# BAB I PENDAHULUAN

* 1. **Latar Belakang**

Skripsi merupakan karya tulis ilmiah berdasarkan hasil penelitian lapangan dan atau studi kepustakaan yang disusun mahasiswa sesuai dengan bidang studinya sebagai tugas akhir dalam studi formalnya di Program Studi Teknik Informatika Universitas Pamulang. Prosedur untuk program Skripsi berbeda dengan mata kuliah wajib pada umumnya. Berbedanya prosedur dan pembayaran pada program Skripsi dibanding mata kuliah yang lain, menjadikan sebagian mahasiswa harus mencari informasi lebih lanjut ketika ingin mengikuti atau mengambil mata kuliah tersebut. Informasi tentang Skripsi telah disediakan oleh Universitas Pamulang pada situs resminya (www.myunpam.ac.id), namun informasi yang diberikan masih membuat sebagian mahasiswa kurang paham tentang prosedur mata kuliah tersebut. Kurang pahamnya sebagian mahasiswa dalam menelaah setiap informasi Skripsi menjadikan mereka lambat dalam proses studi. Peran aktif mahasiswa dalam mencari informasi seputar Skripsi sangat dibutuhkan, karena prosedur Skripsi dapat berubah sewaktu - waktu sesuai kebijakan jurusan dan universitas.

Jurusan Teknik Informatika UNPAM mempunyai sebuah layanan bagi mahasiswa yang ingin bertanya seputar informasi akademik tanpa harus datang ke jurusan selain situs web, yaitu layanan telepon dan *email*. Kontak *email* dan nomor telepon yang disediakan oleh jurusan Teknik Informatika UNPAM ditujukan bagi masyarakat umum maupun mahasiswa, layanan ini sebenarnya dapat dimanfaatkan oleh mahasiswa dalam mencari informasi seputar prosedur Skripsi namun pelayanan telepon yang masih dikenakan tarif panggilan dan *email* yang responnya tidak interaktif membuat mahasiswa enggan untuk menggunakan layanan yang telah disediakan oleh jurusan. Penulis mencari tahu masalah mahasiswa Teknik Informatika UNPAM dalam mencari informasi seputar Skripsi diantaranya :

1. Sulitnya mencari informasi tentang persyaratan dan langkah - langkah Skripsi yang harus disiapkan kurang lengkap yang kadang masih membuat mahasiswa bingung
2. Sulitnya mencari informasi kontak dosen pembimbing yang harus mencari sendiri atau harus bertanya kepada mahasiswa lain
3. Mahasiswa yang harus datang keprodi hanya untuk memastikan bahwa informasi yang didapat itu *valid*
4. Mahasiswa harus jeli dalam memilih judul sebuah topik pembahasan yang mereka pilih apakah sudah digunakan sebelumnya atau belum

Berdasarkan permasalahan yang ada maka dilakukan penelitian ini yang bertujuan untuk membuat sebuah aplikasi *web* yang menjadi layanan interaktif dalam melakukan pencarian informasi yang *update* dan *valid* seputar Skripsi di jurusan Teknik Informatika UNPAM. Aplikasi dari penelitian yang dilakukan ini dapat digunakan kapan pun dan dapat diakses diamana pun, agar para mahasiswa yang ingin mengetahui informasi seputar prosedur Skripsi mendapatkan informasi secara cepat dan *valid*. Penelitian ini menggunakan sebuah platform milik *Google Inc* yaitu *Dialogflow*. *Dialogflow* adalah sebuah *platform* *chatbot* yang berbasis *machine learning* yang ditujukan untuk mengolah masukan kata dan akan menghasilkan sebuah jawaban berupa teks berformat JSON (*Javascript Object Notation*), pada aplikasi web yang format JSON yang diterima akan diolah menjadi format teks untuk pengguna. Dalam penerapannya nanti, aplikasi ini dapat menjadi sebuah layanan interaktif yang dapat digunakan untuk mahasiswa dalam mencari informasi seputar Skripsi.

* 1. **Identifikasi Masalah**

Berdasarkan latar belakang masalah di atas, identifikasi masalah dalam penelitian ini adalah sebagai berikut:

1. Sulitnya mencari informasi tentang persyaratan dan langkah - langkah Skripsi yang harus disiapkan kurang lengkap yang kadang masih membuat mahasiswa bingung
2. Sulitnya mencari informasi kontak dosen pembimbing yang harus mencari sendiri atau harus bertanya kepada mahasiswa lain
3. Mahasiswa yang harus datang keprodi hanya untuk memastikan bahwa informasi yang didapat itu *valid*
4. Mahasiswa harus jeli dalam memilih judul sebuah topik pembahasan yang mereka pilih apakah sudah digunakan sebelumnya atau belum
   1. **Rumusan Masalah**

Berdasarkan latar belakang dan identifikasi masalah di atas maka dapat disrumuskan masalah yaitu:

1. Bagaimana simulasi proses/tahapan penyusunan Skripsi yang tersusun dengan baik dan benar di Universitas Pamulang?
2. Bagaimana cara membantu mahasiswa jurusan Teknik Informatika dalam memberikan informasi kontak dosen pembimbing skripsi yang efektif?
3. Bagaimana cara membangun Aplikasi Web Auto Response untuk mejawab pertanyaan mahasiswa yang dapat diakses kapanpun dan dimanapun?
4. Bagaimana cara membantu mahasiswa jurusan Teknik Informatika dalam memberikan informasi referensi judul Skripsi yang sudah pernah dibuat di Universitas Pamulang?
   1. **Batasan Penelitian**

Berikut beberapa batasan masalah pada penelitian ini:

1. Aplikasi ini hanya ditujukan untuk menginformasikan sesuatu yang umum seputar Skripsi pada jurusan Teknik Informatika UNPAM
2. Aplikasi ini hanya menampilkan informasi Dosen jurusan Teknik Informatika UNPAM
3. Aplikasi ini hanya menampilkan judul-judul Skripsi yang sudah pernah dibuat di UNPAM
4. Aplikasi ini hanya dapat diakses jika ada koneksi internet
   1. **Tujuan Penelitian**

Tujuan dari penelitian ini adalah:

1. Perancangan aplikasi *web* untuk layanan interaktif dalam menjawab pertanyaan seputar Skripsi jurusan Teknik Informatika UNPAM
2. Melihat pengaruh adanya aplikasi ini dengan cara membandingkan kondisi mahasiswa saat mencari informasi setelah adanya aplikasi
   1. **Manfaat Penelitian**

Manfaat yang didapat dari penilitian ini adalah:

1. Memudahkan mahasiswa dalam mencari tahu informasi yang berkaitan dengan penyusunan Skripsi jurusan Teknik Informatika UNPAM
2. Jurusan Teknik Informatika UNPAM mempunyai layanan informasi yang interaktif untuk mahasiswa yang masih bermasalah dengan penyusunan Skripsi.
   1. **Metode Penelitian**

Metode penelitian merupakan tahapan-tahapan yang dilalui oleh peneliti mulai dari perumusan masalah sampai kesimpulan, yang membentuk sebuah alur yang sistematis. Metode penelitian yang dilakukan terdiri dari metode pengumpulan data, metode pengembangan perangkat lunak serta metode pengembangan perangkat lunak serta metode pengujian perangkat lunak

* + 1. **Metode Pengumpulan Data**

Metodologi yang digunakan dalam penulisan skripsi ini adalah sebagai berikut:

1. Sumber data primer :
2. Observasi

Teknik mengumpulkan data dengan cara dengan melakukan penelitian langsung mengunjungi lokasi penelitian, yaitu Universitas Pamulang. Hal ini dilakukan untuk melihat dari dekat masalah-masalah yang berhubungan dengan pokok bahasan yang diperlukan dalam penelitian ini.

1. Wawancara

Metode pengumpulan data dengan mengadakan tanya jawab secara sistematis dan berlandaskan pada tujuan pembahasan

1. Sampel data

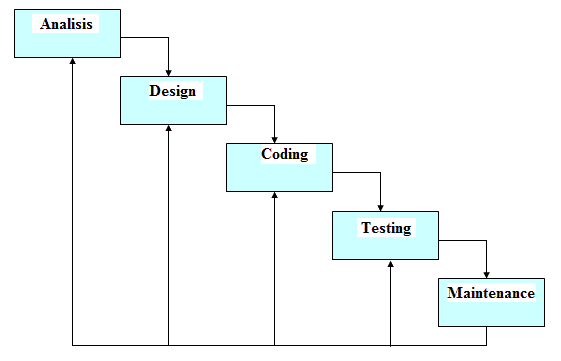
Pengumpulan data dengan cara mengumpulkan data pertanyaan-pertanyaan masalah umum tentang skripsi kepada mahasiswa yang belum, sedang atau sudah skripsi.

1. Sumber data sekunder

Sumber data sekunder diperoleh dengan cara mencari dan mengumpulkan data lengkap dengan mempelajari dan membaca buku-buku yang berhubungan serta menunjang penulisan hasil penelitian.

* + 1. **Metode Pengembangan Perangkat Lunak**

Dalam membangun sistem Aplikasi Web Informasi Auto Response Skripsi Pada Jurusan Tekik Inforamtika Universitas Pamulang, dingunakan metode pengembangan perangkat lunak adalah dengan menggunakan metode *waterfall*. Beberapa proses diagram *waterfall* sebagai berikut :



Gambar 1.1 Skema Model *Waterfall*

* 1. Analisa

Langkah ini merupakan analisa terhadap kebutuhan sistem yang diperlukan dalam pelaksanaan proyek pembuatan aplikasi.

* 1. *Design*

Tahapan dimana dilakukan penuangan fikiran dan perancangan sistem terhadap solusi dari permasalahan yang ada dengan menggunakan permodelan sistem yang nanti mudah dimengerti oleh pengguna.

* 1. *Coding*

Merupakan penerjemah *design* dalam bahasa pemrogaman yang bisa dikenali oleh computer.

* 1. *Testing*

Tahapan akhir dimana sistem dibangun diuji kemampuannya sehingga didapatkan kekurangan dan kelemahan sistem yang kemudian dilakukan pengkajian ulang dan perbaikan terhadap aplikasi tersebut agar menjadi lebih baik dan sempurna.

* 1. *Maintenance*

Tahapan dimana dilalakukan pemeliharaan terhadap aplikasi yang telah dibuat ketika aplikasi sudah diimplementasikan kepada pengguna.

* + 1. **Metode Pengujian Perangkat Lunak**

Pengujian system yang digunakan pada penelitian ini yaitu pengujian *blackbox. Blackbox testing* merupakan pengujian yang mengabaikan mekanisme *internal system* atau komponen dan fokus semata-mata pada *output* yang dihasikan yang merespon input yang dipilih dan kondisi eksekusi. *Blackbox testing* pengujian yang dilakukan untuk mengevaluasi pemenuhan *system* atau komponen dengan kebutuhan fungsional tertentu. Digunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak yang dirancang kebenaran perangkat lunak yang diuji hanya dilihat berdasarkan keluaran yang dihasilkan dari data atau kondisi masukan yang diberikan untuk fungsi yang ada tanpa melihat bagaimana proses untuk mendapatkan keluaran tersebut (Crisdiandiandy Herry Nowor, 2013).

* 1. **Sistematika Penulisan**

Sistematika penulisan yang digunakan dalam penyusunan laporan ini dibedakan dengan pembagian bab sebagai berikut :

BAB I PENDAHULUAN

Pada bab ini memaparkan dan menguraikan tentang latar belakang, identifikasi masalah, rumusan masalah, batasan penelitian, tujuan penelitian, manfaat penelitian, dan sistematika pennulisan.

BAB II LANDASAN TEORI

Pada bab ini mendeskripsikan tentang teori-teori yang menunjang penelitian/peulisan yang bisa di perkuat dengan menunjukkan hasil penelitian sebelumnya.

BAB III ANALISAN DAN PERANCANGAN SISTEM

Pada bab ini membahas analisa system, model system informasi serta menguraikan tentang prosedur system yang akan dibuat.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Pada bab ini menjelaskan tentang rancangan basis data serta rancangan dialog layar yang akan dilakukan pengujian terhadap aplikasi yang dirancang.

BAB V PENUTUP

Pada bab ini membahas tentang kesimpulan dan saran tentang solusi yang akan diterapkan untuk permasalahan yang dibahas pada penelitian ini

# BAB II LANDASAN TEORI

* 1. **Tinjauan Pustaka**

Dalam rangka mendapat hasil penelitian yang baik, selain melakukan penelitian langsung, penulis juga melakukan kajian kepustakaan dari hasil penelitian yang telah dilakukan, berikut hasil penelitian yang menjadi acuan :

Albert Yakobus Chandra, Didik Kurniawan, Rahmat Musa (2020) dalam penelitiannya yang berjudul "Perancangan Chatbot Menggunakan Dialogflow Natural Language Processing (Studi Kasus: Sistem Pemesanan pada Coffee Shop)" pada penelitian ini peneliti menemukan beberapa kasus yang sering dialami di lembaga tertentu seperti Perusahaan Mikro seringkali merupakan staf / karyawan dalam memberikan layanan informasi dan transaksi yang dilakukan secara manual kepada pelanggan terkait dengan kegiatan bisnis ini. Siklus ini selalu berulang dari satu pelanggan ke pelanggan lainnya. Dampaknya jika ada kondisi dimana antrian pelanggan yang cukup ramai dari pada beban kerja staf / karyawan akan lebih tinggi dan risiko kesalahan dalam transaksi juga akan tinggi. Perkembangan teknologi informasi dalam kecerdasan buatan pada era industri 4.0 bergerak maju. Salah satunya adalah Machine Learning - Natural Language Processing (NLP) yang merupakan salah satu ilmu yang berfokus pada bagaimana komputer dapat memahami bahasa manusia dan menanggapinya. Untuk itu dalam penelitian ini sistem chatbot akan dibangun dalam memberikan informasi dan melakukan transaksi dengan pelanggan. Chatbot ini akan dikembangkan menggunakan alat Dialogflow yang disediakan oleh Google. Chatbot yang dibangun ini diharapkan dapat menjadi alternatif yang dapat diimplementasikan di berbagai bisnis untuk memberikan layanan yang lebih baik bagi pelanggan.

Radhian, Dimar (2019) dalam penelitiannya yang berjudul “Pembangunan Aplikasi Chatbot Sebagai Media Pencarian Informasi Dalam Bidang Peternakan.” Pada penelitian ini Dinas Ketahanan Pangan Dan Peternakan Provinsi Jawa Barat merupakan unsur penyelenggara Pemerintah Provinsi Jawa Barat yang melaksanakan urusan pemerintahan bidang pangan dan bidang pertanian. Dinas Ketahanan Pangan Dan Peternakan Provinsi Jawa Barat dalam misinya berupaya meningkatkan pengetahuan peternak dalam peternakan seperti budidaya ternak. Kurangnya media informasi yang dimiliki, membuat peternak kesulitan mendapatkan informasi dan pengetahuan dalam budidaya ternak. Oleh karena itu dibangunlah aplikasi chatbot. Chatbot merupakan salah satu bentuk aplikasi NLP (Natural Language Processing). Sistem ini digunakan untuk memudahkan pengguna dalam melakukan pencarian informasi yang berisikan informasi pengetahuan dalam budidaya ternak untuk membantu peternak dalam menambah pengetahuan peternak tentang budidaya. Platform yang digunakan untuk mengembangkan chatbot adalah Dialogflow. Platform ini menyediakan layanan NLP (Natural Language Processing) yang membuat chatbot lebih cerdas sehingga dapat memahami maksud dari apa yang ditanyakan oleh pengguna. Hasil penelitian menunjukan bahwa 72% peternak setuju aplikasi chatbot ini dapat membantu peternak dalam menambah pengetahuan budidaya ternak dan 78% setuju memudahkan peternak dalam mencari informasi budidaya ternak.

Rahmiati, Rahmiati (2019) dalam penelitiannya yang berjudul “Perancangan Sistem Informasi Desa Tegal Sari Kecamatan Satui Menggunakan Framework Codeigniter.” penelitian ini dibuat suatu situs web untuk menjawab masalah dalam bentuk media informasi. Desa Tegal Sari adalah salah satu desa di Kabupaten Satui Kalimantan Selatan yang belum memiliki situs web sebagai media informasi. Setiap komunitas yang ingin mengakses informasi di Desa Tegal Sari masih terbatas, untuk mengakses informasi setiap warga negara akan mendapatkan SMS (Short Message Service) atau melalui aplikasi Whatsapp oleh masyarakat kabupaten dan prosesnya masih cenderung rumit. Maka dalam penelitian ini dibuat suatu situs web untuk menjawab masalah dalam bentuk media informasi. Aplikasi ini diharapkan mampu menjawab masalah yang ada di Desa Tegal Sari. Aplikasi yang dikembangkan dalam sistem ini menggunakan kerangka kerja CodeIgniter (CI). Peneliti memproses informasi dari Desa Tegal Sari menggunakan kerangka CodeIgniter (CI) yang mendukung model,

Permana, Eko (2017) dalam penelitiannya yang berjudul “Perancangan Sistem Informasi Akuntansi Pada Bina Plus Cisarua Dengan Menggunakan PHP Framework Codeigniter Dan MYSQL” Peneliti dirancang untuk membangun sistem informasi akuntansi pengelolaan aset tetap pada asrama bina plus cisarua diharapkan untuk membantu mengurangi masalah yang ada sehingga pengelolaan aset tetap bisa lebih efektif dan efisien. Asrama Bina Plus Cisarua beralamatkan di jalan di Jl. Terusan Kolonel Masturi No. 64 Desa Pasir Halang, RT 001 RW 07, Cisarua, Lembang, Kabupaten Bandung, Jawa Barat 40551 adalah sebuah lembaga dibidang pendidikan. Proses pengeloalaan aset tetap pada asrama masih manual dengan menggunakan microsoft excel dan microsoft word, sehingga mengakibatkan sulitnya mengetahui kondisi keseluruhan aset tetap yang mengakibatkan data yang dihasilkan kurang akurat. Jenis penelitian yang peneliti gunakan adalah jenis penelitian akademik, jenis desain penelitian adalah deskriptif analistis, metode penelitian adalah metode survei, teknik pengumpulan data yang peneliti gunakan adalah penelitian lapangan yang terdiri dari wawancara, observasi, dan penelitian kepustakaan, struktur pengembangan sistem yang digunakan adalah waterfall, perancangan sistem informasi yang digunakan adalah diagram sequencediagram , BPMN diagram dan use caseDiagram. Peneliti mengusulkan untuk merancang sistem informasi akuntansi pengelolaan aset tetap pada asrama bina plus cisarua diharapkan untuk membantu mengurangi masalah yang ada sehingga pengelolaan aset tetap bisa lebih efektif dan efisien.

Widyantika, Nuke Brilian (2018) dalam penelitiannya yang berjudul “Perancangan Chatbot Menggunakan Rivescript Pada Website E-Commerce Sebagai Virtual Customer Service.” Penelitian ini yang di karenakan teknologi sekarang berkembang dan kebutuhan pemilik toko online mlkpancing.com untuk menyelesaikan permintaan layanan pelanggan, oleh karena itu desain sistem dibuat untuk memberikan informasi cepat kepada pelanggan. Penelitian ini bertujuan untuk membangun chatbot yang dapat menggantikan peran layanan pelanggan manusia dalam hal memberikan informasi kepada pelanggan menggunakan file yang tersimpan yang berisi informasi stok-dalam-perdagangan di mlkpancing.com di dalam sistem. Desain dan implementasi perangkat lunak ini menghasilkan prototipe chatbot yang telah dibangun dengan memanfaatkan basis pengetahuan RiveScript. RiveScript ini membawa input yang dapat diintegrasikan dengan baik oleh chatbot sebagai input teks. Selain itu, komunikasi sistem pelanggan dikembangkan. Pemanfaatan chatbot buatan-intelijen-ditingkatkan ini dapat menyebabkan kemudahan pelanggan untuk mendapatkan informasi dari database dengan cepat. Sistem chatbot ini lulus kompatibilitas pengujian tanggapan. Hasilnya membuktikan bahwa dari 10 topik pertanyaan, ada 9 tanggapan yang kompatibel dan 1 tidak sesuai. Sementara itu, validitas respons dari 10 pertanyaan, ada 8 memberikan informasi valid dan 2 tidak valid. Chatbot dengan RiveScript dapat membantu pelanggan mendapatkan informasi yang mereka inginkan, dengan kesesuaian jawaban hingga 88%. ada 8 memberikan informasi yang valid dan 2 tidak valid. Chatbot dengan RiveScript dapat membantu pelanggan mendapatkan informasi yang mereka inginkan, dengan kesesuaian jawaban hingga 88%. ada 8 memberikan informasi yang valid dan 2 tidak valid. Chatbot dengan RiveScript dapat membantu pelanggan mendapatkan informasi yang mereka inginkan, dengan kesesuaian jawaban hingga 88%.

