

**APLIKASI POJOK KEMAHASISWAAN DI
POLITEKNIK NEGERI INDRAMAYU
BERBASIS WEBSITE**

TUGAS AKHIR

Sebagai salah satu syarat untuk menyelesaikan
Program Diploma III Teknik Informatika
Politeknik Negeri Indramayu



Oleh:

ANNISA KHUSNUL LAILY

NIM 1903062

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI INDRAMAYU
AGUSTUS 2022**

HALAMAN PENGESAHAN

Tugas Akhir ini diajukan oleh:

Nama : ANNISA KHUSNUL LAILY
NIM : 1903062
Program Studi : Diploma III Teknik Informatika
Judul : Aplikasi Pojok Kemahasiswaan Di Politeknik Negeri
Indramayu Berbasis Website
Pembimbing : I. **Munengsih Sari Bunga, S.Kom., M.Eng.**
NIP 198507202019032015
: II. **Iryanto, S.Si., M.Si.**
NIDN 0001089001

Telah berhasil dipertahankan dihadapan dewan penguji pada tanggal dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Ahli Madya Program Studi Diploma III Teknik Informatika, Jurusan Teknik Informatika Politeknik Negeri Indramayu.

DEWAN PENGUJI

Ketua Penguji :
Anggota :
Penguji I
Anggota : **Munengsih Sari Bunga, S.Kom., M.Eng.**
Penguji II NIP 198507202019032015

Indramayu, Agustus 2022
Ketua Jurusan Teknik Informatika

Iryanto, S.Si., M.Si.
NIDN 0001089001

MOTTO

"Jangan ingat lelahnya belajar, tapi ingat buah manis yang bisa dipetik ketika sukses."

PERNYATAAN

Dengan ini saya menyatakan dengan sesungguhnya bahwa Tugas Akhir ini adalah asli hasil karya saya sendiri serta **Tugas Akhir** ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar Ahli Madya di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dirujuk dalam naskah ini dan disebutkan dalam daftar pustaka

Indramayu, Agustus 2022

Yang menyatakan,

Annisa Khusnul Laily

NIM 1903062

ABSTRAK

Program Kreativitas Mahasiswa (PKM) merupakan salah satu program Direktorat Jenderal Pembelajaran dan Kemahasiswaan (Belmawa) melalui direktorat kemahasiswaan untuk meningkatkan kualitas mahasiswa di perguruan tinggi agar dapat memiliki kemampuan akademis. Politeknik Negeri Indramayu merupakan salah satu perguruan tinggi negeri yang ikut serta dalam kegiatan Program Kreativitas Mahasiswa (PKM), terdapat beberapa jenis Program Kreativitas Mahasiswa (PKM) yang ditawarkan. Saat ini proses *review* proposal di Politeknik Negeri Indramayu masih dilakukan manual dengan cara mahasiswa yang sudah bimbingan dengan dosen pembimbing mengenai proposal kemudian mengirim proposal tersebut melalui *email* sehingga merepotkan bagi tim *reviewer*. Tidak hanya itu, penyebaran informasi mengenai kegiatan Program Kreativitas Mahasiswa (PKM) masih melalui aplikasi *instagram* milik Politeknik Negeri Indramayu sehingga penyebaran masih belum efektif. Berdasarkan kendala yang telah dijelaskan diatas, maka penulis memiliki ide untuk membuat sebuah aplikasi yang berjudul “Aplikasi Pojok Kemahasiswaan di Politeknik Negeri Indramayu Berbasis Website” yang bertujuan agar dapat mempermudah pihak Politeknik Negeri Indramayu dalam membuat kegiatan Program Kreativitas Mahasiswa (PKM) menjadi lebih efektif lagi.

Kata Kunci: Program Kreativitas Mahasiswa (PKM), Politeknik Negeri Indramayu, Proses *Review* Proposal, Informasi Kegiatan, Aplikasi.

ABSTRACT

The Student Creativity Program (PKM) is one of the programs of the Directorate General of Learning and Student Affairs (Belmawa) through the student affairs directorate to improve the quality of students in higher education so that they can have academic abilities. Indramayu State Polytechnic is one of the state universities that participates in the Student Creativity Program (PKM), there are several types of Student Creativity Program (PKM) offered. Currently, the review at the Indramayu State Polytechnic is still being carried out manually by means of students who have been guided by their supervisor on the proposal and then send the proposal via email, making it inconvenient for the reviewer. Not only that, the dissemination of information regarding the activities of the Student Creativity Program (PKM) is still through the Instagram belonging to the Indramayu State Polytechnic so that the dissemination is still not effective. Based on the constraints described above, the author has an idea to create an application entitled "Applications for Student Corner at the Website-Based State Polytechnic Indramayu" which aims to make it easier for the Indramayu State Polytechnic to make Student Creativity Program (PKM) activities easier. even more effective.

Keywords: *Student Creativity Program (PKM), Indramayu State Polytechnic, Review, Activity Information, Application.*

KATA PENGANTAR

Puji Syukur kami panjatkan kehadiran Allah SWT. Atas segala rahmat dan hidayah-Nya penulis dapat menyelesaikan Tugas Akhir dengan judul “APLIKASI POJOK KEMAHASISWAAN DI POLITEKNIK NEGERI INDRAMAYU BERBASIS WEBSITE”. Tugas Akhir ini penulis susun sebagai persyaratan untuk menyelesaikan studi program Diploma III Program Studi Teknik Informatika, Jurusan Teknik Informatika, Politeknik Negeri Indramayu.

Kami menyadari tanpa adanya dukungan dan kerja sama dari berbagai pihak, kegiatan Tugas Akhir ini tidak akan dapat berjalan baik. Untuk itu, kami ingin menyampaikan rasa terima kasih kepada:

1. *Allah Subhanahu wa ta'ala.*
2. Bapak Casiman Sukardi, S.T., M.T., selaku Direktur Politeknik Negeri Indramayu.
3. Bapak Iryanto, S.Si.,M.Si., selaku ketua Jurusan Teknik Informatika Politeknik Negeri Indramayu, Dosen Wali Kelas D3TI3C, dan Dosen Pembimbing II Tugas Akhir.
4. Bapak Muhammad Anis Al Hilmi, S.Si., M.T., selaku koordinator Tugas Akhir Jurusan Teknik Informatika.
5. Ibu Munengsih Sari Bunga, S.Kom., M.Eng., selaku Dosen Pembimbing I Tugas Akhir.
6. Bapak dan Ibu Dosen Jurusan Teknik Informatika Politeknik Negeri Indramayu.
7. Kedua orang tua, kakak, dan adik penulis serta keluarga yang selalu mendoakan dan memberi semangat kepada penulis agar dapat menyelesaikan Tugas Akhir dengan lancar dan tepat waktu.
8. Teman-teman kelas D3TI3C dan seluruh rekan-rekan jurusan Teknik Informatika yang telah banyak membantu dalam menyelesaikan Tugas Akhir ini.
9. Dan seluruh pihak yang telah membantu dan mendukung lancarnya pembuatan Tugas Akhir dari awal hingga akhir yang tidak dapat sebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan Tugas Akhir ini, masih banyak terdapat kekurangan dan kelemahan yang dimiliki penulis baik itu sistematika penulisan maupun penggunaan bahasa. Untuk itu penulis mengharapkan saran dan kritik dari berbagai pihak yang bersifat membangun demi penyempurnaan laporan ini. Semoga laporan ini berguna bagi pembaca secara umum dan penulis secara khusus. Akhir kata, penulis ucapkan banyak terima kasih.

