**PERANCANGAN SISTEM PENGELOLAAN PRAKTEK KERJA INDUSTRI DI SMK IKA KARTIKA**

**LAPORAN KERJA PRAKTIK**

Dibuat untuk menyajikan hasil penalaran dari proses pengalaman kerja

**Oleh :**

**Mohamad Iqbal (1406082)**

**Turit Hanafi (1506160)**



**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TINGGI TEKNOLOGI GARUT**

**2019**

**PERANCANGAN SISTEM PENGELOLAAN PRAKTEK KERJA INDUSTRI DI SMK IKA KARTIKA**

**LAPORAN KERJA PRAKTIK**

Dibuat untuk menyajikan hasil penalaran dari proses pengalaman kerja

**Oleh :**

**Mohamad Iqbal (1406082)**

**Turit Hanafi (1506160)**



**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TINGGI TEKNOLOGI GARUT**

**2019**

PEDOMAN PENGGUNAAN LAPORAN

Laporan kerja praktik tersedia untuk umum di Perpustakaan Sekolah Tinggi Teknologi Garut. Hak cipta ada pada kelompok kerja yang dialihkan seluruh hak dan kepentingannya kepada Prodi Informatika Sekolah Tinggi Teknologi Garut. Setiap pengutipan harus menyertakan sitasi yang dapat ditelusuri di dalam daftar pustaka.

PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini :

Nama : Turit Hanafi

NIM : 1506160

Adalah wakil kelompok kerja, dengan ini menyatakan bahwa laporan kerja praktik yang dibuat belum pernah diajukan oleh siapapun, serta mengandung kutipan yang telah dilengkapi dengan sitasi dan tercantum dalam daftar pustaka secara memadai. Kami bersedia menerima sangsi akademik berupa nilai E apabila terbukti melakukan plagiasi, sesuai Peraturan Menteri Pendidikan Nasional Republik Indonesia Nomor 17 Tahun 2010.

Garut, ....................................

Yang Membuat Pernyataan

Turit Hanafi

1506160

LEMBAR PENGESAHAN

**PENGEMBANGAN SISTEM INFORMASI MANAJEMEN DAN AKUNTANSI ASET INSTANSI BERBASIS WEB**

**LAPORAN KERJA PRAKTEK**

**Disusun oleh:**

**Mohamad Iqbal (1406082)**

**Turit Hanafi (1506160)**

Disetujui oleh:

|  |  |
| --- | --- |
| Pembimbing Akademik  Ridwan Setiawan, S.T, M.T  NIDN: 0414128703 | Pembimbing Lapangan  Agus Hermawan, S.T, M.M.Pd  NIP: |
|  |  |
|  | |

Mengetahui,

Ketua Program Studi Teknik Informatika,

Dede Kurniadi, S.Kom.,M.Kom

NIDN: 0402098301

ABSTRAK

Sekolah Menengah Kejuruan adalah bentuk jenjang pendidikan formal yang merupakan lanjutan dari jenjang SMP/MTs, Sekolah Menengah Kejuruan memiliki karakteristik yang berfokus pada pembelajaran kejuruan sesuai program studi yang diambil. ciri khas dari Sekolah Menengah Kejuruan adalah memiliki kegiatan yang merupakan proses pembelajaran yang bernama Praktek Kerja Industri yang mengacu pada Undang- undang no 20 tahun 2003, tentang sistem pendidikan nasional yang berisi bahwa pendidikan adalah usaha sadar dan terencana untuk mewujudkan suasana belajar dan proses pembelajaran agar peserta didik secara aktif mengembangkan potensi dirinya untuk memiliki kekuatan spiritual keagamaan, pengendalian diri, kepribadian, kecerdasan, akhlak mulia, serta keterampilan yang diperlukan dirinya, masyarakat, bangsa dan negara. Pada Proses kegiatan Praktek Kerja Industri diperlukan sebuah panitia atau tim untuk melaporkan segala macam bentuk pengelolaan administrasi praktek kerja industri di bawah tanggung jawab kepala lembaga, maka dari itu diperlukan sebuah sistem informasi pengelolaan PRAKTEK KERJA INDUSTRI yang bertujuan untuk mempermudah panitia atau tim dalam proses pengelolaan kegiatan PRAKTEK KERJA INDUSTRI di SEKOLAH MENENGAH KEJURUAN. Sistem Informasi Pengelolaan PRAKTEK KERJA INDUSTRI yang menjadi hasil atau produk Kerja Praktek penulis memiliki fitur yang cukup atau dapat diandalkan dalam proses pengelolaan kegiatan, sistem mencakup proses penginputan peserta didik yang dilengkapi dengan proses impor data, penginputan data DU/DI (Dunia Usaha / Dunia Industri), alokasi penempatan tempat kerja, monitoring / kunjungan hingga penerapan Sistem Informasi Geografis yang dibangun menggunakan Google Maps API (Application Programming Interface).

**Kata Kunci :** Sekolah Menengah Kejuruan, Praktek Kerja Industri, Sistem Informasi Pengelolaa Praktek Kerja Industri, Sistem Informasi Geografis, *Google Maps API*

KATA PENGANTAR

*Assalamu’alaikum Wr. Wb.*

Ucapan rasa syukur tidak pernah lupa kami ucapkan kepada Allah SWT yang telah menganugerahkan segala nikmat dan karunianya, atas izin-Nya lah sampai saat ini penulis dapat menyelesaikan laporan kerja praktik ini dalam keadaan sehat walafiyat. Sholawat teriring salam tetap tak lupa kami haturkan kepada sang junjungan baginda Nabi Besar Muhammad SAW yang telah membawa umat manusia dari alam kejahiliyaan menuju alam yang terang benderang.

Tulisan ini disusun dengan harapan dapat memberikan laporan mnenai kegiatan-kegiatan yang telah dikerjakan pada kerja praktik, adapun dalam kerja praktik ini kami mengambil judul “***PERANCANGAN SISTEM PENGELOLAAN PRAKTEK KERJA INDUSTRI DI SMK IKA KARTIKA”***

Dalam penyusunan laporan ini penulis begitu banyak mendapatkan dorongan, bantuan dan bimbingan dariberbagai pihak, oleh karenanya penulis mengucapkan terima kasih yang sebanyak-banyaknya kepada yang terhormat :

1. Kedua orang tua tercinta serta seluruh anggota keluarga yang tiada hentinya memberika arahan dan bimbingan serta motivasi selama penyusunan laporan kerja parktik ini.
2. Bapak Ridwan Setiawan, S.T, M.T selaku pembimbing Akademik yang telah memberikan arahan dan masukan sehingga penulis dapat menyelesaikan laporan kerja praktik ini.
3. Bapak Agus Hermawan, S.T, M.M.Pd selaku pembimbing lapangan
4. Sahabat Yang senantiasa membagi ilmunya serta waktunya.
5. Serta semua pihak dengan tidak mengurangi rasa hormat, yang tidak dapat penulis sebutkan satu persatu yang telah memberikan dukungan bagi penulis selama ini.

Sepenuhnya penulis sangat menyadari bahwa dalam penulisan ini masih sangat banyak kesalahan dan kekurangan. Oleh sebab itu kami selaku penulis mohon maaf atas segala kekurangan yang ada di dalam penulisan ini kami selaku penulis menerima segala bentuk saran dan kritik dengan sangat lapang dada.

Akhir kata penulis ucapkan semoga laporan ini dapat bermanfaat khususnya bagi penulis serta umumnya bagi yang membaca.

*Wassalamu’alaikum Wr. Wb.*

Garut, 8 Oktober 2018

Penyusun

DAFTAR ISI

[ABSTRAK i](#_Toc526798116)

[KATA PENGANTAR ii](#_Toc526798117)

[DAFTAR ISI iv](#_Toc526798118)

[DAFTAR GAMBAR v](#_Toc526798119)

[DAFTAR TABEL vi](#_Toc526798120)

[DAFTAR LAMPIRAN vii](#_Toc526798121)

[1.1. Latar Belakang 1](#_Toc526798122)

[1.2. Tujuan Pekerjaan 1](#_Toc526798123)

[1.3. Ruang Lingkup Pekerjaan 2](#_Toc526798124)

[1.4. Tempat dan Waktu Kerja 2](#_Toc526798125)

[1.5. Sistematika Penulisan 3](#_Toc526798126)

[2. LANDASAN TEORI 4](#_Toc526798127)

[3. METODOLOGI PEKERJAAN 39](#_Toc526798128)

[4. HASIL DAN PEMBAHASAN 41](#_Toc526798129)

[4.1. Hasil Pekerjaan **Error! Bookmark not defined.**](#_Toc526798130)

[4.2. Pembahasan **Error! Bookmark not defined.**](#_Toc526798131)

[5. KESIMPULAN DAN SARAN 42](#_Toc526798132)

[5.1. Kesimpulan 42](#_Toc526798133)

[5.2. Saran 42](#_Toc526798134)

[DAFTAR PUSTAKA 43](#_Toc526798135)

DAFTAR GAMBAR

[Gambar 2.1. Logo Sekolah Tinggi Teknologi Garut 37](#_Toc522570688)

[Gambar 3.1. Contoh *Work Breakdown Structure* 39](#_Toc522570689)

[Gambar 3.2. Contoh Diagram Alur Aktivitas 39](#_Toc522570690)

DAFTAR TABEL

[Tabel 2.1. Contoh Tabel 37](#_Toc522570662)

[Tabel 3.1. Sumber Daya Penelitian 40](#_Toc522570663)

DAFTAR LAMPIRAN

[LAMPIRAN A : LANDASAN PEKERJAAN 45](#_Toc526798159)

[LAMPIRAN B : PRESENSI KERJA 46](#_Toc526798160)

[LAMPIRAN C : HASIL PEKERJAAN 47](#_Toc526798161)

[LAMPIRAN D : LAMPIRAN LAIN 48](#_Toc526798162)

**1. PENDAHULUAN**

* 1. Latar Belakang

Sekolah Menengah Kejuruan (SMK) Ika Kartika adalah Sekolah Menengah Kejuruan yang berkonsentrasi pada bidang keahlian teknologi dan rekayasa serta Program Keahlian Teknik Otomotif dan Teknik Komputer Informatika. SMK Ika Kartika mengusung konsep Pembelajaran Berbasis *IT*, Kurikulum Nasional/Kurikulum 2013 serta berkreasi dalam Unit Produksi, menjadikan 3 pondasi dasar yang dapat mewujudkan sekolah unggulan memiliki tamatan yang berkualitas serta menghasilkan tenaga profesional berkompoten dan mandiri. (Admin, 2019).

Permasalahan di SMK Ika Kartika saat ini berada pada sistem pengelolaan administrasi Praktek Kerja Industri (PRAKERIN) yang kurang memadai, maka dari itu kami sebagai peserta praktek kerja berperan sebagai pengelola dan pengembangan Unit Sistem Informasi sekolah yang menjadi bagian dari pusat ICT (*Information and Communication Technology*) satuan pendidikan secara lokal dan mandiri yang terbagi ke dalam bidang IT bagian *Software Enginer* yang bertugas di bidang pembuatan program Aplikasi Pengelolaan Praktek Kerja Industri dan IT bagian Jaringan Komputer yang bertugas di bidang pengelola jaringan komputer.