* 1. **Tinjauan Teori**

Teori umum pada bab ini menjelaskan tentang penjabaran teori-teori yang di dapat penulis berlandaskan studi pustaka maupun dari buku, yaitu sebagai berikut :

1. **Pengertian perancangan**

Perancangan adalah tahapan perancangan (*design*) memiliki tujuan untuk mendesain sistem baru yang dpat menyelesaikan masalah-masalah yang dihadapi dari pemilihan sistem terbaik. (bin Ladjamudin, 2005).

Perancangan adalah proses pengembangan spesifikasi sitem baru berdasarkan hasil rekomendasi analisis sistem. (Kusrini dkk, 2007).

Perancangan adalah penggambaran, perancangan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terspisah kedalam satu kesatuan yang utuh dan berfungsi, perencanaan sistem dapat dirancang dalam bentuk bagan alur sistem (*system flowchart*), yang merupakan alat bentuk grafik yang dapat digunakan untuk menunjukan urutan-urutan proses dari sistem. (Syifaun Nafisah, 2003).

1. **Pengertian Aplikasi**

Aplikasi adalah suatu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktifitas seperti perniagaan, pelayanan, masyarakat, periklanan atau semua proses yang dilakukan manusia. (Henry, 2004).

Aplikasi adalah program yang memiliki aktifitas pemrosesan perintah yang diperlukan untuk melaksanakan permintaan pengguna dengan tujuan tertentu. (Supriyanto, 2005).

1. **Pengertian Perancangan Aplikasi**

Konsep Merancang Multimedia merupakan aplikasi multimedia yang akan dibuat. Untuk dapat merancang konsep dalam membuat aplikasi multimedia dibutuhkan kreatifitas. Kreatifitas adalah kemampuan untuk menyajikan gagasan atau ide baru. Sedangkan inovasi merupakan aplikasi dari gagasan atau ide baru tersebut. Untuk menciptakan ide yang orisinil tidaklah mudah, maka dapat digunakan beberapa teknik untuk menciptakan ide, yaitu penyesuaian (adaptasi), multimedia yang telah ada dianggap belum sesuai dengan lingkungan yang dituju.

Merancang konsep analisis sistem bekerjasama dengan pemakai, mungkin juga bekerjasama dengan professional komunikasi seperti prosedur, sutradara, penulis naskah, editor elektronik terlibat dalam merancang konsep yang menentukan keseluruhan pesan dan membuat aliran (urutan) pada aplikasi multimedia yang akan dibuat. Untuk dapat merancang konsep dalam membuat aplikasi multimedia dibutuhkan kreatifitas. Kreatifitas adalah kemampuan untuk menyajikan gagasan atau ide baru. Sedangkan inovasi merupakan aplikasi dari gagasan atau ide baru tersebut. Untuk menciptakan ide yang orisinil tidaklah mudah, maka dapat digunakan beberapa teknik untuk menciptakan ide, yaitu penyesuaian (adaptasi).

1. **Pengertian Sistem**

Sistem berasal dari bahasa Latin (*Systema*) dan bahasa Yunani (*Sustena*) adalah suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan. Pada prinsipnya, setiap sistem selalu terdiri atas empat elemen :

1. Objek yang dapat berupa bagian, elemen, ataupun *variable.* Ia dapat benda fisik, abstrak, ataupun keduanya sekaligus tergantung kepada sifat sistem tersebut.
2. Atribut yang menentukan kualitas atau sifat kepemilikan sistem dan objeknya.
3. Hubungan internal di antara objek-objek di dalamnya.
4. Lingkungan tempat di mana sistem berada.
   * + 1. **Karakteristik Sstem**

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu mempunyai komponen-komponen, batas sistem, lingkungan luar sistem, penghubung, masukan, keluaran, pengolah, dan sasaran atau tujuan. (Ladjamudin, 2013)

Adapun penjelasan dari masing-masing karakteristik sistem menurut tersebut adalah sebagai berikut :

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerjasama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

1. Batasan Sistem

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan dan menunjukkan ruang lingkup dari sistem tersebut.

1. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan juga merugikan.

1. Penghubung Sistem

Penghubung sistem merupakan media yang menghubungkan antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini kemungkinan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya.

1. Masukan Sistem

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan dan masukan sinyal maintenanceinput adalah energi yang dimasukkan supaya sistem tersebut dapat berjalan. Sinyal input adalah energi yang diproses untuk mendapatkan keluaran dari sistem.

1. Keluaran Sistem

Keluaran sistem adalah energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran dapat merupakan masukan untuk subsistem yang lain.

1. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan menjadi keluaran.

1. Sasaran Sistem

Suatu sistem mempunyai tujuan atau sasaran, kalau sistem tidak mempunyai sasaran maka sistem tidak akan ada. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya. Sasaran sangat berpengaruh pada masukan dan keluaran yang dihasilkan.

1. **Pengertian Informasi**

Informasi merupakan hasil pengolahan data dari satu atau berbagai sumber yang kemudian diolah, sehingga menghasilkan nilai, arti, dan manfaat. Jadi dapat disimpulkan bahwa informasi adalah data yang telah diolah menjadi sebuah bentuk yang lebih berguna dan bermanfaat bagi yang menerimanya.(Pratama E. A., 2014)

Definisi sistem menurut Mulyadi (2016:5), Sistem adalah “suatu jaringan prosedur yang dibuat menurut pola yang terpadu untuk melaksanakan kegiatan pokok perusahaan”. Berdasar kan pengertian diatas dapat disimpulkan bahwa sistem adalah kumpulan dari komponen-komponen yang saling berkaitan satu dengan yang lain untuk mencapai tujuan dalam melaksanakan suatu kegiatan pokok perusahaan.

1. **Komponen Sistem Informasi**

Komponen sistem informasi terdiri dari :

* 1. Blok Masukan (input)

Blok masukan (input) ini mewakili data yang masuk kedalam sistem informasi. Termasuk dalam blok masukan ini adalah metode-metode dan media untuk menangkap data yang akan dimasukkan.

* 1. Blok Model

Blok model ini terdiri dari kombinasi antara prosedur, logika dan model matematika yang akan memanipulasi data inputan agar menghasilkan keluaran (output) seperti yang diinginkan.

* 1. Blok Keluaran

Blok keluaran (output) ini berupa informasi yang berkualitas serta dokumentasi yang bermanfaat untuk semua pemakai sistem.

* 1. Blok Teknologi

Blok teknologi adalah kotak alat (tool-box) dalam sistem informasi yang digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan output serta membantu pengendalian sistem secara keseluruhan. Teknologi ini terdiri dari 3 bagian utama, yaitu teknisi (humanware atau brainware), perangkat lunak (software) dan perangkat keras (hardware).

* 1. Blok Basis Data

Basis data (database) merupakan kumpulan data yang saling terhubung, tersimpan dalam perangkat keras komputer. Yang mana diperlukan perangkat lunak untuk memanipulasinya agar menjadi informasi yang berkualitas.

* 1. Blok Kendali

Blok kendali adalah suatu perancangan yang diterapkan untuk mencegah terjadinya kesalahan-kesalahan, kegagalan sistem atau yang lainnya agar dapat segera teratasi.

1. **Pengertian Web**

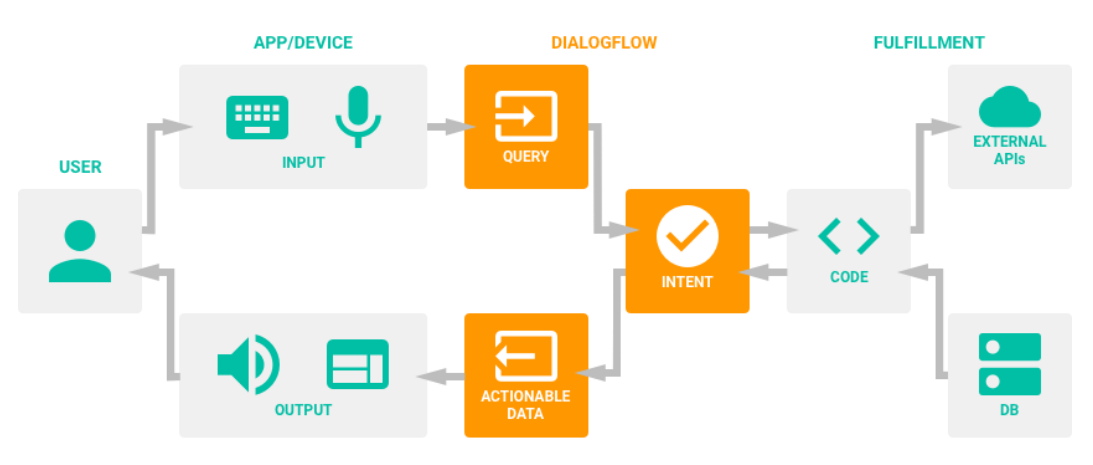
Web adalah (Arief 2011a:7) salah satu aplikasi yang berisikan dokumen-dokumen multimedia (teks, gambar, animasi, video) didalamnya yang menggunakan protocol HTTP (*Hypertext Transfer Protoco*) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut broser, semua dokumen web ditampilkan oleh browser dengan cara diterjemahkan. Beberapa jenis browser yang popular saat ini diantaranya adalah Internet Explorer yang diproduksi oleh Microsoft, Mozilla Firefox, Opera, dan Safari

1. **Google Dialogflow**

Google Dialogflow adalah sebuah platform yang berbasis pengolahan bahasa alami untuk para pengembang perangkat lunak membuat sebuah sistem chatbot. Google Dialogflow sebelum diakuisi oleh perusahaan Google pada September, 2016 bernama Api.ai. Api.ai awalnya dimiliki oleh perusahaan Speaktoit, sebuah perusahaan yang dikenal dengan aplikasi virtual buddy berbasis sistem operasi Android, iOS, dan windows phone yaitu Assistant. Perusahaan Google membeli Api.ai milik Speaktoit, karena platform tersebut menyediakan alat - alat untuk mengembangkan aplikasi - aplikasi pada Google Virtual Assistant. Pada 10 Oktober 2017 Api.ai resmi berganti nama menjadi Dialogflow. Berikut adalah fitur – fitur yang ditawarkan pada Google Dialogflow :

1. Dialogflow Agents

Agents mempunyai definisi yang sama seperti Natural Language Understanding (NLU). Agents dapat digunakan oleh sistem yang pengembang punya untuk dapat mentransformasi permintantaan bahasa alami dari pengguna sistem menjadi sebuah actionable data. Transformasi terjadi ketika masukan dari pengguna cocok dengan salah satu intent yang terdapat pada agent itu sendiri. Intent adalah komponen di agent yang belum ditetapkan atau sudah ditetapkan oleh pengembang yang memproses permintaan – permintaan pengguna. Agent dapat didesain untuk mengatur sebuah alur percakapan dengan bantuan context, intent priorities, slot filling, responsibilities, dan fulfillment dengan perantara webhook. Fitur untuk untuk mengatur pembelajaran mesin yang digunakan untuk sistem pengembang. Dalam menu ini terdapat fitur match mode, yaitu sebuah pengaturan yang menentukan algoritma yang harusnya digunakan untuk pembelajaran mesinnya. Algoritma yang telah ditentukan akan digunakan pada semua intent. Terdapat 2 algoritma yang disediakan oleh Dialogflow agent untuk pembelajaran mesinnya, yaitu : Rule-based Grammar Matching (Hybrid) dan Machine Learning Matching. Pada Gambar di bawah ini adalah alur kerja Dialogflow Agent yang merupakan alur kerja agent pada Google Dialogflow



Gambar 2. 1 Alur Kerja Dialogflow Agent

Sumber : https://dialogflow.com/docs/images/overview/agents/overview.png

1. Dialogflow Intent

Intent adalah sebuah representasi sebuah dialog antara apa yang pengguna katakan atau masukan dan apa tindakan atau balasan yang diberikan oleh sistem. Intent memiliki beberapa 9 bagian di dalamnya, yaitu : Context, Events, Training Phrase, Action and Parameters dan Responses.

1. Dialogflow Entities

Entity pada Dialogflow adalah sebuah alat yang berfungsi untuk mengekstrak parameter values dari masukan pengguna. Entity yang digunakan pada agent tertentu akan bergantung pada parameter values, karena entity akan memberikan output sesuai dengan fungsi parameter yang ada pada intent. Terdapat 3 tipe entity pada Dialogflow, yaitu : System Entities, Developer Entities dan User Entities.

1. Dialogflow Tools

Dialogflow tools merupakan sekumpulan alat – alat untuk membantu pengembangan sebuah chatbot pada Dialogflow. Dialogflow tools mempunyai 4 alat - alat, yaitu : Fulfillment, Intergrations, Training, History, Analytics dan Prebuilt Agents.

* 1. **Aplikasi Pendukung**

Berikut ini adalah beberapa aplikasi pendukung yang penulis gunakan dalam membangun sebuah Aplikasi dalam tugas akhir ini adalah sebagai berikut :

1. **XAMPP**

XAMPP adalah (Choliviana, Triyono, & Sukadi, 2012) paket PHP dan MySQL berbasis *open source*, yang dapat digunakan sebagai *tool* pembantu pengembangan aplikasi berbasis PHP.

Manfaat Xampp sebagai perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsi XAMPP sendiri adalah sebagai server yang berdiri sendiri (*localhost*). XAMPP merupakan perangkat yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dengan menginstall XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi web server Apache, PHP dan MySQL secara manual. XAMPP akan menginstalasi dan mengkonfigurasikannya secara otomatis.

1. **PHP**

PHP adalah (Arief 2011c:43) Bahasa *server-side –scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Karena PHP merupakan *server-side-scripting* maka sintaks dan perintah-perintah PHP akan diesksekusi diserver kemudian hasilnya akan dikirimkan ke *browser* dengan format HTML

PHP singkatan dari PHP *Hypertext Preprocessor* yang merupakan sebuah bahasa *scripting* yang terpasang pada HTML. Sebagian besar sintaks mirip dengan bahasa C, Java dan Perl, ditambah beberapa fungsi PHP yang spesifik. Tujuan utama penggunaan bahasa ini adalah untuk memungkinkan perancang web menulis halaman web dinamik dengan cepat. Hubungan PHP dengan HTML, halaman web biasanya disusun dari kode-kode HTML yang disimpan dalam sebuah file berekstensi .html. File HTML ini dikirimkan oleh server ke *browser*, kemudian *browser* menerjermahkan kode-kode tersebut sehingga menghasilkan suatu tampilan yang indah, Lain halnya dengan program PHP, Program ini harus diterjemahkan oleh web-*server* sehingga menghasilkan kode html yang dikirim ke *browser* agar dapat ditampilkan.

1. **MySQL**

MySQL adalah (Arief 2011:152) salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan *database* sebagai sumber dan pengolahan datanya”.

MySQL dikembangkan oleh perusahaan swedia bernama MySQL AB yang pada saat ini bernama Tcx DataKonsult AB sekitar tahun 1994-1995, namun cikal bakal kodenya sudah ada sejak tahun 1979. Awalnya Tcx merupakan perusahaan pengembang *software* dan konsultan *database*, dan saat ini MySQL sudah diambil alih oleh Oracle Corp.

Kepopuleran MySQL antara lain karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya sehingga mudah untuk digunakan, kinerja *query* cepat, dan mencukupi untuk kebutuhan *database* perusahaan-perusahaan yang berskala kecil sampai menengah, MySQL juga bersifat *open source* (tidak berbayar) . 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.

1. **Codeigniter**

CodeIgniter adalah Kerangka Pengembangan Aplikasi - toolkit - untuk orang yang membangun situs web menggunakan PHP. Tujuannya adalah untuk memungkinkan Anda mengembangkan proyek lebih cepat daripada yang Anda bisa jika Anda menulis kode dari awal, dengan menyediakan serangkaian perpustakaan yang kaya untuk tugas-tugas yang biasanya dibutuhkan, serta antarmuka yang sederhana dan struktur logis untuk mengakses perpustakaan ini. CodeIgniter memungkinkan Anda secara kreatif fokus pada proyek Anda dengan meminimalkan jumlah kode yang diperlukan untuk tugas yang diberikan.

Menurut Hakim (2010:3) CodeIgniter adalah sebuah framework PHP yang dapat membantu mempercepat developer dalam pengembangan aplikasi web berbasis PHP dibandingkan jika menulis semua kode program dari awal.

Sedangkan, menurut Raharjo (2015:3) CodeIgniter adalah framework web untuk bahasa pemrograman PHP, yang dibuat oleh Rick Ellis pada tahun 2006, penemu dan pendiri EllisLab.

Jadi CodeIgniter adalah sebuah framework buatan Rick Ellis yang digunakan untuk mempermudah pada developer dalam mengembangkan suatu aplikasi web.

1. **Bootstrap**

Bootstrap sebuah *library framework* CSS yang dibuat khusus untuk bagian pengembang *front-end* website. Menurut Alatas (2013:2) “bootstrap merupakan *framework* ataupun *tools* untuk membuat aplikasi web atupun situs web *responsive* secara tepat, mudah dan gratis”. Lain pula menurut Riyanto (2014:18)

Dari penjelasan di atas dapat disimpulkan bahwa bootstrap ialah tampilan untuk membuat halaman web menjadi lebih bagus dan mempercepat seseorang dalam mendesain program.

1. **Sublime Text**

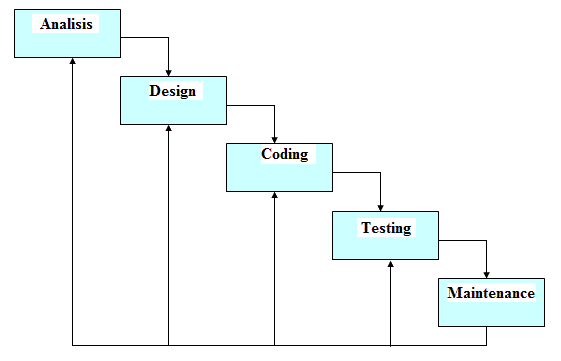
Sublime textsalah satu kode editor yang biasa digunakan oleh para programmer untuk membuat suatu program.

Menurut Supono dan Putratama (2016:14) “*Sublime text* merupakan perangkat lunak *text editor* yang digunakan untuk membuat atau meng-edit suatu aplikasi. *Sublime text* mempunyai fitur plugin tambahan yang memudahkan programmer”.

Selain itu, menurut Faridi (2015:3) menjelaskan bahwa “Sublime Text 3 adalah editor berbasis python, sebuah teks editor yang elegan, kaya akan fitur, *cross platform*, mudah dan simple yang cukup terkenal di kalangan developer (pengembang), penulis dan desainer”.

* 1. **Pengertian Metode Waterfall**

Model *Waterfall* adalah suatu metodologi pengembangan perangkat lunak yang mengusulkan pendekatan kepada perangkat lunak sistematik dan sekuensial yang mulai pada tingkat kemajuan sistem pada seluruh analisis, design, kode, pengujian dan pemeliharaan. Model pengembangan sistem yang digunakan adalah *Waterfall Strategy*. Strategi ini mengisyaratkan ’penyelesaian’ tiap proses satu per satu (Whitten et al, 2004), Model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap analisis, desain, coding, *testing* atau*verification*, dan *maintenance*. Sebagai contoh tahap desain harus menunggu selesainya tahap sebelumnya yaitu tahap requirement. Berikut gambar model waterfall.

****

Gambar 2. 2 Metode Waterfall

* 1. **Teori Perancangan Basis Data**

Menurut (Kristanto, 2012), Basis data adalah Suatu sistem penyusunan dan pengelolaan *record-record* dengan menggunakan komputer dengan tujuan untuk menyimpan serta memelihara data operasional lengkap suatu perusahaan atau organisasi, sehingga mampu menyediakan informasi yang optimal yang diperlukan maka untuk proses pengambilan keputusan.

Untuk mengelola basis data diperlukan perangkat lunak yang disebut *Database Management System* (DBMS). DBMS adalah perangkat lunak sistem yang memungkinkan para pemakai membuat, memelihara, mengontrol dan mengakses basis data dengan cara yang praktis dan efisien. DBMS juga dapat digunakan untuk mengakomondasikan berbagai macam pemakai yang memiliki kebutuhan akses yang berbeda-beda.

