sombong sekali</td> <td>Milenium Kim</td> <td>tidak divalidasi</td> <td></td> </tr> <tr> <td>Nabila Fauziah Sari S.H, M.H</td> <td>pengacara tidak serius melayani kasus saya.</td> <td>Molaayo</td> <td>belum divalidasi</td> <td><a href="#">Validasi</a>   <a href="#">Tolak</a></td> </tr> </tbody> </table> </div> | Nama Pengacara | Alasan Pelaporan | Pelapor  | Status | Aksi | Entin Endah Cahyati S.H | pengacara ini sombong sekali | Milenium Kim | tidak divalidasi |  | Nabila Fauziah Sari S.H, M.H | pengacara tidak serius melayani kasus saya. | Molaayo | belum divalidasi | <a href="#">Validasi</a>   <a href="#">Tolak</a> |
| Nama Pengacara               | Alasan Pelaporan   | Pelapor        | Status           | Aksi   |        |      |                         |                              |              |                  |  |                              |   |         |                  |  |
| Entin Endah Cahyati S.H      | pengacara ini sombong sekali   | Milenium Kim   | tidak divalidasi |  |        |      |                         |                              |              |                  |  |                              |   |         |                  |  |
| Nabila Fauziah Sari S.H, M.H | pengacara tidak serius melayani kasus saya.  | Molaayo        | belum divalidasi | <a href="#">Validasi</a>   <a href="#">Tolak</a> |        |      |                         |                              |              |                  |  |                              |   |         |                  |  |
| Status                       | Valid  |                |                  |  |        |      |                         |                              |              |                  |  |                              |   |         |                  |  |

### 6.3.38 Pengujian Validasi Hapus Pengacara

#### 1. Pengujian validasi hapus pengacara

Tabel 6.70 merupakan hasil dari pengujian validasi *use case* hapus pengacara yang berhasil.

Tabel 6.70 Pengujian Validasi Hapus Pengacara

|                 |  |
|-----------------|--|
| Kode Kebutuhan  | SKPP-F-38  |
| Kasus Uji       | Hapus Pengacara  |
| Prosedur        | 1. Mengakses halaman Laporan Pengacara.<br>2. Menekan tombol Validasi pada nama pengacara "Nabila Fauziah Sari, S.H, M.H".             |
| Expected Result | Sistem menghapus pengacara pada <i>database</i> dan menampilkan halaman Laporan Pengacara dengan notifikasi "Pengacara Sudah Dihapus". |
| Result          | Sistem menghapus pengacara pada <i>database</i> dan menampilkan halaman Laporan Pengacara dengan notifikasi "Pengacara Sudah Dihapus". |





|        |       |
|--------|-------|
|        |       |
| Status | Valid |

## 2. Pengujian validasi hapus pengacara alternatif satu

Tabel 6.71 merupakan hasil dari pengujian validasi *use case* pengacara yang tidak berhasil karena aktor menolak laporan pengacara.

**Tabel 6.71 Pengujian Validasi Hapus Pengacara Alternatif Satu**

|                 |   |
|-----------------|---|
| Kode Kebutuhan  | SKPP-F-38   |
| Kasus Uji       | Hapus Pengacara dengan kondisi alternatif satu  |
| Prosedur        | <ol style="list-style-type: none"> <li>Mengakses halaman Laporan Pengacara.</li> <li>Menekan tombol Tolak pada nama pengacara "Nabila Fauziah Sari, S.H, M.H".</li> </ol> |
| Expected Result | Sistem menampilkan pesan "Pengacara Tidak Dihapus".   |
| Result          | <p>Sistem menampilkan pesan "Pengacara Tidak Dihapus".</p>  |
| Status          | Valid   |

## 6.3.39 Pengujian Validasi Lihat Laporan Klient

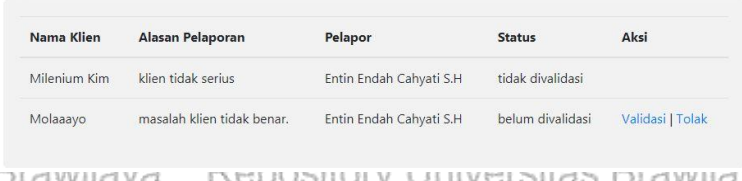
Tabel 6.72 merupakan hasil dari pengujian validasi *use case* lihat laporan klien yang berhasil.

**Tabel 6.72 Pengujian Validasi Lihat Laporan Klient**

|                |                                   |
|----------------|-----------------------------------|
| Kode Kebutuhan | SKPP-F-39                         |
| Kasus Uji      | Lihat Laporan Klient              |
| Prosedur       | Mengakses halaman Laporan Klient. |





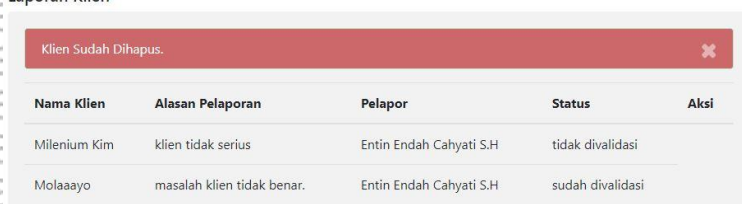
|                        |   |
|------------------------|---|
| <i>Expected Result</i> | Sistem menampilkan daftar klien yang dilaporkan.  |
| <i>Result</i>          | Sistem menampilkan daftar klien yang dilaporkan.<br>Laporan Klien<br> |
| <i>Status</i>          | Valid   |

### 6.3.40 Pengujian Validasi Hapus Klien

#### 1. Pengujian validasi hapus klien

Tabel 6.73 merupakan hasil dari pengujian validasi *use case* hapus klien yang berhasil.

Tabel 6.73 Pengujian Validasi Hapus Klien

|                        |   |
|------------------------|---|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-40</b>  |
| <b>Kasus Uji</b>       | Hapus Klien   |
| <b>Prosedur</b>        | 1. Mengakses halaman Laporan Klien.<br>2. Menekan tombol Validasi pada nama klien "Molaayo".  |
| <i>Expected Result</i> | Sistem menghapus klien pada <i>database</i> dan menampilkan halaman Laporan Pengacara dengan notifikasi "Klien Sudah Dihapus".  |
| <i>Result</i>          | Sistem menghapus klien pada <i>database</i> dan menampilkan halaman Laporan Pengacara dengan notifikasi "Klien Sudah Dihapus".<br>Laporan Klien<br> |
| <i>Status</i>          | Valid   |

#### 2. Pengujian validasi hapus klien alternatif satu

Tabel 6.74 merupakan hasil dari pengujian validasi *use case* hapus klien yang tidak berhasil karena aktor menolak laporan klien.





