

**PEMBUATAN SISTEM INFORMASI BEASISWA BERBASIS
WEB PADA PT. PUNDI MAS BERJAYA**

LAPORAN PRAKTIK KERJA INDUSTRI

Diajukan untuk memenuhi salah satu syarat UAS (Ujian Akhir Sekolah)

Oleh:

FLOURA EXVELERANSA

NIS. 22161022



**SMK PERMATA HARAPAN BATAM
JURUSAN REKAYASA PERANGKAT LUNAK
2024**

LEMBARAN PENGESAHAN SEKOLAH

**PEMBUATAN SISTEM INFORMASI BEASISWA BERBASIS
WEB PADA PT. PUNDI MAS BERJAYA**

Oleh:

FLOURA EXVELERANSA

NIS. 22161022

Jurusan Rekayasa Perangkat Lunak

Laporan ini telah disetujui dan mendapat pengesahan dari kepala sekolah dan
pembimbing SMK Permata Harapan pada tanggal 26 Mei 2023.

Disetujui oleh,

Pembimbing I

Pembimbing II

Miftahul Ilmi, S.Pd., S.Kom., M.Pd.T
NIK. 111603221

Julia Mutiara Rizkosa, S.Pd
NIK. 112207371

Kepala Sekolah

Srima Deliana Damanik, S.Pd
NUPTK. 7662762663130132

LEMBARAN PENGESAHAN PERUSAHAAN
PEMBUATAN SISTEM INFORMASI BEASISWA BERBASIS
WEB PADA PT. PUNDI MAS BERJAYA

Oleh:

FLOURA EXVELERANSA

NIS. 22161022

Jurusan Rekayasa Perangkat Lunak

Laporan ini telah disetujui dan mendapat pengesahan dari PT. Pundi Mas Berjaya
serta pembimbing SMK Permata Harapan pada tanggal 26 Januari 2024.

Disetujui oleh,

Koordinator Prakerind

Pembimbing

Misriyadi, S.Kom

Misriyadi, S.Kom

Kepala Bagian Personalia

Anju Martunas Sinaga, S.E

KATA PENGANTAR

Puji dan syukur kepada Tuhan Yang Maha Esa telah memberikan rahmat-nya sehingga penulis dapat menyelesaikan PRAKERIND dengan lancar dan sukses. Penyusunan laporan ini tentunya tidak terlepas dari berbagai bantuan dan dukungan dari berbagai pihak sehingga laporan ini dapat diselesaikan dengan baik. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Bapak Miftahul Ilmi, S.Pd., S. Kom., M.Pd.T. dan Ibu Julia Mutiara Rizkosa, S.Pd., selaku Ketua Jurusan Rekayasa Perangkat Lunak, sekaligus sebagai guru pembimbing I dan selaku guru Bahasa Indonesia, sekaligus sebagai guru pembimbing II.
2. Bapak Misriyadi, s.kom selaku pembimbing di PT. Pundi Mas Berjaya.
3. Bapak Misriyadi, s.kom selaku Koordinator Praktik Kerja Industri di PT. Pundi Mas Berjaya.
4. Ibu Srima Deliana Damanik, S.Pd, selaku Kepala Sekolah SMK Permata Harapan.
5. Seluruh guru dan teman-teman yang berada di SMK Permata Harapan.
6. Orang tua dan seluruh pihak yang telah membantu dan mendukung secara langsung maupun tidak langsung.

Akhir kata, penulis ingin mengucapkan terima kasih kepada semua pihak yang telah membantu dan mendukung selama kegiatan PRAKERIND berlangsung. Penulis juga memohon maaf atas kekurangan dalam laporan ini dan berharap kritik serta saran yang membangun dapat diberikan untuk perbaikan di masa depan.

Batam, 26 Januari 2024

Penulis,

Floura Exveleransa

NIS. 22161022

ABSTRAK

PEMBUATAN SISTEM INFORMASI BEASISWA BERBASIS WEB PADA PT. PUNDI MAS BERJAYA

Oleh:

FLOURA EXVELERANSA

NIS. 22161022

Laporan ini mengimplementasikan sistem informasi beasiswa berbasis web pada PT. Pundi Mas Berjaya. Ini mencakup cara pembuatannya dan penggunaannya. Penelitian ini memberikan pemahaman tentang pembuatan sistem informasi beasiswa berbasis web untuk meningkatkan efisiensi pengelolaan data beasiswa dan mempermudah akses informasi bagi penggunanya.