* 1. Tujuan Pekerjaan

Tujuan dari pekerjaan ini terbagi menjadi beberapa bagian, yaitu :

1. Membangun Sistem Informasi Praktek Kerja Industri berbasis Daring Perangkat Lunak (*Web Based Software*) yang menerapkan Sistem Informasi Geografis didalamnya, mengingat bahwa pada kegiatan Praktek Kerja Industri sangat diperlukan data penempatan yang akurat dan pemetaan secara baik.
2. Membantu dalam pengelolaan server secara daring agar aplikasi dapat di buka oleh komputer lain melalui akses jaringan.
   1. Ruang Lingkup Pekerjaan

Proses perancangan sistem pada pekerjaan yang diberikan oleh instansi dibagi menjadi 2 kelompok bagian kerja yang dimana peserta pertama merancang arsitektur teknologi bahasa program dan *tools* untuk membangun perangkat lunak, sedangkan peserta kedua merancang arsitektur perangkat keras dan jaringan yang kompatibel untuk menjalankan Sistem Informasi Praktek Kerja Industri, mengingat bahwa sistem yang dibangun mengusung konsep server dan klien.

Spesifikasi yang digunakan pada proses pembuatan sistem adalah perangkat komputer portable (Laptop) dengan spesifikasi prosesor intel core i5 generasi 5 dan penyimpanan sementara (RAM) yang berkapasitas 12 *gigabyte*.

Sedangkan cakupan pada sistem yang dibuat adalah sebagai berikut;

* Sistem dibangun menggunakan bahasa program khusus pengembangan *platform web* yaitu bahasa pemograman PHP menggunakan framework Laravel
* Sistem yang berjalan hanya digunakan untuk jenjang menengah kejuruan
* Metodologi pembuatan sistem yang digunakan adalah Agile Methodology dengan pendekatan Scrum dan dengan menggunakan pemodelan diagram Unified Modelling Language (Use case diagram, Activity diagram dan Class diagram).
  1. Tempat dan Waktu Kerja

Tempat kerja yang dilaksanakan beralamat di Jl. H. Abdurahman No. 40 Ds. Tanggulun Kec. Kadungora Kab. Garut Prov. Jawa Barat 44153. Dilakukan dari mulai tanggal 12 September 2019 sampai dengan 12 Nopember 2019. Sedangkan waktu kerja disesuaikan berdasarkan hasil kesepakatan antara pembimbing lapangan dan mahasiswa yaitu 3 hari kerja dari pukul 08:00 s.d 16:00 WIB. Kebijakan yang diterapkan di SMK IKA KARTIKA khusus di bagian Unit Sistem Informasi yaitu mahasiswa harus memperhatikan jam masuk serta K3 (Kesehatan dan Keselamatan Kerja) mengingat ruang Unit Sistem Informasi menyatu dengan ruang laboratorium hardware komputer dan server.

1.5. Sistematika Penulisan

Laporan kerja praktik ini disusun dengan sistematika sebagai berikut :

1. **PENDAHULUAN**, berisi latar belakang, tujuan pekerjaan, ruang lingkup pekerjaan, tempat dan waktu kerja, serta sistematika penulisan;
2. **LANDASAN TEORI**, berisi kutipan konsep, teori dan metode terkait tujuan pekerjaan yang bersumber dari sejumlah pustaka;
3. **METODOLOGI PEKERJAAN**, berisi penjelasan tentang tahapan kerja berikut aktivitas dan teknik yang digunakan, serta pelaksana pekerjaan, waktu, tempat, dan sumber daya yang digunakan, dengan memperhatikan landasan teori;
4. **HASIL DAN PEMBAHASAN**, berisi penjelasan tentang hasil pekerjaan, pengetahuan dan keterampilan empiris yang diperoleh, serta bagaimana kearifan lokal diperhatikan di dalam pekerjaan dengan merujuk kepada pustaka;
5. **KESIMPULAN DAN SARAN,** berisi ringkasan penelitian berdasarkan hasil dan pembahasan, serta rekomendasi pekerjaan terkait kelemahankemampuan mahasiswa atau hambatan di tempat kerja.
6. LANDASAN TEORI
   1. Profil Sekolah
   2. Sistem Informasi

Sistem merupakan objek dari desain tertentu, secara garis besar, sistem melibatkan pengorganisasian dari berbagai hal, logical dan fisikal. Sistem meliputi data, proses, kebijakan, protocol, keahlian, perangkat keras, perangkat lunak, tanggung jawab dan komponen lain yang menentukan kemampuan dari suatu organisasi (University of Georgia, 2007). Sedangkan informasi adalah data yang telah di olah sehingga menjadi sebuah informasi dan komunikasi yang bermanfaat bagi yang membutuhkan (THAKUR). Dari penjelasan tentang sistem dan informasi tersebut, dapat disimpulkan bahwa sistem informasi adalah penunjang berbagai kebutuhan terkait data dan informasi, yang mampu disajikan dalam model pengorganisasian terpadu, sehingga aktifitas pengelolaan kebutuhan menjadi lebih efektif dan efisien.

* 1. Prakerin

Prakerin (Praktek Kerja Industri) ialah suatu kegiatan pendidikan yang wajib di ikuti untuk siswa/siswi SMK, pelatihan dan pembelajaran yang di laksanakan di dunia usaha dan dunia industri dalam upaya pendekatan ataupun untuk meningkatkan mutu siswa-siswi SMK dengan kompetensi/kemampuan siswa sesuai bidangnya. Dalam pelaksanaanya di lakukan dengan prosedur tertentu, bagi siswa yang bertujuan untuk magang di suatu tempat kerja baik di dunia usaha ataupun dunia industri setidaknya sudah mempunyai kemampuan dasar sesuai dengan bidang yang di gelutinya atau sudah mendapatkan bekal dari pembimbing di sekolah untuk mempunyai ilmu – ilmu dasar yang akan di terapkan dalam dunia usaha atau dunia industri. Pelaksanaan Prakerin di SMK dilandasi oleh KEPMEN Pendidikan Dan Kebudayaan No.323/u/1997, Tentang UU No.20 tahun 2003 Tentang sistem pendidikan Nasional : pendidikan adalah usaha dasar terencana untuk mewujudkan suasana belajar dan proses pembelajaran agar peserta didik secara aktif mengembangkan potensi diri nya untuk mempunyai kekuatan spiritual keagamaan, pengendalian diri kepribadian, kecerdasan, akhlak mulia, dan keterampilan yang diperlukan diri nya, masyarakat Bangsa Dan Negara, dan penyelengaaraan Prakerin SMK

* 1. Agile Scrum

Agile Scrum adalah metode atau *project management* yang gesit dan cerdas, menunjukan kesederhanaan serta familiar. Dengan *Agile Scrum*, seorang *Project Manager* dapat menciptakan sebuah daftar fitur prioritas dengan jaminan pengembangan produk yang sukses. Dipandu dengan jaminan kualitas produk (*Product-Backlog*), seorang *Project Manager* diharuskan untuk selalu mengerjakan prioritas terpenting atau tertinggi terlebih dahulu. Agile Scrum memiliki karakteristik unik dalam proses pengembangan seperti contoh jika tim pengembang mengalami kehabisan sumber daya misalkan dari segi waktu, pekerjaan apapun yang tidak selesai akan menjadi prioritas lebih rendah dari pekerjaan yang selesai, Agile Scrum tidak hanya memberi intruksi, tetapi memberikan kesempatan kepada tim pengembang untuk berinspirasi dan berinovasi dalam proses pengembangan produk. (Rubin, 2012) (Edwards, Bickerstaff, & Bartsch, 2017)

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

UML *(Unified Modeling Language)* adalah tujuan umum, perkembangan, bahasa pemodelan dibidang rekayasa perangkat lunak, yang dimaksudkan untuk menyediakan cara standar untuk memvisualisasikan desain sistem. Menurut Nugroho (2010:6), UML (Unified Modeling Language) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek”. Pemodelan *(modelling)* sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. UML awalnya termotivasi oleh keinginan untuk membakukan sistem notasi yang berbeda dan pendekatan untuk desain perangkat lunak yang dikembangkan oleh Grady Booch , Ivar Jacobson dan James Rumbaugh Rational Software ditahun 1994-1995, dengan pengembangan lebih lanjut yang dipimpin oleh mereka melalui tahun 1996.

Adapun jenis UML *(Unified Modeling Language)* terdiri dari 13 bagian, antara lain :

* + 1. Use Case Diagram

*Use case* merupakan rangkaian/uraian sekelompokkyang saling terkait dan membentuk sistemmsecara teratur yang dilakukan atau diawasi oleh sebuah aktor. *Use case* digunakanmuntuk membentuk tingkahlakubbenda dalam sebuah mode serta direalisasikan olehssebuahh*collaborator*, umumnya *use case* digambarkan dengan sebuah *elips* dengan garis yang solid, biasanyaamengandunggnama. *Use case* menggambarkan proses sistem (kebutuhan sistem dari sudut pandang *user*). Secaraaumum *use case* merupakan polaaperilaku sistemgdan urutan transaksi yanggberhubungan yang dilakukan oleh aktor. *Use case* diagram terdiri dari: *use case*, aktor, *relationship*, sistem *boundary boxes, packages* (Ambler, 2019). Berikut adalah simbol-simbol *use case* diagram:

**Tabel 2.5.1 Simbol- simbol *useecaseddiagram***

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| 1 | 1234074-200 | Aktor | Menspesifikasikan himpunan pesan yang pengguna mainkan ketika berinteraksi |
| 2 | index | *Include* | Menspesifikasikan *use case* secara eksplisit |
| 3 |  | *Association* | Apa yanggmenghubungkannantara objek satu dengannobjek yang lainnya |
| 4 |  | *System* | Menspesifikasikangpaket yang menampilkanssistem-sistem secara terbatas |

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | main-qimg-af9553d254d703613b62550cd758dd40 | *Use case* | Deskripsi dariuurutan aksi-aksi yang ditampilkan sistemyyang menghasilkan suatu hasil yang terukurbbagi suatu actor |
| 6 |  | *Collaboration* | Interaksi aturan-aturan dan elemennlain yang bekerja sama untukkmenyediakan perilaku yangllebih besar dari jumlah dan elemen-elemennya (sinergi). |

Sumber: (Ambler, 2019)

* + 1. Activity Diagram

*Activity* diagram adalah lebihffokus kepada menggambarkanpproses bisnis dan urutan aktivitasddalam sebuah proses. Dipakai pada *businessmmodeling* untuk memperlihatkan urutan aktifitaspproses bisnis. Memiliki struktur diagram yang mirip *flowchart* atau data *flow* diagram pada perancangannterstruktur. Memiliki pula manfaat yaitu apabila penulis membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan (Ambler, 2019). Berikut adalah simbol-simbol *activity* diagram:

**Tabel 2.5.2 Simbol-simbol *activity* diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| 1 |  | *Activity* | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain. |
| 2 |  | *Action* | State dari sistem yang mencerminkan eksekusi dari suatu aksi |
| 3 |  | *Initial Node* | Bagaimana objek dibentu atau di awali. |
| 4 |  | *Activity Final Node* | Bagaimanaoobjekkdibentuk atau di akhiri. |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| 5 |  | *fork Node* | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |

Sumber: (Ambler, 2019)

* + 1. Class Diagram

*Class* diagramhmerupakan hubungan antarkkelas dan penjelasan detailttiap-tiap kelas di dalamgmodel desain dari suatu sistem, jugammemperlihatkan aturan-aturan dan tanggung jawabbentitas yang menentukannperilaku sistem. *Class* diagram juga menunjukkan atribut-atributddan operasi-operasi dari sebuahkkelas dan *constraint* yangbberhubungan dengan objek yang dikoneksikan. *Class* diagram secara khas meliputi: *class, relasi assosiations, generalitation* dan *aggregation, attributes, operation/method* dan *visibility*, tingkat aksessobjek eksternal kepadassuatu operasi atau atribut. Hubungan antarkkelas mempunyai keterangan yang disebut dengan *multiplicity* atau *cardinality*.