Tabel 6.74 Pengujian Validasi Hapus Klien Alternatif Satu

|                 |   |
|-----------------|---|
| Kode Kebutuhan  | SKPP-F-40   |
| Kasus Uji       | Hapus Klien dengan kondisi alternatif satu  |
| Prosedur        | <ol style="list-style-type: none"> <li>Mengakses halaman Laporan Klien.</li> <li>Menekan tombol Tolak pada nama klien "Molaayo".</li> </ol> |
| Expected Result | Sistem menampilkan pesan "Klien Tidak Dihapus".   |
| Result          | <p>Sistem menampilkan pesan "Klien Tidak Dihapus".</p>    |
| Status          | Valid   |

### 6.3.41 Pengujian Validasi Beri Komentar

#### 1. Pengujian validasi beri komentar

Tabel 6.75 merupakan hasil dari pengujian validasi *use case* beri komentar yang berhasil.

Tabel 6.75 Pengujian Validasi Beri Komentar

|                 |   |
|-----------------|---|
| Kode Kebutuhan  | SKPP-F-41   |
| Kasus Uji       | Beri Komentar   |
| Prosedur        | <ol style="list-style-type: none"> <li>Mengakses halaman Blog.</li> <li>Memilih informasi hukum dengan judul "RUU Hukum Acara Perdata Bukan Prioritas dalam Prolegnas, Mengapa?".</li> <li>Mengisikan komentar "berita yang menarik".</li> <li>Menekan tombol Kirim.</li> </ol> |
| Expected Result | Sistem menyimpan data komentar pada <i>database</i> dan menampilkan halaman info hukum.   |
| Result          | Sistem menyimpan data komentar pada <i>database</i> dan menampilkan halaman info hukum.   |





|        |   |
|--------|---|
|        | <p>2 Comments</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <p><b>entincahya</b><br/>berita yang sangat bermanfaat<br/>2019-11-13 03:56:24</p> </div> <div style="border: 1px solid #ccc; padding: 5px;"> <p><b>chrisevans</b><br/>berita yang menarik<br/>2019-11-13 03:56:24</p> </div> |
| Status | Valid   |

## 2. Pengujian validasi beri komentar alternatif satu

Tabel 6.76 merupakan hasil dari pengujian validasi *use case* beri komentar yang tidak berhasil karena ada kolom pada *form* yang kosong.

**Tabel 6.76 Pengujian Validasi Beri Komentar**

|                 |  |
|-----------------|--|
| Kode Kebutuhan  | SKPP-F-41  |
| Kasus Uji       | Beri Komentar dengan kondisi alternatif satu   |
| Prosedur        | <ol style="list-style-type: none"> <li>Mengakses halaman Blog.</li> <li>Memilih informasi hukum dengan judul "RUU Hukum Acara Perdata Bukan Prioritas dalam Prolegnas, Mengapa?".</li> <li>Menekan tombol Kirim tanpa mengisi komentar.</li> </ol> |
| Expected Result | Sistem menampilkan pesan " <i>Please fill out this field</i> ".  |
| Result          | <p>Sistem menampilkan pesan "<i>Please fill out this field</i>".</p>   |
| Status          | Valid  |

## 6.3.42 Pengujian Validasi Laporkan Info Hukum

### 1. Pengujian validasi laporkan info hukum


Tabel 6.77 merupakan hasil dari pengujian validasi *use case* laporkan info hukum yang berhasil.

**Tabel 6.77 Pengujian Validasi Laporkan Info Hukum**

|                |                     |
|----------------|---------------------|
| Kode Kebutuhan | SKPP-F-42           |
| Kasus Uji      | Laporkan Info Hukum |





|                 |   |
|-----------------|---|
| Prosedur        | <ol style="list-style-type: none"> <li>1. Mengakses halaman detail info hukum.</li> <li>2. Menekan ikon <i>Report</i>.</li> <li>3. Mengisikan Alasan Melaporkan Info Hukum “info hukum memuat info yang tidak jelas”.</li> <li>4. Menekan tombol Simpan.</li> </ol> |
| Expected Result | Sistem menyimpan laporan pengacara ke dalam <i>database</i> .   |
| Result          | <p>Sistem menyimpan laporan pengacara ke dalam <i>database</i>.</p>   |
| Status          | Valid   |

2. Pengujian validasi laporkan info hukum alternatif satu

Tabel 6.78 merupakan hasil dari pengujian validasi *use case* laporkan info hukum yang tidak berhasil karena ada kolom pada *form* laporkan info hukum yang kosong.

**Tabel 6.78 Pengujian Validasi Laporkan Info Hukum Alternatif Satu**

|                 |   |
|-----------------|---|
| Kode Kebutuhan  | SKPP-F-42   |
| Kasus Uji       | Laporkan Info Hukum dengan kondisi alternatif satu  |
| Prosedur        | <ol style="list-style-type: none"> <li>1. Mengakses halaman detail info hukum.</li> <li>2. Menekan ikon <i>Report</i>.</li> <li>3. Menekan tombol Simpan tanpa mengisi alasan melaporkan info hukum.</li> </ol> |
| Expected Result | Sistem menampilkan pesan “ <i>Please fill out this field</i> ”.   |
| Result          | Sistem menampilkan pesan “ <i>Please fill out this field</i> ”.   |





|        |       |
|--------|-------|
|        |       |
| Status | Valid |

### 6.3.43 Pengujian Validasi Lihat Profil

Tabel 6.79 merupakan hasil dari pengujian validasi *use case* lihat profil yang berhasil.

**Tabel 6.79 Pengujian Validasi Lihat Profil**

|                 |   |
|-----------------|---|
| Kode Kebutuhan  | SKPP-F-43   |
| Kasus Uji       | Lihat Profil  |
| Prosedur        | <ol style="list-style-type: none"> <li>1. Menekan <i>username</i> yang ada pada menu <i>bar</i>.</li> <li>2. Memilih Profil yang ada pada <i>dropdown</i>.</li> </ol> |
| Expected Result | Sistem menampilkan halaman profil.  |
| Result          | <p>Sistem menampilkan halaman profil.</p>   |
| Status          | Valid   |

### 6.3.44 Pengujian Validasi Edit Profil

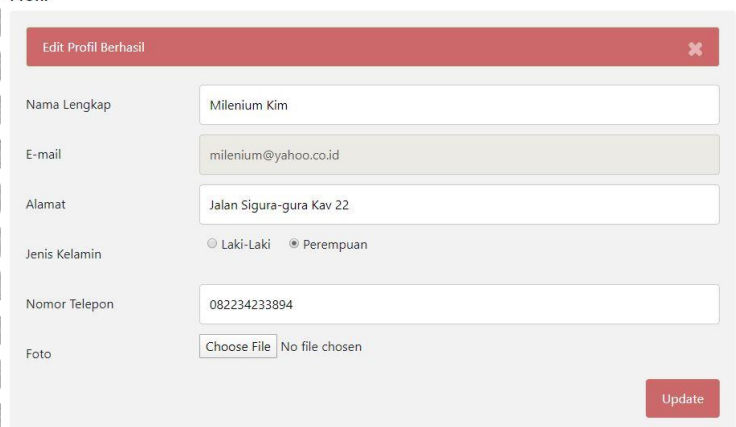
1. Pengujian validasi edit profil





Tabel 6.80 merupakan hasil dari pengujian validasi *use case* edit profil yang berhasil.