PT Pundi Mas Berjaya adalah perusahaan IT yang fokus pada pengembangan solusi perangkat lunak seperti, *website*, aplikasi, dan sistem informasi. Solusi berbasis layanan untuk para pelanggan yang tersebar di beberapa negara dan memiliki model pengembangan on-site dan off-site. Sejak berdiri pada tahun 2014, Perusahaan telah merancang, mengembangkan dan digunakan banyak solusi di bidang properti, otomotif, transportasi, pengiriman makanan, pengiriman barang dan ecommerce.

Hasil dari implementasi adalah sebuah *website* sistem informasi beasiswa yang dapat membantu pengguna dalam mengelola data beasiswa seperti informasi siswa, pendapatan beasiswa, nilai, dan data lainnya melalui antarmuka *website*. Dalam antarmuka *website* beasiswa juga terdapat berbagai menu yang disajikan secara intuitif dan mudah digunakan oleh pengguna.

Kata kunci: Sistem Informasi Beasiswa, *Website*

DAFTAR ISI

LEMBARAN PENGESAHAN SEKOLAH.....	i
LEMBARAN PENGESAHAN PERUSAHAAN.....	ii
KATA PENGANTAR.....	iii
ABSTRAK	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	vi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penulisan Laporan.....	3
1.4 Manfaat PRAKERIND.....	3
1.4.1 Bagi Penulis	3
1.4.2 Bagi Perusahaan	3
1.4.3 Bagi Sekolah	3
1.5 Sistematika Penulisan.....	4
BAB II TINJAUAN UMUM	5
2.1 Latar Belakang Perusahaan	5
2.2 Visi dan Misi Perusahaan	6
2.3 Struktur Perusahaan.....	6
2.3 Peraturan Perusahaan	7
BAB III LANDASAN TEORI.....	8
3.1 Pembuatan Sistem	8
3.2 Sistem Informasi	10
3.3 Sistem Informasi Beasiswa Berbasis <i>Web</i>	12
3.4 Bahasa Pemograman	17
3.4.1 PHP Menggunakan Konsep OOP	17
3.4.2 HTML (<i>HyperText Markup Language</i>)	22
3.4.3 CSS (<i>Cascading Style Sheet</i>)	25
3.5 Sistem Keamanan Informasi	28
3.6 Data Flow Diagram (DFD)	30
3.7 Unified Modelling Language (UML).....	32

3.8	Entity Relationship Diagram (ERD)	34
3.9	Normalisasi.....	35
3.10	Database	36
3.10.1	Contoh – contoh Database	37
3.10.1	MYSQL.....	37
<u>DAFTAR PUSTAKA</u>		
<u>LAMPIRAN</u>		
<u>BIODATA</u>		
<u>MOTTO</u>		

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1 Latar Belakang

Praktik Kerja Industri (PRAKERIND) adalah kegiatan akademik yang wajib dilakukan oleh seluruh siswa/i yang mengambil sekolah SMK (Sekolah Menengah Kejuruan). Adanya PRAKERIND, para siswa/siswi dapat mengetahui dan mempraktikkan secara langsung bagaimana proses kerja dan juga pelajaran pertama untuk beradaptasi dengan dunia kerja sebelum nanti para siswa bekerja setelah lulus dari sekolah. Oleh karena itu, diharapkan siswa/siswi dapat memperoleh ilmu pengetahuan yang lebih luas dan benar-benar siap untuk menghadapi dunia kerja setelah lulus dari sekolah sebab sudah mendapatkan pengetahuan dan pengalaman dari PRAKERIND.

Di era modern ini, sistem informasi telah berkembang pesat yang membawa pengaruh besar bagi kehidupan manusia sehari-hari di hampir segala bidang, salah satunya bidang pendidikan. Kebutuhan akan sistem informasi yang cepat dan tepat menuntut beberapa lembaga, salah satunya lembaga pendidikan untuk meningkatkan sistem mereka menjadi yang lebih efisien untuk memberi pelayanan terbaik. Bukan hanya dengan penggunaan komputer dengan spesifikasi mumpuni saja, tetapi dibutuhkan juga sistem informasi yang akurat untuk pengelolaan data.

Saat ini masih banyak lembaga pendidikan, contohnya sekolah, yang mengandalkan sistem informasi yang bersifat manual yang akan memakan banyak waktu dan berjalan cukup lama, di mana sistem ini juga memerlukan ketelitian saat penulisan data. Apabila terdapat kesalahan dengan data yang dicatat, maka harus dilakukan pengecekan dan perbaikan data yang ditulis sebelumnya. Masalah lain yang timbul adalah pencarian data yang dilakukan secara manual juga akan memakan waktu yang lama.