Indramayu, Agustus 2022

Penulis

DAFTAR ISI

HALAMAN PENGESAHAN	ii
MOTTO	iii
PERNYATAAN.....	iv
ABSTRAK	v
ABSTRACT	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	3
1.5. Manfaat Penelitian.....	3
1.6. Sistematika Penulisan.....	4
BAB II LANDASAN TEORI	6
2.1. Program Kreativitas Mahasiswa (PKM)	6
2.2. Politeknik Negeri Indramayu	6
2.3. Aplikasi.....	7
2.4. Website	8
2.5. Visual Studio Code	9
2.6. Metode Prototype	9
2.7. Laravel.....	12
2.8. XAMPP	14
2.9. MariaDB	15
2.10. Hypertext Preprocessor (PHP).....	16
2.11. HyperText Markup Language (HTML).....	17
2.12. Javascript	18

2.13. <i>Flowchart</i>	18
2.14. <i>Unified Model Language (UML)</i>	22
2.15. <i>Entity Relationship Diagram (ERD)</i>	26
2.15.1. <i>Pengertian Entity Relationship Diagram (ERD)</i>	26
2.15.2. <i>Simbol Entity Relationship Diagram (ERD)</i>	27
2.16. <i>Bootstrap</i>	29
2.17. <i>Cascading Style Sheet (CSS)</i>	29
2.18. <i>Balsamiq Mockup</i>	29
2.19. <i>Black Box Testing</i>	30

DAFTAR TABEL

Tabel 2. 1 Simbol-Simbol Arus	20
Tabel 2. 2 Simbol-Simbol Proses.....	20
Tabel 2. 3 Simbol-Simbol I/O	21
Tabel 2. 4 Simbol-Simbol Use Case Diagram	23
Tabel 2. 5 Simbol-Simbol Class Diagram.....	24
Tabel 2. 6 Simbol-Simbol Sequence Diagram.....	25
Tabel 2. 7 Simbol-Simbol Activity Diagram	26
Tabel 2. 8 Simbol-Simbol Entity Relationship Diagram	27

DAFTAR GAMBAR

Gambar 2. 1 Logo Politeknik Negeri Indramayu	6
Gambar 2. 2 Logo Visual Studio Code	9
Gambar 2. 3 Metode Prototype	9
Gambar 2. 4 Logo Laravel	12
Gambar 2. 5 Logo XAMPP	14
Gambar 2. 6 Logo PHP	16
Gambar 2. 7 Logo HTML	17

DAFTAR LAMPIRAN

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan era digitalisasi dan penerapan revolusi industri 4.0 saat ini berkembang pesat diberbagai bidang termasuk di dunia pendidikan. Program Kreativitas Mahasiswa (PKM) merupakan salah satu program Direktorat Jenderal Pembelajaran dan Kemahasiswaan (Belmawa) melalui direktorat kemahasiswaan untuk meningkatkan kualitas mahasiswa di perguruan tinggi agar dapat memiliki kemampuan akademis dan/atau profesional yang dapat menerapkan, mengembangkan dan menyebarluaskan ilmu pengetahuan, teknologi dan/atau kesenian serta memperkaya budaya nasional. Lulusan sebuah perguruan tinggi dituntut untuk memiliki *academic knowledge, skill of thinking, management skill, dan communication skill*. Kekurangan salah satu dari keempat keterampilan atau kemahiran tersebut dapat menyebabkan berkurangnya mutu lulusan (Ditjen Belmawa, 2016).

Politeknik Negeri Indramayu merupakan salah satu perguruan tinggi negeri yang ikut serta dalam kegiatan Program Kreativitas Mahasiswa (PKM). Terdapat beberapa jenis Program Kreativitas Mahasiswa (PKM) yang ditawarkan yaitu Program Kreativitas Mahasiswa Riset Eksakta (PKM-RE), Program Kreativitas Mahasiswa Riset Sosial Humaniora (PKM-RSH), Program Kreativitas Mahasiswa Kewirausahaan (PKM-K), Program Kreativitas Mahasiswa Pengabdian kepada Masyarakat (PKM-PM), Program Kreativitas Mahasiswa Penerapan Iptek (PKM-PI), Program Kreativitas Mahasiswa Karsa Cipta (PKM-KC), Program Kreativitas Mahasiswa Karya Inovatif (PKM-KI), dan Program Kreativitas Mahasiswa Video Gagasan Konstruktif (PKM-VGK). Namun untuk jenis Program Kreativitas Mahasiswa (PKM) terkadang setiap tahun dapat berubah. Alur terakhir dari kegiatan Program Kreativitas Mahasiswa (PKM) adalah Pekan Ilmiah Mahasiswa Nasional (Pimnas), yang diikuti oleh mahasiswa dengan Program Kreativitas Mahasiswa (PKM) yang sudah dinyatakan lolos dan didanai Dikti.

Mahasiswa yang ingin mengikuti kegiatan Program Kreativitas Mahasiswa (PKM) terlebih dahulu mendaftarkan diri beserta timnya untuk didata dan diberi arahan baik mengenai proposal maupun rangkaian kegiatan Program Kreativitas Mahasiswa (PKM). Proposal yang diikutsertakan dalam Program Kreativitas Mahasiswa (PKM) akan dilakukan *review* oleh Bidang Kemahasiswaan sebelum diunggah ke Sim-Belmawa yang merupakan sistem informasi manajemen untuk Program Kreativitas Mahasiswa (PKM) milik Direktorat Jenderal Pembelajaran dan Kemahasiswaan Kemenristek. Tujuan dilakukannya *review* adalah untuk mengoreksi beberapa hal yang perlu ditambahkan ataupun diperbaiki sehingga proposal dapat disusun lebih baik.

Dalam kegiatan Program Kreativitas Mahasiswa (PKM) yang dilaksanakan di Politeknik Negeri Indramayu, peserta yang mengikuti kegiatan Program Kreativitas Mahasiswa (PKM) setiap tahunnya berbeda jumlahnya sehingga menyebabkan munculnya permasalahan-permasalahan baru dalam proses pelayanan dan administrasi Program Kreativitas Mahasiswa (PKM). Saat ini proses *review* proposal di Politeknik Negeri Indramayu masih dilakukan manual dengan cara mahasiswa yang sudah bimbingan dengan dosen pembimbing mengenai proposal kemudian mengirim proposal tersebut melalui *email* sehingga merepotkan bagi tim *reviewer*. Tidak hanya itu, penyebaran informasi mengenai kegiatan Program Kreativitas Mahasiswa (PKM) masih melalui aplikasi *instagram* milik Politeknik Negeri Indramayu sehingga penyebaran masih belum efektif.

Berdasarkan kendala yang telah dijelaskan diatas, maka penulis memiliki ide untuk membuat sebuah aplikasi yang berjudul “Aplikasi Pojok Kemahasiswaan di Politeknik Negeri Indramayu Berbasis Website” yang bertujuan agar dapat mempermudah pihak Politeknik Negeri Indramayu dalam membuat kegiatan Program Kreativitas Mahasiswa (PKM) menjadi lebih efektif lagi.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang sudah dijabarkan, maka diperoleh suatu rumusan masalah yang menjadi dasar pembuatan sistem tersebut, diantaranya sebagai berikut:

1. Bagaimana membuat dan mengimplementasikan aplikasi agar pengelolaan kegiatan Program Kreativitas Mahasiswa (PKM) yang dilaksanakan oleh Politeknik Negeri Indramayu dapat lebih efektif?
2. Bagaimana agar proses *review* proposal yang dilakukan oleh Politeknik Negeri Indramayu dapat memudahkan *reviewer* dalam mengoreksi proposal?

1.3. Batasan Masalah

Untuk memfokuskan pembahasan, maka diperoleh beberapa batasan masalah, diantaranya:

1. Pengguna aplikasi berbasis *website* ini adalah admin, *reviewer*, dosen pembimbing, dan mahasiswa Politeknik Negeri Indramayu yang mengikuti kegiatan Program Kreativitas Mahasiswa (PKM).
2. Aplikasi ini berfokus pada proses *review* proposal internal.
3. Untuk pembuatan proposal dilakukan secara mandiri.
4. Terdapat fitur bimbingan dengan dosen pembimbing.
5. Disediakan *template* proposal Program Kreativitas Mahasiswa (PKM) dalam bentuk pdf.
6. Aplikasi ini dibuat menggunakan *framework Laravel*.

1.4. Tujuan Penelitian

Berikut merupakan tujuan dibuatnya “Aplikasi Pojok Kemahasiswaan di Politeknik Negeri Indramayu Berbasis Website” ini meliputi:

1. Terwujudnya “Aplikasi Pojok Kemahasiswaan di Politeknik Negeri Indramayu Berbasis Website” yang dapat membantu pengelolaan kegiatan Program Kreativitas Mahasiswa (PKM) lebih efektif lagi.
2. Untuk meningkatkan proses *review* proposal Program Kreativitas Mahasiswa (PKM).