### ERD ( *Entity Relationship Diagram* )

ERD atau *Entity Relationship Diagram* merupakan suatu model jaringan yang menggunakan data yang disimpan pada sistem secara abstrak. ERD juga menggambarkan hubungan antara entitas yang memiliki jumlah atribut dengan entitas yang lain dalam suatu sistem yang terintrigasi. ERD digunakan oleh perancang sistem untuk memodelkan data yang nantinya akan dikembangkan menjadi basis data (database). Model data ini juga akan membantu pada saat melakukan analisis dan perancangan basis data, karena model data ini akan menunjukan brmacam-macam data yang dibutuhkan dan hubungan antara data. EDR juga merupakan model konseptual yang dapat mendeskripsikan hubungan antara file yang akan digunkan untuk memodelkan struktur data serta hubungan antara data (Membara, 2014).

|  |  |
| --- | --- |
| **Notasi** | **Keterangan** |
| Entitas | **Entitas** adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai. |
| Relasi | **Relasi** menunjukan adanya hubungan di antara sejumlah entitas yang berbeda. |
| Atribut | **Atribut** yaituberfungsi sebagai mendeskripsikan karakter entitas (atribut yang berfungsi sebagai key diberi garis bawah). |
| Garis | **Garis** penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut. |
| 1 N  Pegawai  Kantor  dikerjakan | **One to Many** adalah perbandingan antara entity pertama dengan entity kedua berbanding satu berbanding banyak. |
| M M  Pegawai  Dinas  mengikuti | **Many to Many** yaitu perbandingan antara entity pertama dengan entity kedua berbanding banyak berbanding banyak. |

Tabel 2. 1 ERD ( Entity Relationship Diagram )

1. **LRS ( *Logical Record Structure* )**

LRS (*Logical Record Structure*) adalah representasi dari struktur *record* pada table-table yang terbentuk dari hasil antar himpunan entitas. LRS dibentuk dengan nomor dari tipe *record*. LRS terdiri dari link-link diantara tipe *record*. *Link* ini menunjukan arah dari satu tipe *record* lainnya. (Riyanto, 2005).

* 1. **Unified Modelling Language (UML)**

Menurut (Pandawa, 2016) UML*(Unified Modelling Language)* adalah bahasa garis untuk mendokumentasikan menspesifikasikan, dan membangun system perangkat lunak UML berorienasi objek menerapkan banyak level absraksi, tidak tergantung pada Bahasa dan teknologi, pemanduan beberapa notasi diberagam metodologi, usaha bersama dari banyak pihak. Standar UML dikelola oleh OMG *(Object Management Group*).

Sedangkan menurut (Akil, 2018, p. 3), *Unified Model Language* (UML) adalah bahasa pemodelan visual yang digunakan untuk mendefinisikan, memvisualisasikan, membangun dan mendokumentasikan rancangan dari suatu sistem perangkat lunak.

Pada UML terdiri dari beberapa macam diagram berikut penjelasan masing – masing diagram yang penulis gunakan dalam mendesain sistem :

1. **Activity Diagram**

*Activity Diagram* ialah sebuah diagram yang digunakan untuk menggambarkan jalur aktivitas pada sistem yang sedang dirancang. *Activity diagram* juga digunakan untuk menggambarkan *decision* yang mungkin terjadi pada saat sistem dirancang serta digunakan untuk menggambarkan sebuah proses parallel yang mungkin terjadi pada saat eksekusi.

Menurut (Anatasia, 2018, p. 49), *Activity Diagram* yaitu aktivitas yang menggambarkan urutan kegiatan atau urutan aktivitas dari sebuah sistem. Tujuan dibuatnya *activity diagram* adalah untuk memudahkan dalam memahami proses bisnis sistem. Adapun simbol – simbol *activity diagram* dapat dilihat pada tabel berikut :

|  |  |  |
| --- | --- | --- |
| No. | Simbol | Keterangan |
| 1 |  | ***Start*** merupakan status awal sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
| 2 |  | ***Activity*** merupakan aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| 3 |  | ***Decision*** merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari 1. |
| 4 |  | ***Swimlane*** digunakan untuk memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |
| 5 |  | ***Finish*** merupakan status akhir sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
| 6 |  | ***Asosiasi*** penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| 7 |  | ***Join*** digunakan untuk menunjukkan kegiatan yang digabungkan. |
| 8 |  | ***Fork*** digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel. |
| 9 |  | ***Miracle Activities*** digunakan untuk masukan dan keluaran, dipakai pada waktu *start point.* |
| 10 |  | ***Black Hole Activities*** digunakan untuk masukan dan keluaran. |

Tabel 2. 2 Simbol Activity Diagram

1. **Use Case Diagram**

Menurut (Anatasia, 2018, p. 45), *Use Case Diagram* adalah diagram yag menggambarkan interaksi antara pengguna *(actor)* dengan sistem informasi yang akan dibuat. *Use case* menggambarkan siapa saja actor yang terlibat dan fungsi apa saja yang dapat digunakan *actor* pada sistem informasi tersebut. *Use Case Diagram* terdiri dari *Use Case, Actor, Relationship, System Boundary* atau batas sistem (opsional). Adapun simbol – simbol *Use Case Diagram* dapat dilihat pada tabel berikut :

|  |  |  |  |
| --- | --- | --- | --- |
| No | Gambar | Nama | Keterangan |
| 1 |  | *Actor* | Menspesifikasi antara himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case. |
| 2 |  | *Depedency* | Hubungan perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri. |
| 3 |  | *Generalization* | Hubungan dimana objek anak*(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk *(ancestor).* |
| 4 |  | *Include* | Menspesifikasikan atau bahwa *use case* sumber secara *eksplisit.* |
| 5 |  | *Extend* | Menspesifikasikan atau bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan. |
| 6 |  | *Association* | Apa sajakah yang menghubungkan antara objek satu dengan objek lainnya. |
| 7 |  | *System* | Menspesifikasi paket yang menampilkan sistem secara terbatas. |
| 8 |  | *Use case* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor. |
| 9 |  | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerjasama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemen. |
| 10 |  | *Note* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi. |

Tabel 2. 3 Simbol Use Case Diagram

1. **Class Diagram**

Menurut (Anatasia, 2018, p. 56), *Class diagram*menggambarkan struktur dan deskripsi *class, package,* dan objek serta hubungan satu sama lain seperti *inheritance, association,* dan lain – lain. Dibawah ini adalah gambaran dari simbol – simbol *class* diagram :

|  |  |  |  |
| --- | --- | --- | --- |
| No | Gambar | Nama | Keterangan |
| 1 |  | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagai perilaku dan struktur data dari objek yang ada diatasnya objek induk (*ancestor*). |
| 2 |  | *Nary Association* | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek. |
| 3 |  | *Class* | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. |
| 4 |  | *Collaboration* | Deskripsi dari urutan aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor. |
| 5 |  | *Realization* | Operasi yang benar-benar dilakukan oleh suatu objek. |
| 6 |  | *Dependency* | Hubungan yang dimana ada perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri. |
| 7 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek yang lainya. |

Tabel 2. 4 Simbol Class Diagram

1. **Sequence Diagram**

Menurut (Anatasia, 2018, p. 51), *Sequence Diagram* menggambarkan interaksi antar objek dalam sistem. *Sequence Diagram* digunakan untuk menggambarkan scenario pada use case. Jumlah *Sequence Diagram* harus sama dengan jumlah use case. Adapun simbol – simbol *Sequence Diagram* dapat dilihat pada gambar dibawah ini :

|  |  |  |  |
| --- | --- | --- | --- |
| No | Gambar | Nama | Keterangan |
| 1 |  | *Actor* | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case.* |
| 2 |  | *Life Line* | Objek *entity,* antarmuka yang saling berinteraksi. |
| 3 |  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi. |
| 4 |  | *Message to Self* | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi. |
| 5 |  | *Boundary* | Untuk menunjukkan adanya *interfaces requirement,* bukan untuk menunjukkan bagaimana *interface* yang akan di implementasikan. |
| 6 |  | *Controller* | Sebuah control adalah meniru suatu kelas yang menggambarkan suatu pengendalian manajer atau kesatuan. Suatu control mengorganisir dan menjadwalkan aktivitas. |
| 7 |  | *Entity* | Suatu entity adalah tempat atau ketekunan mekanisme yang menangkap pengetahuan dan informasi di dalam suatu sistem. |

Tabel 2. 5 Simbol Sequence Diagram

* 1. **Notasi UML**

Notasi UML dturunkan dari tiga notasi yang sudah ada sebelumnya yaitu Grady Booch OOD (Object Oriented Design), Jim Rumbough OMT (Object Modelling Technique), dan Ivar Jacobson OOSE (Object Oriented Software Engineering). Notasi UML yang digunakan sekarang merupakan penggabungan dari tiga notasi tersebut, yang terdiri dari: (Nurhasanah, 2014)

* + 1. Aktor ( *Actor* )

Dalam pemodelan sistem dengan UML, actor adalah seseorang atau sesuatu yang berinteraksi dengan sistem yang sedang kita kembangkan. Actor berada diluar lingkup sistem atau perangkat lunak yang sedang kita kemangkan, bersifat eksternal.

Secara prinsip dapat kita kenali 3 jenis actor untuk hampir semua sistem atau perangkat lunak yang kita kembangkan: para pengguna sistem, perangkat lunak yang berinteraksi dengan sistem, dan perangkat lunak yang kita kembangkan serta waktu. Jadi aktor ini bisa berupa orang, perangkat keras atau mungkin juga objek lain dalam sistem yang sama. Biasanya yang dilakukan oleh aktor adalah memberikan informasi pada sistem dan atau memerintahkan sistem untuk melakukan sesuatu.

Kelas seperti juga objek, adalah sesuatu yang membungkus (encapsulate) informasi dan perilaku dalam dirinya. Dalam pengembangan system tradisional, kita mengadakan pandekatan dengan cara memisahkan informasi-informasi pada sisi basis data dan perilaku yang mengaksesnya di sisi aplikasi pemasup atau pengakses. Pendekatan berorientasi objek berbeda, yaitu menggabungkan potongan-potongan informasi dengan perilaku yang akan mengaksesnya dalam apa yang dinamakan kelas.

* + 1. Usecase

Usecase adalah peringkat tertinggi dari fungsional yang dimiliki system. Dengan kata lain, usecase menggambarkan bagaimana seseorang akan menggunakan sistem. Usecase menjelaskan suatu urutan kegiatan yang dilakukan oleh aktor dan sistem untuk mencapai tujuan tertentu walaupun menjelaskan kegiatan namun usecase hanya menjelaskan apa yang dilakukan oleh aktor dan sistem, bukan bagaimana aktor dan sistem melakukan kegiatan tersebut.

Keunggulan dari cara memandang sistem sebagai kumpulan usecase adalah kemampuannya untuk memisahkan implementasi sistem dari alasan mengapa sistem harus ada. Ia akan membantu kita untuk berfokus pada apa yang paling penting, yaitu menentukan apa yang dibutuhkan serta apa harapan pengguna terhadap sistem atau peranggkat lunak yang sedang dikembangkan.

* + 1. Interaction

Interaction digunakan untuk menunjukan baik aliran pesan atau informasi antara objek. Biasanya interection ini dilengkapi juga dengan teks bernama operationsignature yang tersusun dari nama operasi, perameter yang dikirim dan tipe parameter yang dikembalikan.

* + 1. Interface

Interface merupakan kumpulan opersi tanpa implementasi dari suatu class. Implementasi operasi dalam interface dijabarkan dalam operasi dalam class. Oleh karena itu kebaradaan interface selalu disertai oleh class yang mengimplementasikan operasinya. Interface ini merupakan salah satu cara mewujudkan prinsip enkapsulasi dalam objek.

1. Package

Package adalah container atau wadah konseptual yang digunakan untuk mengelompokkan elemen-elemen dari suatu sistem yang sedang dibangun, sehingga bisa dibuat model yang lebih sederhana. Tujuannya adalah untuk mempermudah penglihatan (visibility) dari suatu model yang sedang dibangun.

1. Note

Note dibangun untuk memberikan keterangan dan komentar tambahan dari suatu elamen sehingga bisa langsung terlampir dalam model. Note ini bisa ditempelkan ke semua elemen notasi yang lain.

1. Dependency

Merupakan relasi yang menunjukan bahwa perubahan pada suatu elemen menberikan penngaruh pada elemen yang lain. Elemen yang ada di bagian tanda panah adalah elemen yang tergantung pada elemen yang ada di bagian tanpa ada tanda panah. Terdapat dua stereotype dari dependency, yaitu include dan extend. Include menunjukan bahwa suatu bagian dari elemen (yang ada di garis tanpa panah) memicu eksekusi bagian dari elemen yang lain(yang ada di garis dengan panah), misalnya untuk notasi A B operasi yang ada di class A memicu dieksekuensinya operasi yang berada di class B.

Extend menunjukan bahwa suatu bagian dari elemen di garis tanpa panah bisa disiapkan kedalam elemen yang ada di garis dengan panah, misalnya untuk notasi A B suatu fungsi dari usecase A bisa disisipkan ke dalam usecase B atau dengan kata lain A optimal untuk B. Kedua stereotype ini di representasikan dengan menambahkan textinclude atau extend di notasi dependency.

1. Association

Association menggambarkan navigasi antara class (navigation), berapa banyak objek lain yang bisa berhubungan dengan satu objek (multiplicity antar class) dan apakah suatu class menjadi bagian dari class lainnya (aggregation).

Navigation di lambangkan dengan penambahan tanda panah di akhir garis. Bidirectional navigation menunjukan bahwa dengan mengetahui salah satu class bisa didapatkan dari informasi lainnya. Sementara dengan unidirectional navigation hanya dengan mengetahui class di ujung garis association tanpa panah kita bisa medapatkan informasi dari class di ujung dengan panah, tetapi tidak sebaliknya.

Pada penelitian kali ini penulis menggunakan visual paradigm untuk merancang aplikasi. Visual paradigm adalah perangkat lunak yang digunakan untuk perancangan aplikasi dengan tools UML.

* 1. **Pengujian Perangkat Lunak**

Pengujian adalah aktivitas yang digunakan untuk dapat melakukan evaluasi suatu atribut atau kemampuan dari program atau sistem dan menentukan apakah telah memenuhi kebutuhan atau hasil yang diharapkan. (Nidhra and Dondetti, 2012)

Menurut Munawar (2018, p. 38), Pengetesan software adalah kegiatan yang sangat penting sebelum diimplementasikan. Beberapa organisasi besar, biasanya memiliki bagian khusus untuk pengetesan software sebelum diimplementasikan. Namun tidak semua organisasi memiliki bagian khusus untuk pengetesan software. Oleh karena itu, beberapa organisasi melibatkan analisis dalam pengetesan, karena analisis lebih memahami kebutuhan organisasi sehingga bisa mengatur kinerja software dari sisi kebutuhan fungsional dan non-fungsional.

Beberapa metode yang biasa digunakan untuk melakukan pengujian antara lain *Black Box* dan *White Box Testing.*

* 1. **Black Box Testing**

Black Box Testing merupakan sebuah metode yang digunakan untuk menemukan kesalahan dan mendemonstrasikan fungsional aplikasi saat dioperasikan, apakah input diterima dengan benar dan output yang dihasilkan telah sesuai dengan yang diharapkan. (Nidhra and Dondetti, 2012).

Pengujian perangkat lunak mempunyai beberapa level, untuk pengujian menggunakan metode Black Box, terdapat enam level yaitu Integration, Functional, System, Acceptance, Beta, dan Regression .

Salah satu dari pengujian Black Box yang dapat dilakukan oleh seorang penguji independen adalah Functional Testing. Basis uji dari Functional Testing ini adalah pada spesifikasi dari komponen perangkat lunak yang diuji.

Fokus dari pengujian menggunakan metode Black Box adalah pada pengujian fungsionalitas dan output dihasilkan aplikasi. Pengujian Black Box didesain untuk mengungkap kesalahan pada persyaratan fungsional dengan mengabaikan mekanisme internal atau komponen dari suatu program. Functiona Testing memastikan bahwa semua kebutuhan – kebutuhan telah dipenuhi dalam sistem aplikasi. Dengan demikian fungsinya adalah tugas – tugas yang didesain untuk dilaksanakan sistem. Functional Testing berkonsentrasi pada hasil dari proses bukan bagaimana proses terjadi

Black Box Testing cenderung untuk menemukan hal-hal berikut:

* 1. Fungsi yang tidak benar atau tidak ada.
  2. Keslahan anarmuka (interface errors).
  3. Kesalahan pada struktur data dan akses basis data.
  4. Kesalahan performansi (performance errors).
  5. Kesalahan inisialisasi dan terminasi.

Pengujian didesain untuk menjawab pertanyaan-pertanyaan berikut:

* 1. Bagaimana fungsi-fungsi diuji agar dapat dinyatakan valid?
  2. Input seperti apa yang dapat menjadi bahan kasus uji yang baik?
  3. Apakah sistem sensitif pada input-input tertentu?
  4. Bagaimana sekumpulan data dapat diisolasi?
  5. Berapa banyak rata-rata dan jumlah data yang dapa ditangani sistem?
  6. Efek apa yang dapat membuat kombinasi data ditangani spesifik pada operasi sistem?

Saat ini terdapat banyak metode atau teknik untuk melaksanakan Black Box Teting, antara lain:

* 1. Equivalence Partitioning
  2. Boundary Value Analysis/Limit Testing
  3. Comparison Testing
  4. Sample Testing
  5. Robustness Testing
  6. Behavior Testing
  7. Requirement Testing
  8. Performance Testing
  9. Uji Ketahanan (Endurance Testing)
  10. Uji Sebab-Akibat (Cause-Effect Relationship Testing)

1. **White Box Testing**

White box testing adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur control dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Secara sekilas dapat diambil kesimpulan white box testing merupakan petunjuk untuk mendapatkan program yang benar secara 100%.

Pengujian white bertujuan untuk:

1. Untuk mengetahui cara kerja suatu perangkat lunak secara internal.
2. Untuk menjamin operasi-operasi internal sesuai dengan spesifikasi yang telah ditetapkan dengan menggunakan struktur kendali dari prosedur yang dirancang.

Pelaksanaan pengujian white box:

1. Menjamin seluruh independent path dieksekusi paling sedikit satu kali. Independent path adalah jalur dalam program yang menunjukkan paling sedikit satu kumpulan proses ataupun kondisi baru.
2. Menjalani logical decision pada sisi dan false.
3. Mengeksekusi pengulangan (looping) dalam batas-batas yang ditentukan.
4. Menguji struktur data internal.

Kelebihan White Box Testing antara lain:

1. Kesalahan Logika

Menggunakan sintax ‘if dan sintax pengulangan. Langkah selanjutnya metode white box testing ini akan mencari dan mendeteksi segala kondisi yang di percaya tidak sesuai dan mencari kapan suatu proses perulangan diakhiri.

1. Ketidaksesuaian Asumsi

Menampilkan dan memonitor beberapa asumsi yang diyakini tidak sesuai dengan yang diharapkan atau yang akan diwujudkan, untuk selanjutnya akan dianalisa kembali dan kemudian diperbaiki.

1. Kesalahan Pengetikan

Mendeteksi dan mencari bahasa-bahasa pemrograman yang dianggap bersifat case sensitif.

Kelemahan White Box Testing adalah pada perangkat lunak yang jenisnya besar, metode white box testing ini dianggap boros karena melibatkan banyak sumberdaya untuk melakukannya.

**BAB III  
ANALISA DAN PERANCANGAN**

* 1. **Analisa Sistem**

Tahap analisis dilakukan setelah tahap perencanaan system dan sebelum tahap desain system. Tahap ini merupakan tahap yang kritis karena kesalahan dalam tahap ini menyebabkan kesalahan pada tahap selanjutnya.

Analisa system informasi akan membantu dalam mengetahui informasi – informasi tentang system yang sedang berjalan. Sehingga dengan analisa system, diharapkan dapat diketahui sejauh mana kebutuhan yang telah ditangani oleh system yang berjalan dan bagaimana agar kebutuhan – kebutuhan yang belum dapat terpenuhi dapat diberikan solusinya dan dditerapkan dalam tahapan perancangan system.