Sistem informasi beasiswa ini adalah aplikasi berbasis *website* yang dibuat untuk melakukan pengolahan data beasiswa salah satunya di sekolah atau kampus dengan cepat dan akurat. Dibantu dengan penerapan teknologi komputer, data yang diolah dapat menjadi data yang berharga, dan tersimpan dengan aman. Dengan adanya sistem informasi beasiswa ini diharapkan dapat membantu pihak sekolah atau kampus dalam pendataan siswa/mahasiswa, pengolahan nilai, dan pembuatan rapor dengan efisien. Sistem informasi beasiswa ini juga dapat mempermudah kegiatan operasional sehari-hari, seperti absensi, dan penilaian.

Data-data yang tersimpan di dalam sistem informasi beasiswa ini dapat diakses di mana pun dan kapan pun. Hal ini sangat efisien karena guru tidak perlu meluangkan waktu dan tenaga untuk ke sekolah jika membutuhkan data. Selain itu, sistem informasi siswa juga dapat memberikan kemudahan dalam *monitoring* perkembangan siswa secara individu dan memberikan informasi yang akurat mengenai prestasi dan kekurangan yang dimiliki siswa. Hal ini dapat membantu guru dalam memberikan perhatian khusus dan membantu siswa dalam meningkatkan prestasi akademiknya. Dengan adanya sistem informasi beasiswa, diharapkan dapat membantu lembaga pendidikan, salah satunya kampus dalam mengelola data mahasiswa maupun informasi akademik lainnya secara efisien. Dengan alasan tersebut, laporan PKL telah dirancang dan diimplementasikan dengan judul "Pembuatan Sistem Informasi Beasiswa Berbasis Web".

1.2 Rumusan Masalah

Tujuan dari rumusan masalah adalah untuk mengidentifikasi masalah yang akan dipecahkan. Berdasarkan latar belakang di atas maka dapat kita rumuskan masalah sebagai berikut.

1. Bagaimana proses pembuatan sebuah Sistem Informasi Beasiswa yang berbasis Web ?
2. Bagaimana cara penggunaan Sistem Informasi Beasiswa yang berbasis Web ?
3. Bagaimana sistem keamanan pada Sistem Informasi Beasiswa yang berbasis Web?

1.3 Tujuan Penulisan Laporan

Penulisan laporan pelaksanaan PRAKERIND ini bertujuan untuk melatih siswa yang melaksanakan PRAKERIND dapat berpikir secara kritis dan kreatif. Berdasarkan masalah di atas maka dapat kita rumuskan tujuan sebagai berikut.

1. Mengetahui cara pembuatan sebuah Sistem Informasi Beasiswa yang berbasis Web.
2. Mengetahui cara penggunaan Sistem Informasi Beasiswa yang berbasis Web.
3. Mengetahui sistem keamanan pada Sistem Informasi Beasiswa yang berbasis Web.

1.4 Manfaat PRAKERIND

Adapun manfaat dari PRAKERIND bagi penulis, perusahaan, dan sekolah adalah sebagai berikut.

1.4.1 Bagi Penulis

Manfaat yang diperoleh penulis setelah melakukan PRAKERIND adalah sebagai berikut.

1. Melatih keterampilan yang dimiliki dalam lapangan kerja dengan menerapkan teori yang sudah dipelajari di sekolah.
2. Melatih diri untuk menjadi lebih disiplin dan profesional dalam dunia kerja.
3. Menambah wawasan dan pengalaman dalam bekerja.

1.4.2 Bagi Perusahaan

Manfaat yang diperoleh perusahaan setelah penulis melakukan PRAKERIND adalah sebagai berikut.

1. Meningkatkan produktivitas perusahaan.
2. Meningkatkan citra perusahaan.

1.4.3 Bagi Sekolah

Manfaat yang diperoleh sekolah setelah penulis melakukan PRAKERIND adalah sebagai berikut.

1. Meningkatkan popularitas sekolah.
2. Meningkatkan citra sekolah.

1.5 Sistematika Penulisan

Sistematika penulisan dalam laporan PRAKERIND ini bertujuan untuk menjelaskan setiap bagian dalam laporan secara rinci dan terstruktur agar pembaca dapat dengan mudah memahami isi laporan ini. Selain itu, penulisan akan disajikan secara formal dan menggunakan bahasa yang jelas serta mudah dipahami. Sistematika penulisan dalam laporan ini adalah sebagai berikut.

1. BAB I PENDAHULUAN

Pada BAB ini diterangkan mengenai Latar Belakang, Rumusan Masalah, Tujuan Pembuatan Laporan, Manfaat PRAKERIND, dan Sistematika Penulisan. Selain itu, bab ini juga berfungsi sebagai pengantar untuk memberikan gambaran umum tentang isi dari laporan PRAKERIND ini.