1.5. Manfaat Penelitian

Manfaat yang dapat diperoleh dengan dibuatnya “Aplikasi Pojok Kemahasiswaan di Politeknik Negeri Indramayu Berbasis Website” ini adalah:

1. Dapat memberikan solusi untuk memaksimalkan pengelolaan kegiatan Program Kreativitas Mahasiswa (PKM) lebih efektif.
2. Memudahkan tim *reviewer* dalam mengoreksi proposal Program Kreativitas Mahasiswa (PKM) yang diajukan oleh tim pengusul Program Kreativitas Mahasiswa (PKM).

1.6. Sistematika Penulisan

Dalam sistematika penulisan laporan tugas akhir ini dibagi menjadi beberapa bagian bab yang dimana setiap bagian memiliki pembahasan yang berbeda-beda tetapi saling terkait antara satu dengan lainnya. Untuk memudahkan penulisan laporan tugas akhir ini, penulis mengurutkan serta menjabarkan setiap bagian secara sistematis. Adapun urutan penulisan laporannya sebagai berikut:

BAB I PENDAHULUAN

Pada bab pendahuluan berisi pemaparan latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian “Aplikasi Pojok Kemahasiswaan di Politeknik Negeri Indramayu Berbasis Website”, serta sistematika penulisan laporan.

BAB II LANDASAN TEORI

Bab ini menguraikan tentang teori-teori dasar yang berkaitan secara langsung dalam proses pembuatan dan penyelesaian masalah berdasarkan sumber referensi yang valid terkait penyusunan laporan tugas akhir. Sumber yang dijadikan referensi berasal dari buku, jurnal, maupun internet dari sumber yang dapat dipercaya.

BAB III METODE PELAKSANAAN

Metode pelaksanaan berisi penjelasan tentang tahapan-tahapan pelaksanaan serta metode pelaksanaan yang digunakan untuk mencapai tujuan yang ditetapkan yaitu membangun “Aplikasi Pojok Kemahasiswaan di Politeknik Negeri Indramayu Berbasis Website”. Pada bab ini terdapat rancangan aplikasi seperti *flowchart*, *Unified Model Language* (UML), dan rancangan *user interface* yang dibuat untuk menjelaskan tahapan serta gambaran nyata dari sistem.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini membahas hasil-hasil dari tahap pelaksanaan implementasi “Aplikasi Pojok Kemahasiswaan di Politeknik Negeri Indramayu Berbasis

Website” berdasarkan rancangan yang telah dipaparkan pada tahap sebelumnya serta pengujian dari “Aplikasi Pojok Kemahasiswaan di Politeknik Negeri Indramayu Berbasis Website”.

BAB V PENUTUP

Bab ini berisi kesimpulan dan saran yang berkaitan dengan analisa berdasarkan uraian pada bab-bab sebelumnya. Kesimpulan dihasilkan berdasarkan pada penulisan laporan tugas akhir “Aplikasi Pojok Kemahasiswaan di Politeknik Negeri Indramayu Berbasis Website”, sedangkan saran berisi pendapat atau masukan dari pembaca.

BAB II

LANDASAN TEORI

2.1. Program Kreativitas Mahasiswa (PKM)

Pemberdayaan masyarakat merupakan peningkatan kemampuan dan kemandirian masyarakat dalam meningkatkan taraf hidupnya. Pemberdayaan masyarakat menjadi salah satu tanggung jawab Perguruan Tinggi, hal ini tertuang pada Undang-Undang No.12 Tahun 2012 tentang Pendidikan Kementerian Riset, Teknologi, dan Pendidikan Tinggi membuat suatu kebijakan berupa Program Kreativitas Mahasiswa (PKM).

Program Kreativitas Mahasiswa (PKM) dilaksanakan pertama kali pada tahun 2001 yang kegiatannya meliputi pendidikan, penelitian, dan pengabdian kepada masyarakat yang selama ini syarat dengan partisipasi aktif mahasiswa, diintegrasikan ke dalam satu wahana, yaitu Program Kreativitas Mahasiswa (Ristekdikti, 2017).

Program Kreativitas Kemahasiswaan (PKM) merupakan sebuah kegiatan oleh Kementerian Riset, Teknologi, dan Pendidikan Tinggi melalui Direktorat Jenderal Pembelajaran dan Kemahasiswaan (Ditjen Belmawa) sebagai unit pelaksanaannya. Kegiatan ini sebagai upaya agar mahasiswa mencapai level kreatif. Makna kreatif disini adalah memiliki kemampuan dalam kecepatan menemukan solusi atas persoalan atau permasalahan yang dihadapinya serta dapat diwujudkan. Kemampuan berpikir dan bertindak kreatif pada hakikatnya dapat dilakukan setiap manusia apalagi yang menikmati pendidikan tinggi (Ditjen Belmawa, 2016).

2.2. Politeknik Negeri Indramayu



Gambar 2. 1 Logo Politeknik Negeri Indramayu

Politeknik Negeri Indramayu atau yang dikenal dengan nama Polindra merupakan perguruan tinggi negeri yang berada di Jl. Lohbener Lama No. 08, Lohbener, Indramayu. Pada awalnya pendidikan vokasi ini memiliki 3 Program Studi diantaranya adalah Jurusan Teknik Mesin (D3), Teknik Informatika (D3), dan Teknik Pendingin dan Tata Udara (D3). Dan kemudian Politeknik Negeri Indramayu mempunyai program studi baru yaitu Perancangan Manufaktur (D4), Rekayasa Perangkat Lunak (D4), dan Akademi Keperawatan (D3).

Pada tahun 2021, Politeknik Negeri Indramayu menerima 505 mahasiswa yang terbagi menjadi 6 Prodi. Pertama Prodi Teknik Mesin menerima mahasiswa baru berjumlah 99 mahasiswa, kedua Prodi Teknik Pendingin dan Tata Udara menerima mahasiswa baru berjumlah 113 mahasiswa, ketiga Prodi Teknik Informatika menerima mahasiswa baru berjumlah 85 mahasiswa, keempat Prodi Perancangan Manufaktur menerima mahasiswa baru berjumlah 59 mahasiswa, kelima Prodi Rekayasa Perangkat Lunak menerima mahasiswa baru berjumlah 60 mahasiswa, dan keenam Prodi Keperawatan menerima mahasiswa baru berjumlah 89 mahasiswa.

Menurut data tahun 2021 dari bidang kemahasiswaan Politeknik Negeri Indramayu, mahasiswa atau tim yang mengikuti dan mengusulkan proposal sebanyak 80 proposal. Namun pada tahun 2021 Kementerian Riset, Teknologi, dan Pendidikan Tinggi memberikan kuota Program Kreativitas Kemahasiswaan (PKM) ke Politeknik Negeri Indramayu sebanyak 60 proposal sehingga ada proses seleksi proposal berdasarkan kuota yang diberikan.

2.3. Aplikasi

Pengertian aplikasi menurut Kamus Besar Bahasa Indonesia (KBBI) adalah penerapan dari rancangan sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu.

Aplikasi berasal dari bahasa Inggris yaitu “*To applicate*” yang merupakan suatu program yang memiliki perintah untuk dapat mengolah suatu data. Aplikasi merupakan suatu sub kelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna (Aris, 2016).

Aplikasi adalah suatu sub kelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Aplikasi merupakan suatu *software* untuk digunakan sebagai kebutuhan aktifitas, seperti aktifitas instansi pemerintah, toko, dan dapat membantu mempermudah pekerjaan seseorang (Supardi & Herfianti, 2019).

2.4. Website

World Wide Web (WWW) atau lebih dikenal dengan *Website* adalah suatu sistem informasi yang mendukung interaksi pengguna melalui antarmuka berbasis *website*. Interaksi pengguna dengan *website* dibagi kedalam tiga tahap, yaitu permintaan, pemrosesan, dan jawaban. *Website* merupakan kumpulan halaman-halaman yang digunakan untuk menampilkan informasi berupa teks, gambar, animasi, suara, ataupun gabungan dari semuanya, baik bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman (Aditya, 2019).