**Tabel 2.5.3 Simbol-simbol *class* diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| 1 |  | *Package* | *Package* merupakan sebuah bungkusan dari suatu atau lebih kelas |
| 2 |  | *Class* | *Class* pada struktur sistem |
| 3 |  | *Interface* | Sama dengan konsep *interface* dalam pemrograman berorientasi objek |
| 4 |  | *Association* | Relasi antar kelasddengan makna umum, asosiasi biasanyajjuga disertai dengan *multiplicity* |
| 5 |  | *Directed Association* | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, |
| **No** | **Gambar** | **Nama** | **Keterangan** |
|  |  |  | asosiasi biasanya juga disertai dengan *multiplicity* |
| 6 |  | *Generalization* | Relasi antar kelas dengan makna generalisai-spesialisasi (umum khusus) |
| 7 |  | *Aggregation* | Relasi antarkkelas dengannmakna semua bagian (*whole-part*) |
| 8 |  | *Depedency* | Relasi antar kelas dengan makna kebergantungan antar kelas |

Sumber: (Ambler, 2019)

* + 1. Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dengan massage yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefinisikan interaksi jalannya pesan sudah dicakup dalam diagram sekuen sehingga semakin banyak use case yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| 1 |  | *Object* | Object merupakan instance dari sebuah class dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah class (kotak) dengan nama obyek  didalamnya yang diawali dengan sebuah titik koma |
| 2 |  | *Actor* | Actor juga dapat berkomunikasi dengan object, maka actor juga dapat diurutkan sebagai kolom. Simbol Actor sama dengan simbol pada Actor Use Case Diagram. |
| 3 |  | *Lifeline* | Lifeline mengindikasikan keberadaan sebuah object dalam basis waktu. Notasi untuk Lifeline adalah garis putus-putus vertikal yang ditarik dari sebuah obyek. |
| 4 |  | *Activation* | Activation dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah lifeline. Activation mengindikasikan sebuah obyek yang akan melakukan sebuah aksi. |
| 5 |  | *Message* | Message, digambarkan dengan anak panah horizontal antara Activation. Message mengindikasikan komunikasi antara object-object. |

* + 1. Component Diagram

Komponen perangkat lunak adalah bagian fisik dari sebuah sistem yang menetap di komputer. komponen merupakan implementasi software dari sebuah class. Komponen bisa berupa tabel, file data, file exe, file DLL, dokumen, dan lain-lain Diagram component mengandung komponen, interface dan relationship. Komponen diagram ini digunakan pada saat pengembang ingin memecah sistem menjadi komponen-komponen dan ingin menampilkan hubungan-hubungan mereka dengan antarmuka atau pemecahan komponen menjadi struktur yang lebih rendah. Secara umum dapat disimpulkan bahwa component diagram yang digunakan untuk menjelaskan kebergantungan antar beragam komponenkomponen software seperti misalnya kebergantungan antara file-file executable dengan file-file sumbernya *(source file).*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| 1 |  | *Komponen* | Sebuah komponen melambangkan sebuah entitas software dalam sebuah sistem. Sebuah komponen dinotasikan sebagai sebuah kotak segiempat dengan dua kotak kecil tambahan yang menempel disebelah  kirinya. |
| 2 |  | *Actor* | Actor juga dapat berkomunikasi dengan object, maka actor juga dapat diurutkan sebagai kolom. Simbol Actor sama dengan simbol pada Actor Use Case Diagram. |

* + 1. Deployment Diagram

Deployment diagram adalah salah satu model diagram dalam UML untuk mengerahkan artifact dalam node. Deployment diagram digunakan untuk memvisualisasikan, menspesifikasikan, dan mendokumentasikan proses yang terjadi pada suatu sistem perangkat lunak berbasis Object Oriented yang akan dibangun. Tujuan atau fungsi dari deployment diagram yaitu untuk menggambarkan/memvisualisasikan secara umum proses yang terjadi pada suatu sistem/software.

Node dalam UML merepresentasikan hardaware atau software execution environments. Node bisa terhubung melalui communication path untuk membuat system jaringan dari arbitrary complexity. Artifacts dalam UML mempresentasikan Spesifikasi dari bentuk physic informasi yang digunakan atau dihasilkan development process.

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| 1 |  | *Komponen* | Pada deployment diagram, komponenkomponen yang ada diletakkan didalam node untuk memastikan keberadaan posisi mereka |
| 2 |  | *Node* | Node menggambarkan bagian-bagian hardware dalam sebuah sistem. Notasi untuk node digambarkan sebagai sebuah kubus 3 dimensi. |
| 3 |  | *Association* | Sebuah association digambarkan sebagai sebuah garis yang menghubungkan dua node yang mengindikasikan jalur komunikasi antara element-elemen hardware. |

* + 1. State Diagram

Diagram status atau state diagram atau statechart diagram menunjukkan kondisi yang dapat dialami atau terjadi pada sebuah objek sehingga setiap objek memiliki sebuah diagram status. Diagram status diadopsi dari penggambaran kondisi mesin status (state machine) yang menggambarkan status apa saja yang dialami oleh mesin, misalnya mesin pembelian kopi dengan uang koin.

Diagram Status mengambarkan seluruh state/status yang memungkinkan obyek-obyek dalam class dapat dimiliki dan kejadian-kejadian yang menyebabkan satus berubah. Perubahan dalam suatu state disebut juga transisi (transition). Suatu transisi juga dapat memiliki sebuah aksi yang dihubungkan pada status, lebih spesifik apa yang harus dilakukan dalam hubungannya dengan transisi status. Pada diagram ini, perilaku sistem ditunjukkan. Sebuah status adalah kondisi selama hidup objek atau interaksi selama memenuhi suatu kondisi, melaksanakan suatu aksi, atau menunggu suatu kejadaian.

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *State* | Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek. |
| 2 |  | *Initial Pseudo State* | Bagaimana objek dibentuk atau diawali |
| 3 |  | *Final State* | Bagaimana objek dibentuk dan dihancurkan |
| 4 |  | *Transition* | Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya |
| 5 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 6 |  | *Node* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi. |

* + 1. Composite Diagram

Diagram komposit menggambarkan struktur internal dari pengklasifikasi terstruktur dengan menggunakan bagian, pelabuhan, dan konektor. Sebuah penggolong terstruktur mendefinisikan pelaksanaan penggolongan dan dapat mencakup kelas, komponen, atau penempatan node. Anda dapat menggunakan diagram struktur komposit untuk menunjukkan rincian internal penggolongan dan untuk menggambarkan objek dan peran yang bekerja sama untuk melakukan perilaku yang mengandung penggolongan.

Sebuah diagram struktur komposit mirip dengan class diagram, tetapi menggambarkan bagian-bagian individu, bukan seluruh kelas. Sebelum Anda dapat menentukan struktur internal penggolongan, Anda juga harus menunjukkan wadah struktur atau membuka sebuah diagram struktur komposit. Anda kemudian dapat memodelkan bagian-bagian yang mewakili hal yang mengandung penggolongan . Anda dapat menambahkan konektor untuk menghubungkan dua atau lebih bagian dalam hubungan asosiasi atau ketergantungan.

|  |  |  |
| --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** |
| 1 |  | *Structured Classifier* |
| 2 |  | *Encapsulated Structured* |
| 3 |  | *Property* |
| 4 |  | *Part* |
| 5 |  | *Service Port* |
| 6 |  | *Behavior Port* |

* + 1. Object Diagram

Diagram Objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. dalam diagram objek harus di pastikan bahwa semua kelas yang sudah didefinisikan pada diagram kelas harus dipakai objeknya, karena jika tidak pendefinisian kelas itu tidak dapat dipertanggung jawabkan. sebuah diagram objek merupakan gambaran objek-objek pada sebuah sistem pada satu titik waktu. karena lebih menonjolkan printah-printah dari pada kelas daigram objek sering disebut juga sebagai diagram perintah. elemen-elemen sebuah diagram objek adalah spesifikasi perintah.

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *Objek* | Objek dari kelas saat system sedang dijalankan. |
| 2 |  | *Link* | Relasi antar objek |

* + 1. Package Diagram

Diagram paket merupakan salah satu jenis UML yang digunakan untuk mengelompokkan elemen-elemen model dari use case ataupun class diagram. biasanya package diagram digunakan pada kumpulan sistem yang besar. atau package diagram disebut sekelompok elemen-elemen model. sebuah paket dapat berisi elemen-elemen model yang berlainan, termasuk paket-paket untuk menciptakan/menggambarkan sifat hiraki. sebuah paket diberi nama yang menggambarkan isinya.

Packages digambarkan sebagai sebuah direktori (file folders) yang berisi model – model elemenPackage merupakan kumpulan atau pengelompokan class – class yang memiliki sifat sama. Penggambaran diagram Package mirip dengan simbol folder dalam Microsoft Windows. Salah satu manfaat package adalah kemampuannya untuk digunakan pada component lainnya.

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *Package* | Sekelompok elemen-elemen model |
| 2 |  | *Import* | Suatu dependency yang mengidikasikan isi tujuan paket secara umum yang ditambahkan ke dalam sumber paket |
| 3 |  | *Access* | Suatu dependency yang mengidikasikan isi tujuan paket secara umum yang bias digunakan pada nama sumber paket |

* + 1. Communication/Collaboration Diagram

Model Komunikasi Diagram interaksi antara benda atau bagian dalam hal mengurutkan. Diagram komunikasi merupakan kombinasi dari informasi yang diambil dari Kelas, urutan, dan Use case diagram menggambarkan baik struktur statis dan perilaku dinamis dari suatu sistem. Namun, diagram komunikasi menggunakan pengaturan bebas-berupa objek dan link seperti yang digunakan dalam diagram Object. Dalam rangka mempertahankan Urutan pesan dalam suatu diagram format bebas, pesan diberi label dengan nomor kronologis dan ditempatkan di dekat link pesan yang dikirim. Membaca diagram komunikasi yaitu mulai dari pesan 1.0, dan mengikuti pesan dari objek ke objek.

Diagram komunikasi menunjukkan banyak informasi yang sama seperti diagram urutan, tetapi karena cara informasi tersebut disajikan, beberapa di antaranya lebih mudah untuk mencari salah satu diagram daripada yang lain. Diagram komunikasi menunjukkan tiap elemen berinteraksi dengan baik, namun diagram urutan menunjukkan urutan di mana interaksi berlangsung lebih jelas.