2. BAB II TINJAUAN UMUM

Pada BAB ini diterangkan mengenai Latar Belakang Perusahaan, Visi Dan Misi Perusahaan, Dan Peraturan Perusahaan. Selain itu, pada bab ini juga akan dibahas mengenai struktur organisasi perusahaan dan deskripsi singkat mengenai setiap departemen atau bagian yang ada dalam perusahaan.

3. BAB III LANDASAN TEORI

Pada BAB ini diterangkan mengenai Pengertian Pembuatan Sistem, Pengertian Sistem Informasi, Pengertian Sistem Informasi Beasiswa Berbasis Web, Pengertian Bahasa Pemrograman.

4. BAB IV PERANCANGAN

Pada BAB ini diterangkan mengenai Pembuatan Dan Implementasi Sistem Informasi Beasiswa Berbasis Web.

5. BAB V PENUTUP

Pada BAB ini, dijelaskan mengenai kesimpulan dan saran yang diambil dari hasil penelitian atau pengalaman yang telah dijabarkan sebelumnya. Selain itu, disampaikan pula saran-saran yang dapat menjadi acuan untuk perbaikan dan pengembangan di masa depan.

BAB II

TINJAUAN UMUM

2.1 Latar Belakang Perusahaan

PT. Pundi Mas Berjaya adalah salah satu penyedia solusi perangkat lunak di pasar global yang memberikan solusi bisnis mengadopsi teknologi informasi terkini. Solusi berbasis layanan untuk para pelanggan yang tersebar di beberapa negara dan memiliki model pengembangan *on-site* dan *off-site*. Sejak berdiri pada tahun 2014, Perusahaan telah merancang, mengembangkan dan digunakan banyak solusi di bidang properti, otomotif, transportasi, pengiriman makanan, pengiriman barang dan *ecommerce*.

PT.Pundi Mas Berjaya memiliki tim yang kuat dengan ukuran pelaksanaan proyek besar dan mengerahkan seluruh kemampuannya untuk memberikan pelayanan yang berkualitas tinggi kepada pelanggan. Salah satu kunci keberhasilan adalah adaptasi terhadap berbagai macam dan aneka kebutuhan akan sistem informasi komputer sehingga menghasilkan manajemen kualitas, manajemen proyek, kebutuhan Infrastruktur dengan sempurna untuk memberikan kepuasan kepada para pelanggan. Pelayanan menggunakan media responsif *website*, *mobile website*, *mobile apps* untuk mendukung aktifitas yang lebih optimal.



Gambar 2.1 Latar Belakang Perusahaan

PT. Pundi Mas Berjaya selalu mengutamakan pelayanan secara profesional dengan memberikan citra yang baik untuk perusahaan. *Software-software* yang pernah dikeluarkan oleh perusahaan seperti.

1. *Listing Property Online*
2. *Transportasi Online*
3. *Aplikasi Bank Data*
4. *Flowers E-Commerce*
5. *Aplikasi E-Commerce*

2.2 Visi dan Misi Perusahaan

Sebuah perusahaan pasti memiliki visi dan misi, tidak terlewatkan juga perusahaan ini. Visi dan Misi perusahaan ini adalah sebagai berikut.

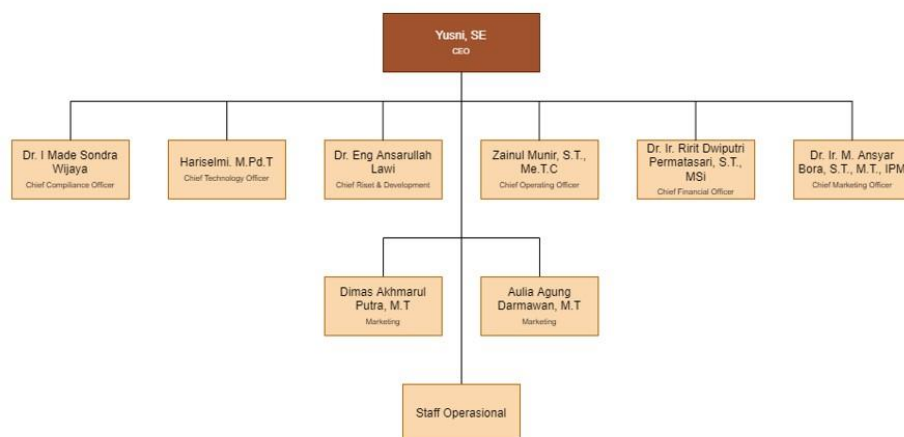
1. Visi Perusahaan

“Menjadi sebuah perusahaan Teknologi Informasi ternama berkelas dunia dan terus berinovasi menjadi yang terbaik”.