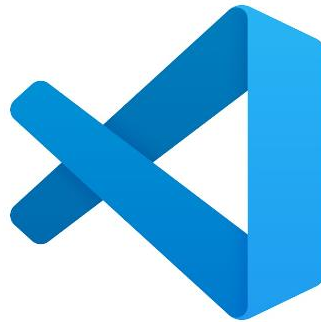
Website menampilkan informasi bisa berupa teks, gambar, video, maupun suara yang dapat diakses melalui internet. *Website* dapat dibuka dimanapun dan kapanpun sehingga dapat menjangkau ruang lingkup yang lebih luas. Suatu halaman *website* yang sudah terhubung dengan suatu halaman *website* lain biasanya disebut dengan *hyperlink*, sedangkan teks yang terhubung oleh teks lain disebut sebagai *hypertext*.

Menurut Indoweb (2021), *Website* adalah suatu halaman *web* yang saling berhubungan yang umumnya berisikan kumpulan informasi berupa teks, gambar, animasi, audio, video maupun gabungan dari semuanya yang biasanya dibuat untuk personal, organisasi, maupun perusahaan.

Berikut ini merupakan jenis-jenis *website* yang berdasarkan sifatnya, diantaranya:

1. *Website* dinamis, merupakan sebuah *website* yang menyediakan *content* atau isi yang selalu berubah-ubah setiap saat. Contoh nya seperti *website* berita.
2. *Website* statis, merupakan *website* yang *content* nya sangat jarang diubah. Contoh nya seperti *website* profil organisasi.

2.5. Visual Studio Code

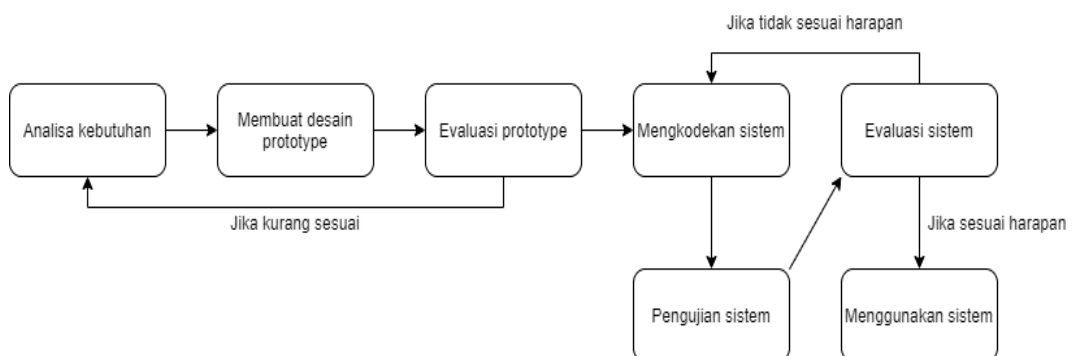


Gambar 2. 2 Logo Visual Studio Code

Visual studio code adalah kode editor yang dikembangkan oleh Microsoft untuk Windows, Linux, dan MacOS. Inu termasuk dukungan untuk *debugging*, kontrol git yang tertanam dan GitHub, penyorotan sintaksis, penyelesaian kode cerdas, *snippet*, dan *refactoring* kode. Ini sangat dapat disesuaikan, memungkinkan pengguna untuk mengubah tema, pintasan *keyboard*, preferensi, dan menginstall *ektensi* yang menambah fungsionalitas tambahan (Agustini, 2019).

Menurut Yulianto (2019), *visual studio code* adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft dan secara langsung mendukung bahasa pemrograman *JavaScript*, *Typescript*, *Node.js*, serta bahasa pemrograman lainnya dengan bantuan *plugin* yang dapat dipasang via *marketplace visual studio code* (seperti C++, C#, Python, Java, dan lain-lain).

2.6. Metode Prototype



Gambar 2. 3 Metode Prototype

Menurut Yanuarti (2017), “*Prototype* adalah salah satu pendekatan dalam rekayasa perangkat lunak yang secara langsung mendemonstrasikan bagaimana sebuah perangkat lunak atau komponen-komponen perangkat lunak akan bekerja dalam lingkungannya sebelum tahapan konstruksi actual dilakukan. Model *prototype* digunakan sebagai indikator dari gambaran yang akan dibuat pada masa yang akan datang dan membedakan dua fungsi eksplorasi dan demonstrasi”.

Menurut Novitasari (2020), Metode *Prototype* merupakan salah satu metode siklus hidup sistem yang didasarkan pada konsep model bekerja (*working model*). Berikut ini merupakan tahapan-tahapan dalam metode *prototype*, yaitu:

1. Pengumpulan kebutuhan

Langkah pertama kali yang harus dilakukan dalam tahapan metode *prototype* adalah mengidentifikasi seluruh perangkat dan permasalahan. Tahapan metode *prototype* yang sangat penting adalah analisis dan identifikasi kebutuhan garis besar dari sistem. Setelah itu akan diketahui apa dan permasalahan yang akan dibuat dan dipecahkan.

2. Membangun *prototype*

Langkah selanjutnya adalah langkah metode *prototype* membangun *prototype* yang berfokus pada penyajian. Misalkan membuat *input* dan *output* hasil sistem. Sementara hanya *prototype* saja dulu selanjutnya akan ada tindak lanjut yang harus dikerjakan.

3. Evaluasi *prototype*

Sebelum melangkah ke langkah selanjutnya, ini bersifat wajib yaitu memeriksa langkah 1, karena ini adalah penentu keberhasilan dan proses yang sangat penting. Ketika langkah 1 dan 2 terdapat ada yang kurang atau salah maka kedepannya akan sulit sekali melanjutkan langkah selanjutnya.

4. Mengkodekan sistem

Sebelum pengkodean biasanya hal yang perlu dilakukan, yaitu memahami terlebih dahulu bahasa pemrograman yang akan digunakan. Dalam tahap ini merancang, membangun, dan mengaplikasikan *website* atau aplikasi disesuaikan dengan kebutuhan dalam bentuk kode program.

5. Menguji sistem

Setelah pengkodean, yang akan dilakukan selanjutnya yaitu *testing* program. Banyak sekali cara untuk *testing*, misalkan menggunakan *white box* atau *black box*. Menggunakan *white box* berarti menguji kodingan, sedangkan *black box* berarti menguji fungsi-fungsi tampilan apakah sudah benar dengan aplikasinya atau tidak.

6. Evaluasi sistem

Mengevaluasi dari semua langkah yang pernah dilakukan, sudah sesuai dengan kebutuhan atau belum. Jika belum atau masih ada revisi maka dapat mengulangi dan kembali ketahap 1 dan 2.

7. Menggunakan sistem

Sistem sudah selesai diimplementasi, sebaiknya dilakukan upaya untuk *maintenance system* agar sistem terjaga dan berfungsi dengan baik dan dapat meningkatkan produktifitas dan kinerja.

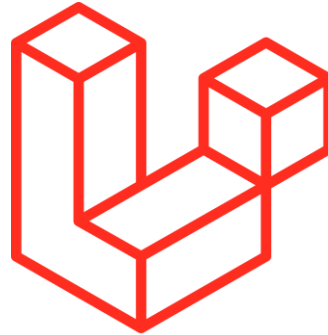
Terdapat beberapa kelebihan menggunakan metode *prototype*, diantaranya sebagai berikut:

1. Adanya komunikasi yang baik antara pengembang dan pelanggan.
2. Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan.
3. Pelanggan berperan aktif dalam pengembangan sistem.
4. Lebih menghemat waktu dalam pengembangan sistem.
5. Penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkannya membuat *client* mendapat gambaran awal dari *prototype*.

Terdapat juga beberapa kelemahan menggunakan metode *prototype*, diantaranya sebagai berikut:

1. Pelanggan tidak melihat bahwa perangkat lunak belum mencerminkan kualitas perangkat lunak secara keseluruhan dan belum memikirkan pemeliharaan dalam jangka waktu yang lama.
2. Pengembang biasanya ingin cepat menyelesaikan proyek sehingga menggunakan algoritma dan bahasa pemrograman sederhana.
3. Hubungan pelanggan dengan komputer mungkin tidak menggambarkan teknik perancangan yang baik.

2.7. *Laravel*



Gambar 2. 4 Logo *Laravel*

Pengertian *framework* menurut Naista adalah suatu struktur konseptual dasar yang digunakan untuk memecahkan atau menangani suatu masalah yang kompleks. Singkatnya, *framework* adalah wadah atau kerangka kerja dari sebuah *website* yang akan dibangun. Dengan menggunakan kerangka tersebut waktu yang digunakan dalam membuat *website* lebih singkat dan memudahkan dalam melakukan perbaikan. Salah satu *framework* yang banyak digunakan oleh *programmer* adalah *framework laravel*. *Laravel* adalah *framework* berbasis PHP (*Hypertext Preprocessor*) yang sifatnya *open source*, dan menggunakan konsep *model-view-controller*. *Laravel* berada di bawah lisensi MIT *license* dengan menggunakan Github sebagai tempat berbagai *code* menjalankannya (Naista,2017).