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **SIMBOL** | **NAMA** | **KETERANGAN** |
| 1 |  | **Object** | Object merupakan instance dari sebuah class dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah class (kotak) dengan nama obyek  didalamnya yang diawali dengan sebuah titik koma |
| 2 |  | **Actor** | Actor juga dapat berkomunikasi dengan object, maka actor juga dapat diurutkan sebagai kolom. Simbol Actor sama dengan simbol pada Actor Use Case Diagram. |
| 3 |  | **Message** | Message, digambarkan dengan anak panah yang  mengarah antar obyek dan diberi label urutan  nomor yang mengindikasikan urutan  komunikasi yang terjadi antar obyek. |

* + 1. *Interaction Overview* Diagram

Interaction Overview Diagram/Diagram Interaksi digunakan untuk memodelkan interaksi objek di dalam sebuah proses. diagram interaksi memperlihatkan interaksi yang memuat himpunan dari objek dan relasi yang terjadi antar objek tersebut, termasuk juga bagaimana pesan mengalir diantar objek interaction overview diagram ini sangat mirip dengan diagram aktivitas. Sementara diagram aktivitas menunjukkan urutan proses interaction overview diagram menunjukkan urutan diagram interaksi. Dalam istilah sederhana mereka bisa disebut kumpulan diagram interaksi dan urutan mereka terjadi

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *Decisiom* | Decision adalah unsur suatu diagram yang menunjukkan titik perkembangan syarat. |
| 2 |  | *Initial* | Mendefinisikan awal aliran ketika sebuah kegiatan dipanggil |
| 3 |  | *Final State* | Menujukkan oenyelesaian sebuah kegiatan. |
| 4 |  | *Activity* | Menunjukkan aktifitas yang sedang berlangsung |
| 5 |  | *Control Flow* | Konektor yang menghubungkan dua node dalam sebuah diagram activity. |

* + 1. Timing Diagram

Timing Diagram digunakan untuk menyelidiki tingkah laku objek sepanjang periode waktu tertentu. Sebuah timing diagram merupakan berntuk khusus dari sequence diagram. Perbedaan antara timing diagram dan diagram sequence adalah sumbu dibalik sehingga waktu meningkat dari kiri ke kanan dan jalur hidup yang ditampilkan dalam ruang terpisah yang diatur secara vertikal.

Timing diagram UML digunakan untuk menampilkan perubahan dalam keadaan atau nilai dari satu atau lebih elemen dari waktu ke waktu. Hal ini juga dapat menunjukkan interaksi antara peristiwa waktunya dan kendala waktu dan durasi yang mengaturnya.

* 1. **Tools yang digunakan**

Adapun beberapa tools yang digunakan dalam peracncangan system aplikasi ini, antara lain :

* + 1. ***PHP (Hypertext Preprocessor)***

PHP adalah bahasa program di sisi server paling populer yang cocok digunakan untuk pengembangan web, memiliki kemampuan yang cepat, fleksibel dan fragmatis (Lerdorf)

* + 1. ***Laravel***

Dikarenakan PHP memiliki fleksibilitas pada pusat inti bahasa program maka memungkinkan bagi seorang pemogram untuk dapat membuat kerangka kerja tersendiri untuk mempermudah proses pengembangan. Seperti halnya Laravel yang yang merupakan sebuah kerangka kerja dibawah Bahasa Program PHP, *Laravel* sendiri adalah kerangka kerja untuk aplikasi berbasis web dengan sintaksis yang ekspresif dan elegan, laravel memilik fondasi dasar dan canggih yang mampu mempermudah pemogram dan mempercepat proses pengembangan aplikasi berbasis web (Otwell, 2019)

* + 1. **MySql**

MySQL adalah tool yang digunakan khusus untuk mengolah SQL *(Structured Query Language)*. SQL sendiri merupakan sebuah bahasa yang digunakan untuk mengakses baris data relasi. Mudahnya adalah untuk mengakses bahasa dalam komputer. Karena SQL dan MySQL sifatnya khusus, maka hanya orang-orang yang berkecimpung dalam dunia IT lah yang familiar dengan bahasa ini.

Jika SQL adalah bahasa yang digunakan untuk mengakses data dalam database, maka MySQL adalah tool atau software atau alat yang digunakan untuk mengolah SQL. SQL adalah bahasanya, MySQL adalah alatnya. Yang perlu dipahami, MySQl bukanlah alat satu-satunya yang bisa mengolah SQL. Masih ada banyak alat lain. Sebut saja Microsoft Acces. Keduanya hanya berbeda pada cara penggunaan dan pengolahan databasenya saja.

* + 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 pemrogramman 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*. Biasanya bagi bahasa pemrograman yang didukung ataupun belum terdukung secara default dapat lebih dimaksimalkan atau didukung dengan menggunakan add-ons yang bisa didownload sesuai kebutuhan user.

* + 1. **Browser**

Browser adalah salah satu jenis perangkat lunak *(software)* yang umumnya digunakan untuk membuka halaman website di internet. Browser disebut juga dengan peramban web dan web browser. Secara fundamental browser mempunyai kemampuan untuk menampilkan kode semantik (bahasa pemrograman) halaman website seperti; *HTML, CSS, Js,* dan lainnya menjadi halaman yang dimengerti oleh semua orang. Browser yang populer digunakan saat ini adalah *Google Chrome* dan *Mozilla Firefox*.

* + 1. **Argo UML**

ArgoUML merupakan software UML modeling tool open source dan menyediakan dukungan untuk semua standar diagram UML *(Unified Modeling Language).* Kode-kode program ArgoUML ditulis sepenuhnya dengan menggunakan bahasa pemrograman Java dan menggunakan Java Foundation Classess. Selain itu, ArgoUML juga kompatibel dengan standar OMG untuk UML 1.4. Software UML modeling tool ini berjalan pada semua platform Java dan tersedia dalam berbagai bahasa (sekitar 10 bahasa).

*Computer Aided Software Engineering (Case)* mengotomatisasikan proses pembuatan software dari depan ke belakang. Sedangkan *Unified Modeling Language (UML)* menyediakan interoperabilitas *(interoperability)* yang memungkinkan mendapat manfaat dari CASE. ArgoUML merupakan suatu UML CASE tool untuk menganalisa dan merancang software berorientasi objek *(object oriented software)*. Selain merupakan software yang bersifat open source, ArgoUML memiliki banyak aspek unik yang membedakannya dari tool-tool komersial, termasuk desain yang dapat meningkatkan produktivitas programmer serta dikembangkan pada sifat alami bahasa pemrograman Java yang memungkinkan akses dan kompatibilitas yang luas.

1. **METODOLOGI PEKERJAAN**

**3.1** ***Work Breakdown Structure***

Proses tahapan kerja dalam perancangan aplikasi yang dilakukan selama kerja praktek digambarkan dengan *work breakdown structure* sebagai tujuan untuk menjelaskan tahapan-tahapan pekerjaan yang disesuaikan dengan *scrum*, serta aktivitas yang akan dilakukan pada setiap tahapan-tahapan tersebut yang disajikan dalam skema WBS. Berikut gambar tahapan dari *Work Breakdown Structure*.



**Gambar 3.1.1 *Work Breakdown Structure***

Pada tahapan pertama dalam WBS ini adalah *product* *backlog* pada tahapan ini mempunyai aktivitas membuat proses bisnis, yang diperoleh dari dokumen, hasil wawancara, buku referensi dan jurnal peneliti sebelumnya, lalu aktifitas selanjutnya yaitu identifikasi aktor yang akan digambarkan dengan *use case* diagram, aktivitas selanjutnya yaitu identifikasi aktivitas bisnis yang digambarkan dengan *activity* diagram, aktivitas selanjutnya yaitu membuat struktur sistem yang akan digambarkan dengan *class* diagram, dan mengidentifikasi kebutuhan sistem yang akan digambarkan dengan tabel *backlog item*.

Pada tahapan *sprint* yaitu merancang proses kegiatan yang akan dilakukan untuk membuat sistem dari *backlog* *item* yang dikembangkan dengan membuat tabel perencanaan. Selanjutnya dilakukan tahapan *daily scrum* pada tahapan ini menghasilkan tabel evaluasi hasil dari *sprint* yang telah diselesaikan, Selanjutnya tahapan *sprint review* di mana tahapan ini merupakan tahapan terakhir yaitu mengimplementasikan hasil dari sistem yang telah dibuat.

Berikut merupakan diagram alur aktivitas yang menggambarkan dari aktifitas yang dilakukan pada pembuatan aplikasi, sebagaimana disajikan pada Gambar 3.3



Gambar 3.1.3. Diagram Alur Aktivitas

Pada alur diagram aktivitas ini, menggambarkan langkah-langkah yang akan dibahas pada bab selanjutnya. Langkah awal diagram aktivitas ini adalah *product backlog*, yang mengidentifikasi komponen-komponen yang berkaitan dengan penelitian, dengan melakukan observasi dan studi literatur untuk proses kegiatan yang dilakukan. Selanjutnya tahap *sprint*, pada tahap ini melakukan analisis lebih lanjut dari tahapan *product* *backlog* pada tahap *sprint* ini dilakukan *sprint planning* dan *sprint backlog.* Tahap selanjutnya *daily scrum*, pada tahapan ini dilakukan *meeting* untuk mengevaluasi hasil kerja yang telah dilakukan. Dan tahap terakhir yaitu *sprint review*, yaitu pada tahap ini dilakukan *meeting* untuk menampilkan hasil pekerjaan yang telah selesai.

Adapun aktivitas yang dilaksanakan serta perangkat yang digunakan pada saat penelitian ini disajikan pada Tabel 3.1 sumber daya penelitian sebagai berikut:

Tabel 3.1.2. Sumber Daya Penelitian

|  |  |  |  |
| --- | --- | --- | --- |
| No | Aktivitas | Manusia | *Tools* |
| 1. | Membuat proses bisnis | Peneliti | Wawancara, Jurnal, *e-book*, |
| 2. | Identifikasi aktor | Peneliti | *Microsoft Visio* |
| 3. | Identifikasi aktivitas bisnis | Peneliti | *Microsoft Visio* |
| 4. | Membuat struktur sistem | Peneliti | *Microsoft Visio* |
| 5. | Identifikasi kebutuhan sistem | Peneliti | *Microsoft Visio* |
| 6. | *Sprint planning* | Peneliti | *Microsoft Word* |
| 7. | *Sprint backlog* | Peneliti | *Microsoft Word* |
| 8.  9. | *Daily scrum meeting*  *Sprint review meeting* | Peneliti  Peneliti | *Microsoft Word*  *PHP Laravel Framework* |

1. **HASIL PEKERJAAN DAN PEMBAHASAN**

**4.1 Hasil Pekerjaan**

Permasalahan pada sistem pengelolaan praktek kerja industri di SMK IKA KARTIKA secara konvensional diantaranya administrasi awal seperti pembuatan surat permohonan prakerin ke instansi masih harus membuat secara pada umumnya menggunakan aplikasi Ms. Word, pengelolaan data tempat instansi masih melihat pada riwayat pengeluaran surat permohonan sehingga waktu yang dibutuhkan tidak terlalu cepat serta lokasi penempatan belum dapat dipastikan akurat karena pengelamatan belum menggunakan koordinat.

Pada proses pembuatan sistem ini menggunakan *Agile Methodology*dengan pendekatan *Scrum.* Selain itu, pemodelan yang akan digambarkan pada pembuatan sistem ini antara lain mencakup *use case* diagram, *activity* diagram, *class* diagram dan tabel perencanaan.