* + 1. **Analisis Masalah**

Berdasarkan masalah yang diuraikan di bab pendahuluan, maka analisis masalah dapat dijabarkan sebagai berikut :

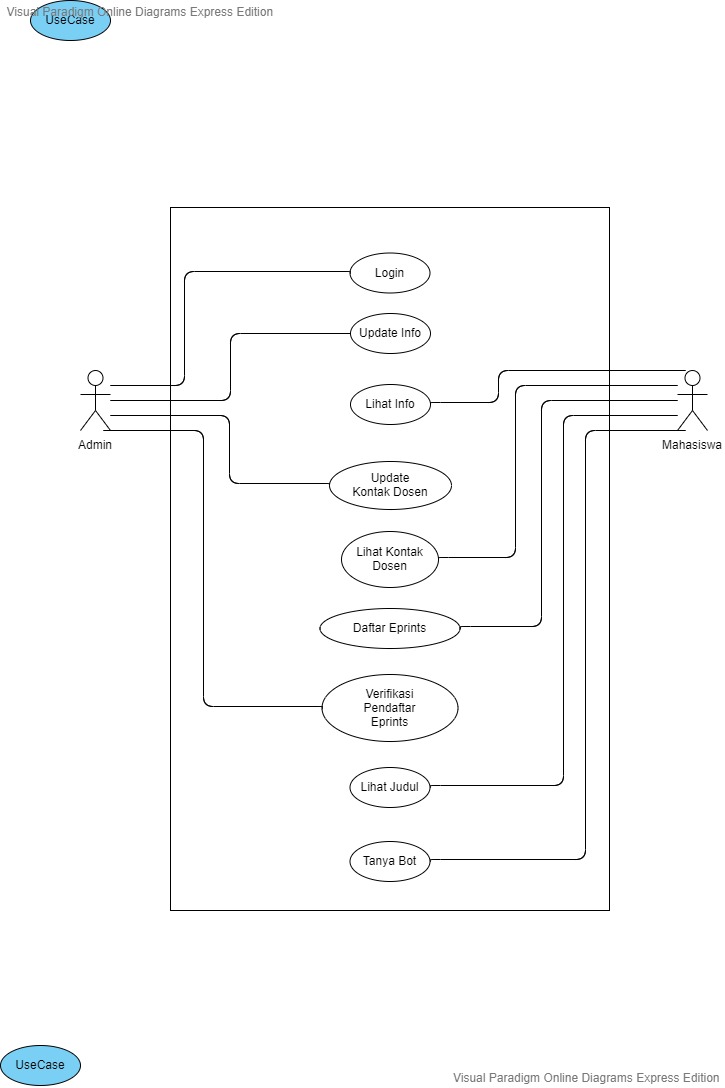
* + 1. Dibutuhkan alternative sistem yang efektif untuk menjawab pertanyaan-pertanyaan mahasiswa seputar skripsi, dimana banyak mahasiswa yang masih bingung dengan alur pendafaran skripsi dari mulai mendaftar, mendapatkan dosen pembimbing hingga sidang skripsi. Solusi yang ditawarkan ialah sebuah aplikasi *chatbot* untuk membantu mahasiswa dalam menjawab pertanyaan mahasiswa seputar skripsi
    2. Untuk memberikan informasi yang efektif dalam suatu wadah yang tepat, mahasiswa membutuhkan tempat untuk bertanya dan biasa bertanya yang paling mudah ialah bertanya kepada dosen, tetapi kembali lagi dengan mahasiswanya yang kesulitan mendapatkan kontak dosen pembimbingnya sendiri. Adapun solusi yang ditawarkan adalah adanya fitur mendapatkan info kontak dosen pembimbing jurusan Teknik Informatika UNPAM.
    3. Mahasiswa harus berupaya untuk bertanya hal yang mereka butuhkan langsung ke bagian prodi jurusan Teknik Informatika yang ada di kampus pastinya, dan hanya bisa dilakukan pada jam operasional kerja. Untuk aplikasi *chatbot*  ini mahasiswa dengan mudah bertanya kapanpun dan dimanapun hanya dengan koneksi internet.

* + 1. **Solusi Yang Dibangun**

Berdasarkan analisis terhadap masalah yang ada, maka diperlukan sebuah media yang dapat digunakan untuk memberikan informasi seputar skripsi kepada mahasiswa secara *realtime*. Media tersebut ialah berupa aplikasi *chatbot.* Teknologi *chatbot* dipilih karena dalam penerapannya dapat bertindak selayaknya manusia atau dalam konteks ini ialah sebagai lawan bicara.

Adapun karakter Sistem Informasiyang dibangun adalah :

1. Mahasiswa dapat berinteraksi dengan *bot* mengenai informasi seputar skripsi dimanapun dan kapanpun dengan koneksi internet
2. Dapat menampilkan kontak dosen pembimbing yang dibutuhkan.
3. Disertakan tahapan-tahapan alur proses penyusunan skripsi yang terstruktuk dalam bentuk chat
4. Dapat melihat judul skripsi sebagai refensi penyusunan skripsi



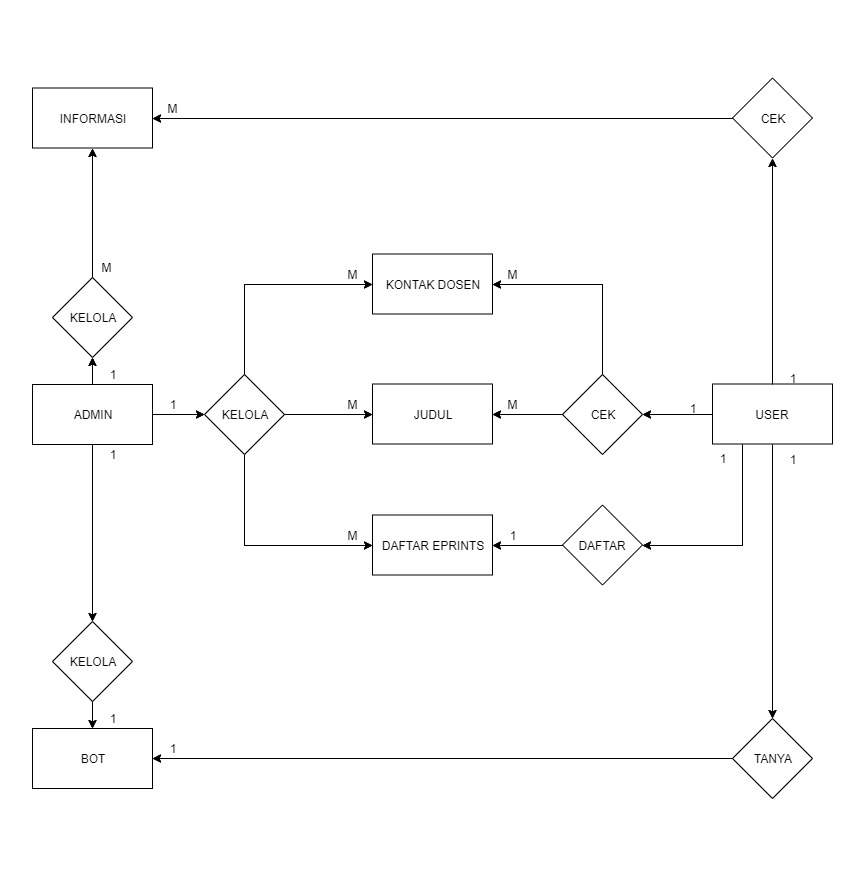
Gambar 3. 1 Use Case Diagram

* 1. **Perancangan Basis Data**

Dalam membuat system ini, ada database sebagai media penyimpanan data. Berikut adalah beberapa rancangan dalam membuat database dari system yang diusulkan.

* + 1. **Entity Relationship Diagram (ERD)**

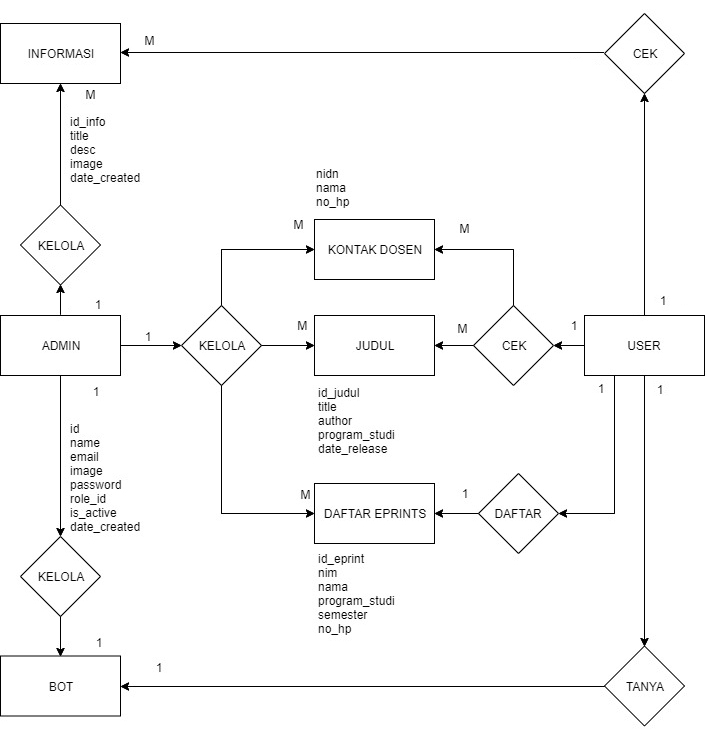
Diagram erd dibawah ini menggambarkan struktur database yang akan dibangun dalam aplikasi ini.



Gambar 3. 2 Entity Relationshio Diagram

* + 1. **Transformasi ERD Ke Logical Record Structure (LRS)**

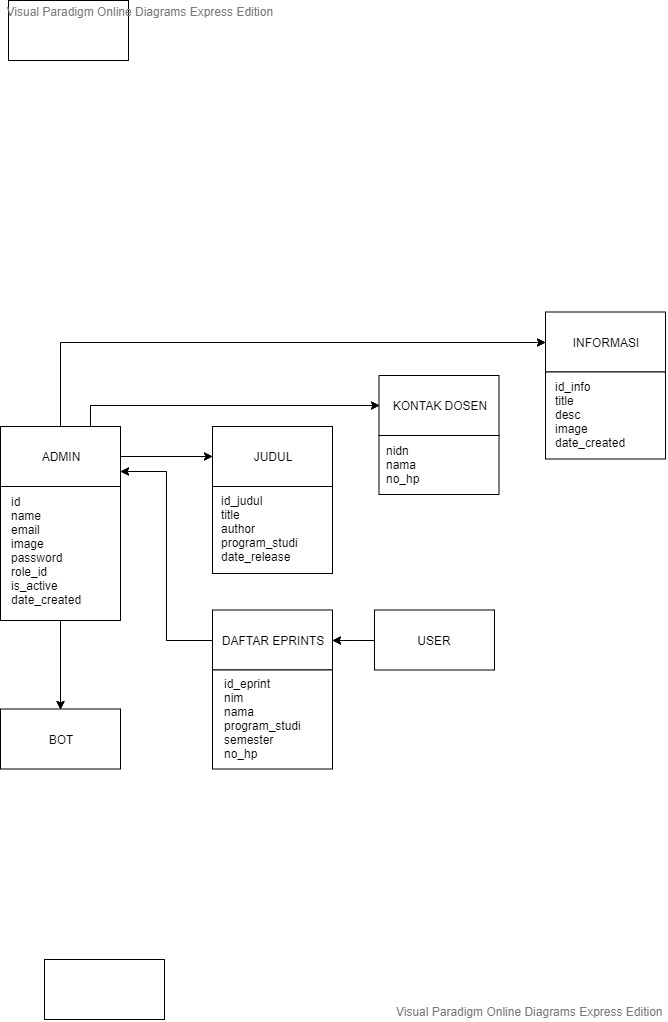
Berikut adalah transformasi *entitas relationship diagram* (LRS) ke *logical record structure* (LRS) digambarkan sebagai berikut :



Gambar 3. 3 ERD ke Logical Record Structure

* + 1. **Logical Record Structure (LRS)**

Berikut adalah *logical record structure* (LRS) digambarkan sebagai berikut :



Gambar 3. 4 Logical Record Structure

* + 1. **Spesifikasi Basis Data**

Berikut spesifikasi basis data yang menjelaskan masing-masing relasi berisi nama file, media penyimpanan yang dipakai, isi atau keterangan dari file, primary key, panjang record, jumlah record, dan struktur file.

* + 1. Nama Tabel : user

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Field** | **Type** | **Length** | **Keterangan** |
| 1 | Id | int | 11 | *Primary Key* |
| 2 | Name | varchar | 128 | Nama Admin |
| 3 | Email | varchar | 128 | Email Admin |
| 4 | Password | varchar | 256 | Password Admin |
| 5 | role\_id | int | 11 | Role Access |
| 6 | is\_active | int | 1 | Kondisi Akun |
| 7 | date\_created | int | 11 | Tanggal Daftar |

Tabel 3. 1 Tabel user

* + 1. Nama Tabel : post\_dosen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Field** | **Type** | **Length** | **Keterangan** |
| 1 | nidn | Int | 20 | *Primary Key* |
| 2 | nama | Varchar | 100 | Nama Dosen |
| 3 | hp | Varchar | 50 | Kontak Dosen |

Tabel 3. 2 Tabel post\_dosen

* + 1. Nama Tabel : post\_article

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Field** | **Type** | **Length** | **Keterangan** |
| 1 | id\_info | Int | 11 | *Primary Key* |
| 2 | title | Varchar | 128 | Judul Info |
| 3 | desc | Text |  | Deskripsi Info |
| 4 | image | Varchar | 256 | Gambar Info |
| 5 | date\_created | Int | 11 | Tanggal Info Dibuat |

Tabel 3. 3 Tabel post\_acrticle

* + 1. Nama Tabel : post\_eprints

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Field** | **Type** | **Length** | **Keterangan** |
| 1 | id\_eprint | Int | 11 | *Primary Key* |
| 2 | nim | Varchar | 128 | NIM User |
| 3 | nama | Varchar | 256 | Nama User |
| 4 | program\_studi | Varchar | 256 | Program Studi User |
| 5 | semester | Varchar | 128 | Semester User |
| 6 | no­­\_hp | Varchar | 256 | No Hp User |

Tabel 3. 4 Tabel post\_eprints

* + 1. Nama Tabel : post\_judul

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Field** | **Type** | **Length** | **Keterangan** |
| 1 | id\_judul | Int | 11 | *Primary Key* |
| 2 | title | Varchar | 512 | Judul Skripsi |
| 3 | author | Varchar | 256 | Nama Penulis |
| 4 | program\_studi | Varchar | 256 | Program Studi |
| 5 | Date\_release | Int | 11 | Tanggal Rilis |

Tabel 3. 5 Tabel post\_judul

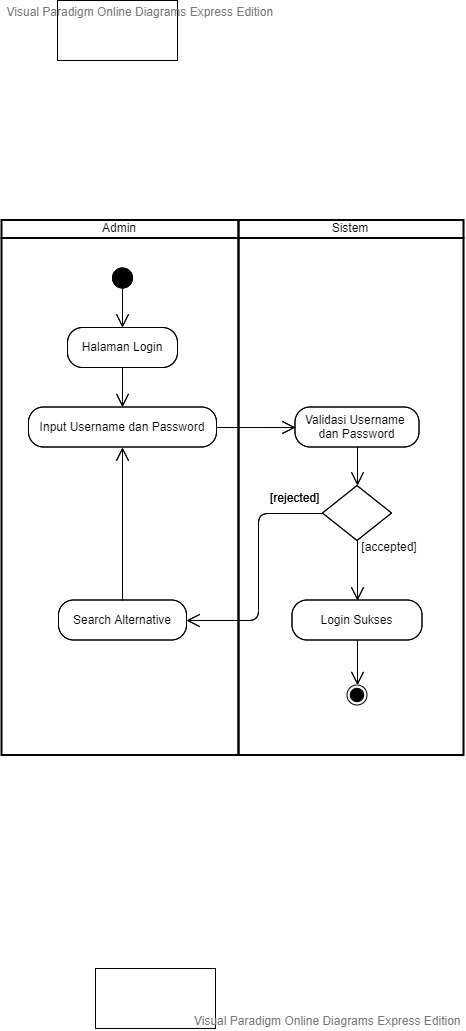
* 1. **Pengajuan Unified Modelling Language (UML)**

Perancangan pengembangan perangkat dengan menggunakan metode grafis serta merupakan bahasa untuk visualisasi, spesifikasi, kontruksi serta dokumentasi. *unified modelling language* (UML) meliputi perancangan *usecase* diagram, *activity* diagram, *class* diagram, *sequence* diagram yang akan digunakan.

* + 1. **Activity Diagram**

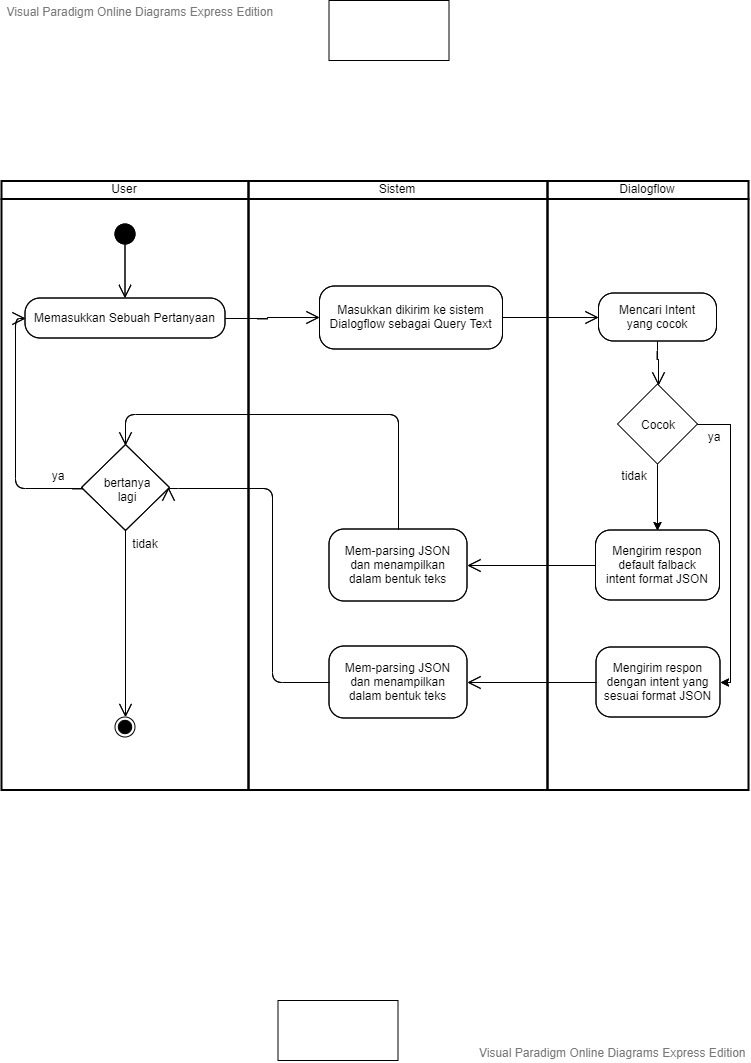
Activity diagram bertujuan sebagai gambaran sebuah alur kerja pada aplikasi berdasarkan aktifitas yang dilakukan oleh pengguna terhadap sistem. Pada penelitian ini *activity* diagram yang akan dibuat mempunyai alur yang sama dengan setiap kemungkinan percakapan yang akan terjadi. Pada Gambar 3.8 Activity Diagram merupakan activity diagram yang terdapat pada aplikasi :