Banyak keuntungan yang didapatkan seperti dengan menggunakan *framework* dapat memberikan struktur yang baik dalam program yang dibuat karena memiliki *library* atau fungsi yang bisa digunakan, serta mempermudah dalam pengerjaan program secara tim karena dalam membangun *website* harus melakukan penyesuaian dengan gaya *framework* yang dipakai. Namun, *Laravel* memiliki beberapa kekurangan seperti ukuran *file* yang cukup besar, dibutuhkan koneksi internet untuk instalasi dan mengunduh *library Laravel*, dan untuk PHP (*Hypertext Preprocessor*) minimal versi 5.4 untuk menjalankannya (Naista, 2017). Berikut beberapa dasar-dasar pada *Laravel*:

1. *Artisan*

Artisan adalah *command line* atau perintah yang dijalankan melalui terminal dan disediakan beberapa perintah-perintah yang dapat digunakan selama melakukan pembuatan sistem. Salah satu fungsi dari PHP (*Hypertext*

Preprocessor) *artisan* yaitu “*php artisan serve*” yang berfungsi untuk membuka *website* yang telah dibuat tanpa menggunakan *website server* lokal.

2. *Routing*

Routing adalah suatu proses yang bertujuan agar suatu *item* yang diinginkan dapat sampai ketujuan. Dengan menggunakan *routing* dapat ditentukan halaman-halaman yang akan muncul ketika dibuka oleh *user*. Pengaturan *routing* di *Laravel* biasanya terletak di *file web.php*. *File web.php* terletak didalam folder *routes*.

3. *Controller*

Controller adalah suatu proses yang bertujuan untuk mengambil permintaan, menginisialisasi, memanggil model untuk dikirimkan ke *view*. Permintaan yang dibuat dalam *laravel* harus berada didalam *controller*, kemudian dilempar melalui *routing* untuk mendapat permintaan yang diinginkan.

4. *View (blade templating)*

Blade adalah *template* bawaan dari *laravel*. *Blade* memiliki kode-kode yang lebih mudah untuk menghasilkan *laravel*. Cara membuat *file blade* dilakukan secara manual dengan membuat *nama_file.blade.php* didalam folder *views*.

Didalam *blade* terdapat *template* master dan *inheritance*. Pembuatan *template* master dan turunannya ini bertujuan agar elemen yang sama tidak ditulis secara berulang-ulang. Pada *template inheritance* diberikan kode “*extend (nama_layout)* dan *section (nama_content)*”.

5. *Middleware*

Middleware adalah penengah antara *request* yang masuk dengan *controller* yang dituju. Cara membuat *middleware* menggunakan *artisan* dengan mengetikkan “*php artisan make:middleware nama_file*”. *File middleware* berada didalam folder *middleware*.

6. *Session*

Session adalah sebuah cara yang digunakan untuk penyimpanan pada *server* dan penyimpanan tersebut digunakan pada beberapa halaman termasuk halaman itu sendiri. Dalam menggunakan *session* terdapat dua acara yaitu

session dapat dibuat menggunakan *request* dan dapat digunakan fungsi global *helper session*.

7. *Migration*

Migration adalah sebuah fitur yang ada dalam *laravel*, dan merupakan *control version system* untuk *database*. Dengan menggunakan *migration*, bisa lebih mudah dan cepat dalam membuat *table* data. *Migration* membuat atau meng-generate *file-file migration* sebagai *control system*. Cara membuat *migration* menggunakan *artisan* dengan mengetikkan “*php artisan make:migration create_namatable_table -create=namatable*”.

8. *Model*

Model merupakan salah satu dari bagian MVC (*Model-View-Controller*) yang bertugas berhubungan langsung dengan *database*. Bisa dikatakan juga bahwa *model* adalah penghubung setiap alur program yang berhubungan dengan data. Nantinya, *model* yang sudah terhubung ke *database* akan dipanggil melalui *controller* sebagaimana konsep MVC (*Model-View-Controller*) itu berjalan. Cara membuat *model* menggunakan *artisan* dengan mengetikkan “*php artisan make:model nama_model*”.

2.8. XAMPP



Gambar 2. 5 Logo XAMPP

Menurut Bunafit (dalam Putera dan Ibrahim, 2018), XAMPP merupakan paket PHP (*Hypertext Preprocessor*) berbasis *open source* yang dikembangkan oleh sebuah komunitas *Open Source*. Dalam menggunakan XAMPP sudah disediakan berbagai kebutuhan sehingga tidak perlu melakukan penginstalan

program lainnya. Beberapa paket yang sudah disediakan adalah *phpmyadmin*, *filezilla*, PHP (*Hypertext Preprocessor*), MySQL, dan *apache*.

Pengertian XAMPP adalah perangkat lunak (*free software*) bebas, yang mendukung untuk banyak sistem operasi, yang merupakan kompilasi dari beberapa program. Fungsi XAMPP sendiri adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri beberapa program antara lain *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP (*Hypertext Preprocessor*) dan Perl. Nama XAMPP sendiri merupakan singkatan dari X (tempat sistem operasi apapun seperti *windows*, *linux*, dan sebagainya), *Apache*, *MySQL*, *PHP*, dan *Perl*. Program ini tersedia dalam *GNU General Public License* dan bebas, merupakan *web server* yang mudah untuk digunakan yang dapat menampilkan halaman *web* yang dinamis (Umagapi & Ambarita, 2018).

2.9. MariaDB

Pada tahun 2010, MySQL dibeli oleh perusahaan Oracle. Hal ini membuat khawatir banyak kalangan karena Oracle merupakan perusahaan *database* terbesar didunia. Untuk mengantisipasi jika terjadi pengembangan MySQL dihentikan oleh Oracle, maka dari itu dikembangkanlah MariaDB sebagai pengganti MySQL. MariaDB dibuat oleh tim yang dulunya juga membuat MySQL.

Pada dasarnya, MariaDB adalah *cloningan* MySQL atau MariaDB berisi MySQL yang diberi “*merk*” MariaDB serta penambahan fitur dan perbaikan performa. Seluruh materi yang dibahas dalam tutorial MySQL bisa berjalan baik di MariaDB maupun MySQL. Faktanya, jika kita sudah *men-download* aplikasi XAMPP, sekarang sudah di *bundle* dengan *database* MariaDB bukan lagi MySQL. Untuk sisi *programming* dan *query* tidak ada perbedaan antara MySQL dengan MariaDB (Anugerah, 2018).

MariaDB adalah sistem manajemen *database* relasional yang dikembangkan dari MySQL. MariaDB dikembangkan oleh komunitas pengembang yang sebelumnya berkontribusi untuk *database* MySQL. MariaDB tetap mempertahankan kompatibilitas dan API layaknya MySQL versi sebelumnya (Permana, 2016).

2.10. Hypertext Preprocessor (PHP)



Gambar 2. 6 Logo PHP

PHP adalah singkatan dari *Hypertext Preprocessor*, yaitu bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs *website* dan bisa digunakan bersamaan dengan HTML (*HyperText Markup Language*). PHP (*Hypertext Preprocessor*) diciptakan oleh Rasmus Lerdorf pertama kali tahun 1994. Pada awalnya PHP adalah singkatan dari “*Personal Home Page Tools*”, selanjutnya diganti menjadi FI (*Forms Interpreter*). Sejak versi 3.0, nama bahasa ini diubah menjadi PHP (*Hypertext Preprocessor*) dengan singkatan PHP, PHP versi terbaru adalah versi-5 (Umagapi & Ambarita, 2018).

PHP (*Hypertext Preprocessor*) merupakan sebuah *script open source* yang digunakan untuk mengembangkan sebuah *website* dan PHP (*Hypertext Preprocessor*) dapat digabungkan ke dalam HTML (*HyperText Markup Language*). PHP (*Hypertext Preprocessor*) mengeksekusi setiap kodenya dilakukan didalam *server*, dengan cara seperti ini *client* tidak bisa mengetahui pemrograman yang akan dibuat. PHP (*Hypertext Preprocessor*) digunakan untuk merancang yang sifatnya dinamis dan dapat bekerja secara otomatis (Setiawan, dkk, 2019).