* + 1. *Product* *Backlog*

Product backlog merupakan tahapan pertama, di mana pada tahapan ini menghasilkan beberapa aktivitas di antaranya: Identifikasi Spesifikasi Kebutuhan sistem dengan cara membuat identifikasi kebutuhan sistem yang akan disajikan dalam bentuk kebutuhan funcitional sistem dan non funcitional sistem, menganalisa proses bisnis, identifikasi aktor, identifikasi aktivitas bisnis, dan menentukan daftar product Backlog yang disepakati dengan para pemangku keputusan (Product Owner):

* + - 1. . Spesifiksi Kebutuhan Sistem

1. Spesifikasi Fungsional Sistem

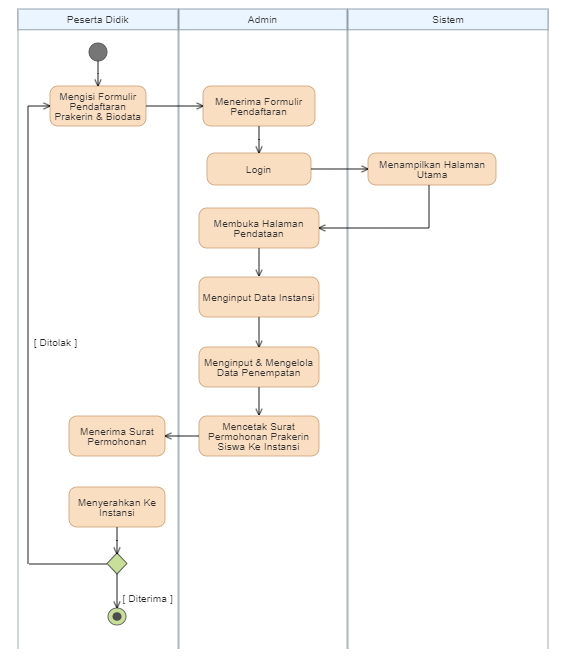
Berdasarkan hasil pekerjaan yang sudah dijelaskan di atas, maka Sistem Informasi Prakerin ini memiliki fungsional sistem sebagai berikut:

1. Deskripsi Sistem : Sistem Informasi Prakerin ini merupakan aplikasi yang berfungsi sebagai pengelola data pada ruang lingkup kegiatan prakerin peserta didik di SMK IKA KARTIKA.
2. Fungsional Sistem : berdasarkan kepada deskripsi sistem pada point sebelumnya, maka sistem yang dibuat memiliki spesifikasi fungsional sebagai berikut:
3. Sistem dapat mengelola data instansi beserta bidangnya
4. Sistem dapat mengelola peserta didik hingga menjadi angota peserta prakerin
5. Sistem dapat mengelola lokasi tempat prakerin peserta didik
6. Sistem dapat mencetak kebutuhan laporan yang dibutuhkan
7. Sistem dapat mengelola administrasi kegiatan monitoring prakerin
8. Spesifikasi Non Fungsional Sistem

Berdasarkan kebutuhan sebelumnya, maka dapat digambarkan spesifikasi non fungsional sistem sebagai berikut:

1. Sistem yang dibangun berbasis platform web
2. Sistem menggunakan bahasa pemograman PHP dengan *Framework Laravel*
3. Sistem menggunakan basis data manajemen MySQL
4. Sistem mampu saling terhubung antar jaringan lokal maupun online
   * + 1. Proses Bisnis

Proses bisnis yang dilakukan yaitu membuat rancangan yang dibangun dengan menggambarkan aktivitas yang akan dilaksanakan oleh administrator dan siswa. Berikut ini rancangan diagram aktivitas mengenai proses bisnis.



Gambar 4.1.1.2. Proses bisnis yang akan dibangun

Berikut deskripsi proses bisnis sistem berdasarkan diagram diatas:

Tabel 4.1.1.2. Penjelasan proses bisnis yang akan dibangun

| No | Aktifitas |
| --- | --- |
| 1 | Peserta Didik mengisi formulir pendaftaran prakerin dan biodata |
| 2 | Admin menerima formlir pendaftaran dari peserta didik |
| 3 | Admin melakukan login kedalam sistem / aplikasi |
| 4 | Admin membuka halaman pendaftaran |
| 5 | Admin membuka melakukan pengelolaan data instansi |
| 6 | Admin melakukan pengelolaan data penempatan |
| 7 | Admin mencetak surat permohonan prakerin ke instansi untuk peserta didik |
| 8 | Peserta didik menerima surat permohonan dari admin |
| 9 | Peserta didik menyerahkan surat permohonan kepada instansi tempat kerja |
| 10 | Jika Peserta Didik diterima oleh instansi tempat kerja, maka Peserta Didik diterima dan menunggu hingga waktu keberangkatan prakerin tiba, tetapi jika tidak diterima instansi, Peserta Didik kembali mengisi formulir pendaftaran dengan instansi yang baru. |

* + - 1. Identifikasi Aktor

Identifikasi aktor bertujuan untuk mengidentifikasi target yang akan terlibat pada sistem ini. Adapun aktor yang teridentifikasi adalah sebagai berikut:

1. Administrator adalah aktor yang bertugas sebagai pengelola sistem dan seluruh modul dalam aplikasi.
2. Peserta Didik adalah aktor yang menjadi peserta prakerin, peserta mengisi biodata lengkap serta mengisi formulir tempat kerja untuk menjadi bahan inputan oleh administrator.
3. Tenaga Pendidik adalah aktor yang bertugas sebagai pembimbing dan pemonitoring peserta prakerin.

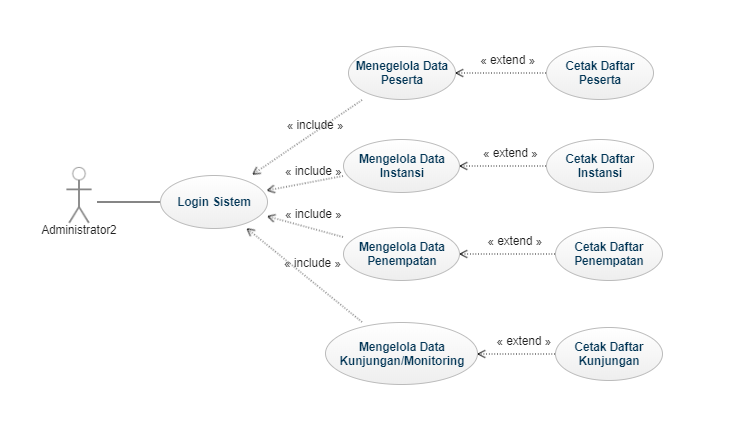
Adapun aktifitas yang dilakukan oleh aktor akan dijelaskan pada tabel 4.2 berikut ini:

Tabel 1. 4.1.1.3 Aktifitas Aktor

|  |  |  |
| --- | --- | --- |
| No | Aktor | Aktivitas |
| 1 | Administrator | 1. Mebuka Aplikasi 2. Login Kedalam Aplikasi 3. Mengelola Data Peserta 4. Mengelola Data Instansi 5. Mengelola Data Instansi 6. Mengelola Data Penempatan 7. Mengelola Data Kunjungan |
| 2 | Peserta Didik | 1. Mengisi Formulir Biodata 2. Mengisi Formulir Tempat Kerja 3. Menerima Surat Permohonan Kerja |

|  |  |  |
| --- | --- | --- |
| No | Aktor | Aktivitas |
| 3 | Tenaga Pendidik | 1. Menjadi Pembimbing Akademik 2. Administrasi Kunjungan Prakerin |

Adapun *use case* diagram untuk menggambarkan susunan serta mendefinisikan interaksi peran aktor, berikut merupakan peran aktor yang digambarkan dengan *use case* diagram:



Gambar 4.1.1.3. *Use Case* Diagram

Adapun narasi dan scenario *use case* diagram sebagai berikut:

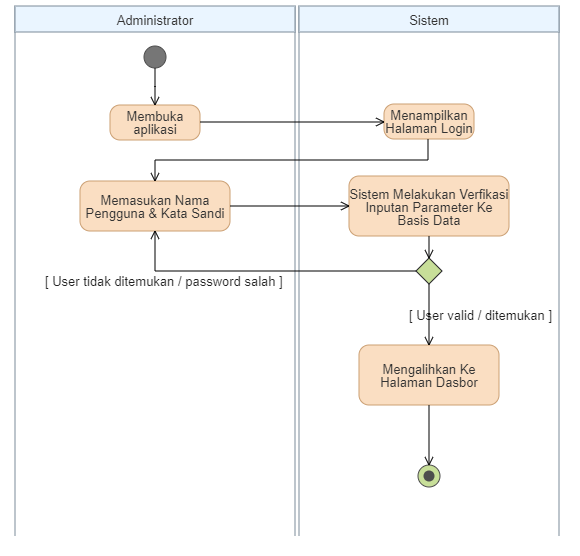
|  |  |
| --- | --- |
| Administrator | Sistem |
| 1. Administrator login kedalam sistem | 1. Sistem merespon tindakan login dan mengalihkan halaman ke halaman dasbor |
| 1. Administrator menginput data dan mengelola peserta prakerin | 1. Sistem melakukan penyimpanan dan pembaruan ke entitas |
| 1. Admistrator menginput dan mengelola data instansi | 1. peserta didik dalam basis data |

|  |  |
| --- | --- |
| Administrator | Sistem |
| 1. Adminstrator menginput dan mengelola data penempatan | 1. Sistem melalukan penyimpanan dan pembaruan ke entitas instansi dalam basis data |
| 1. Mengelola data kunjungan monitoring | 1. Sistem melakukan penyimpanan dan pembaruan ke entitas penempatan dalam basis data |

Tabel 2 4.1.1.3 Skenario *Use Case* Diagram

* + - 1. Identifikasi Aktifitas Sistem

Identifikasi aktifitas sistem merupakan aktifitas dari setiap modul yang dijalankan oleh setiap aktor, identifikasi aktifitas sistem digambarkan melalui Aktifitas Diagram yang beberbentuk flowchart alur aktifitas yang akan disajikan pada berikut ini:



Gambar 4.1.1.4 Aktifitas Diagram *Login* administrator

Berikut ini merupakan penjelasan dari *Activity Diagram login* adminstrator4.1.1.4 pada Sistem Pengelolaan Praktek Kerja Industri yang akan disajikan pada tabel 4.1.1.4

Tabel 4.1.1.4 Deskripsi *Activity Diagram Login* Administrator

|  |  |  |
| --- | --- | --- |
| No. | Aktifitas | Deskripsi |
| 1 | Membuka aplikasi | Administrator membuka aplikasi Sistem Pengelolaan Prakerin |
| 2 | Melakukan Autentikasi | Setelah sistem menampilkan halaman login, administrator menginputkan data nama pengguna dan kata sandi untuk dapat mengakses sistem secara penuh |
| 3 | Validasi autentikasi | Sistem melakukan verifikasi dan validasi pengguna terdaftar pada basis data. |
| 4 | Peringatan parameter inputan tidak valid | Jika hasil verfikasi dan validasi administrator tidak valid atau data administrator tidak terdaftar pada basis data maka sistem akan menampilkan pesan error bahwa proses autentikasi tidak valid |
| 5 | Parameter inputan valid / ditemukan | Jika hasil verfikasi dan validasi administrator dinyatakan valid atau data administrator tersedia pada basis data maka sistem akan mengalihkan halaman ke halaman dasbor |