* + 1. Activity Diagram Login



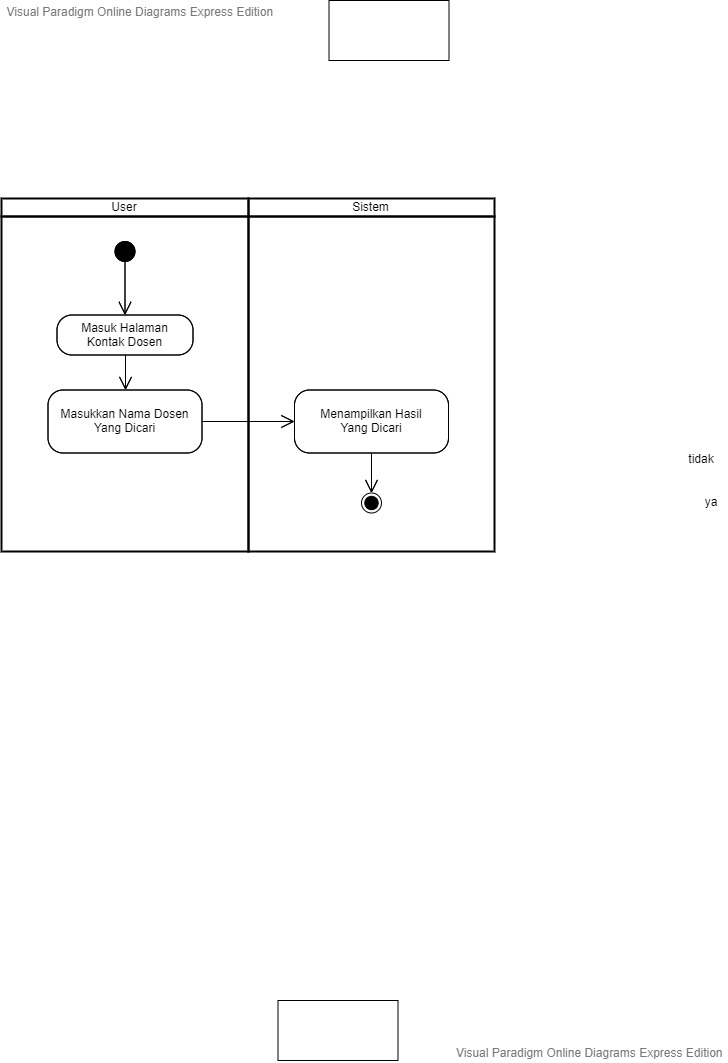
Gambar 3. 5 Activity Diagram Login

* + 1. Activity Bot



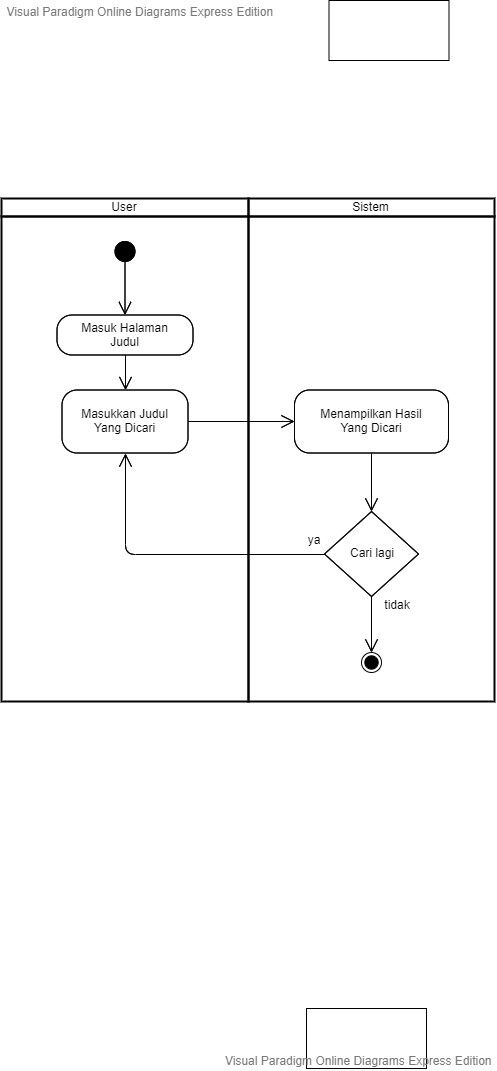
Gambar 3. 6 Activity Diagram Bot

* + 1. Activity Kontak Dosen



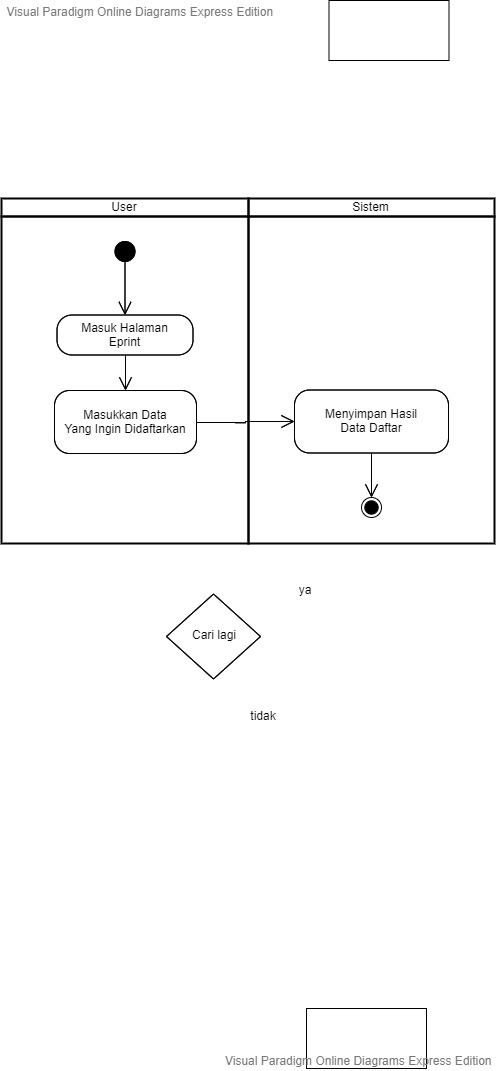
Gambar 3. 7 Activity Diagram Kontak Dosen

* + 1. Activity Judul



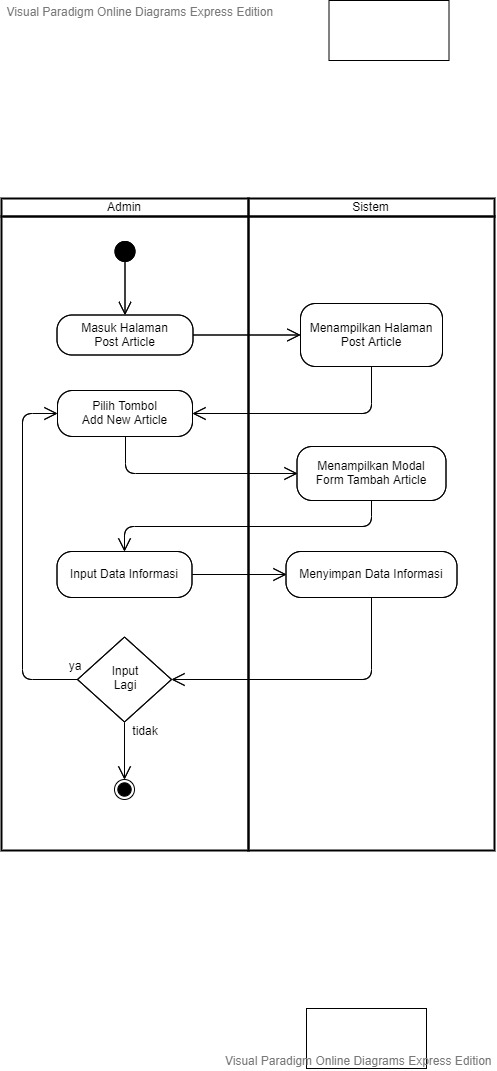
Gambar 3. 8 Activity Diagram Judul

* + 1. Activity Eprints



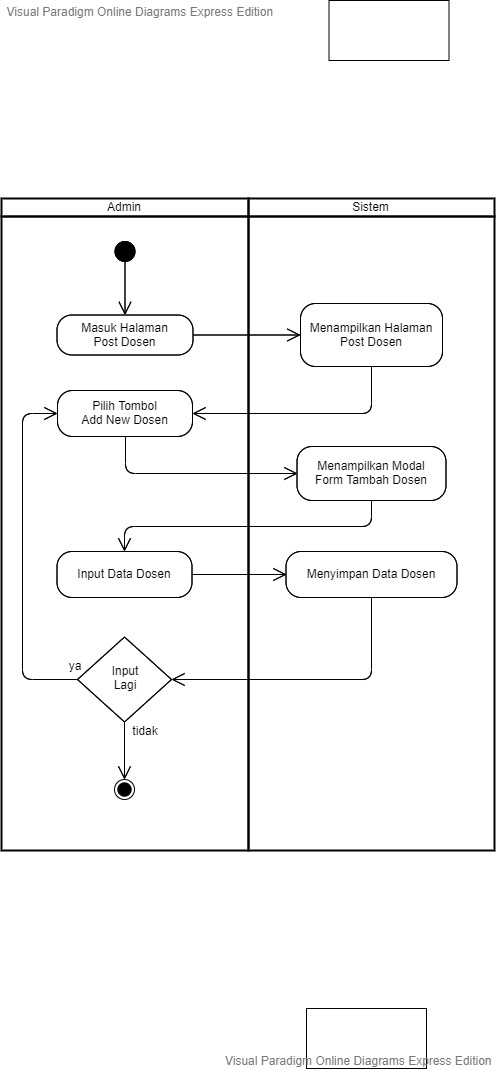
Gambar 3. 9 Activity Diagram Eprints

* + 1. Activity Post Informasi



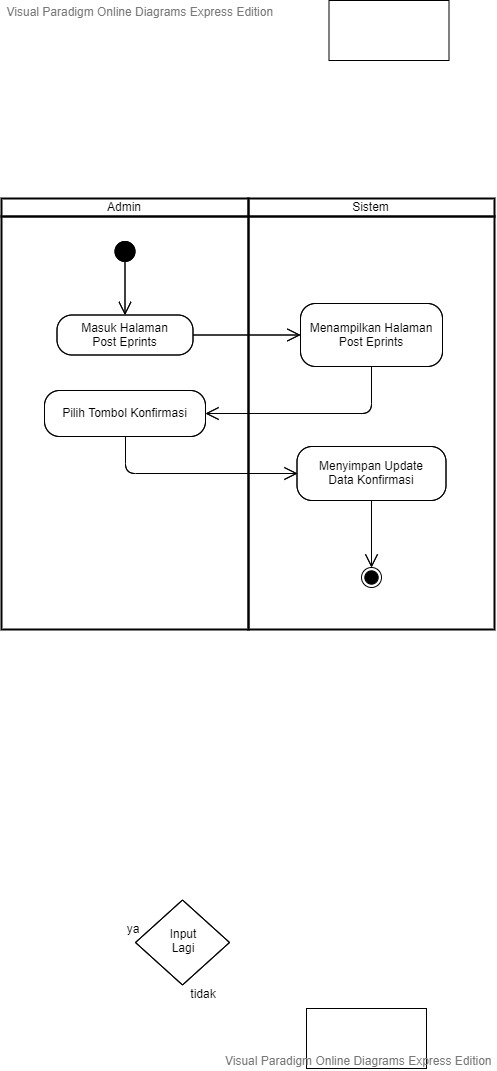
Gambar 3. 10 Activity Diagram Post Informasi

* + 1. Activity Post Dosen



Gambar 3. 11 Activity Diagram Post Dosen

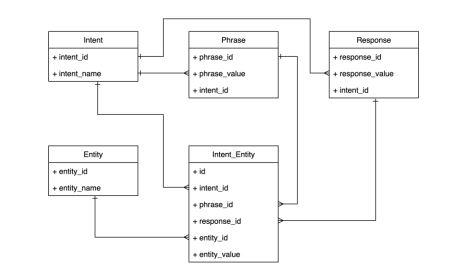
* + 1. Activity Update Data Eprints



Gambar 3. 12 Activity Diagram Update Data Eprints

* + 1. **Class Diagram**

Penulis menggambarkan *class* diagram bertujuan untuk menampilkan kelas-kelas dan paket didalam *system* aplikasi ini. Class diagram memberikan gambaran *system* aplikasi dan relasi antar mereka. Biasanya, dibuat beberapa *class* diagram untuk *system* tunggal. Beberapa diagram akan menampilkan subset dari kelas-kelas dan relasinya.

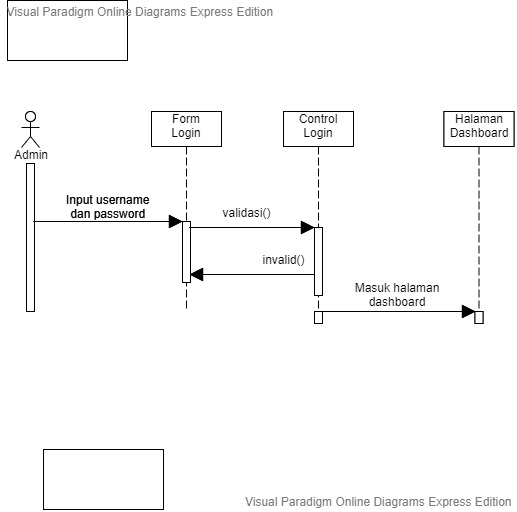


Gambar 3. 13 Class Diagram Bot

* + 1. **Sequence Diagram**

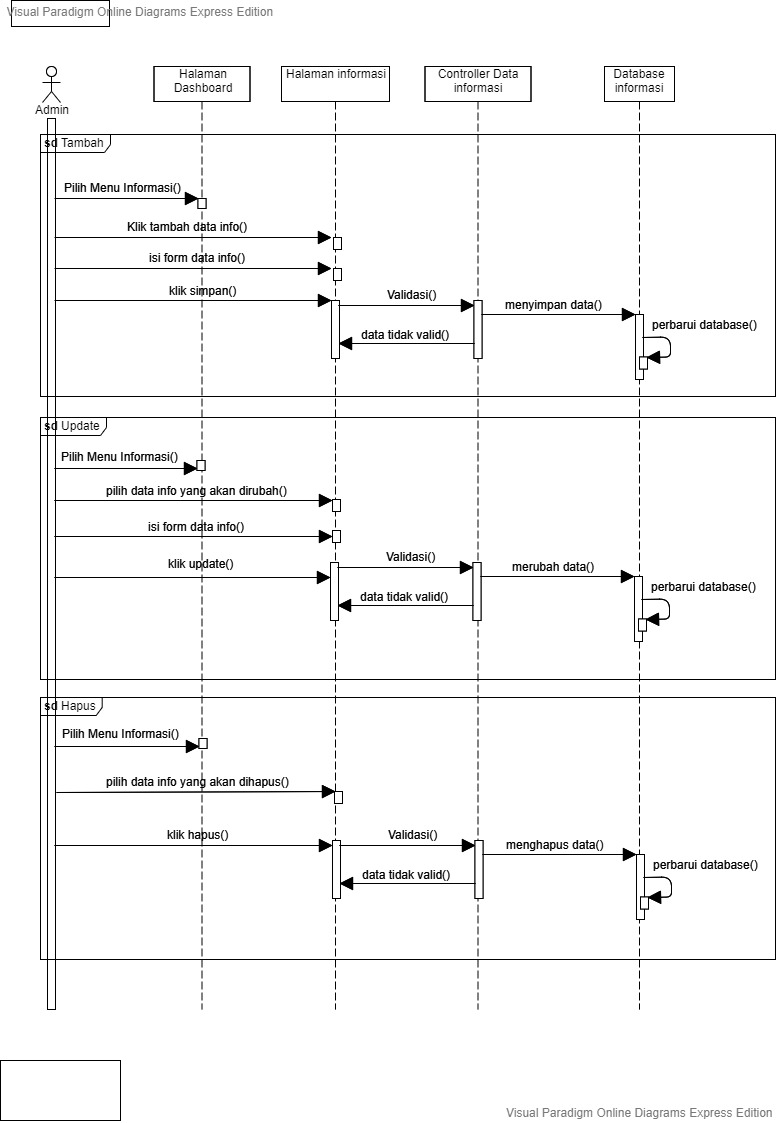
Sequence diagram menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus berasosiasi dengan *usecase*. *Sequence* diagram juga menggambarkan kelakuan objek pada *usecase* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan yang diterima oleh objek oleh karena itu untuk menggambarkan diagram *sequence* harus diketahui objek-objek yang terlibat dalam sebuah *usecase* beserta metode-metode yang dimiliki *class* yang diinstansiasi menjadi objek itu.

1. Sequence Diagram Login Admin



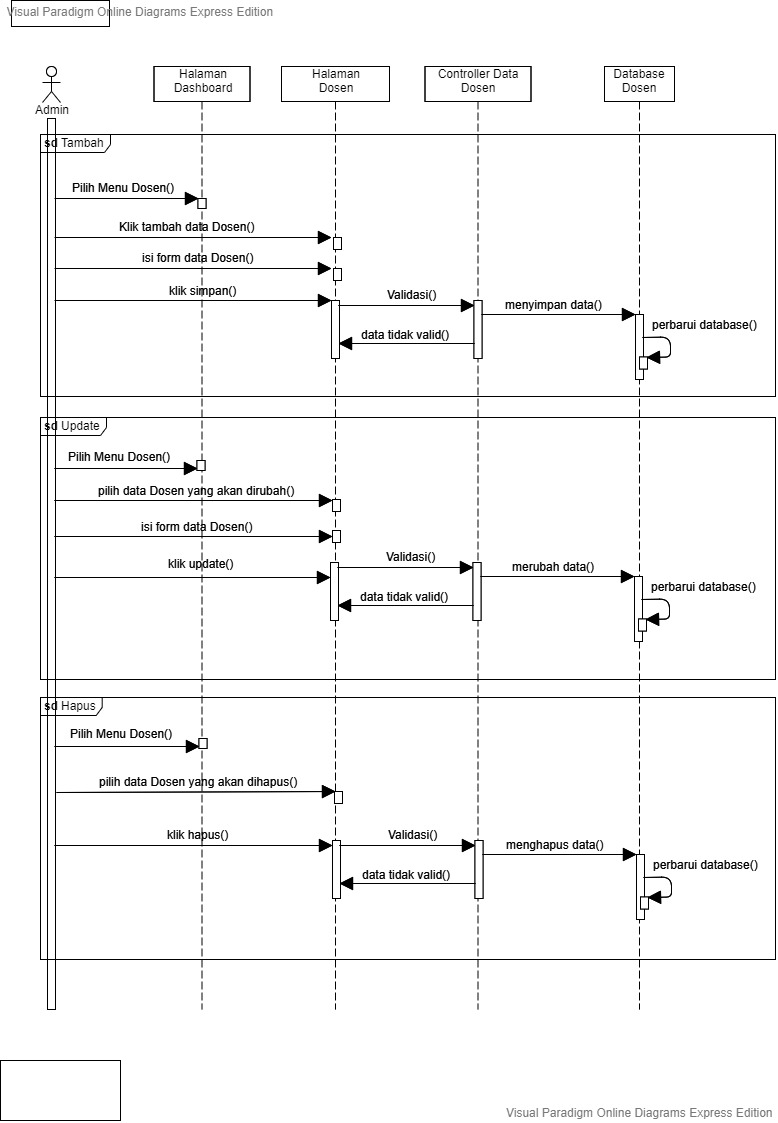
Gambar 3. 14 Sequence Diagram Login Admin

1. Sequence Diagram Post Informasi



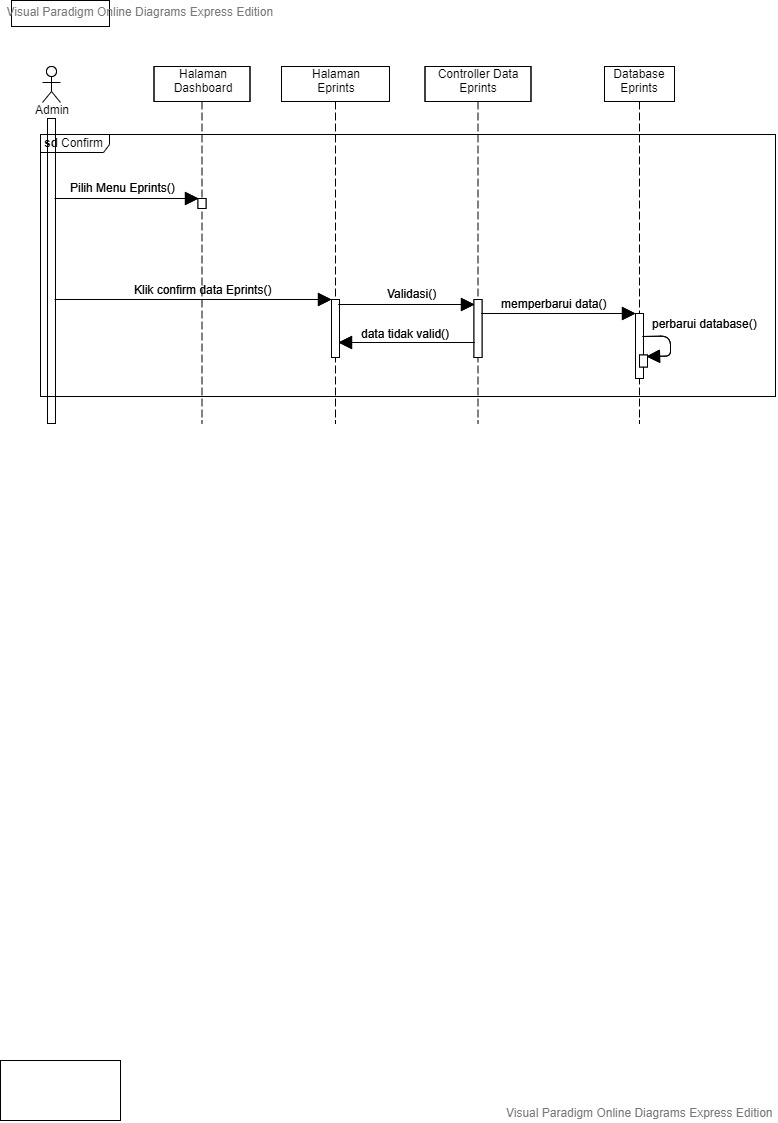
Gambar 3. 15 Sequence Diagram Post Informasi