Website dinamis yang bisa dibuat dengan menggunakan PHP (*Hypertext Preprocessor*) adalah situs *web* yang bisa menyesuaikan tampilan konten tergantung kebutuhan dan situasi. *Website* dinamis juga dapat menyimpan data kedalam *database*, membuat halaman yang berubah-ubah sesuai dengan permintaan pengguna, dan lain sebagainya. Dalam pembuatan *web*, kode PHP (*Hypertext Preprocessor*) biasanya disisipkan kedalam dokumen HTML (*HyperText Markup Language*). PHP (*Hypertext Preprocessor*) disebut juga sebagai *scripting language* karena memiliki fitur tersebut (Setiawan, 2017).

2.11. *HyperText Markup Language* (HTML)



Gambar 2. 7 Logo HTML

HTML atau singkatan dari *Hyper Text Markup Language*, yaitu skrip yang berupa *tag-tag* untuk membuat dan mengatur struktur *website*. Beberapa tugas utama HTML (*HyperText Markup Language*) dalam membangun *website* diantaranya seperti menentukan *layout website*, membuat *list*, membuat tabel, membuat *link*, menyisipkan gambar, video, maupun audio (Josi, 2017).

Hypertext Markup Language (HTML) adalah halaman yang menciptakan dokumen-dokumen *hypertext* atau *hypermedia*. HTML (*HyperText Markup Language*) memasukkan kode-kode pengendali dalam sebuah dokumen pada berbagai poin yang dapat dispesifikasikan, yang dapat menciptakan hubungan (*hyperlink*) dengan bagian yang lain dari dokumen tersebut atau dengan dokumen lain yang berada di *Word Wide Web* (WWW). Tag HTML tidak *case sensitive*, jadi dapat menggunakan `<html>` atau `<HTML>` keduanya mendapatkan *output* yang sama. HTML berawal dari bahasa GSML (*Standart Generalized Markup Language*) yang penulisannya disederhanakan. HTML (*HyperText Markup Language*) dapat dibaca oleh berbagai macam *platform*. HTML (*HyperText Markup Language*) juga merupakan bahasa pemrograman yang fleksibel, dapat disisipi atau digabungkan dengan bahasa pemrograman lain, seperti PHP (*Hypertext Preprocessor*), ASP, JSP, *JacaScript*, dan lainnya. Jika ada kesalahan dalam penulisan HTML (*HyperText Markup Language*), *browser* tidak akan memperlihatkan *syntax error*, tetapi hanya tidak menampilkannya. HTML (*HyperText Markup Language*) terus berkembang seiring perkembangan *browser* (Musli, 2016).

2.12. Javascript

Javascript adalah bahasa pemrograman yang sederhana. Dengan *javascript* dapat dengan mudah membuat sebuah halaman *website* yang interaktif. Program *javascript* dituliskan pada file HTML (*.htm, *.html) (Efendi, dkk, 2020).

Javascript adalah bahasa pemrograman untuk sisi *client* atau *client side*. *Javascript* adalah bahasa pemrograman yang mendekati bahasa manusia atau bisa dikatakan bahasa tingkat tinggi, maka dari itu *javascript* mudah dipelajari. *Javascript* sendiri tujuannya dibuat untuk memperkaya fitur pada *website* agar lebih dinamis, seperti untuk menampilkan dan menghilangkan objek-objek pada *website* kemudian dengan fungsi *javascript* dapat menggali kembali objek yang dihilangkan tersebut (Silvia, 2019).

2.13. Flowchart

Flowchart atau sering disebut dengan diagram alir merupakan suatu jenis diagram yang mempresentasikan *algoritma* atau langkah-langkah instruksi yang berurutan dalam sistem. Seorang analis sistem menggunakan *flowchart* sebagai bukti dokumentasi untuk menjelaskan gambar logis sebuah sistem. Dengan begitu, *flowchart* dapat membantu untuk memberikan solusi terhadap masalah yang bisa saja terjadi dalam membangun sistem. Pada dasarnya, *flowchart* digambarkan dengan menggunakan simbol-simbol, dimana setiap simbol mewakili suatu proses sistem. Sedangkan untuk menghubungkan satu proses ke proses selanjutnya digambarkan dengan menggunakan garis penghubung. Dengan adanya *flowchart*, setiap urutan proses dapat digambarkan dengan lebih jelas. Selain itu, ketika ada penambahan proses baru dapat dilakukan dengan mudah menggunakan *flowchart* ini. Setelah proses membuat *flowchart* selesai, maka giliran *programmer* yang akan menerjemahkan desain logis tersebut kedalam bentuk program dengan berbagai bahasa pemrograman yang telah disepakati (Rosaly, 2019).

Berikut merupakan beberapa fungsi *flowchart* yang bisa disimak dibawah ini, seperti:

1. Merancang proyek baru

Ketika akan merancang suatu proyek, maka hal selanjutnya adalah memetakan proyek tersebut ke bentuk *flowchart*. Itu dapat membantu untuk

merancang serangkaian langkah-langkah yang melibatkan keputusan bersama.

2. Mengelola alur kerja

Untuk mengelola alur kerja, *flowchart* adalah cara yang paling penting dilakukan. Karena *flowchart* berperan dalam penentuan integritas dari proses tersebut, yaitu dapat menciptakan hasil yang berkualitas berdasarkan prosedur.

3. Memodelkan proses bisnis

Proses bisnis yang dimaksud bukan hanya berkaitan dengan keuntungan, melainkan serangkaian tugas baik itu yang sederhana sampai yang rumit juga termasuk kedalam proses bisnis. Tujuan pemodelan *flowchart* dapat dilakukan untuk memberikan hasil yang konsisten dan dapat juga diprediksi.

4. Mendokumentasikan setiap proses

Dalam menyelesaikan suatu proyek perlu adanya dokumentasi proses. Dengan begitu, *flowchart* menjadi media yang bagus untuk memenuhi tujuan tersebut.

5. Merepresentasikan *algoritma*

Biasanya, para perancang sistem terlebih dahulu menentukan *algoritma* untuk menyelesaikan proyek tersebut menggunakan SDL (*Specification and Description Language*) yang merupakan suatu spesifikasi bahasa yang digunakan untuk mendeskripsikan perilaku dari sistem tersebut. Maka dari itu, *flowchart* dapat memenuhi kebutuhan tersebut karena *flowchart* menawarkan berbagai simbol yang unik yang dapat digunakan untuk memetakan sistem yang dirancang.

6. Mengaudit proses

Secara umum, *flowchart* dapat digunakan untuk mendeteksi kerusakan yang terjadi dalam setiap proses. *Flowchart* dapat membantu menyelesaikan permasalahan dengan cara membagi setiap langkah dari proses itu kedalam segmen-segmen yang lebih kecil, kemudian memeriksa bagian mana yang tidak berfungsi atau perlu diadakan perbaikan.


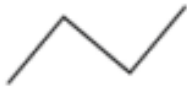
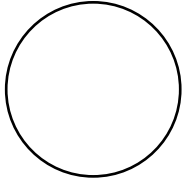
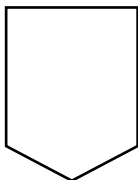
Flowchart dibuat berdasarkan pemikiran untuk menganalisa suatu permasalahan dalam bisnis. Hanya saja, untuk dapat merancang *flowchart* ketika

telah mengetahui simbol-simbol standar yang umum digunakan dalam proses pembuatan *flowchart*. Berikut ini merupakan simbol-simbol *flowchart* yang dibagi kedalam 3 kategori, diantaranya:

1. Simbol Arus (*Flow Direction Symbols*)

Biasanya simbol yang termasuk kedalam kategori ini digunakan sebagai simbol penghubung. Simbol yang termasuk kedalam kategori ini, yaitu sebagai berikut:



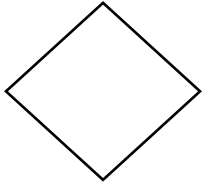


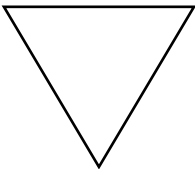

Tabel 2. 1 Simbol-Simbol Arus

Simbol	Nama	Fungsi
	<i>Flow Direction Symbol</i> atau <i>Connecting Line</i>	Berfungsi untuk menghubungkan simbol yang satu dengan yang lainnya, menyatakan arus suatu proses.
	<i>Communication Link</i>	Berfungsi untuk transmisi data dari satu lokasi ke lokasi lain.
	<i>Connector</i>	Digunakan untuk menyatakan sambungan dari proses yang satu ke proses berikutnya di halaman yang sama.
	<i>Offline Connector</i>	Digunakan untuk menyatakan sambungan dari proses yang satu ke proses berikutnya di halaman yang berbeda.