* + - 1. Struktur Sistem

Setelah mengidentifikasi aktor dan aktifitas aktor maka selanjutnya membuat struktur sistem yang akan disajikan dengan menggunakan struktur menu, *class* diagram, dan *layout* antarmuka pengguna pada gambar berikut ini:

a). Struktur Menu

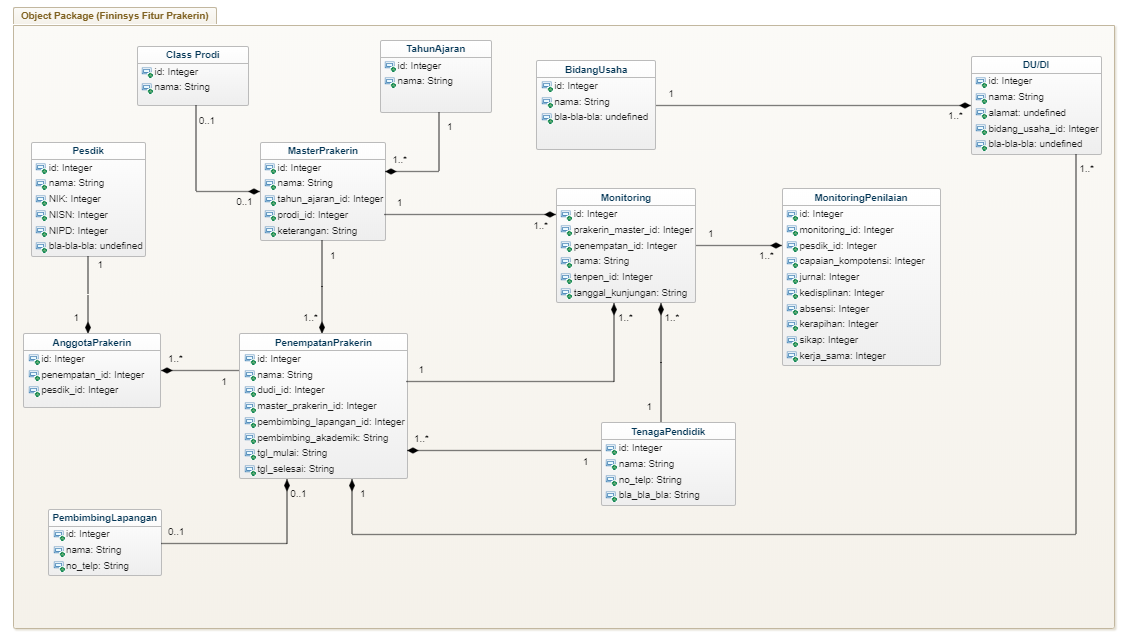


Adapun susunan menu setelah administrator login adalah sebagai berikut:

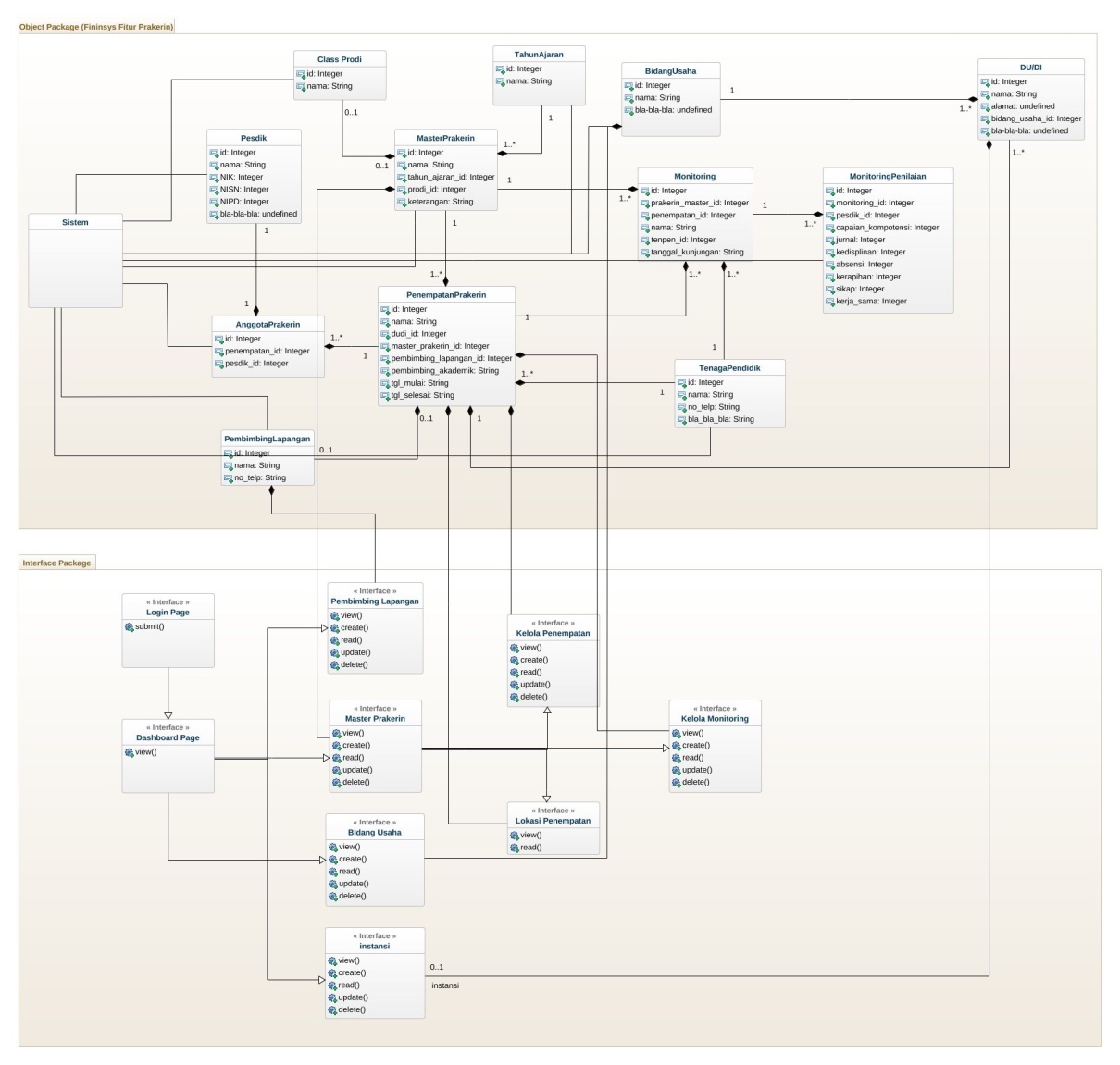
**Gambar a) Struktur Menu Aplikasi**

Gambar diatas merupakan struktur menu yang ada pada halaman administrator pada aplikasi yang hanya dapat di akses oleh administrator.

b). Class Diagram



**Gambar 1 b) *Class Diagram System Package***



**Gambar 2 b) *Class Diagram Relasi Interface Package***

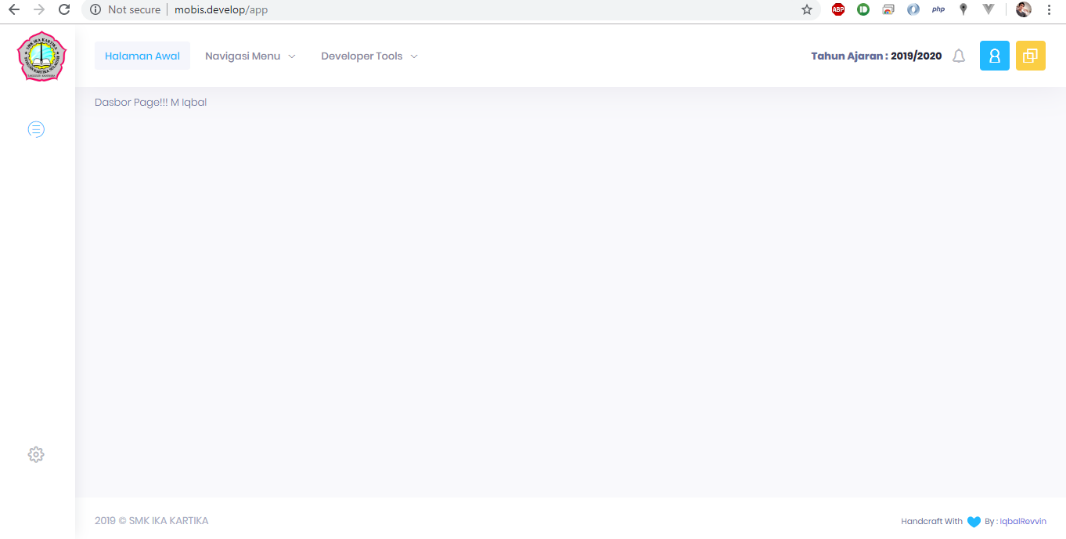
Adapun penjelasan class diagram yang akan disajikan pada tabel 4.8 berikut ini:

Tabel b) Deskripsi *Class Diagram*

|  |  |  |
| --- | --- | --- |
| **No.** | **Nama Kelas** | **Penjelasan** |
| 1 | Kelas Prodi | Kelas prodi menampilkan data program studi yang berfungsi sebagai pelengkap entitas peserta didik |
| 2 | Kelas Tahun Ajaran | Kelas tahun ajaran menampilkan tahun ajaran serta sebagai riwayat data dan identifikasi tahun ajaran yang sedang berjalan |
| 3 | Kelas Peserta Didik | Kelas pesdik merupakan data dari peserta didik yang menjadi peserta prakerin |
| 4 | Kelas Tenaga Pendidik | Kelas Tenaga Pendidik merupakan data dari tenaga pendidik yang menjadi pembimbing akademik prakerin |
| 5 | Kelas Bidang Usaha | Kelas Bidang Usaha merupakan data jenis bidang usaha sebagai pelengkap DU/DI |
| 6 | Kelas DU/DI | Kelas DU/DI adalah data instansi untuk tempat prakerin peserta |
| 7 | Kelas Master Prakerin | Kelas Master Prakerin berfungsi sebagai master data untuk pengolahan prakerin berkala berdasarkan tahun ajaran yang aktif |
| 8 | Kelas Penempatan Prakerin | Kelas Penempatan Prakerin adalah data relasi yang di ambil dari data instansi dan mengacu pada master prakerin |
| 9 | Kelas Anggota Prakerin | Kelas Anggota Prakerin adalah data yang berelasi dari penempatan dan peserta didik |
| 10 | Kelas Pembimbing Lapangan | Kelas Pembimbing Lapangan adalah data pembimbing yang di ambil dari data tenaga pendidik dan berelasi ke kelas Penempatan Prakerin |
| 11 | Kelas Monitoring | Kelas Monitoring adalah data untuk monitoring atau kunjungan prakerin oleh pembimbing akademik |
| 12 | Kelas Monitoring Penilaian | Kelas Monitoring Penilaian adalah data penilaian dari hasil monitoring |

*c). Layout* Antarmuka

*Layout* antarmuka pengguna merupakan rancangan antarmuka yang akan digunakan sebagai perantara pengguna dengan sistem yang di bangun. *Layout* antarmuka pengguna dari Sistem Pengelolaan Praktek Kerja Industri akan disajikan pada gambar berikut ini:



**Gambar c. *Layout* Antarmuka**

Gambar di atas merupakan layout antarmuka tampilan utama atau rancangan untuk halaman dasbor admin, di mana pada halaman ini dapat di akses oleh administrator.