2. Misi Perusahaan

- a. Mengembangkan produk inovatif yang hemat biaya dan berkualitas baik.
- b. Menitik beratkan pada layanan dan solusi di bidang Teknologi Informasi kepada klien di seluruh dunia melalui tim yang kreatif dan profesional.

2.3 Struktur Organisasi Perusahaan



Gambar 2.1 Latar Belakang Perusahaan

2.4 Peraturan Perusahaan

Peraturan dalam perusahaan dibuat untuk melatih anggota perusahaan agar tercapainya kedisiplinan serta ketertiban yang lebih baik. Peraturan-peraturan yang ada di PT. Pundi Mas Berjaya yaitu sebagai berikut.

1. Jam kerja

Jam Kerja	Keterangan
08.00 s/d 17.00	Senin s/d Kamis
12.00 s/d 13.00	Istirahat
08.00 s/d 16.00	Sabtu s/d Minggu

Gambar 2.1 Jam Kerja

- Karyawan wajib minta izin terlebih dahulu kepada pemimpin kerjanya, apabila akan datang terlambat, sakit, meninggalkan pekerjaan atau pulang sebelum waktunya.
- Karyawan yang tidak masuk kerja dikarenakan sakit atau karena alasan lain wajib memberitahukan ketidakhadirannya pada hari pertama tidak hadir bekerja dengan cara apa pun dan sudah memberikan surat keterangan dokter / bukti ketidakhadiran yang sah dan kuat selambatnya pada hari pertama kerja.
- Wajib melaksanakan tugas yang diberikan oleh pemimpin perusahaan. Melaksanakan tugas yang diberikan oleh pemimpin perusahaan merupakan bagian dari tanggung jawab karyawan dalam menjalankan tugas dan fungsi kerjanya. Sebagai karyawan yang profesional, kita harus selalu siap dan bertanggung jawab untuk menjalankan tugas-tugas yang diberikan oleh pemimpin perusahaan dengan baik dan tepat waktu.

BAB III

LANDASAN TEORI

3.1 Pembuatan Sistem

Pembuatan sistem adalah proses merancang, mengembangkan, dan mengimplementasikan sebuah sistem komputer atau informasi yang terdiri dari perangkat lunak, perangkat keras, prosedur, dan orang-orang yang terlibat untuk menjadi tujuan tertentu dalam suatu organisasi atau lingkungan. Proses ini melibatkan analisis kebutuhan, perencanaan, desain, pengkodean, pengujian, dan pemeliharaan sistem tersebut.

Menurut Michael S.Scott Morton, pembuatan sistem adalah proses yang melibatkan transformasi organisasi menjadi solusi teknologi informasi yang efektif. Morton menekankan pentingnya pemahaman yang mendalam terhadap kebutuhan bisnis dalam merancang sistem yang sesuai. Menurut James A.Senn, memberikan sudut pandang yang menekankan pada aspek desain dan pengembangan sistem. Baginya, pembuatan sistem tidak hanya tentang menghasilkan produk teknologi, tetapi juga menciptakan struktur yang dapat mendukung proses bisnis dengan efisien. Senn menekankan perlunya kolaborasi yang kuat antara pemangku kepentingan bisnis dan pengembang teknologi dalam proses pembuatan sistem.

Sedangkan Mary Shaw mengatakan, pentingnya pendekatan sistematis dan metodologi yang terstruktur dalam pembuatan sistem. Baginya, sebuah sistem yang baik tidak hanya berfungsi dengan baik disaat pertama kali diimplementasikan, tetapi juga dapat beradaptasi dan berkembang seiring waktu sesuai dengan perubahan kebutuhan dan lingkungan. Adanya pembuatan sistem berfungsi sebagai peningkatan efisiensi operasional, akurasi dan konsistensi, memfasilitasi pengambilan keputusan, dan meningkatkan layanan pelanggan. Dari ketiga pengertian para ahli diatas, dapat diartikan bahwa pembuatan sistem adalah proses yang melibatkan berbagai disiplin ilmu serta pemangku kepentingan.