**Tabel 6.80 Pengujian Validasi Edit Profil**

|                        |  |
|------------------------|--|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-44</b>   |
| Kasus Uji              | Edit Profil  |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakses halaman Profil.</li> <li>2. Mengganti Alamat menjadi "Jalan Sigura-gura Kav 22".</li> <li>3. Menekan tombol <i>Update</i>.</li> </ol>                |
| <i>Expected Result</i> | Sistem menyimpan perubahan profil ke dalam <i>database</i> dan menampilkan notifikasi "Edit Profil Berhasil".  |
| <i>Result</i>          | <p>Sistem menyimpan perubahan profil ke dalam <i>database</i> dan menampilkan notifikasi "Edit Profil Berhasil".</p>  |
| Status                 | Valid  |

2. Pengujian validasi edit profil alternatif satu

Tabel 6.81 merupakan hasil dari pengujian validasi *use case* edit profil yang tidak berhasil karena ada kolom pada *form* edit profil yang kosong.

**Tabel 6.81 Pengujian Validasi Edit Profil Alternatif Satu**

|                        |   |
|------------------------|---|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-44</b>  |
| Kasus Uji              | Edit Profil   |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakses halaman Profil.</li> <li>2. Menekan tombol <i>Update</i> dengan mengosongkan Alamat.</li> </ol> |
| <i>Expected Result</i> | Sistem menampilkan pesan " <i>Please fill out this field</i> ".   |
| <i>Result</i>          | Sistem menampilkan pesan " <i>Please fill out this field</i> ".   |





|        |  |
|--------|--|
| Profil | Nama Lengkap: Milenium Kim<br>E-mail: milenium@yahoo.co.id<br>Alamat: <input type="text"/><br>Jenis Kelamin: <input type="radio"/> Laki-Laki <input checked="" type="radio"/> Perempuan <span style="border: 1px solid black; padding: 2px;">Please fill out this field.</span><br>Nomor Telepon: 082234233894<br>Foto: <input type="button" value="Choose File"/> No file chosen<br><input type="button" value="Update"/> |
| Status | Valid  |

### 6.3.45 Pengujian Validasi Lihat Daftar Pengacara

Tabel 6.82 merupakan hasil dari pengujian validasi *use case* lihat daftar pengacara yang berhasil.

Tabel 6.82 Pengujian Validasi Lihat Daftar Pengacara

|                        |   |
|------------------------|---|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-45</b>  |
| Kasus Uji              | Lihat Daftar Pengacara  |
| Prosedur               | <ol style="list-style-type: none"> <li>Mengakses halaman <i>Home</i>.</li> <li>Menekan tombol Lihat Semua.</li> </ol> |
| <i>Expected Result</i> | Sistem menampilkan halaman daftar pengacara.  |
| <i>Result</i>          | Sistem menampilkan halaman daftar pengacara.  |





|        |       |
|--------|-------|
|        |       |
| Status | Valid |

### 6.3.46 Pengujian Validasi Lihat Profil Pengacara

Tabel 6.83 merupakan hasil dari pengujian validasi *use case* lihat profil pengacara yang berhasil.

Tabel 6.83 Pengujian Validasi Lihat Profil Pengacara

|                 |   |
|-----------------|---|
| Kode Kebutuhan  | SKPP-F-46   |
| Kasus Uji       | Lihat Profil Pengacara  |
| Prosedur        | <ol style="list-style-type: none"> <li>1. Mengakses halaman Daftar Pengacara.</li> <li>2. Memilih pengacara dengan nama "Entin Endah Cahyati S.H".</li> </ol> |
| Expected Result | Sistem menampilkan halaman profil pengacara yang dipilih.   |
| Result          | Sistem menampilkan halaman profil pengacara yang dipilih.   |





|        |       |
|--------|-------|
|        |       |
| Status | Valid |

### 6.3.47 Pengujian Validasi Cari Pengacara

#### 1. Pengujian validasi cari pengacara

Tabel 6.84 merupakan hasil dari pengujian validasi *use case* cari pengacara yang berhasil.

**Tabel 6.84 Pengujian Validasi Cari Pengacara**

|                        |   |
|------------------------|---|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-47</b>  |
| Kasus Uji<br>Prosedur  | Cari Pengacara<br>1. Mengakses halaman Cari Pengacara<br>2. Memilih Jenis Pengacara Teknologi Informasi.<br>3. Memilih Kabupaten Malang.<br>4. Menekan tombol Cari Pengacara. |
| <i>Expected Result</i> | Sistem menampilkan hasil pencarian pada halaman Cari Pengacara.   |
| <i>Result</i>          | Sistem menampilkan hasil pencarian pada halaman Cari Pengacara.   |






|        |   |
|--------|---|
|        |  <p>Entin Endah<br/>Cahyati S.H<br/>mantulll</p> |
| Status | Valid   |

## 2. Pengujian validasi cari pengacara alternatif satu

Tabel 6.85 merupakan hasil dari pengujian validasi *use case* cari pengacara yang tidak berhasil karena ada kolom pada *form* yang kosong.

**Tabel 6.85 Pengujian Validasi Cari Pengacara Alternatif Satu**

|                        |   |
|------------------------|---|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-47</b>  |
| Kasus Uji              | Cari Pengacara dengan kondisi alternatif satu   |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakses halaman Cari Pengacara</li> <li>2. Memilih Jenis Pengacara Teknologi Informasi.</li> <li>3. Memilih Kabupaten Malang.</li> <li>4. Menekan tombol Cari Pengacara tanpa mengisi jenis pengacara dan kabupaten.</li> </ol> |
| <i>Expected Result</i> | Sistem menampilkan pesan "Pengacara Tidak Ditemukan".   |
| <i>Result</i>          | <p>Sistem menampilkan pesan "Pengacara Tidak Ditemukan".</p>    |
| Status                 | Valid   |

## 6.3.48 Pengujian Validasi Lihat Info Hukum

Tabel 6.86 merupakan hasil dari pengujian validasi *use case* lihat info hukum yang berhasil.