* + - 1. Identifikasi Kebutuhan Sistem

Identifikasi kebutuhan sistem merupakan tahap yang dilakukan untuk menganilisis kebutuhan sistem, dari sisi kebutuhan program aplikasi yang akan disajikan pada tabel-tabel berikut ini:

Tabel 4.1.1.6. *Backlog Item*

|  |  |
| --- | --- |
| **No.** | ***Backlog Item*** |
| 1 | Sistem mampu menyimpan data peserta didik |
| 2 | Sistem mampu meyimpan data instansi tempat kerja peserta |
| 3 | Sistem memiliki data-data master penunjang untuk data instansi |
| 4 | Sistem mampu mengelola penempatan tempat kerja peserta |
| 5 | Sistem mampu mencetak surat permohonan kerja peserta ke instansi |
| 6 | Sistem mampu mencetak daftar instansi |
| 7 | Sistem mampu menampilkan titik koordinat lokasi tempat kerja |

* + 1. *Sprint*

Sprint merupakan tahap kedua, di mana pada tahap ini memiliki dua aktifitas yaitu:*Sprint planning* dan *sprint backlog* yang akan disajikan berikut ini:

* + - 1. ***Sprint Planning***

Setelah menentukan *product backlog* lalu diadakan pertemuan/rapat oleh *scrum team* di setiap awal *sprint* untuk mengevaluasi *product backlog*, mendiskusikan tujuan dan misi dari setiap fitur, pada bagian ini *scrum team* menentukan waktu yang dibutuhkan tiap anggota untuk melaksanakan setiap fiturnya, yang akan disajikan pada Tabel 4.5 berikut ini:

**Tabel 4.1.2.1 *Sprint Planning***

|  |  |  |
| --- | --- | --- |
| No | Sprint Planning | Estimasi Waktu (Hari) |
| 1 | *Templating/Layouting* | 7 |
| 2 | Halaman Login | 3 |
| 3 | Halaman Utama / Dasbor | 1 |
| 4 | Halaman Pengelolaan Bidang Usaha | 2 |
| 5 | Halaman Pengelolaan Pembimbing Lapangan | 2 |
| 6 | Halaman Pengelolaan Instansi | 5 |
| 7 | Halaman Master Prakerin | 5 |
| 8 | Halaman Pengelolaan Peserta Prakerin | 7 |
| 9 | Halaman Pengaturan Titik Lokasi | 5 |

* + - 1. ***Sprint Backlog***

Pada tahap ini yaitu merancang *sprint backlog* di mana pada tahap ini yaitu membuat kebutuhan yang jadi prioritas pengguna dengan waktu kerja yang telah ditentukan oleh *team*, pada bagian ini yaitu menyelesaikan fitur yang telah didefinisikan pada *product backlog*  dan *sprint planning*, yang akan disajikan pada tabel berikut ini:

**Tabel 4.1.2.2 *Sprint Backlog Sprint* 1- Templating**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint 1** | **Pengerjaan** | **Estimasi (Waktu/Jam)** | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| Templating | Penyesuaian Template *Front End* | 2 |  |  |  |  |  |  |
| Pengaturan Komponen Tata Letak | 6 | 8 | 8 |  |  |  |  |
| Penyesuaian *Stylesheet* |  |  |  | 8 |  |  |  |
| Integrasi *Blade Engine* |  |  |  |  | 8 |  |  |
| Integrasi *Package* |  |  |  |  |  | 8 | 8 |
| **Total Estimasi** | | **56 Jam** | | | | | | |

*Sprint* 1 terdiri dari 5 pekerjaan dengan total estimasi waktu sebanyak 56 jam, pada *sprint* ini adalah *templating* yang berfungsi sebagai kerangka UI & UX awal yang terintegrasi, maka selanjutnya akan dilanjutkan dengan *sprint* 2.

**Tabel 4.1.2.2 *Sprint Backlog Sprint* 2 – Login**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint 2** | **Pengerjaan** | **Estimasi (Waktu/Jam)** | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| Login | UI & UX | 8 |  |  |  |  |  |  |
| Desain *Database* |  | 4 |  |  |  |  |  |
| *Migration Database* |  | 4 |  |  |  |  |  |
| *Logic Codding* |  |  | 6 |  |  |  |  |
| *Testing* |  |  | 2 |  |  |  |  |
| **Total Estimasi** | | **24 Jam** | | | | | | |

*Sprint* 2 terdiri dari 5 pekerjaan dengan total estimasi waktu 3 hari dan 24 jam, *sprint* ini adalah pembuatan halaman login yang berfungsi sebagai jembatan pengguna dengan sistem autentikasi, setelah *sprint* 2 selesai maka akan dilanjutkan ke *sprint* 3.

**Tabel 4.1.2.2 *Sprint Backlog Sprint* 3 – Halaman Utama**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint 3** | **Pengerjaan** | **Estimasi (Waktu/Jam)** | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| Halaman Utama | UI & UX | 3 |  |  |  |  |  |  |
| Desain *Database* |  |  |  |  |  |  |  |
| *Migration Database* |  |  |  |  |  |  |  |
| *Logic Codding* | 2 |  |  |  |  |  |  |
| *Testing* | 1 |  |  |  |  |  |  |
| **Total Estimasi** | | **6 Jam** | | | | | | |

*Sprint* 3 terdiri dari 3 pekerjaan dengan total estimasi waktu sebanyak 6 jam, *sprint* ini membuat tampilan awal untuk administrator dan sebagai halaman penentu untuk menentukan apakah administrator sudah melakukan autentikasi atau belum.

**Tabel 4.1.2.2 *Sprint Backlog Sprint* 4 – Halaman Utama**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint 4** | **Pengerjaan** | **Estimasi (Waktu/Jam)** | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| Bidang Usaha | UI & UX | 3 |  |  |  |  |  |  |
| Desain *Database* | 3 |  |  |  |  |  |  |
| *Migration Database* |  | 1 |  |  |  |  |  |
| *Logic Codding* |  | 3 |  |  |  |  |  |
| *Testing* |  | 2 |  |  |  |  |  |
| **Total Estimasi** | | **12 Jam** | | | | | | |

*Sprint* 4 terdiri dari 5 pekerjaan dengan total estimasi waktu sebanyak 12 jam, *sprint* ini membuat halaman untuk pengelolaan data bidang usaha, setelah *sprint* 3 selesai maka akan dilanjutkan oleh *sprint* 4.

**Tabel 4.1.2.2 *Sprint Backlog Sprint* 5 – Halaman Pembimbing Lapangan**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint 5** | **Pengerjaan** | **Estimasi (Waktu/Jam)** | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| Pembimbing Lapangan | UI & UX | 3 |  |  |  |  |  |  |
| Desain *Database* | 3 |  |  |  |  |  |  |
| *Migration Database* |  | 1 |  |  |  |  |  |
| *Logic Codding* |  | 3 |  |  |  |  |  |
| *Testing* |  | 2 |  |  |  |  |  |
| **Total Estimasi** | | **12 Jam** | | | | | | |

*Sprint* 5 terdiri dari 5 pekerjaan dengan total estimasi waktu sebanyak 12 jam, *sprint* ini membuat halaman untuk pengelolaan data Pembimbing Lapangan, setelah *sprint* 5 selesai maka akan dilanjutkan oleh *sprint* 6.

**Tabel 4.1.2.2 *Sprint Backlog Sprint* 6 – Halaman Pengelolaan Instansi**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint 6** | **Pengerjaan** | **Estimasi (Waktu/Jam)** | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| Pengelolaan Instansi | UI & UX | 8 | 4 |  |  |  |  |  |
| Desain *Database* |  | 4 |  |  |  |  |  |
| *Migration Database* |  |  | 2 |  |  |  |  |
| *Logic Codding* |  |  |  | 8 | 6 |  |  |
| *Testing* |  |  |  |  | 2 |  |  |
| **Total Estimasi** | | **34 Jam** | | | | | | |

*Sprint* 6 terdiri dari 5 pekerjaan dengan total estimasi waktu sebanyak 34 jam, *sprint* ini membuat halaman untuk pengelolaan data instansi serta penentuan lokasi menggunakan *google maps* setelah *sprint* 6 selesai maka akan dilanjutkan oleh *sprint* 7.

**Tabel 4.1.2.2 *Sprint Backlog Sprint* 7 – Halaman Master Prakerin**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint 7** | **Pengerjaan** | **Estimasi (Waktu/Jam)** | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| Master Prakerin | UI & UX | 8 | 4 |  |  |  |  |  |
| Desain *Database* |  | 4 |  |  |  |  |  |
| *Migration Database* |  |  | 2 |  |  |  |  |
| *Logic Codding* |  |  |  | 8 | 6 |  |  |
| *Testing* |  |  |  |  | 2 |  |  |
| **Total Estimasi** | | **34 Jam** | | | | | | |

*Sprint* 7 terdiri dari 5 pekerjaan dengan total estimasi waktu sebanyak 34 jam, *sprint* ini membuat halaman untuk master data dari pengelolaan prakerin secara lanjut*,* setelah *sprint* 7 selesai maka akan dilanjutkan oleh *sprint* 8.

**Tabel 4.1.2.2 *Sprint Backlog Sprint* 8 – Pengelolaan Peserta Prakerin**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint 8** | **Pengerjaan** | **Estimasi (Waktu/Jam)** | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| Pengelolaan Peserta Prakerin | UI & UX | 8 | 4 |  |  |  |  |  |
| Desain *Database* |  | 4 | 8 |  |  |  |  |
| *Migration Database* |  |  |  | 4 |  |  |  |
| *Logic Codding* |  |  |  | 4 | 8 | 8 |  |
| *Testing* |  |  |  |  |  |  | 2 |
| **Total Estimasi** | | **50 Jam** | | | | | | |

*Sprint* 8 terdiri dari 5 pekerjaan dengan total estimasi waktu sebanyak 50 jam, *sprint* ini membuat halaman untuk pengelolaan peserta prakerin termasuk generate surat permohonan, dan alokasi penempatan peserta*,* setelah *sprint* 8 selesai maka akan dilanjutkan oleh *sprint* 9.

**Tabel 4.1.2.2 *Sprint Backlog Sprint* 9 – Pengaturan Titik Lokasi**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint 9** | **Pengerjaan** | **Estimasi (Waktu/Jam)** | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| Pengaturan Titik Lokasi | UI & UX | 8 | 8 |  |  |  |  |  |
| Desain *Database* |  |  | 2 |  |  |  |  |
| *Migration Database* |  |  |  | 2 |  |  |  |
| *Logic Codding* |  |  |  | 6 | 6 |  |  |
| *Testing* |  |  |  |  | 2 |  |  |
| **Total Estimasi** | | **40 Jam** | | | | | | |

*Sprint* 9 terdiri dari 5 pekerjaan dengan total estimasi waktu sebanyak 40 jam, *sprint* ini membuat halaman untuk penentuan titik lokasi penempatan prakerin.