Dalam pembuatan sistem, pendekatan yang holistik dan terstruktur menjadi kunci dalam merancang, mengembangkan, dan mengimplementasikan solusi teknologi informasi yang efektif. Berdasarkan konsep yang disampaikan oleh Michael S.Scott Morton, proses ini bukan hanya tentang transformasi teknologi, tetapi juga mengintegrasikan kebutuhan bisnis dengan solusi yang sesuai. Pentingnya memahami kebutuhan bisnis yang mendalam memungkinkan pengembang untuk menciptakan sistem yang tidak hanya memenuhi persyaratan teknis, tetapi juga memberikan nilai tambah kepada organisasi. Selain itu, perspektif James A.Senn menekankan kolaborasi antara pemangku kepentingan bisnis dan pengembang teknologi. Dengan demikian, pembuatan sistem menjadi lebih dari sekadar menghasilkan produk teknologi, tetapi juga menciptakan struktur yang dapat mengoptimalkan proses bisnis. Kolaborasi yang kuat memungkinkan pemahaman yang lebih baik terhadap kebutuhan dan tujuan bisnis, sehingga sistem yang dihasilkan dapat mengakomodasi perubahan dan meningkatkan efisiensi operasional secara berkelanjutan. Mary Shaw menyoroti pentingnya pendekatan sistematis dan metodologi yang terstruktur dalam pembuatan sistem. Sistem yang baik bukan hanya berfungsi saat pertama kali diimplementasikan, tetapi juga mampu beradaptasi dan berkembang seiring waktu. Fleksibilitas ini menjadi kunci dalam menghadapi dinamika lingkungan bisnis yang terus berubah, sehingga sistem dapat terus memberikan nilai tambah kepada organisasi.

Dari sudut pandang para ahli tersebut, dapat disimpulkan bahwa pembuatan sistem adalah proses yang kompleks dan multi disiplin. Dalam upaya mencapai tujuan tertentu dalam suatu organisasi atau lingkungan, diperlukan pemahaman yang mendalam terhadap kebutuhan bisnis, kolaborasi yang kuat antara pemangku kepentingan, dan pendekatan sistematis yang memungkinkan sistem untuk berkembang seiring waktu. Dengan demikian, pembuatan sistem bukan hanya tentang menghasilkan produk teknologi, tetapi juga mengoptimalkan proses bisnis, meningkatkan efisiensi operasional, dan memberikan nilai tambah yang berkelanjutan kepada organisasi.

3.2 Sistem informasi

Sistem informasi merupakan sebuah kerangka kerja atau struktur yang terdiri dari berbagai elemen yang saling terkait dan bekerja bersama-sama untuk mengumpulkan, menyimpan, memanipulasi, dan menyebarkan informasi dalam sebuah organisasi atau entitas tertentu. Pengertian ini mencakup aspek teknologi informasi, proses bisnis, dan manusia yang terlibat dalam pengelolaan informasi. Pada tingkat yang lebih mendalam, sistem informasi mencakup perangkat lunak, perangkat keras, jaringan komunikasi, serta data yang digunakan untuk mengumpulkan, menyimpan, dan mengelola informasi. Namun, sistem informasi tidak hanya tentang teknologi tetapi juga mencakup proses bisnis dan kebijakan organisasi yang mengatur bagaimana informasi dikumpulkan, diproses, digunakan. Selain itu, sistem informasi juga melibatkan peran manusia dalam mengelola, mengoperasikan, dan menggunakan sistem tersebut. Ini termasuk penggunaan sistem yang memasukkan, mengolah, dan mengambil informasi dari sistem, serta personel TI yang bertanggung jawab atas pengembangan, pemeliharaan (*maintenance input*), dan dukungan sistem.

James Edward memandang sistem informasi merupakan kombinasi dari tiga komponen utama, yaitu teknologi informasi, proses bisnis, dan orang-orang yang terlibat. Teknologi informasi mencakup perangkat keras, perangkat lunak, dan infrastruktur jaringan yang digunakan untuk mengelola informasi, sedangkan proses bisnis mengacu pada serangkaian langkah atau aktivitas yang dilakukan untuk mengubah *input* menjadi *output* yang bernilai bagi organisasi. Sementara itu, orang-orang yang terlibat dalam sistem informasi termasuk pengguna sistem yang menginput, memproses, dan mengambil informasi dari sistem, serta personel TI yang bertanggung jawab atas pengembangan, pemeliharaan, dan dukungan sistem. Pengertian ini mencakup pemahaman bahwa sistem informasi tidak hanya berfokus pada teknologi semata, tetapi juga melibatkan interaksi kompleks antara teknologi, proses bisnis, dan manusia. Dan kesimpulan sistem informasi adalah menjadi sebuah kerangka kerja yang holistik.

Menurut Kenneth C. Laudon dan Jane P. Laudon, sistem informasi adalah kumpulan dari komponen-komponen yang saling terkait, yang bekerja bersama-sama untuk mengumpulkan, memproses, menyimpan, dan menyebarkan informasi yang diperlukan untuk mendukung pengambilan keputusan, koordinasi, kontrol, analisis, dan visualisasi dalam suatu organisasi.