**Tabel 6.86 Pengujian Validasi Lihat Info Hukum**

|                       |  |
|-----------------------|--|
| <b>Kode Kebutuhan</b> | <b>SKPP-F-48</b>   |
| Kasus Uji             | Lihat Info Hukum   |
| Prosedur              | <ol style="list-style-type: none"> <li>1. Mengakses halaman Blog.</li> </ol> |





|                        |   |
|------------------------|---|
|                        | 2. Memilih Info Hukum Dengan Judul “RUU Hukum Acara Perdata Bukan Prioritas dalam Prolegnas, Mengapa?”  |
| <i>Expected Result</i> | Sistem menampilkan detail info hukum yang dipilih.  |
| <i>Result</i>          | Sistem menampilkan detail info hukum yang dipilih.<br><br><div style="text-align: center;"> <p><b>RUU Hukum Acara Perdata Bukan Prioritas dalam Prolegnas, Mengapa?</b></p> <p>By: <a href="#">pratamaputra</a> 2019-11-13 03:56:24 2 Report</p> </div> |
| Status                 | Valid   |

### 6.3.49 Pengujian Validasi Tambah Hasil Pertemuan Konsultasi Tatap Muka

#### 1. Pengujian validasi tambah hasil pertemuan konsultasi tatap muka

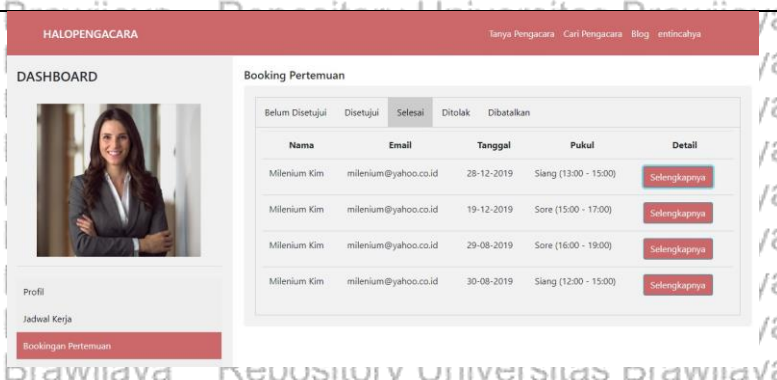
Tabel 6.87 merupakan hasil dari pengujian validasi *use case* tambah hasil pertemuan konsultasi tatap muka yang berhasil.

**Tabel 6.87 Pengujian Validasi Tambah Hasil Pertemuan Konsultasi Tatap Muka**

|                        |   |
|------------------------|---|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-49</b>  |
| Kasus Uji              | Tambah Hasil Pertemuan Konsultasi Tatap Muka  |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakses halaman Booking Pertemuan.</li> <li>2. Memilih <i>tablink</i> Disetujui.</li> <li>3. Memilih bookingan klien yang sudah selesai melakukan konsultasi tatap muka.</li> <li>4. Menekan <i>link</i> Selesai pada bookingan klien.</li> <li>5. Memasukkan hasil pertemuan konsultasi tatap muka “sudah melakukan konsultasi dengan lancar.”</li> <li>6. Menekan tombol Simpan.</li> </ol> |
| <i>Expected Result</i> | Sistem menyimpan data hasil pertemuan konsultasi tatap muka pada <i>database</i> dan menampilkan halaman Bookingan Pertemuan.   |
| <i>Result</i>          | Sistem menyimpan data hasil pertemuan konsultasi tatap muka pada <i>database</i> dan menampilkan halaman Bookingan Pertemuan.   |





|  |       |
|--|-------|
|  |       |
| Status   | Valid |

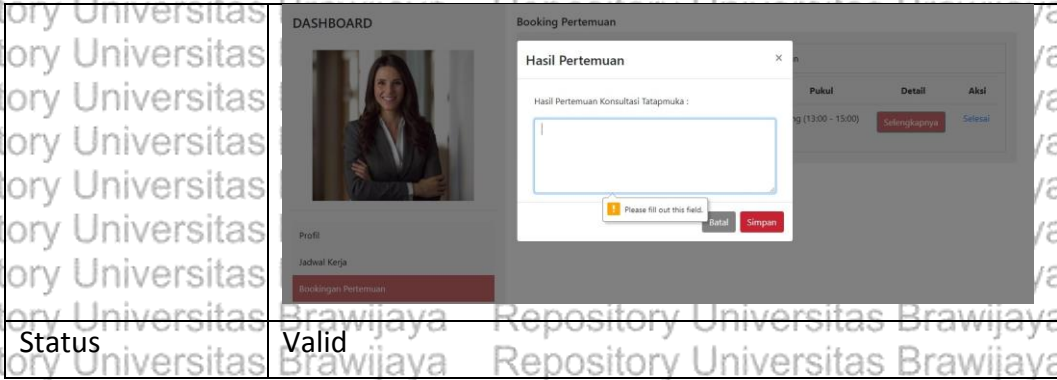
2. Pengujian validasi tambah hasil pertemuan konsultasi tatap muka alternatif satu

Tabel 6.88 merupakan hasil dari pengujian validasi *use case* hasil pertemuan konsultasi tatap muka yang tidak berhasil karena ada kolom pada *form* berita acara pertemuan konsultasi tatap muka yang kosong.

**Tabel 6.88 Pengujian Validasi Tambah Hasil Pertemuan Konsultasi Tatap Muka Alternatif Satu**

|                        |   |
|------------------------|---|
| Kode Kebutuhan         | SKPP-F-14   |
| Kasus Uji              | Tambah Hasil Pertemuan Konsultasi Tatap Muka dengan kondisi alternatif satu   |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakses halaman Booking Pertemuan.</li> <li>2. Memilih <i>tablink</i> Disetujui.</li> <li>3. Memilih bookingan klien yang sudah selesai melakukan konsultasi tatap muka.</li> <li>4. Menekan <i>link</i> Selesai pada bookingan klien.</li> <li>5. Menekan tombol Simpan tanpa memasukkan hasil pertemuan konsultasi tatap muka.</li> </ol> |
| <i>Expected Result</i> | Sistem menampilkan pesan " <i>Please fill out this field</i> ".   |
| <i>Result</i>          | Sistem menampilkan pesan " <i>Please fill out this field</i> ".   |





Status Valid

### 6.3.50 Pengujian Validasi Tambah Progres Kasus

#### 1. Pengujian validasi tambah progres kasus

Tabel 6.89 merupakan hasil dari pengujian validasi *use case* tambah progres kasus yang berhasil.

**Tabel 6.89 Pengujian Validasi Tambah Progres Kasus**

|                        |  |
|------------------------|--|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-50</b>   |
| <b>Kasus Uji</b>       | Tambah Progres Kasus   |
| <b>Prosedur</b>        | <ol style="list-style-type: none"> <li>1. Mengakses Manajemen Kasus.</li> <li>2. Menekan tombol Selengkapnya pada klien bernama "Chris Evans".</li> <li>3. Menekan tombol Tambah Progres Kasus</li> <li>4. Memasukkan tanggal "22-06-2020".</li> <li>5. Memasukkan progres "mempersiapkan dokumen terkait"</li> <li>6. Menekan tombol Simpan.</li> </ol> |
| <b>Expected Result</b> | Sistem menyimpan data tambah progres kasus pada <i>database</i> dan menampilkan halaman Manajemen Kasus dengan pesan "Progres Berhasil Ditambahkan".   |
| <b>Result</b>          | Sistem menyimpan data tambah progres kasus pada <i>database</i> dan menampilkan halaman Manajemen Kasus dengan pesan "Progres Berhasil Ditambahkan".   |
|                        |  |
| <b>Status</b>          | <b>Valid</b>   |

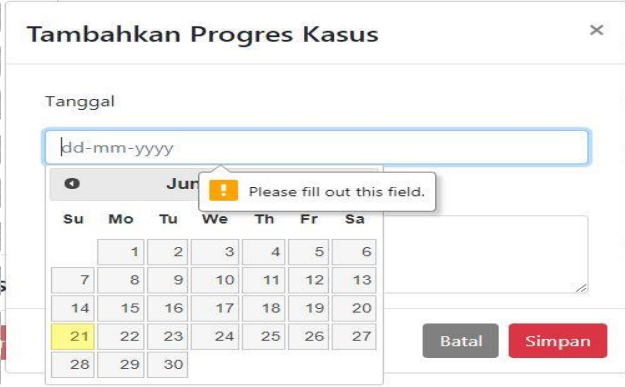




## 2. Pengujian validasi tambah progres kasus alternatif satu

Tabel 6.90 merupakan hasil dari pengujian validasi *use case* tambah progres kasus yang tidak berhasil karena ada kolom pada *form* tambah progres kasus yang kosong.