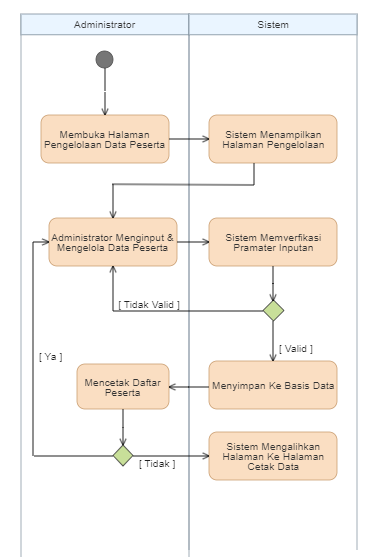
* + 1. *Daily* *Scrum*

*Daily scrum* merupakan tahap ke tiga di mana pada tahap ini menghasilkan aktivitas *daily scrum meeting* disajikan sebagai berikut:

1. *Daily scrum meeting*

Pada tahap ini praktik yang dilakukan untuk memantau kinerja, *meeting* yang dilakukan untuk pelaporan fitur yang sedang dikembangkan, dan pada akhir *meeting* akan dilakukan pembaharuan pada setiap waktu penyelesaian untuk mengetahui sisa pengerjaan yang dilakukan selama proses *sprint* berjalan, yang akan disajikan pada tabel berikut ini:

Lampiran B : Activity Diagram

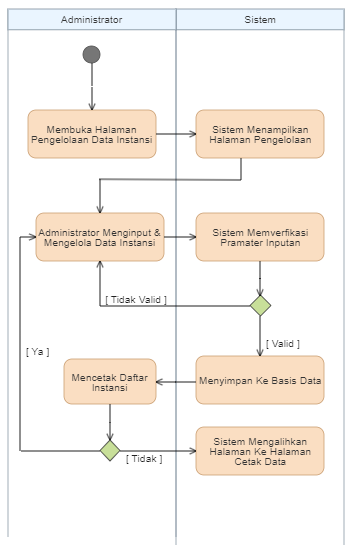


Gambar B.1 Aktifitas Diagram Kelola Data Peserta

Berikut ini merupakan penjelasan dari *Activity Diagram* Pengelolaan Data Pesertapada B.1 Sistem Pengelolaan Praktek Kerja Industri yang akan disajikan pada table B.1

Tabel B.1 Deskripsi *Activity Diagram Login* Administrator

|  |  |  |
| --- | --- | --- |
| No. | Aktifitas | Deskripsi |
| 1 | Membuka Halaman Pengelolaan Data Peserta | Administrator membukan halaman data pengelolaan data peserta |
| 2 | Sistem Menampilkan Halaman Pengelolaan | Sistem menampilkan halaman pengelolaan data peserta |
| 3 | Mengelola data peserta | Administrator menginput dan mengelola data peserta prakerin |
| 4 | Sistem melakukan verifikasi penginputan | Setelah administrator mengirim data inputan, maka sistem melakukan verifikasi data di belakang layar, jika parameter valid maka sistem akan menyimpan data ke database, jika tidak valid maka sistem mengembalikan tampilan ke halaman pegelolaan |
| 5 | Mencetak data peserta | Setalah administrator menginput dan mengelola data peserta, admin dapat mencetak data peserta sebagai bahan laporan |

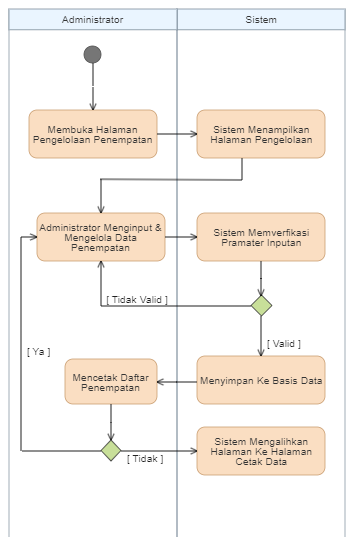


Gambar B.2 Aktifitas Diagram Kelola Data Instansi

Berikut ini merupakan penjelasan dari *Activity Diagram* Pengelolaan Data InstansiB.2 pada Sistem Pengelolaan Praktek Kerja Industri yang akan disajikan pada tabel B.2

Tabel B.2 Deskripsi *Activity Diagram* Data Instansi

|  |  |  |
| --- | --- | --- |
| No. | Aktifitas | Deskripsi |
| 1 | Membuka Halaman Pengelolaan Data Instansi | Administrator membukan halaman data pengelolaan data instansi |
| 2 | Sistem Menampilkan Halaman Pengelolaan | Sistem menampilkan halaman pengelolaan data instansi |
| 3 | Mengelola data instansi | Administrator menginput dan mengelola data instansi |
| 4 | Sistem melakukan verifikasi penginputan | Setelah administrator mengirim data inputan, maka sistem melakukan verifikasi data di belakang layar, jika parameter valid maka sistem akan menyimpan data ke database, jika tidak valid maka sistem mengembalikan tampilan ke halaman pegelolaan |
| 5 | Mencetak data instansi | Setalah administrator menginput dan mengelola data instansi, admin dapat mencetak data instansi sebagai bahan laporan |

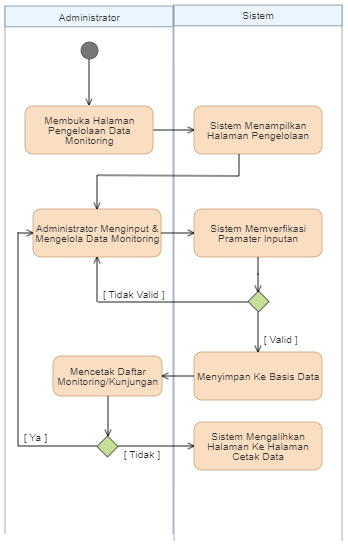


Gambar B.3 Aktifitas Diagram Kelola Data Penempatan

Berikut ini merupakan penjelasan dari *Activity Diagram* Pengelolaan Data PenempatanB.3 pada Sistem Pengelolaan Praktek Kerja Industri yang akan disajikan pada tabel B.3

Tabel B.3 Deskripsi *Activity Diagram* Data Penempatan

|  |  |  |
| --- | --- | --- |
| No. | Aktifitas | Deskripsi |
| 1 | Membuka Halaman Pengelolaan Penempatan | Administrator membukan halaman data pengelolaan data penempatan |
| 2 | Sistem Menampilkan Halaman Pengelolaan | Sistem menampilkan halaman pengelolaan data penempatan |
| 3 | Mengelola data penempatan | Administrator menginput dan mengelola data penempatan |
| 4 | Sistem melakukan verifikasi penginputan | Setelah administrator mengirim data inputan, maka sistem melakukan verifikasi data di belakang layar, jika parameter valid maka sistem akan menyimpan data ke database, jika tidak valid maka sistem mengembalikan tampilan ke halaman pegelolaan |
| 5 | Mencetak data penempatan | Setalah administrator menginput dan mengelola data penempatan, admin dapat mencetak data penempatan sebagai bahan laporan |



Gambar B.4 Aktifitas Diagram Kelola Data Penempatan

Berikut ini merupakan penjelasan dari *Activity Diagram* Pengelolaan Data PenempatanB.4 pada Sistem Pengelolaan Praktek Kerja Industri yang akan disajikan pada tabel B.4

Tabel B.4 Deskripsi *Activity Diagram* Data Penempatan

|  |  |  |
| --- | --- | --- |
| No. | Aktifitas | Deskripsi |
| 1 | Membuka Halaman Pengelolaan Monitoring | Administrator membukan halaman data pengelolaan data monitoring |
| 2 | Sistem Menampilkan Halaman Monitoring | Sistem menampilkan halaman pengelolaan data monitoring |
| 3 | Mengelola data monitoring | Administrator menginput dan mengelola data monitoring |
| 4 | Sistem melakukan verifikasi penginputan | Setelah administrator mengirim data inputan, maka sistem melakukan verifikasi data di belakang layar, jika parameter valid maka sistem akan menyimpan data ke database, jika tidak valid maka sistem mengembalikan tampilan ke halaman pegelolaan |
| 5 | Mencetak data monitoring | Setalah administrator menginput dan mengelola data penempatan, admin dapat mencetak data penempatan sebagai bahan laporan |

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Tuliskan point-point ringkasan penting dari bagian Hasil dan Pembahasan.

5.2. Saran

Tuliskan point-point rekomendasi untuk pekerjaan selanjutnya terkait kelemahandiriatau hambatan di tempat kerja.

DAFTAR PUSTAKA

Bill. (2019, October 24). *What is Google Maps?* Retrieved from GCF Learn Free: https://edu.gcfglobal.org/en/google-maps/what-is-google-maps/1/

Cahyana, R. (2018). *Panduan Penelitian Mahasiswa dan Penyusunan Skripsi.* Garut: Sekolah Tinggi Teknologi Garut.

Catlin, H. (2019, October 24). *Sass Homepage*. Retrieved from Sass Official: https://sass-lang.com/

Dawson, C. W. (2005). *Project on computing and information system : a student’s guide.* England: Pearson Education Limited.

Edwards, L., Bickerstaff, R., & Bartsch, C. (2017). *Contracting for agile sotfware development projects.* London: Bird & Bird.

Emilia, E. (2009). *Menulis Tesis dan Disertasi.* Bandung: Penerbit Alfabeta.

Github. (n.d.). *About project boards*. (Git Version Control) Retrieved October 23, 2019, from https://help.github.com/en/github/managing-your-work-on-github/about-project-boards

Google. (2019, October 24). *About Google Maps*. Retrieved from About Google Maps: https://www.google.com/maps/about/

Kho, B. (2016, October 15). *Pengertian Kanban dan 6 Aturan Utamanya*. (Ilmu Manajemen) Retrieved October 23, 2019, from https://ilmumanajemenindustri.com/pengertian-kanban-dan-6-aturan-utamanya/

Lerdorf, R. (n.d.). *PHP Official*. (Programming) Retrieved October 23, 2019, from https://www.php.net/

Otwell, T. (2019, October 24). *The PHP Framework for Web Artisans*. Retrieved from Laravel Official: https://laravel.com

Rubin, K. S. (2012). *Essential Scrum: A Practical Guide to the Most Popular Agile Process.* Boston: Addison-Wesley Professional.

SMK IKA KARTIKA. (n.d.). *SMK IKA KARTIKA*. (Sekolah Menengah Kejuruan) Retrieved October 21, 2019, from SMK IKA KARTIKA: https://smkikakartika.sch.id

Team, B. (2019, October 24). *Get Bootstrap*. Retrieved from Bootstrap Official: https://getbootstrap.com/

THAKUR, D. (n.d.). *What do you mean by Data and Information ?* (Information Technologi) Retrieved October 24, 2019, from http://ecomputernotes.com/fundamental/information-technology/what-do-you-mean-by-data-and-information

University of Georgia. (2007). *Information System.* Denver: Global Text Project.

LAMPIRAN A : LANDASAN PEKERJAAN

Sajikan bukti surat balasan atau lainnya yang menjadi landasan pekerjaan.

LAMPIRAN B : PRESENSI KERJA

Sajikan semua presensi kerja, termasuk di dalamnya kartu bimbingan lapangan dan akademik.

LAMPIRAN C : HASIL PEKERJAAN

Sajikan semua hasil pekerjaan yang tidak berkaitan dengan pembahasan.

LAMPIRAN D : LAMPIRAN LAIN

Sajikan lampiran lainnya di sini.