Sistem informasi adalah gabungan dari orang-orang, teknologi, prosedur, database, dan model matematika yang diorganisasi untuk memecahkan masalah bisnis dan menyediakan dukungan bagi pengambilan keputusan dan kontrol dalam organisasi. Sistem informasi memiliki peran penting dalam mengelola informasi, mengoptimalkan proses bisnis, dan mendukung pengambilan berbagai keputusan di berbagai tingkatan organisasi. Melalui integrasi komponen-komponen yang berbeda, sistem informasi membantu organisasi dalam mengumpulkan, menyimpan, memproses, dan menyebarkan informasi yang diperlukan untuk mencapai tujuan. pemahaman yang komprehensif tentang sistem informasi menurut berbagai ahli dapat memberikan landasan yang kokoh untuk analisis dan evaluasi sistem informasi dalam suatu organisasi dan dapat mengevaluasi sistem informasi yang dimiliki oleh organisasi memenuhi kebutuhan bisnis dan memberikan kontribusi yang signifikan dalam pencapaian tujuan organisasi tersebut. Selain itu, pemahaman yang mendalam tentang sistem informasi yang dapat membantu dalam merancang dan mengimplementasikan perubahan atau peningkatan yang diperlukan dalam sistem informasi organisasi untuk meningkatkan efisiensi, produktivitas, dan daya saing.

Fungsi-fungsi dari sistem informasi sangat beragam dan meliputi berbagai aspek organisasi, salah satunya seperti pengumpulan dan penyimpanan data. Sistem informasi mengumpulkan data dari berbagai sumber baik internal maupun eksternal organisasi. Data ini dapat berupa transaksi bisnis, informasi pelanggan, informasi pasar, dan lainnya. Keberadaan sistem informasi memungkinkan organisasi untuk meningkatkan kinerja, mengambil keputusan yang lebih baik, dan mencapai tujuan pengguna dengan lebih efektif dan efisien.

3.3 Sistem informasi Beasiswa Berbasis Web

Sistem informasi beasiswa berbasis *website* adalah sebuah platform digital yang dirancang untuk mengelola informasi seputar pemberian beasiswa. Dalam konteks ini, “sistem informasi” merujuk pada infrastruktur teknologi yang terdiri dari perangkat keras, perangkat lunak, basis data, serta prosedur-prosedur yang digunakan untuk mengumpulkan, menyimpan, mengolah, dan menyebarkan informasi terkait beasiswa.

Berikut adalah penjelasan tentang cara penggunaan *website* sistem informasi beasiswa.

1. Halaman utama (*index*)

Ketika pertama kali membuka *website* sistem informasi beasiswa, anda akan diarahkan ke halaman utama atau *index*. Halaman ini biasanya dirancang untuk memberikan ringkasan tentang berbagai fitur yang tersedia di situs tersebut. Disini, anda mungkin akan menemukan informasi tentang jenis beasiswa yang tersedia, syarat dan ketentuan, serta tautan menuju halaman-halaman penting lainnya seperti daftar beasiswa, panduan aplikasi, dan informasi kontak.

2. Daftar beasiswa

Pada halaman ini terdapat halaman khusus yang berisi daftar semua beasiswa yang tersedia. Pada halaman ini, anda dapat menemukan informasi tentang setiap beasiswa termasuk nama beasiswa, penyelenggara, deskripsi singkat, syarat pendaftaran, batas waktu, dan tautan untuk mengajukan permohonan.

3. Proses aplikasi

Cara mengajukan beasiswa, Anda perlu mengikuti proses aplikasi yang biasanya terdiri dari beberapa langkah. Termasuk pembuatan akun pengguna, mengisi formulir aplikasi *online*, mengunggah dokumen pendukung seperti transkrip nilai, dan menyerahkan aplikasi sesuai batas waktu yang ditentukan. Setiap langkah biasanya dijelaskan secara detail di situs *website*.

4. Layanan Bantuan dan Dukungan

Jika pengguna mengalami kesulitan atau memiliki pertanyaan tentang penggunaan situs *website* atau proses aplikasi, biasanya akan ada layanan bantuan dan dukungan yang tersedia. Layanan ini bisa berupa pusat bantuan *online*, formulir kontak, atau nomor telepon yang dapat dihubungi untuk mendapatkan bantuan dari staf administratif.