**Tabel 6.90 Pengujian Validasi Tambah Progres Kasus Alternatif Satu**

| Kode Kebutuhan  | SKPP-F-50   |
|-----------------|---|
| Kasus Uji       | Tambah Progres Kasus dengan kondisi alternatif satu   |
| Prosedur        | <ol style="list-style-type: none"> <li>1. Mengakses Manajemen Kasus.</li> <li>2. Menekan tombol Selengkapnya pada klien bernama "Chris Evans".</li> <li>3. Menekan tombol Tambah Progres Kasus.</li> <li>4. Menekan tombol Simpan tanpa mengisi tanggal dan progres.</li> </ol> |
| Expected Result | Sistem menampilkan pesan "Please fill out this field".  |
| Result          | <p>Sistem menampilkan pesan "Please fill out this field".</p>   |
| Status          | Valid   |

## 6.3.51 Pengujian Validasi Edit Progres Kasus

### 1. Pengujian validasi edit progres kasus

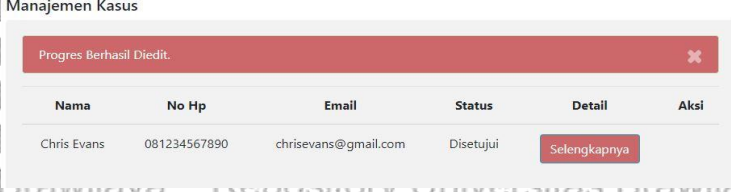
Tabel 6.91 merupakan hasil dari pengujian validasi *use case* edit progres kasus yang berhasil.

**Tabel 6.91 Pengujian Validasi Edit Progres Kasus**

| Kode Kebutuhan | SKPP-F-51   |
|----------------|---|
| Kasus Uji      | Edit Progres Kasus  |
| Prosedur       | <ol style="list-style-type: none"> <li>1. Mengakses Manajemen Kasus.</li> </ol> |





|                        |  |
|------------------------|--|
|                        | <ol style="list-style-type: none"> <li>Menekan tombol Selengkapnya pada klien bernama "Chris Evans".</li> <li>Menekan tombol Edit Pada progres tanggal "22-06-2020".</li> <li>Mengedit progres "mempersiapkan dokumen terkait sebagaimana hal yang disampaikan pekan lalu".</li> <li>Menekan tombol Simpan.</li> </ol> |
| <i>Expected Result</i> | Sistem menyimpan data edit progres kasus pada <i>database</i> dan menampilkan halaman Manajemen Kasus dengan pesan "Progres Berhasil Diedit".  |
| <i>Result</i>          | <p>Sistem menyimpan data edit progres kasus pada <i>database</i> dan menampilkan halaman Manajemen Kasus dengan pesan "Progres Berhasil Diedit".</p>    |
| <i>Status</i>          | Valid  |

## 2. Pengujian validasi edit progres kasus alternatif satu

Tabel 6.92 merupakan hasil dari pengujian validasi *use case* edit progres kasus yang tidak berhasil karena ada kolom pada *form* edit progres kasus yang kosong.

**Tabel 6.92 Pengujian Validasi Edit Progres Kasus Alternatif Satu**

|                        |   |
|------------------------|---|
| <b>Kode Kebutuhan</b>  | <b>SKPP-F-51</b>  |
| Kasus Uji              | Edit Progres Kasus dengan kondisi alternatif satu   |
| Prosedur               | <ol style="list-style-type: none"> <li>Mengakses Manajemen Kasus.</li> <li>Menekan tombol Selengkapnya pada klien bernama "Chris Evans".</li> <li>Menekan tombol Edit Pada progres tanggal "22-06-2020".</li> <li>Menekan tombol Simpan tanpa mengisi progres.</li> </ol> |
| <i>Expected Result</i> | Sistem menampilkan pesan "Please fill out this field".  |
| <i>Result</i>          | Sistem menampilkan pesan "Please fill out this field".  |





|        |   |
|--------|---|
|        | <div style="border: 1px solid black; padding: 5px;"> <div style="text-align: right; border-bottom: 1px solid black; margin-bottom: 5px;"> <span style="float: right;">x</span> </div> <div style="margin-bottom: 5px;"> <p>Tanggal</p> <input type="text" value="22-06-2020"/> </div> <div style="margin-bottom: 5px;"> <p>Progres</p> <div style="border: 1px solid gray; height: 20px; width: 100%;"></div> </div> <div style="text-align: right; margin-top: 5px;"> <span style="font-size: 12px;">!</span> Please fill out this field.                 <span style="margin-left: 10px; border: 1px solid gray; padding: 2px 5px;">Batal</span> <span style="margin-left: 10px; border: 1px solid gray; padding: 2px 5px; color: white;">Simpan</span> </div> </div> |
| Status | Valid   |

### 3. Pengujian validasi edit progres kasus alternatif dua

Tabel 6.93 merupakan hasil dari pengujian validasi *use case* edit progres kasus yang tidak berhasil karena membatalkan edit progres.

**Tabel 6.93 Pengujian Validasi Edit Progres Alternatif Dua**

| <b>Kode Kebutuhan</b>  | <b>SKPP-F-51</b>  |  |         |      |            |  |  |
|------------------------|---|--|---------|------|------------|--|--|
| Kasus Uji              | Edit dengan kondisi alternatif dua  |  |         |      |            |  |  |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakses Manajemen Kasus.</li> <li>2. Menekan tombol Selengkapnya pada klien bernama "Chris Evans".</li> <li>3. Menekan tombol Batal.</li> </ol>   |  |         |      |            |  |  |
| <i>Expected Result</i> | Sistem menampilkan <i>modal</i> Selengkapnya dari manajemen kasus.  |  |         |      |            |  |  |
| <i>Result</i>          | <p>Sistem menampilkan <i>modal</i> Selengkapnya dari manajemen kasus.</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 5px;"> <p>Progres Kasus</p> <div style="text-align: center; margin-bottom: 5px;"> <span style="background-color: #e74c3c; color: white; padding: 2px 10px; border-radius: 3px;">Tambah Progres Kasus</span> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Tanggal</th> <th style="width: 60%;">Progres</th> <th style="width: 20%;">Aksi</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">22-06-2020</td> <td style="font-size: 10px;">mempersiapkan dokumen terkait yang dibutuhkan dalam persidangan.</td> <td style="text-align: center; font-size: 10px;"> <a href="#">Edit</a>   <a href="#">Hapus</a> </td> </tr> </tbody> </table> </div> | Tanggal                                      | Progres | Aksi | 22-06-2020 | mempersiapkan dokumen terkait yang dibutuhkan dalam persidangan. | <a href="#">Edit</a>   <a href="#">Hapus</a> |
| Tanggal                | Progres   | Aksi   |         |      |            |  |  |
| 22-06-2020             | mempersiapkan dokumen terkait yang dibutuhkan dalam persidangan.  | <a href="#">Edit</a>   <a href="#">Hapus</a> |         |      |            |  |  |
| Status                 | Valid   |  |         |      |            |  |  |


### 6.3.52 Pengujian Validasi Hapus Progres Kasus

Tabel 6.94 merupakan hasil dari pengujian validasi *use case* hapus progres kasus yang berhasil.