2. Simbol Proses (*Processing Symbols*)

Simbol proses merupakan simbol yang berkaitan dengan serangkaian proses yang dilakukan. Simbol yang termasuk kedalam bagian proses, yaitu sebagai berikut:

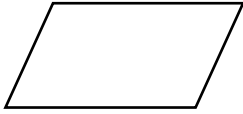


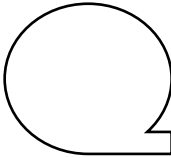


Tabel 2. 2 Simbol-Simbol Proses

Simbol	Nama	Fungsi
	<i>Processing</i>	Digunakan untuk menunjukkan pengolahan yang akan dilakukan dalam komputer.
	<i>Manual Operation</i>	Simbol ini digunakan untuk menunjukkan atau memperlihatkan pengolahan yang tidak dilakukan oleh komputer.
	<i>Decision</i>	Digunakan untuk memilih proses yang akan dilakukan berdasarkan kondisi tertentu.
	<i>Predefined Process</i>	Digunakan untuk mempersiapkan penyimpanan yang sedang/akan digunakan dengan memberikan harga awal.
	<i>Terminal</i>	Digunakan untuk memulai atau mengakhiri program.
	<i>Offline Storage</i>	Berfungsi untuk menunjukkan bahwa data akan disimpan ke media tertentu.
	<i>Manual Input Symbol</i>	Digunakan untuk menginputkan data secara manual dengan <i>keyboard</i> .

3. Simbol I/O (*Input-Output*)

Simbol yang termasuk kedalam bagian *input-output* berkaitan dengan masukan dan keluaran. Berikut beberapa simbol yang termasuk, yaitu:

Tabel 2. 3 Simbol-Simbol I/O

Simbol	Nama	Fungsi
	<i>Input/Output</i>	Digunakan untuk menyatakan <i>input</i> dan <i>output</i> tanpa melihat jenisnya.
	<i>Punched Card</i>	Digunakan untuk menyatakan masukan dan keluaran yang berasal dari <i>card</i> .
	<i>Disk Storage</i>	Digunakan untuk menyatakan masukan dan keluaran yang berasal dari <i>disk</i> .
	<i>Magnetic Tape</i>	Digunakan untuk menyatakan masukan dan keluaran yang berasal dari pita magnetis.
	<i>Document</i>	Digunakan untuk menyatakan masukan dan keluaran yang berasal dari dokumen,
	<i>Display</i>	Digunakan untuk menyatakan keluaran melalui layar monitor.

2.14. *Unified Model Language (UML)*

Menurut Ade Hendini (2016), *Unified Model Language (UML)* adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan, dan membangun perangkat lunak. *Unified Model Language (UML)* merupakan metodologi dalam mengembangkan sistem yang berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.

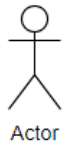

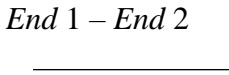
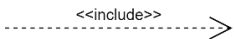
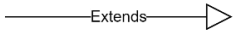
Menurut Sulianta (2017), “*Unified Model Language (UML)* merupakan kumpulan diagram-diagram yang sudah memiliki standar untuk membangun

perangkat lunak berbasis objek”. Berikut merupakan diagram-diagram yang dimiliki oleh *Unified Model Language* (UML):

1. *Use Case Diagram*

Use case diagram merupakan diagram yang harus dibuat pertama kali saat pemodelan perangkat lunak berorientasi objek dilakukan. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut ini merupakan simbol-simbol yang digunakan untuk membuat *use case diagram*, antara lain:

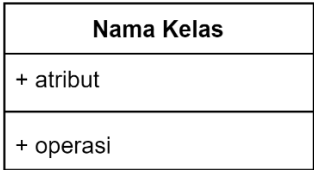



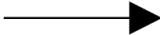
Tabel 2. 4 Simbol-Simbol *Use Case Diagram*

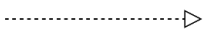
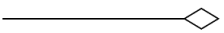
Simbol	Nama	Keterangan
	Aktor	Merupakan pengguna dari sistem. Penamaan aktor menggunakan kata benda.
	<i>Use Case</i>	Merupakan pekerjaan yang dilakukan oleh aktor. Penamaan <i>use case</i> dengan kata kerja.
	Asosiasi	Hubungan antara aktor dengan <i>use case</i> .
	<i>Include</i>	Hubungan antara <i>use case</i> dengan <i>use case</i> , <i>include</i> menyatakan bahwa sebelum pekerjaan dilakukan harus mengerjakan pekerjaan lain terlebih dahulu.
	<i>Extends</i>	Hubungan antara <i>use case</i> dengan <i>use case</i> , <i>extends</i> menyatakan bahwa jika pekerjaan yang dilakukan tidak sesuai atau terdapat kondisi khusus, maka lakukan pekerjaan itu.

2. *Class Diagram* (Diagram Kelas)

Class diagram dibuat setelah *use case diagram* dibuat. *Class diagram* merupakan hubungan antar kelas dan penjelasan *detail* tiap-tiap kelas didalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. Pada diagram ini harus menjelaskan hubungan apa saja yang terjadi antara suatu objek dengan objek lainnya sehingga terbentuklah suatu sistem aplikasi. Berikut ini merupakan simbol-simbol yang digunakan untuk membuat *class diagram*, antara lain:

Tabel 2. 5 Simbol-Simbol *Class Diagram*

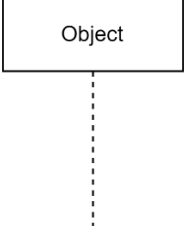

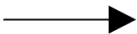
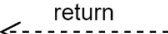
Simbol	Keterangan
<p>Kelas</p> 	Kelas pada struktur sistem.
<p>Antar muka / <i>Interface</i></p> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
<p>Asosiasi / <i>Association</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya disertai dengan <i>multiplicity</i> .
<p>Asosiasi berarah / <i>Redirected Association</i></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i> .
<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).

Kebergantungan / <i>dependency</i> 	Relasi antar kelas dengan makna ketergantungan antar kelas.
Agresiasi / <i>Aggregation</i> 	Relasi antar kelas dengan makna semua-bagian.

3. *Sequence Diagram* (Diagram Urutan)

Sequence diagram adalah diagram yang dibuat untuk mengetahui alur dari interaksi antar objek. Isi dari *sequence diagram* harus sama dengan *use case* dan *class diagram*. Berikut ini merupakan simbol-simbol yang digunakan untuk membuat *sequence diagram*, antara lain:

Tabel 2. 6 Simbol-Simbol *Sequence Diagram*




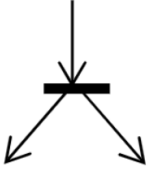



Simbol	Nama	Keterangan
	Objek / Aktor	Sebuah objek yang berasal dari kelas, atau dapat dinamai dengan kelasnya saja. Aktor termasuk objek. Garis putus-putus menunjukkan garis hidup suatu objek.
	Aktivasi	Menunjukkan masa hidup dari objek.
<i>Message 1</i> 	Pesan	Interaksi antara suatu objek dengan objek lainnya. Objek dapat mengirimkan pesan ke objek lain. Interaksi antar objek ditunjukkan pada bagian operasi pada diagram kelas.
<i>Message 2</i> 	<i>Return</i>	Pesan kembalian dari komunikasi antar objek.