1. Sequence Diagram Post Dosen



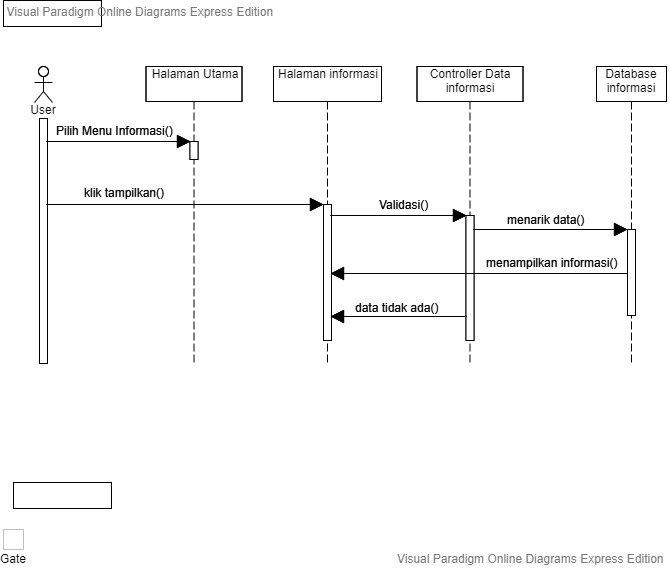
Gambar 3. 16 Sequence Diagram Post Dosen

1. Sequence Diagram Post Eprints



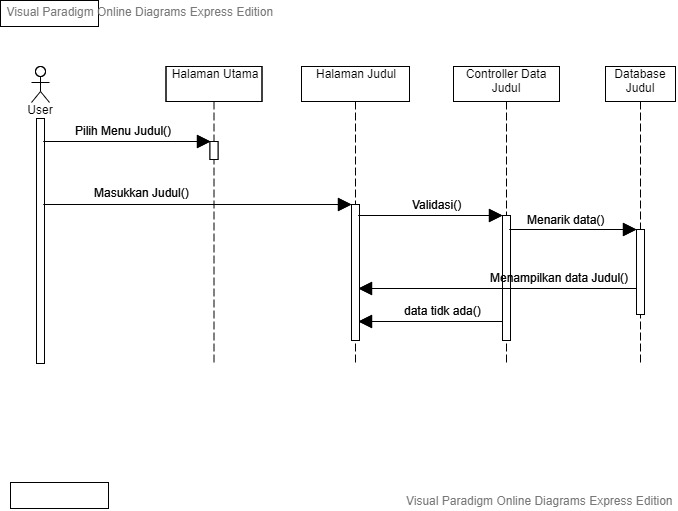
Gambar 3. 17 Sequence Diagram Post Eprints

1. Sequence Diagram Informasi



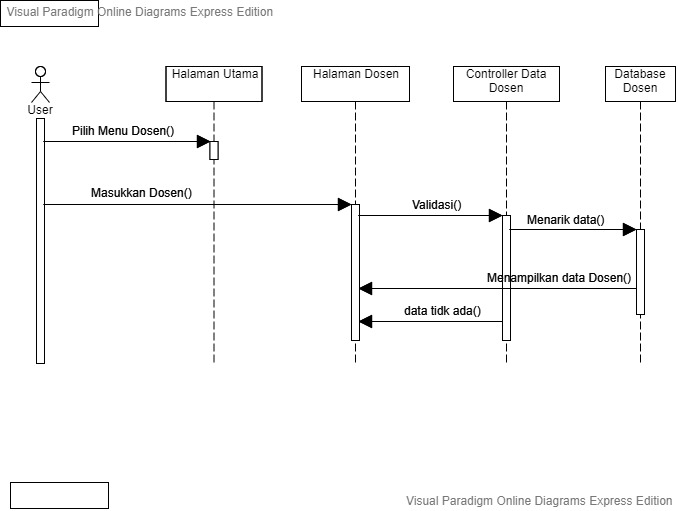
Gambar 3. 18 Sequence Diagram Informasi

1. Sequence Diagram Judul



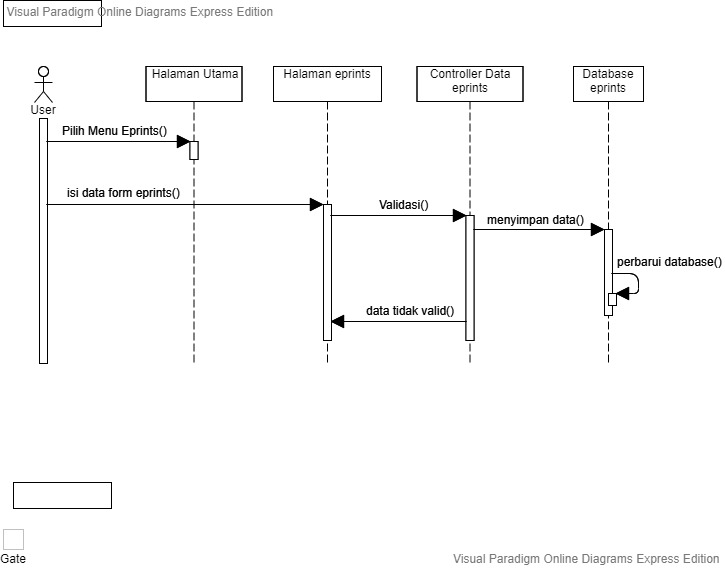
Gambar 3. 19 Sequence Diagram Judul

1. Sequence Diagram Dosen



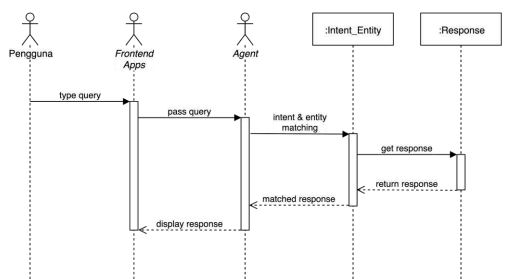
Gambar 3. 20 Sequence Diagram Dosen

1. Sequence Diagram Eprints



Gambar 3. 21 Sequence Diagram Eprints

1. Sequence Diagram Chatbot

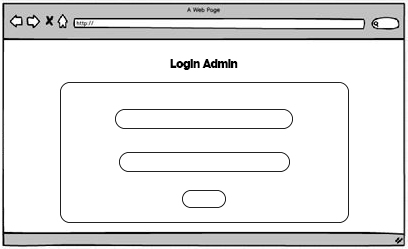


Gambar 3. 22 Sequence Diagram Chatbot

* 1. **Perancangan Antar Muka (User Interface)**

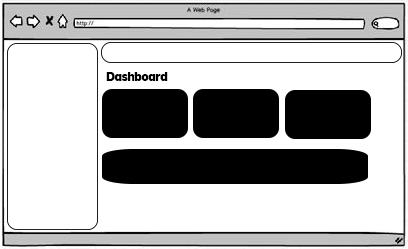
perancangan antarmuka merupakan suatu deskripsi layout antar muka system yang akan dibuat. Dengan adanya perancangan antar muka diharapkan baik pengembang aplikasi maupun pengguna dapat menyatukan pandangan mengenai layout tampilan aplikasi yang dibuat.

* + 1. Halaman Login



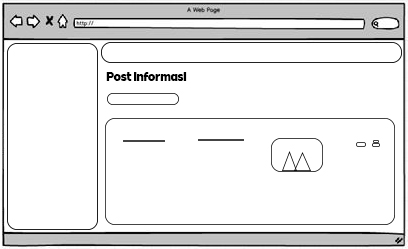
Gambar 3. 23 Rancangan Tampilan Halaman Login

* + 1. Halaman Dasbord



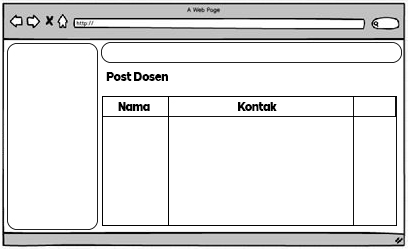
Gambar 3. 24 Rancangan Tampilan Halaman Dashboard

* + 1. Halaman Post Info



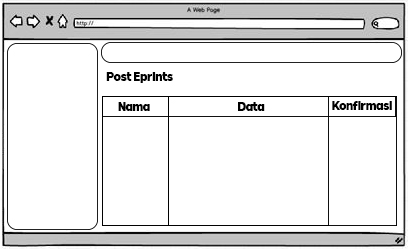
Gambar 3. 25 Rancangan Tampilan Halaman Post Info

* + 1. Halaman Post Dosen



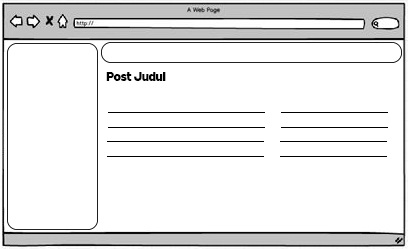
Gambar 3. 26 Rancangan Tampilan Halaman Post Dosen

* + 1. Halaman Post Eprints



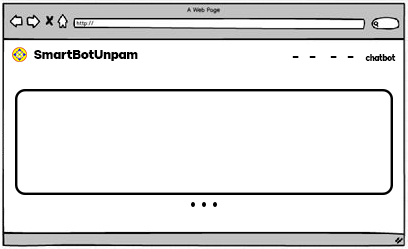
Gambar 3. 27 Rancangan Tampilan Halaman Post Eprints

* + 1. Halaman Post Judul



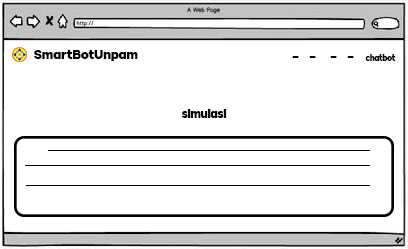
Gambar 3. 28 Rancangan Tampilan Halaman Post Judul

* + 1. Halaman Utama



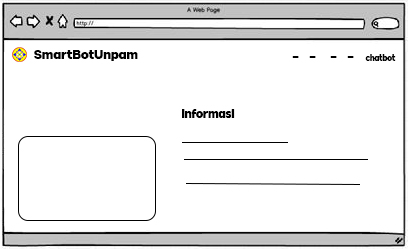
Gambar 3. 29 Rancangan Tampilan Halaman Utama

* + 1. Halaman Simulasi



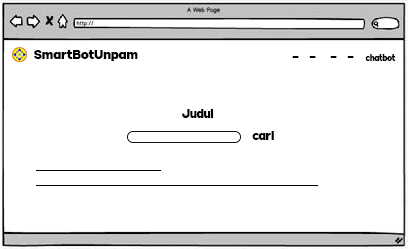
Gambar 3. 30 Rancangan Tampilan Halaman Simulasi

* + 1. Halaman Informasi



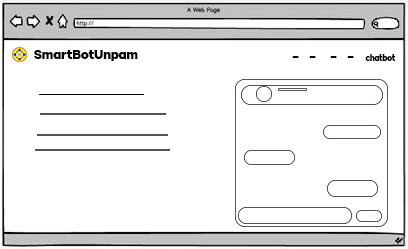
Gambar 3. 31 Rancangan Tampilan Halaman Informasi

* + 1. Halaman Judul



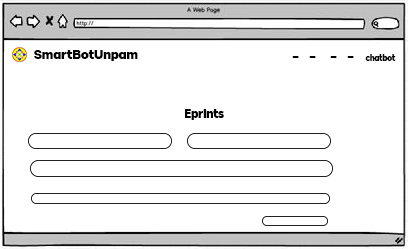
Gambar 3. 32 Rancangan Tampilan Halaman Judul

* + 1. Halaman Chatbot



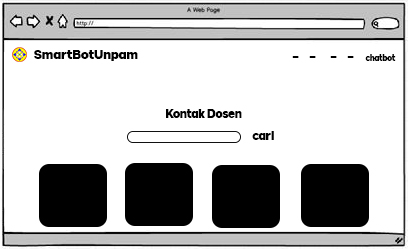
Gambar 3. 33 Rancangan Tampilan Halaman Chatbot

* + 1. Halaman Form Eprints



Gambar 3. 34 Rancangan Tampilan Halaman Form Eprints

* + 1. Halaman Kontak Dosen



Gambar 3. 35 Rancangan Tampilan Halaman Kontak Dosen

# BAB IV IMPLEMENTASI DAN PENGUJIAN

* 1. **Spesifikasi Sistem**

Implementasi system merupakan dokumen yang berfungsi menggambarkan fungsi dan kinerja system berbasis computer yang akan dikembankan, membatasu elemen-elemen yang telah dialokasikan , serta memberikan indikasi mengenai perangkat lunak dan kontek system keseluruhan dan informasi data dan control yang dimasukkan dan dikeluarkan oleh system yang telah digambarkan dalam diagram aliran arsitektur. Berikut salah satu format dokumen spesifikasi sitem yang biasa digunakan.

* + 1. **Spesifikasi Perangkat Keras**

Hardware atau disebut dengan perangkat keras yaitu salah satu komponen dari sebuah computer yang sifatnya bisa dilihat dan diraba secara langsung atau yang berbentuk nyata, yang berfungsi untuk mendukung proses komputerisasi. Hardware dapat bekerja berdasarkan perintah yang telah ditentukan ada padanya, atau yang juga disebut dengan istilah instruction set. Dengan adanya perintah yang dapat dimengerti oleh hardware tersebut, maka hardware tersebut dapat melakukan berbagai kegiatan yang telah ditentukan oleh pemberi perintah.

Adapun perangkat keras yang digunakan untuk mendukung pembuatan program ini adalah sebagai berikut :

1. Processor : Intel Core2duo
2. Monitor : 14-inch HD
3. Memori : 2GB
4. Hardisk : 500 GB
   * 1. **Implementasi Perangkat Lunak**

Sekumpulan data elektronik yang disimpan oleh computer itu dapat berupa program atau instruksi yang akan menjalankan suatu perintah. Perangkat lunak disebut juga sebagai penerjemah perintah-perintah yang dijalankan pengguna computer untuk diteruskan atau diproses oleh perangkat keras. Melalui software atau perangkat lunak inilah suatu computer dapat menjalankan perintah.

Adapun perangkat lunak (software) yang dibutuhkan untuk menjalankan aplikasi system informasi skripsi adalah sebagai berikut :

1. sistem operasi windows 10
2. sublime text 3
3. xampp control panel versi 3.2.2
   1. **Implementasi Antarmuka (Interface)**

Implementasi antar muka dilakukan dengan setiap lempiran program yang dibangun dan pengkodeannya dalam bentuk file program, berikut beberapa tampilan program yang terdapat didalam aplikasi media pemberlajaran pengenalan hardware computer. Berikut ini adalah implementasi antarmuka dari aplikasi sistem informasi skripsi di unversitas pamulang :

* + 1. **Halaman Utama**

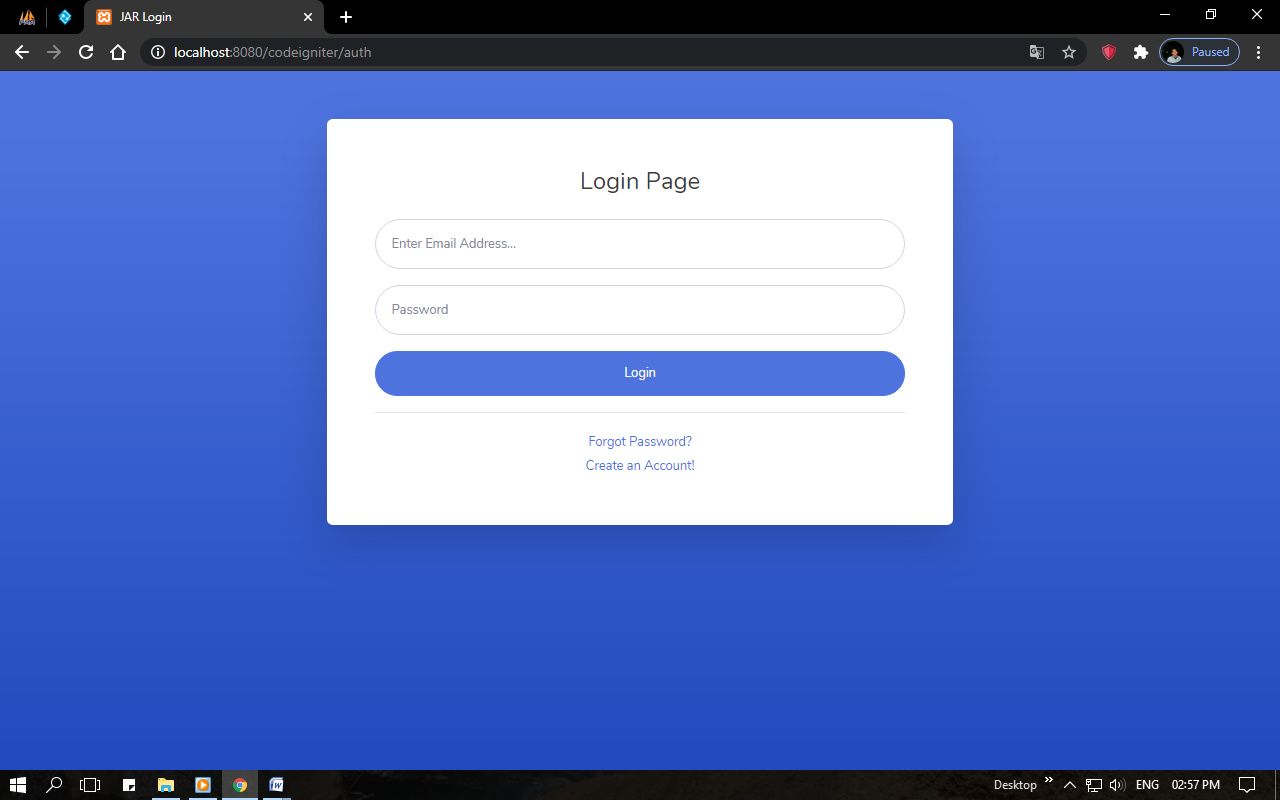
Tampilan halaman utama ini berisi tentang informasi singkat tentang website sistem informasi skripsi unpam (smartbot unpam).



Gambar 4. 1 Tampilan Halaman Utama

* + 1. **Halaman Login**

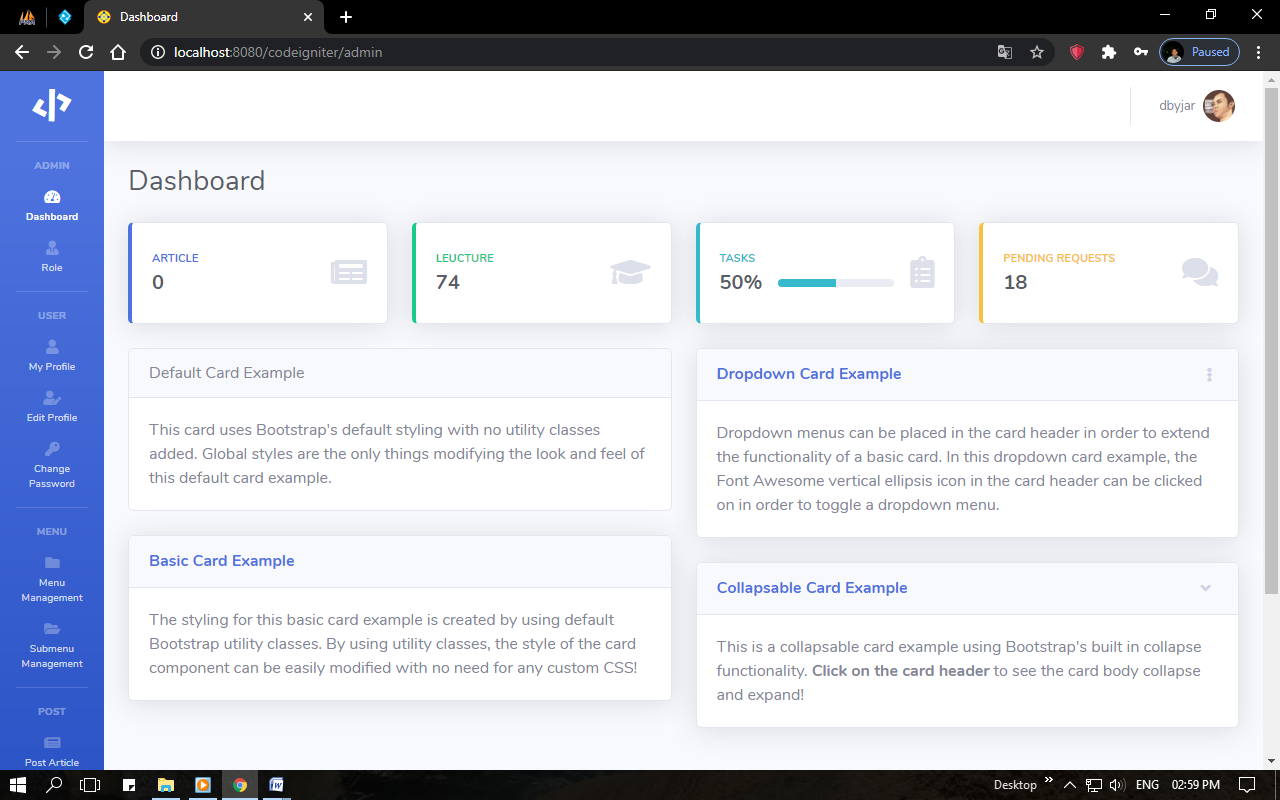
Tampilan halaman login diperuntukan bagi admin website untuk mengatur data didalam web sistem informasi skripsi ini.