Tabel 6.94 Pengujian Validasi Hapus Progres Kasus

|                 |  |
|-----------------|--|
| Kode Kebutuhan  | SKPP-F-52  |
| Kasus Uji       | Hapus Progres Kasus  |
| Prosedur        | <ol style="list-style-type: none"> <li>1. Mengakses Manajemen Kasus.</li> <li>2. Menekan tombol Selengkapnya pada klien bernama "Chris Evans".</li> <li>3. Menekan tombol hapus pada tanggal "22-06-2020".</li> </ol>                      |
| Expected Result | Sistem menyimpan data hapus progres kasus pada <i>database</i> dan menampilkan halaman Manajemen Kasus dengan pesan "Progres Berhasil Dihapus".  |
| Result          | <p>Sistem menyimpan data hapus progres kasus pada <i>database</i> dan menampilkan halaman Manajemen Kasus dengan pesan "Progres Berhasil Dihapus".</p>  |
| Status          | Valid  |

### 6.3.53 Pengujian Validasi Lihat Progres Kasus

Tabel 6.95 merupakan hasil dari pengujian validasi *use case* lihat progres kasus yang berhasil.

Tabel 6.95 Pengujian Validasi Lihat Progres Kasus

|                 |   |
|-----------------|---|
| Kode Kebutuhan  | SKPP-F-53   |
| Kasus Uji       | Lihat Progres Kasus   |
| Prosedur        | <ol style="list-style-type: none"> <li>1. Mengakses halaman Progres Kasus.</li> <li>2. Memilih tombol Selengkapnya pada pengacara Pratama Putra S.H M.H.</li> </ol> |
| Expected Result | Sistem menampilkan <i>modal</i> yang berisikan progres kasus.   |
| Result          | Sistem menampilkan <i>modal</i> yang berisikan progres kasus  |





|            |   |         |               |            |  |
|------------|---|---------|---------------|------------|--|
|            | <p>Progres Kasus</p> <div style="border: 1px solid #ccc; padding: 5px;"> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%; text-align: center;">Tanggal</td> <td style="text-align: center;">Progres Kasus</td> </tr> <tr> <td style="text-align: center;">22-06-2020</td> <td style="text-align: center;">mempersiapkan dokumen terkait yang dibutuhkan dalam persidangan.</td> </tr> </table> </div> <p style="text-align: right; color: red; font-weight: bold;">Close</p> | Tanggal | Progres Kasus | 22-06-2020 | mempersiapkan dokumen terkait yang dibutuhkan dalam persidangan. |
| Tanggal    | Progres Kasus   |         |               |            |  |
| 22-06-2020 | mempersiapkan dokumen terkait yang dibutuhkan dalam persidangan.  |         |               |            |  |
| Status     | Valid   |         |               |            |  |

### 6.3.54 Pengujian Validasi Lihat Hasil Pertemuan Konsultasi Tatap Muka

Tabel 6.96 merupakan hasil dari pengujian validasi *use case* lihat hasil pertemuan konsultasi tatap muka yang berhasil.

**Tabel 6.96 Pengujian Validasi Lihat Hasil Pertemuan Konsultasi Tatap Muka**

|                        |   |                 |   |
|------------------------|---|-----------------|---|
| Kode Kebutuhan         | SKPP-F-54   |                 |   |
| Kasus Uji              | Lihat Hasil Pertemuan Konsultasi Tatap Muka   |                 |   |
| Prosedur               | <ol style="list-style-type: none"> <li>1. Mengakses halaman Bookingan Saya.</li> <li>2. Memilih salah satu daftar konsultasi tatap muka</li> <li>3. Menekan tombol Selengkapnya.</li> </ol>   |                 |   |
| <i>Expected Result</i> | Sistem menampilkan modal hasil pertemuan.   |                 |   |
| <i>Result</i>          | <p>Sistem menampilkan modal hasil pertemuan.</p> <div style="border: 1px solid #ccc; padding: 5px;"> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%; text-align: center;">Hasil Pertemuan</td> <td style="text-align: center;">sudah didapatkan titik temu dalam pertemuan</td> </tr> </table> </div> | Hasil Pertemuan | sudah didapatkan titik temu dalam pertemuan |
| Hasil Pertemuan        | sudah didapatkan titik temu dalam pertemuan   |                 |   |
| Status                 | Valid   |                 |   |

### 6.4 Pengujian *Compatibility*

Tahap pengujian *compatibility* merupakan tahap untuk menguji sistem konsultasi dan pencarian pengacara yang dikembangkan dapat dijalankan dalam berbagai peramban web (*browser*). pada pengujian ini menggunakan aplikasi *sortSite* Trial 5.37.868.0. Indikator masalah *compatibility* yang ada pada *SortSite* dibagi menjadi tiga. *Priority 1* merupakan *Critical Issue* dimana fungsionalitas ataupun konten yang tidak bisa digunakan/didukung pada suatu *browser*. *Priority 2* merupakan *Major Issue* dimana terdapat masalah utama pada performa ataupun tampilan pada suatu *browser*. *Priority 3* merupakan *Minor Issue* dimana terdapat masalah minor/kecil yang terdapat pada performa ataupun tampilan pada suatu *browser*. Kategori masalah *compatibility* yang menjadi kategori di *SortSite* ditunjukkan pada gambar 6.5.