4. *Activity Diagram* (Diagram Aktivitas)

Menurut Hendini (2016), “*Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis”. Berikut ini

merupakan simbol-simbol yang digunakan untuk membuat *activity diagram*, antara lain:

Tabel 2. 7 Simbol-Simbol *Activity Diagram*

Simbol	Keterangan
	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
	<i>End Point</i> , akhir aktivitas.
	<i>Activities</i> , menggambarkan suatu proses / kegiatan bisnis.
	<i>Fork</i> / Percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara <i>paralel</i> atau untuk menggabungkan kedua kegiatan <i>paralel</i> menjadi satu.
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>True</i> dan <i>False</i> .
	<i>Swimline</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

2.15. Entity Relationship Diagram (ERD)

2.15.1. Pengertian Entity Relationship Diagram (ERD)

Menurut Sukamto dan Shalahuddin (2018), “*Entity Relationship Diagram* (ERD) merupakan pemodelan awal basis data yang dikembangkan berdasarkan teori himpunan dalam bidang matematika untuk pemodelan basis data relasional”.

Simbol-simbol dalam *Entity Relationship Diagram* (ERD) adalah sebagai berikut:

1. Entitas: Suatu yang nyata atau abstrak yang mempunyai karakteristik dimana kita akan menyimpan data.
2. Atribut: Ciri umum semua atau sebagian besar instansi pada entitas tertentu.
3. Relasi: Hubungan alamiah yang terjadi antara satu atau lebih entitas.
4. *Link*: Garis penghubung atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi.

Kardinalitas relasi *Entity Relationship Diagram* (ERD) adalah sebagai berikut:

1. Satu ke satu (*One to One*)

Setiap elemen dari entitas A berhubungan paling banyak dengan elemen pada entitas B. Demikian juga sebaliknya setiap elemen B berhubungan paling banyak satu elemen pada entitas A.

2. Satu ke banyak (*One to Many*)

Setiap elemen dari entitas A berhubungan dengan maksimal banyak elemen pada entitas B. Dan sebaliknya setiap elemen dari entitas B berhubungan dengan paling banyak satu elemen di entitas A.

3. Banyak ke satu (*Many to One*)

Setiap elemen dari entitas A berhubungan paling banyak dengan satu elemen pada entitas B. Dan sebaliknya setiap elemen dari entitas B berhubungan dengan maksimal banyak elemen di entitas A.

4. Banyak ke banyak (*Many to Many*)

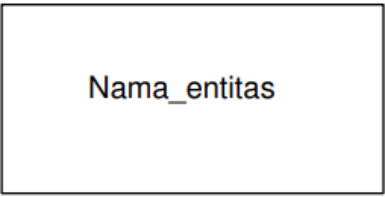
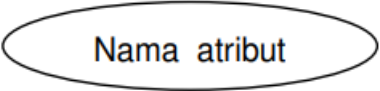
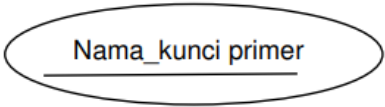
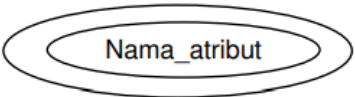
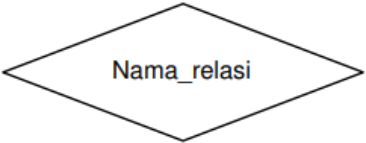

Setiap elemen dari entitas A berhubungan maksimal banyak elemen pada entitas B demikian sebaliknya.

2.15.2. Simbol *Entity Relationship Diagram* (ERD)

Berikut merupakan simbol-simbol yang digunakan pada *Entity Relationship Diagram* (ERD) dengan notasi *Chen*:

Tabel 2. 8 Simbol-Simbol *Entity Relationship Diagram*

No.	Simbol	Deskripsi
1.	Entitas/ <i>Entity</i>	Entitas merupakan data inti yang akan disimpan; bakal <i>table</i> pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses

		oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
2.	<p>Atribut</p> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
3.	<p>Atribut Kunci Primer</p> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).
4.	<p>Atribut Multinilai/<i>Multivalued</i></p> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
5.	<p>Relasi</p> 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.
6.	<p>Asosiasi/<i>Association</i></p> 	Penghubung antara relasi dan entitas dimana dikedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan kardinalitas 1 ke N (<i>one to many</i>) menghubungkan entitas A dan entitas B.

2.16. *Bootstrap*

Menurut Eko (2016), “*Bootstrap* merupakan salah satu *framework* HTML (*HyperText Markup Language*), CSS, dan JS yang digunakan untuk membuat *website* yang bersifat *responsive* atau bisa menyesuaikan tampilan *layout* nya berdasarkan ukuran *viewport* dari *device* pengaksesnya, mulai dari *smartphone*, *tablet*, maupun layer PC”.

Bootstrap adalah sebuah *framework* yang dibuat dengan menggunakan bahasa dari HTML (*HyperText Markup Language*) dan CSS (*Cascading Style Sheet*), namun juga menyediakan efek *javascript* yang dibangun dengan menggunakan *jquery*. *Bootstrap* telah menyediakan kumpulan komponen *class interface* dasar yang telah dirancang sedemikian rupa untuk menciptakan tampilan yang menarik, bersih, dan ringan. Selain itu, *bootstrap* juga memiliki fitur *grid* yang berfungsi untuk mengatur *layout* yang bisa digunakan dengan sangat mudah dan cepat (Sanjaya, 2017).

2.17. *Cascading Style Sheet (CSS)*

Cascading Style Sheet (CSS) adalah salah satu dari bahasa desain *website Style Sheet Language* dimana mampu mengontrol format tampilan sebuah halaman *website* yang ditulis dengan menggunakan penanda atau disebut dengan *markup language*. Umumnya *Cascading Style Sheet (CSS)* ini digunakan untuk mendesain sebuah halaman HTML (*HyperText Markup Language*) dan juga XHTML, akan tetapi sekarang *Cascading Style Sheet (CSS)* dapat diaplikasikan pula untuk segala dokumen XML, termasuk juga didalamnya SVG dan juga XUL bahkan sampai dengan sistem operasi android (Abror, 2018).

2.18. *Balsamiq Mockup*

Balsamiq Mockup adalah alat *wireframing UI* dengan kualitas rendah dan cepat yang mereproduksi pengalaman membuat sketsa di *notepad* atau papan tulis, tetapi menggunakan komputer. *Balsamiq Mockup* benar-benar memaksa anda untuk fokus pada struktur dan konten (Balsamiq.com, 2019). Fitur dan kelebihan *balsamiq mockup* adalah sebagai berikut:

1. Menciptakan tampilan antarmuka *user* dan komponen lainnya dengan mudah.

2. *Import* dan *export* hasil proyek ke pdf atau png.
3. Dapat digunakan secara *offline* disertai fitur berbagi ke *user* lainnya.
4. Dapat dengan mudah menggunakan *drag* dan *drop*.
5. Fitur detail lebih banyak dibandingkan *software low fidelity* lainnya.

2.19. **Black Box Testing**

Black box testing merupakan sebuah pengujian fungsional dimana proses pengujiannya dikembangkan atas dasar program atau fungsi sistem secara luar. Pengujian didasarkan pada *detail* aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi dengan bisnis proses yang diinginkan oleh *customer*. *Tester* atau penguji melakukan pengujian dengan mencoba memasukkan data *input* dan mengamati *output* dari sistem tanpa perlu mengetahui struktur internal dan cara kerja dari program atau sistem (William, 2017).

Black box testing merupakan salah satu teknik pengujian perangkat lunak yang fokusnya pada spesifikasi fungsional dari perangkat lunak. Dengan adanya *black box testing* memungkinkan pengembang perangkat lunak untuk membuat kumpulan suatu keadaan *input* pada suatu program yang akan melatih seluruh syarat-syarat fungsionalnya (Achmad Yani, 2020).

Ada beberapa keuntungan menggunakan perangkat lunak dengan pengujian *black box testing*, yaitu:

1. Penguji tidak harus mempunyai pengetahuan yang khusus tentang bahasa pemrograman.
2. Pengujian ini membantu untuk mengungkapkan *ambigulitas* atau *inkonsistensi* dalam spesifikasi persyaratan, yang jika dilihat dari sudut pandang penggunaan.
3. *Programmer* dan *tester* memiliki ketergantungan satu sama lain.

Selanjutnya ada beberapa kekurangan dalam pengujian *black box testing*, yaitu:

1. Adanya kesulitan uji kasus tanpa spesifikasi yang jelas.
2. Adanya kemungkinan pengulangan tes yang sudah dilakukan oleh *programmer*.