Gambar 4. 2 Tampilan Halaman Login Admin

* + 1. **Halaman Dashboard Admin**

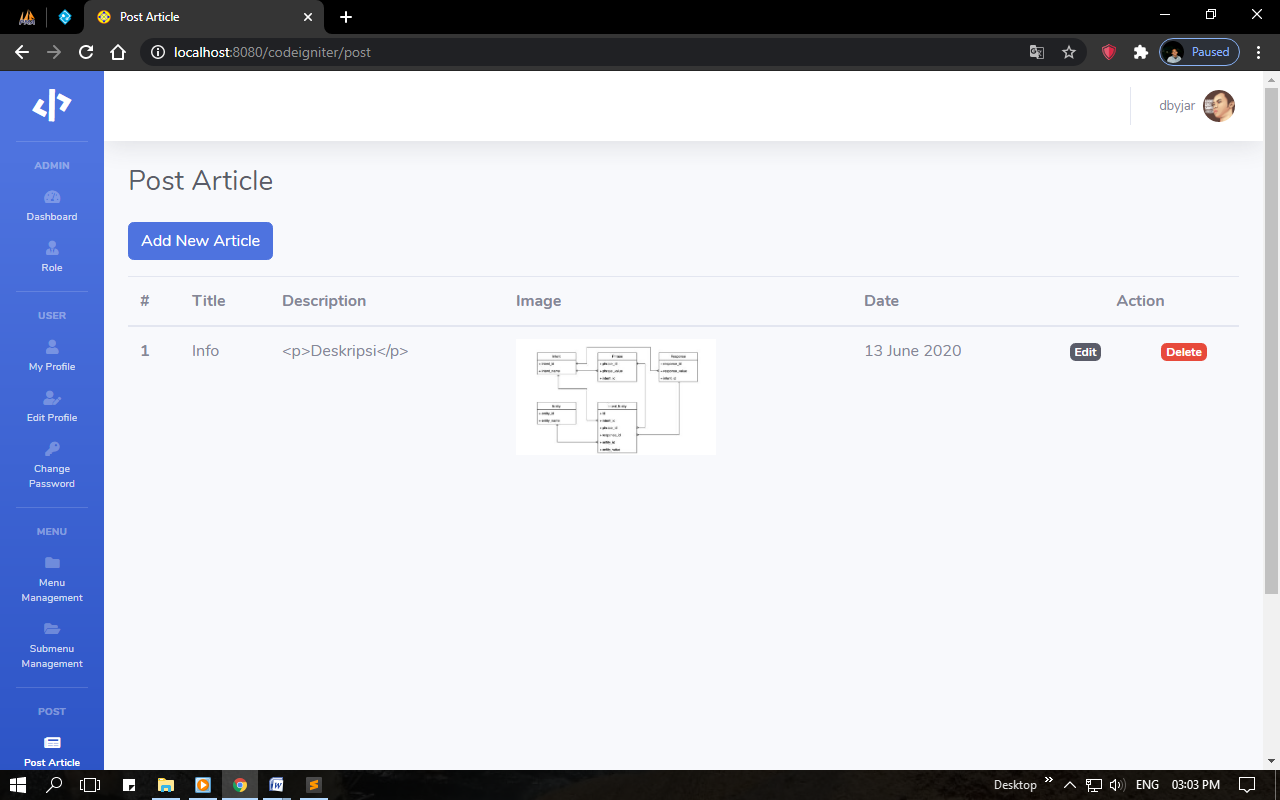
Sebuah halaman untuk menampilkan jumlah data yang terdapat di dalam web sistem informasi skripsi ini.



Gambar 4. 3 Tampilan Halaman Dashboard

* + 1. **Halaman Post Info**

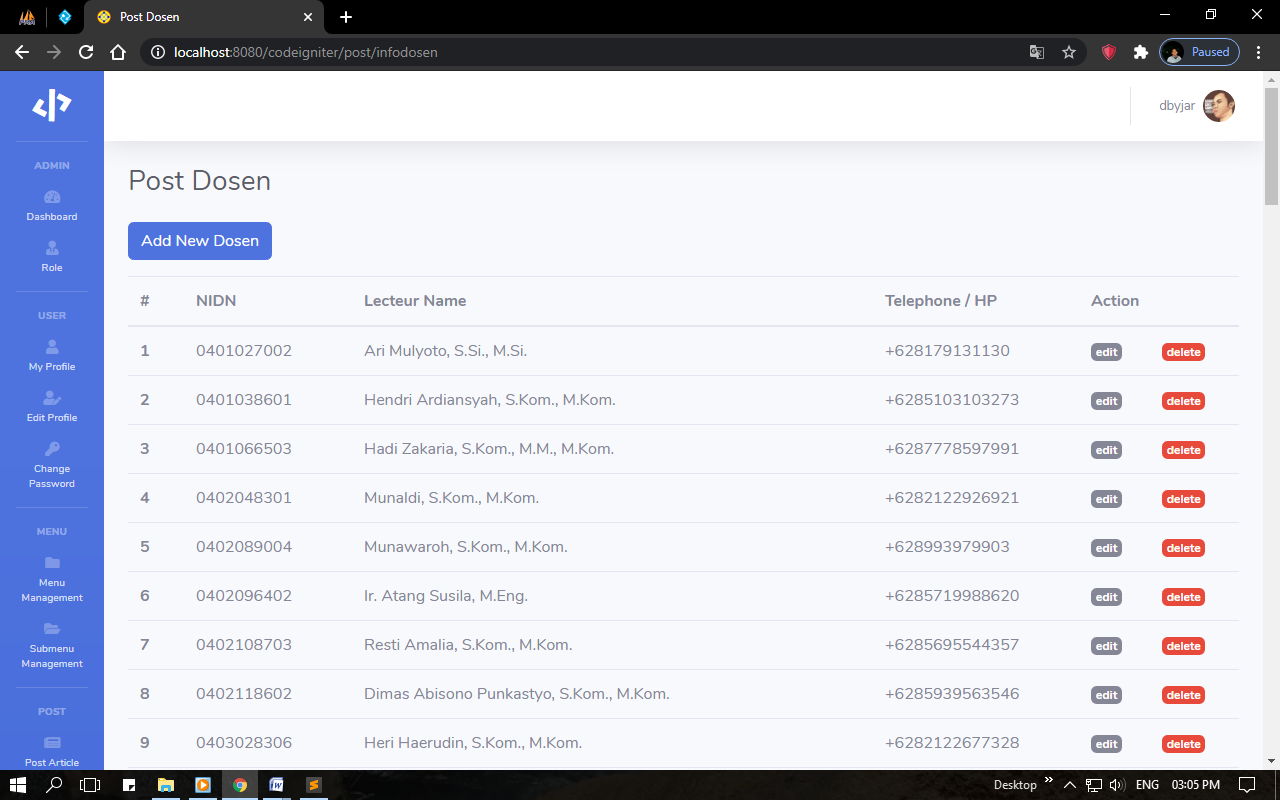
Halaman post info atau artikel adalah sebuah halaman admin untuk mengelola sebuah postingan informasi untuk pengguna web.



Gambar 4. 4 Tampilan Halaman Post Info

* + 1. **Halaman Post Dosen**

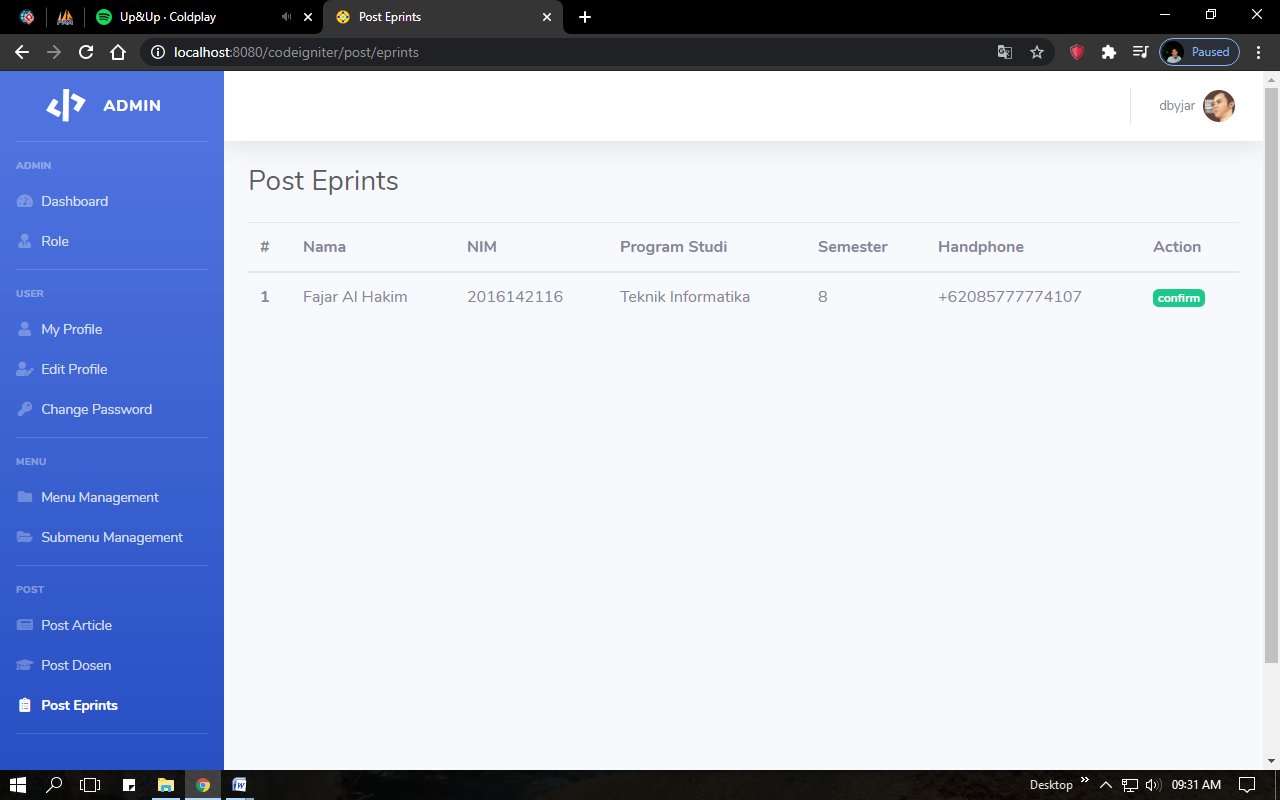
Halaman post dosen adalah halaman untuk mengelola data-data dosen teknik informatika di unversitas pamulang.



Gambar 4. 5 Tampilan Halaman Post Dosen

* + 1. **Halaman Post Eprints**

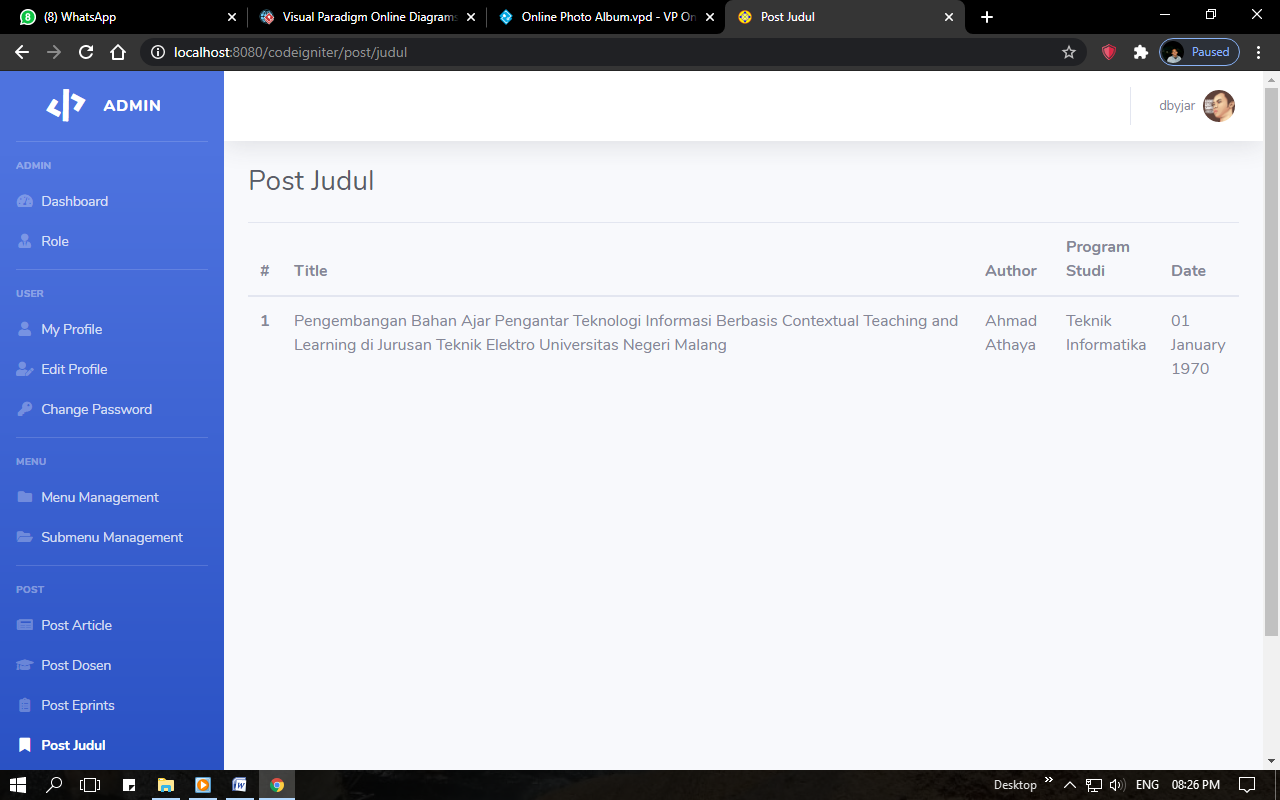
Halaman post eprints yaitu halaman untuk mengelola data pendaftar eprints untuk dikonfirmasi oleh admin untuk didaftarkan ke bagian eprints universitas pamulang.



Gambar 4. 6 Tampilan Halaman Post Eprints

* + 1. **Halaman Post Judul**

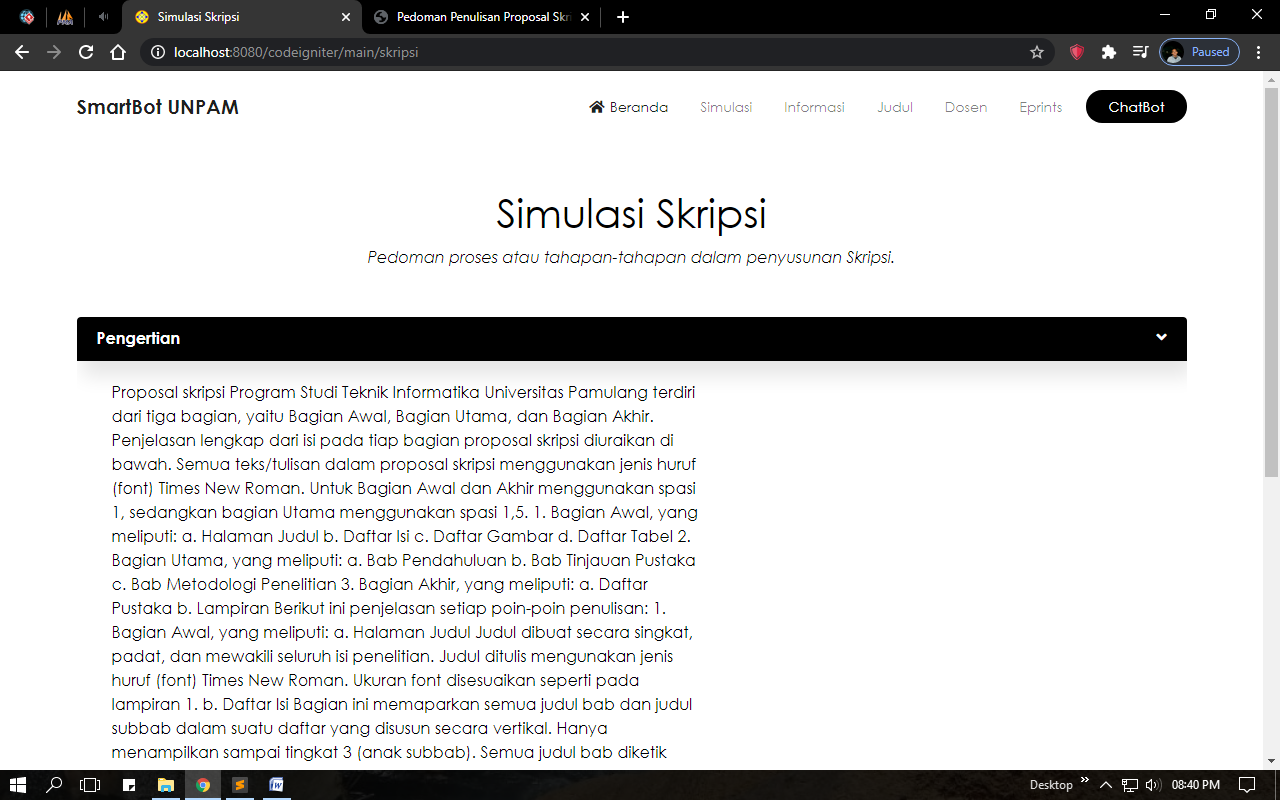
Halaman post judul yaitu halaman untuk mengelola data judul –judul skripsi yang ada di Universitas Pamulang Jurusan Teknik Informatika



Gambar 4. 7 Tampilan Halaman Post Judul

* + 1. **Halaman Skripsi**

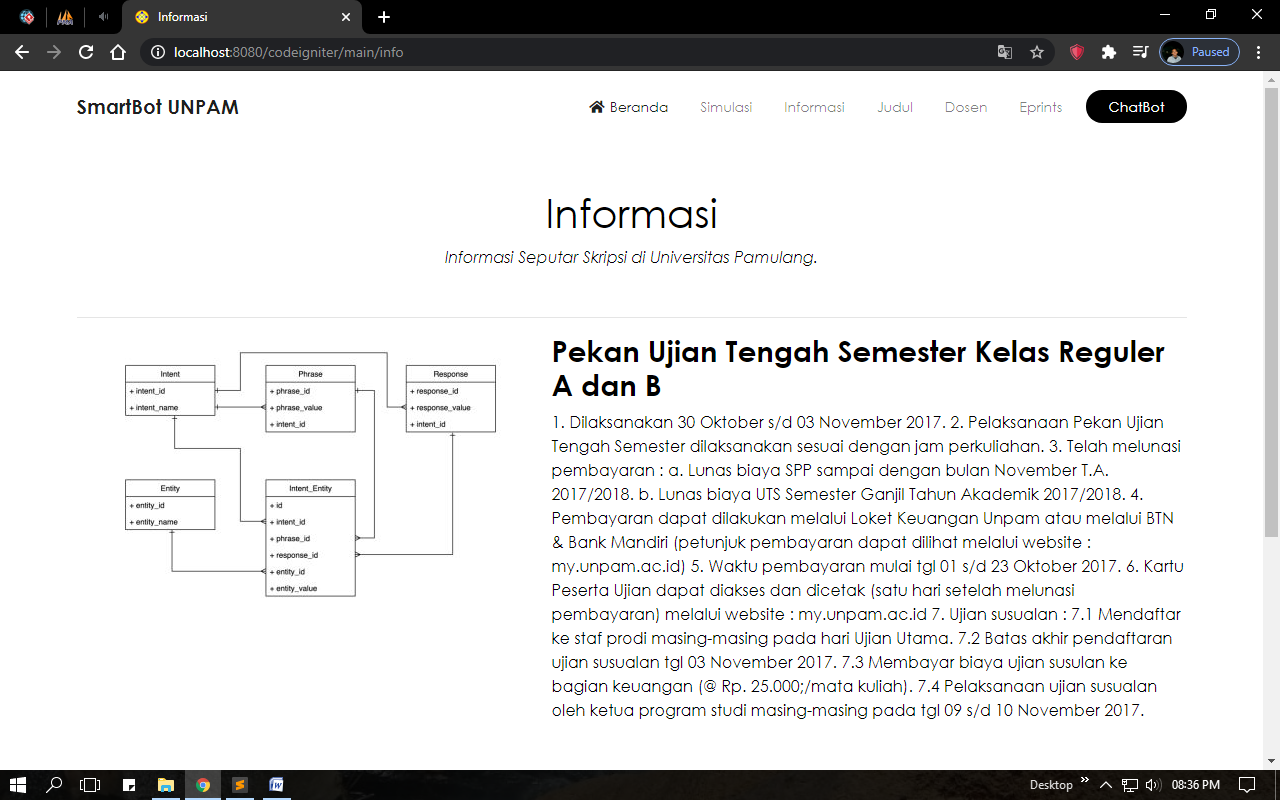
Halaman ini adalah halaman dimana terdapat simulasi atau tahapan pedoman penyusunan skripsi.