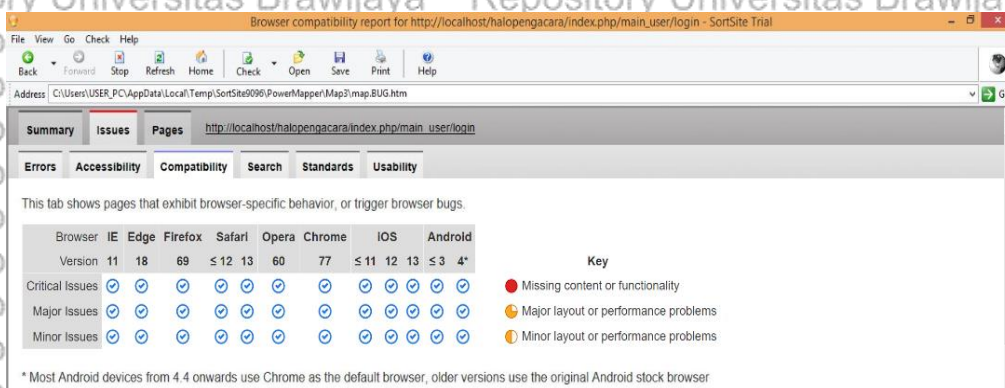




### Key

- Missing content or functionality
- Major layout or performance problems
- Minor layout or performance problems

Gambar 6.5 Indikator Masalah *Compatibility* SortSite



Gambar 6.6 Hasil Pengujian *Compatibility*

Pada gambar 6.6 menunjukkan hasil dari pengujian *compatibility* sistem konsultasi dan pencarian pengacara yang telah dibuat. Hasil pengujian menunjukkan sistem dapat berjalan di pada enam browser berbeda yang memiliki versi berbeda pula dan juga dapat berjalan pada *operating system* yang berbeda. Sistem konsultasi dan pencarian pengacara dapat berjalan dengan baik dalam berbagai macam peramban dan *operating system* sesuai dengan hasil dari pengujian yang tidak mengandung indikator masalah.



## BAB 7 PENUTUP

Pada bab penutup membahas mengenai kesimpulan dari pembuatan penelitian berjudul pengembangan sistem konsultasi dan pencarian pengacara dan saran saran yang digunakan untuk bahan pertimbangan pengembangan penelitian selanjutnya.

### 7.1 Kesimpulan

Kesimpulan merupakan hasil pengamatan dari proses rekayasa kebutuhan, perancangan, implementasi, dan pengujian. Kesimpulan yang didapatkan dari penelitian berjudul Sistem Konsultasi dan Pencarian Pengacara adalah sebagai berikut:

1. Pada tahap rekayasa kebutuhan, sistem konsultasi dan pencarian pengacara mempunyai 4 aktor. Spesifikasi kebutuhan iterasi pertama dan kedua menghasilkan 54 kebutuhan fungsional dan 1 kebutuhan non fungsional, total terdapat 55 kebutuhan. Pada pemodelan kebutuhan dibagi menjadi pemodelan *use case diagram* dan *use case scenario*.
2. Pada tahap perancangan, perancangan *sequence diagram* yang menghasilkan 3 *sequence diagram*. Bagian perancangan *class diagram* hasil dari iterasi pertama dan iterasi kedua menghasilkan 5 kelas *controller* dan 5 kelas *model*. Perancangan basis data dilakukan dengan pembuatan CMD yang dijelaskan menggunakan ERD, pada iterasi pertama dan kedua pembuatan ERD dihasilkan 18 entitas yang di transformasikan ke PMD menghasilkan 18 tabel. Perancangan komponen iterasi pertama dan kedua menggunakan empat sampel. Sedangkan untuk perancangan antarmuka merupakan rancangan antarmuka yang akan diterapkan pada sistem.
3. Pada tahap implementasi, spesifikasi sistem yang terbagi menjadi spesifikasi perangkat lunak yang digunakan untuk pengembangan sistem konsultasi dan pencarian pengacara ini. Implementasi basis data menghasilkan 18 tabel sesuai dengan perancangan basis data yang dijelaskan sebelumnya. Implementasi komponen didasarkan pada perancangan komponen yang dituliskan menjadi kode program, dalam implementasi antarmuka dihasilkan berdasarkan perancangan antarmuka yang dirancangan sebelumnya.
4. Pada tahap pengujian, terdapat klas *Driver* yang digunakan untuk menjalankan pengujian unit. Pengujian unit terdapat tiga sampel yang menghasilkan valid dalam semua jalur sehingga dinyatakan lolos. Pengujian validasi terdapat 87 kasus uji didefinisikan dan menghasilkan 100% valid pada semua kasus uji tersebut didapatkan kesimpulan bahwa seluruh spesifikasi kebutuhan telah terpenuhi. Pada pengujian *compatibility* menghasilkan bahwa sistem berjalan dengan baik pada 6 peramban web (*browser*) dan 2 *operating system*.





## 7.2 Saran

Saran merupakan bahan pertimbangan dalam pengembangan penelitian selanjutnya. Saran yang diberikan bagi bahan pertimbangan penelitian pengembangan sistem konsultasi dan pencarian pengacara adalah sebagai berikut:

1. Terdapat notifikasi jika mendekati waktu konsultasi tatap muka via email untuk memberikan peringatan waktu untuk konsultasi tatap muka.
2. Sistem dapat ditambahkan fitur *chat* untuk memudahkan berkomunikasi antara pengacara dan klien.



## DAFTAR REFERENSI

Ali, A., Heryani, W., 2012. *Menjelajahi Kajian Empiris Terhadap Hukum Edisi Pertama*. Jakarta: Kencana Prenadamedia Group.

Asnawi, I., 2018. *Pengembangan Sistem Informasi Lembaga Pengkajian dan Konsultan Bantuan Hukum (LPKBH) Menggunakan Zachman Framework (Studi Pada LPKBH Al-Baihaqy)*. S1. Universitas Brawijaya.

Malik, M., 2017. *Rancang Bangun Sistem Informasi Manajemen Pengelolaan dan Penjadwalan Layanan Konsultasi Bantuan Hukum Pada LBH Makassar*. S1. UIN Alauddin Makassar. Tersedia di <<http://repositori.uin-alauddin.ac.id/3547/1/Maulana%20Malik.pdf>> [Diakses 27 Juli 2020]

Sitepu, D., 2015. *Sistem Konsultasi Jenis Kejahatan Information Technology Berbasis Web*. S1. Universitas Pakuan. Tersedia di <<http://perpustakaan.fmipa.unpak.ac.id/file/e-jurnal%20domas%20065111328.pdf>> [Diakses 4 Februari 2019]

Wysocki, R. K., 2006. *Effective Software Project Management*. Indiana: Wiley Publishing, Inc.

S, Rosa A. dan Shalahuddin, M., 2018. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Edisi 2. Bandung: Informatika Bandung.

Kendall, K.E., and Kendall, J.E., 2011. *Systems Analysis and Design*. 8th Edition. New York: Pearson Education.

Pressman, Rogers S., 2010. *Rekayasa Perangkat Lunak, buku 1*. Edisi 7. Diterjemahkan dari Bahasa Inggris oleh A Nugroho, et. al. 2012. Yogyakarta: Andi.

Whitten, J.L., Bentley, L.D. & Dittman, K.C., 2004. *Metode Desain dan Analisis Sistem*. Edisi 6. Diterjemahkan dari Bahasa Inggris oleh Tim Penerjemah ANDI. 2004. Yogyakarta: Andi.

Booch, G., James, R., and Ivar, J., 2005. *The Unified Modeling Language User Guide*. Second Edition. Boston: Addison Wesley Professional.

Badiyanto., 2013. *Buku Pintar Framework Yii Cara Mudah Membangun Aplikasi Web PHP*. Yogyakarta: Mediakom.

Septian, G., 2011. *Trik Pintar Menguasai CodeIgniter*. Jakarta: PT Elex Media Komputindo.

Kurniadi, Adi., 2011. *Pemrograman Microsoft Visual Basic 6*. Jakarta: PT Elex Media Komputindo.

Nugroho, Adi., 2006. *E-commerce*. Bandung: Informatika Bandung.

Bride, Mac., 2007. *Javascript*. London: Holder Headline.

Arief, M. Rudianto., 2011. *Pemrograman Web Dinamis Menggunakan Php dan Mysql*. Yogyakarta: ANDI.





Sommerville, I., 2001. *Software Engineering (Rekayasa Perangkat Lunak), jilid 2.*

Edisi 6. Diterjemahkan dari Bahasa Inggris oleh Y Hanum. 2003. Jakarta: Erlangga.





## LAMPIRAN A HASIL WAWANCARA

### 1. Hasil Wawancara Narasumber Yayan Riyanto, S.H, M.H.

Tanggal : 12 Februari 2019

Waktu : 13:00 – 14:30

Tempat : Kantor Advokat Yayan Riyanto, S.H, M.H

Narasumber : Yayan Riyanto, S.H, M.H

Pewawancara : Entin Endah Cahyati

P : Pewawancara

N : Narasumber

|   |   |
|---|---|
| P | Kenapa mencari pengacara di Malang sangat susah pak?  |
| N | Karena mereka tidak punya kantor, yang punya kantor cuman beberapa. Di malang sendiri sudah ada 1000 lebih pengacara, tapi yang memiliki kesadaran untuk berusaha hanya sedikit. Banyak pengacara yang berusaha di rumah-rumah, tidak di kantor.  |
| P | Saya kemarin menemukan pengacara tapi di rumah dan dalam rumah itu juga ada klinik gigi, jadi saya ingin masuk ke dalam juga ragu itu benar pengacara atau tidak.   |
| N | Iya memang kebanyakan seperti itu, banyak yang praktek di rumah dikasih plang nama pengacara seperti itu. Banyak kalau kalian mencari di google dan mendapatkan alamat pengacara disana. Kalian datang, nanti keanyakan kalian akan menemukan alamat itu di rumah-rumah bukan di kantor. Mereka tidak serius dalam membuka usaha, harusnya kalau ingin usaha mereka dianggap serius. Mereka harusnya memiliki kantor, agar tidak terlihat pengacara abal-abalan atau beneran. |
| P | Apakah bapak sering mendapatkan kasus?  |
| N | Kalau saya sudah bekerja sebagai pengacara selama 20 tahun, dan sudah ada kantor seperti ini. Untuk pengacara yan terbilang masih belum senior seperti saya ya saya sudah banyak menangani kasus tidak hanya di malang saja tapi klien saya sudah sampai jakarta.   |
| P | Apakah ada pembagian perkara dalam pengacara menangani kasus?   |
| N | Di Indonesia sebenarnya tidak ada pembagian perkara, jadi advokat bisa menangani semua tidak ada spesialis A dan spesialis B. jadi biasanya advokat itu condongnya kemana, kesukaannya menangani kasus apa. Kalau saya tidak mau menangani narkoba tapi suka menangani bisnis, jadi seperti itu.  |





|   |   |
|---|---|
| P | Bagaimana cara memesan pengacara di kantor bapak?   |
| N | Bisa langsung ke kantor nanti ada yang melayani maunya apa. Nanti kalau mau ketemu saya harus janji dulu dengan saya. Atau jika sudah memiliki kontak saya bisa menghubungi kontak saya untuk janji bertemu.  |
| P | Apakah kasus yang sudah ditangani bisa ditampilkan ke dalam sistem pak?   |
| N | Pengacara wajib merahasiakan kasus klien. Tidak bisa diumbar begitu saja dan tidak etis untuk mengumbar kasus klien. Jika di televisi ada kasus yang diumbar berarti kasus itu sudah tidak bisa diselesaikan. Biasanya ada klien yang mau kasusnya cepat selesai dan tidak di umbar-umbarkan. |



**2. Hasil Wawancara Narasumber Aswinda Desovi, S.H**

Tanggal : 15 Juli 2019

Waktu : 18:00 – 19:00

Tempat : Rumah Narasumber

Narasumber : Aswinda Desovi, S.H

Pewawancara : Entin Endah Cahyati

P : Pewawancara

N : Narasumber

|   |   |
|---|---|
| P | Apa saja syarat untuk menjadi pengacara?  |
| N | Pengacara itu banyak macamnya, ada pengacara magang dan ada juga pengacara yang sudah punya sumpah. Syarat untuk sumpah sendiri itu minimal usia 25 tahun. Kalau syarat umum pengacara magang atau sudah sumpah pastinya harus lulus S1 hukum. Jadi singkatnya alurnya seperti ini, sarjana S1 hukum -> pendidikan khusus profesi advokat -> lulus ujian proresi advokat (UPA) -> magang di Law firm minimal 2 tahun -> (apabila usia sudah mencukupi yaitu 25 tahun) ambil sumpah advokat. Ini sudah diatur dalam UU no 18 tahun 2003 tentang advokat. |
| P | Apakah advokat magang sudah bisa menangani kasus?   |
| N | Sudah bisa, tapi tidak boleh bersidang ke pengadilan. Dan biasanya Cuma jadi pendamping lawyer yang sudah sumpah.   |
| P | Apakah perkara yang ditangani pengacara ada jenisnya?   |
| N | Ada. Biasanya yang umum ada 2 macam, litigasi atau corporate. Litigasi itu bersidang di pengadilan sedangkan corporate itu masalah bisnis ga perlu perlu jalur litigasi. Atau mungkin mau di kelompokkan masing-masing ada pidana, perdata, administrasi dan hukum tata negara, international, dan syariah. Pidana juga ada 2 yaitu pidana umum dan pidana khusus. Untuk yang perdata lingkungnya sangat luas.  |





## LAMPIRAN B ROADMAP SKRIPSI

| No             | Deskripsi Task           | Milestone | Progress | Status   |
|----------------|--------------------------|-----------|----------|----------|
| 1.             | <b>Layanan</b>           |           |          |          |
|                | Konsultasi Tatap Muka    |           | 100      | Done     |
| 2.             | Gunakan Jasa Pengacara   |           | 100      | Done     |
|                | <b>Portfolio</b>         |           |          |          |
|                | Profil Pengacara         |           | 100      | Done     |
| 3.             | Upload Portofolio        |           | 100      | Done     |
|                | Ulasan                   |           | 100      | Done     |
|                | <b>Cari Pengacara</b>    |           |          |          |
|                | List Pengacara           |           | 100      | Done     |
| 4.             | <b>Manajemen Perkara</b> |           |          |          |
|                | Manajemen Kasus          |           | 100      | Done     |
|                | Riwayat Tatap Muka       |           | 100      | Done     |
| 5.             | <b>Konsultasi</b>        |           |          |          |
|                | List Pertanyaan          |           | 100      | Done     |
| 6.             | <b>Info Hukum</b>        |           |          |          |
|                | List Berita              |           | 100      | Done     |
| Total Progress |                          |           | 100      | Complete |