Gambar 4. 8 Tampilan Halaman Skripsi

* + 1. **Halaman Informasi**

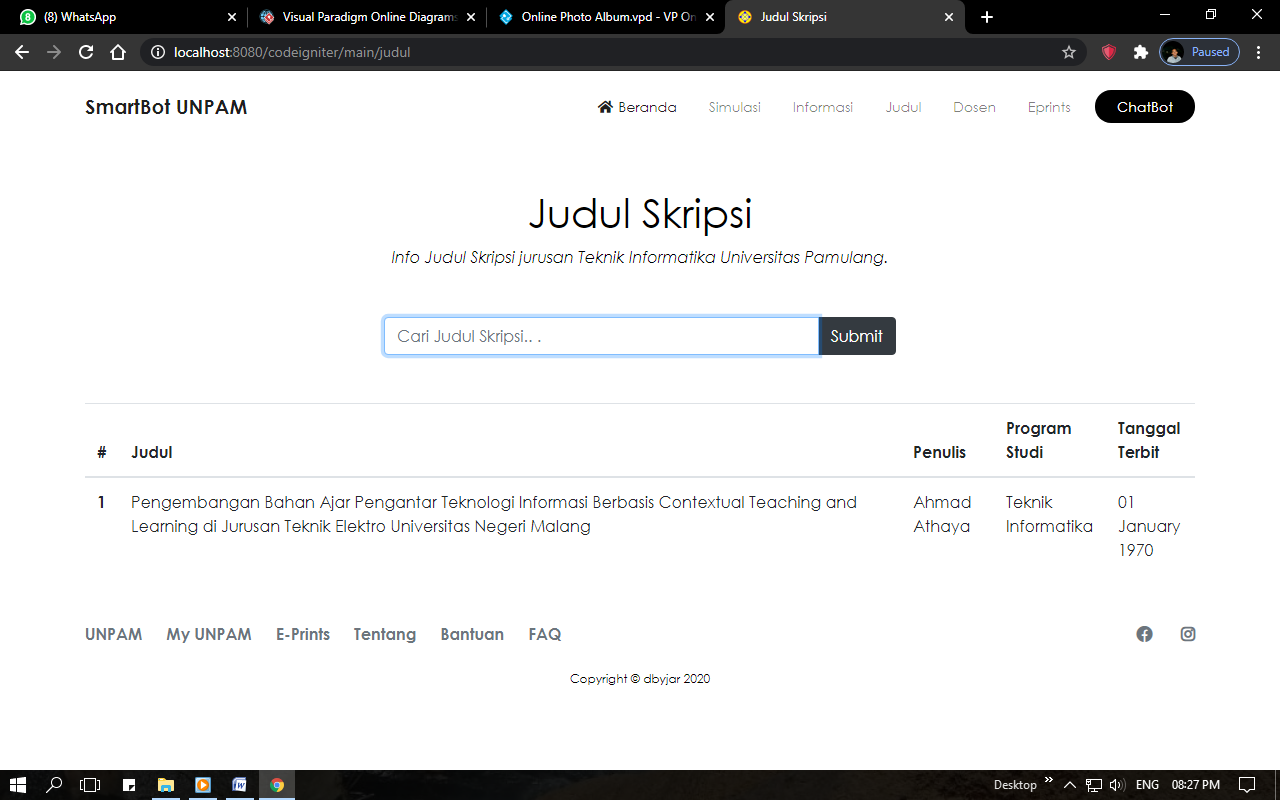
Halaman informasi halaman dimana terdaoat kumpulan informasi yang dibutuhkan pengguna.



Gambar 4. 9 Tampilan Halaman Informasi

* + 1. **Halaman Judul**

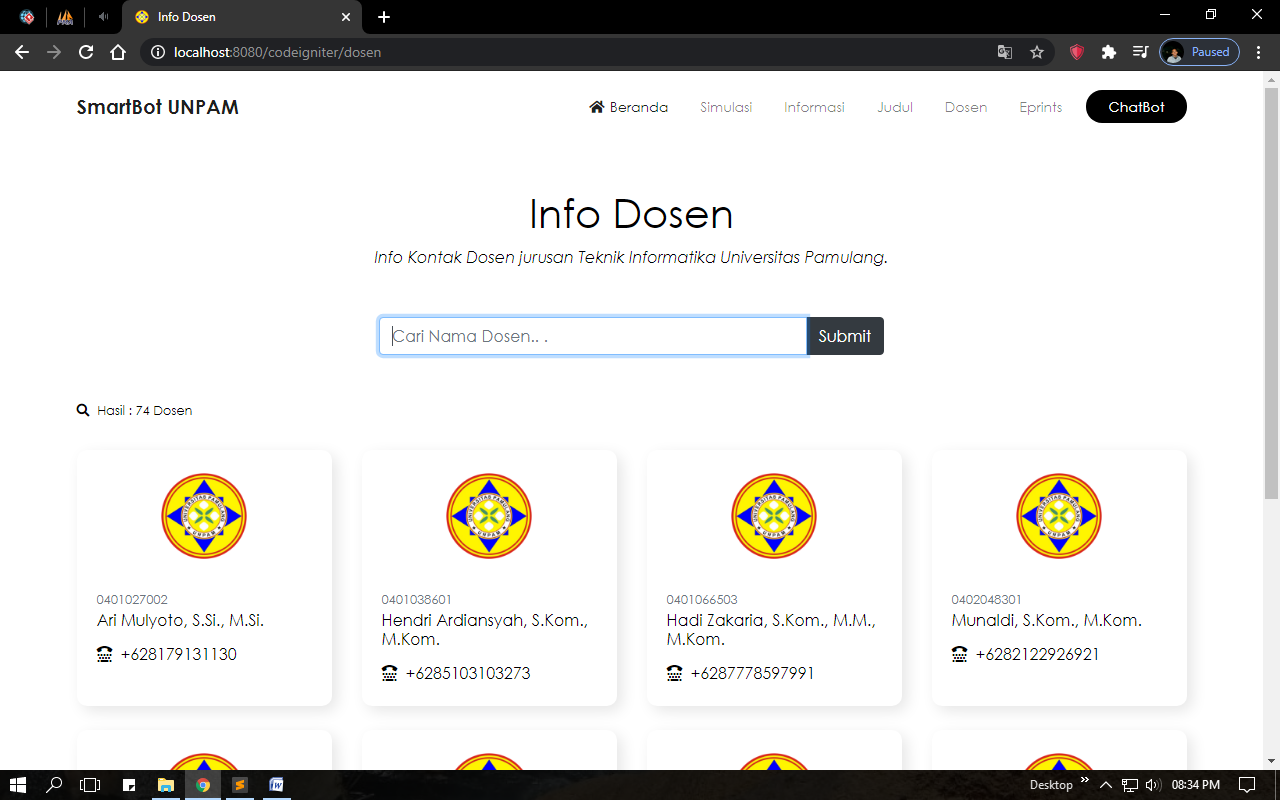
Halaman judul yaitu halaman untuk pencarian judul-judul skripsi yang sudah pernah ada di Universitas Pamulang



Gambar 4. 10 Tampilan Halaman Judul

* + 1. **Halaman Dosen**

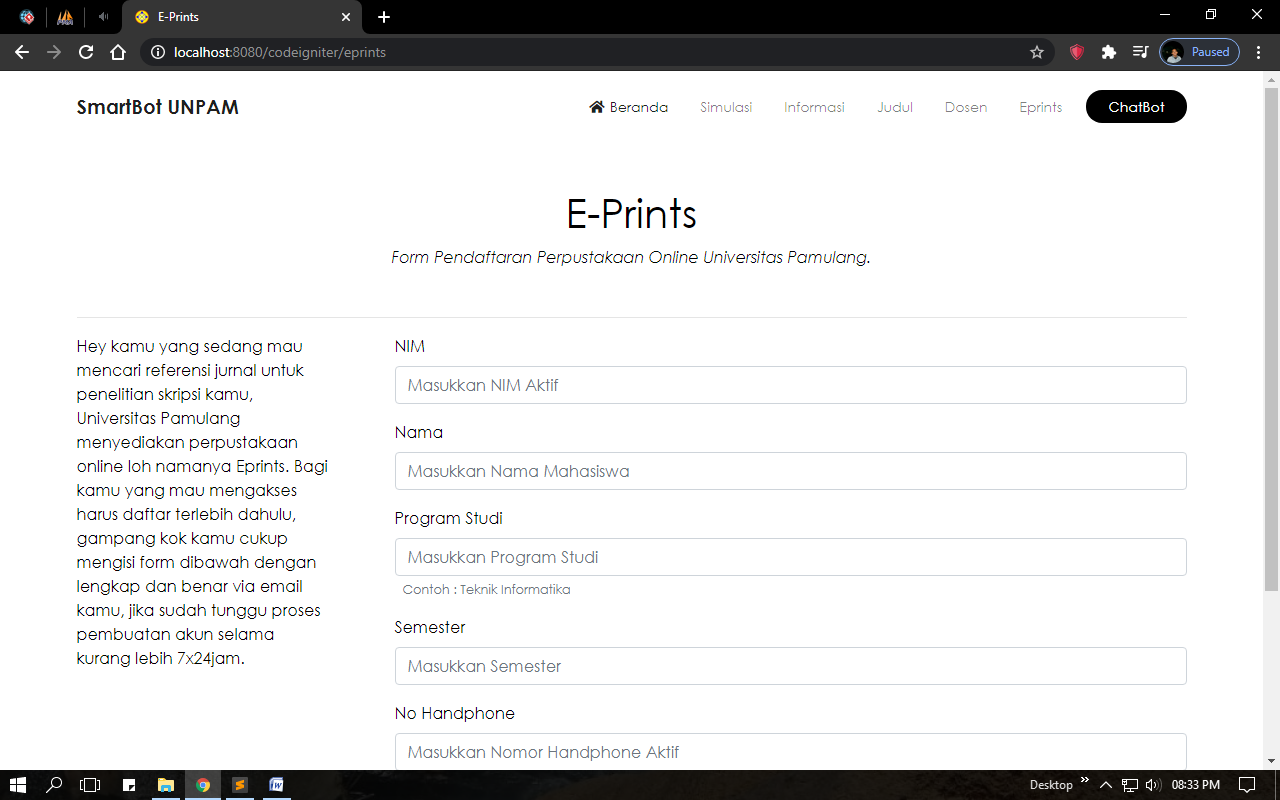
Halaman dosen yaitu halaman yang terdapat nama-nama dosen beserta kontak jurusan teknik informatika yang ada di universitas pamulang.



Gambar 4. 11 Tampilan Halaman Dosen

* + 1. **Halaman Eprints**

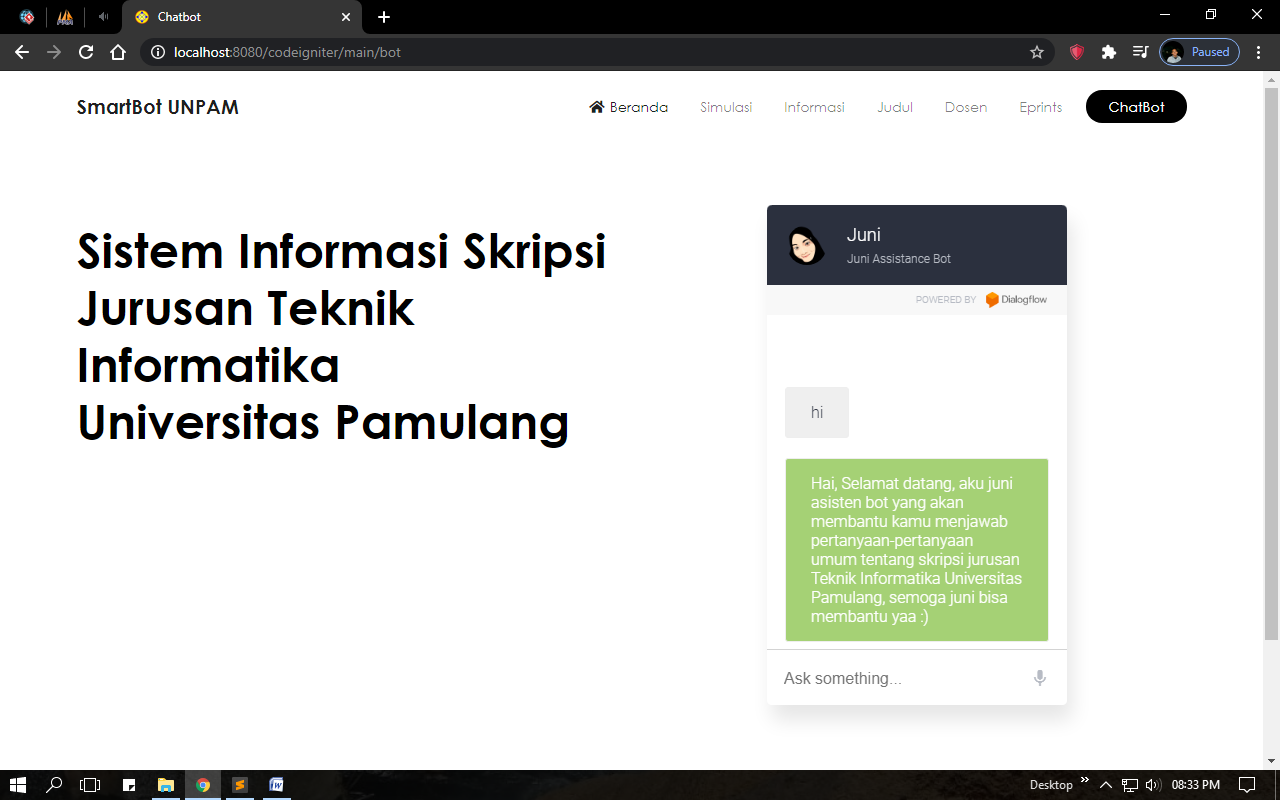
Halaman eprints ini adalah halaman form pendaftaran eprints untuk mendapatkan akses login ke website eprints universitas pamulang.



Gambar 4. 12Tampilan Halaman Eprints

* + 1. **Halaman Chatbot**

Halaman ini hanya terdapat frame chatbot dimana pengguna dapat mengajukan pertanyaan ke smartbot tersebut.



Gambar 4. 13Tampilan Halaman Chatbot

* 1. **Pengujian**

Pengujian merupakan bagian yang terpenting dalam siklus pembangunan perangkat lunak. Pengujian dilakukan untuk menjamin kualitas dan mengetahui kelemahan dari perangkat lunak. Tujua dari pengujian ini adalah untuk menjamin bahwa perangkat lunak yang dibangun memiliki kualitas yang handal, yaitu mampu merepresentasikan kajian pokok dari spesifikasi, analisa, perancangan dan pengkodean dari perangkat lunak itu sendiri.

* + 1. **Pengujian Black Box**

Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Pengujian sistem informasi skripsi ini menggunakan data uji berupa data input dari administrator pada sistem yang telah dibuat.

1. *Black Box* Tampilan *Login* Admin

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Halaman** | ***Test Case*** | **Hasil Yang Diharapkan** | **Hasil** | **Keterangan** |
| 1 | Halaman *Login* | Isi *Username* Admin dan *Password* yang sudah didaftarkan sebelumnya | Akan Berhasil dan akan muncul Halaman *Dashboard* Admin | Sukses | *Valid* |
| 2 | Halaman *Login* | Isi *Username* Admin dan *Password* yang belum didaftarkan sebelumnya | Jika Gagal tidak akan masuk ke Halaman *Dashboard* dan tetap berada di Halaman *Login* | Sukses | *Valid* |
| 3 | Halaman *Login* | Klik Tombol *Logout* | Akan keluar dari halaman *Dashboard* | Sukses | *Valid* |

Gambar 4. 14 Blackbox Login Admin

1. *Black Box* Halaman Post Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Halaman** | ***Test Case*** | **Hasil Yang Diharapkan** | **Hasil** | **Keterangan** |
| 1 | Halaman *Post* Info | Klik Tombol *Add New Article* | Akan mumcul Modal Form untuk menambahkan Arikel / Informasi | Sukses | *Valid* |
| 2 | Halaman *Post* Info | Isi *Form* Artikel / Informasi untuk menambahkan Informasi | Jika Sukses maka Data Informasi akan bertambah | Sukses | *Valid* |
| 3 | Halaman *Post* Info | Klik Tombol Edit pada salah satu Data | Jika Sukses maka akan ke Halaman *Form* Edit Info | Sukses | *Valid* |
| 4 | Halaman *Post* Info | Ubah Data Info | Jika Sukses maka akan kembali ke Halaman Post Info dan Data akan ter*update* | Sukses | *Valid* |
| 5 | Halaman *Post* Info | Klik Tombol *Delete* pada salah satu data | Jika sukses maka akan menghapus data yang dipilih | Sukses | *Valid* |

Gambar 4. 15 Blackbox Post Info

1. *Black Box* Halaman Post Dosen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Halaman** | ***Test Case*** | **Hasil Yang Diharapkan** | **Hasil** | **Keterangan** |
| 1 | Halaman *Post* Dosen | Klik Tombol *Add New* Dosen | Akan mumcul modal *Form* untuk menambahkan Dosen | Sukses | *Valid* |
| 2 | Halaman *Post* Dosen | Isi Form Dosen untuk menambahkan Dosen | Jika sukses maka Data Dosen akan bertambah | Sukses | *Valid* |
| 3 | Halaman *Post* Dosen | Klik Tombol *Edit* pada salah satu data | Jika sukses maka akan ke Halaman *Form Edit* Dosen | Sukses | *Valid* |
| 4 | Halaman *Post* Dosen | Ubah Data info | Jika sukses maka akan kembali ke Halaman *Post* Dosen dan data akan ter *update* | Sukses | *Valid* |
| 5 | Halaman *Post* Dosen | Klik Tombol *Delete* pada salah satu Data | Jika sukses maka akan menghapus data yang dipilih | Sukses | *Valid* |

Gambar 4. 16 Blackbox Post Dosen

1. *Black Box* Halaman Post Eprints

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Halaman** | ***Test Case*** | **Hasil Yang Diharapkan** | **Hasil** | **Keterangan** |
| 1 | Halaman *Post* Eprints | Klik Tombol *Confirm* pada Data yang dipilih | Jika Sukses Data yang dipilih akan terkonfirmasi dan terhapus | Sukses | *Valid* |

Gambar 4. 17 Blackbox Post Eprints

1. *Black Box* Halaman Judul

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Halaman** | ***Test Case*** | **Hasil Yang Diharapkan** | **Hasil** | **Keterangan** |
| 1 | Halaman *Post* Judul | Input Judul Skripsi yang ingin dicari pada form pencarian | Jika Sukses Data yang dicari akan ditampilkan | Sukses | *Valid* |
| 2 | Halaman *Post* Judul | Input Judul Skripsi yang ingin dicari pada form pencarian | Jika Gagal akan menampilkan pemberitahuan tidak adanya data yang dicari | Sukses | *Valid* |

Gambar 4. 18 Blackbox Judul

1. *Black Box* Halaman Utama

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Halaman** | ***Test Case*** | **Hasil Yang Diharapkan** | **Hasil** | **Keterangan** |
| 1 | Halaman Utama | Klik Tombol *Chatbot* | Akan masuk ke Halaman *Chatbot* | Sukses | *Valid* |

Gambar 4. 19 Blackbox Utama

1. *Black Box* Halaman Info Dosen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Halaman** | ***Test Case*** | **Hasil Yang Diharapkan** | **Hasil** | **Keterangan** |
| 1 | Halaman Info Dosen | Isi nama Dosen pada *Form* pencarian | Jika Berhasil maka akan meampilkan data dosen | Sukses | *Valid* |
| 2 | Halaman Info Dosen | Isi nama Dosen yang tidak terdaftar pada *Form* pencarian | Jika Gagal maka tidak ada data Dosen yang muncul | Sukses | *Valid* |

Gambar 4. 20 Blackbox Info Dosen

1. *Black Box* Halaman Pendaftaran Eprints

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Halaman** | ***Test Case*** | **Hasil Yang Diharapkan** | **Hasil** | **Keterangan** |
| 1 | Halaman *Eprints* | Isi data diri lengkap dan benar pada *Form* pendaftaran | Jika Berhasill maka akan muncul *Alert* Sukses | Sukses | *Valid* |
| 2 | Halaman *Eprints* | Isi data diri belum lengkap dan benar pada *Form* pendaftaran | Jika Gagal maka akan muncul Alert Gagal | Sukses | *Valid* |

Gambar 4. 21 Blackbox Form Eprint