5. Pengumuman Beasiswa

Halaman ini digunakan untuk mengumumkan berbagai informasi penting terkait dengan beasiswa, seperti pengumuman pembukaan periode pendaftaran, perubahan dalam persyaratan aplikasi, pengumuman penerimaan beasiswa, dan berita lain yang relevan. Pengguna dapat mengakses halaman ini untuk mendapatkan informasi terkini secara langsung dari penyelenggara beasiswa.

6. Penilaian dan Seleksi

Halaman ini memberikan informasi tentang proses penilaian dan seleksi penerimaan beasiswa. Pengguna dapat mempelajari kriteria yang digunakan untuk menilai aplikasi, bobot nilai dari setiap kriteria, serta informasi tentang siapa yang terlibat dalam proses penilaian tersebut. Hal ini dapat membantu pengguna untuk memahami lebih baik bagaimana keputusan diperoleh dan apa yang diharapkan oleh penyelenggara beasiswa.

7. Evaluasi

Halaman ini memberikan pengguna kesempatan untuk memberi umpan balik tentang pengalaman mereka dalam menggunakan sistem informasi beasiswa. Pengguna dapat memberikan saran, kritik, atau testimoni yang dapat membantu dalam meningkatkan kualitas dan efektivitas platform. Selain itu, halaman ini juga dapat digunakan untuk menyajikan hasil evaluasi dan perbaikan yang telah dilakukan berdasarkan umpan balik sebelumnya.

Simpulan dari halaman-halaman tersebut adalah sistem informasi beasiswa berbasis web merupakan sebuah platform digital yang dirancang untuk memberikan akses yang mudah dan transparan terhadap informasi seputar pemberian beasiswa. Dengan infrastruktur teknologi yang mencakup perangkat keras, perangkat lunak, basis data, dan prosedur-prosedur tertentu, sistem ini bertujuan untuk mengumpulkan, menyimpan, mengolah, dan menyebarkan informasi terkait beasiswa kepada pengguna. Ketika pengguna pertama kali mengunjungi situs *website* sistem informasi beasiswa, mereka akan diarahkan ke halaman utama atau *index*. Halaman ini berfungsi sebagai pintu gerbang utama yang memberikan ringkasan tentang berbagai fitur yang tersedia di platform. Disini, pengguna dapat menemukan informasi tentang jenis beasiswa yang tersedia, syarat dan ketentuan, serta tautan menuju halaman-halaman penting lainnya seperti daftar beasiswa, panduan aplikasi, dan informasi kontak.

Salah satu halaman penting lainnya adalah halaman daftar beasiswa. Halaman ini, pengguna dapat menemukan daftar lengkap semua beasiswa yang tersedia. Informasi yang disediakan mencakup nama beasiswa, penyelenggara, deskripsi singkat, syarat pendaftaran, batas waktu, dan tautan untuk mengajukan permohonan. Pengguna juga dapat menggunakan filter atau pencarian untuk menemukan beasiswa yang sesuai dengan kebutuhan atau kriteria pengguna. Selanjutnya, proses aplikasi menjadi fokus utama pengguna dalam mencari beasiswa. Halaman ini memberikan panduan langkah demi langkah tentang cara mengajukan beasiswa, mulai dari pembuatan akun pengguna, mengisi formulir, hingga mengunggah dokumen pendukung seperti transkrip nilai. Selain itu, sistem informasi beasiswa juga menyediakan layanan bantuan dan dukungan bagi pengguna yang mengalami kesulitan atau memiliki pertanyaan tentang penggunaan situs *website* atau proses aplikasi tersebut. Layanan ini bisa berupa pusat bantuan *online*, formular kontak, atau nomor telepon yang dapat dihubungi untuk mendapatkan bantuan dari staf administratif.

Pengumuman beasiswa juga menjadi bagian penting dari platform ini. Halaman ini digunakan untuk menginformasikan pengguna tentang berbagai informasi penting terkait dengan beasiswa, seperti pengumuman pembukaan periode pendaftaran, perubahan dalam persyaratan aplikasi, pengumuman penerimaan beasiswa, dan berita lainnya yang relevan. Selanjutnya, pengguna diberikan wawasan tentang proses penilaian dan seleksi penerimaan beasiswa melalui halaman khusus. Disini, pengguna dapat mempelajari kriteria yang digunakan untuk menilai aplikasi, bobot nilai dari setiap kriteria, serta informasi tentang siapa yang terlibat dalam proses penilaian tersebut.

Terakhir, halaman evaluasi memberikan pengguna kesempatan untuk memberikan umpan balik tentang pengalaman mereka menggunakan sistem informasi beasiswa. Pengguna dapat memberikan saran, kritik, atau testimoni yang dapat membantu dalam meningkatkan kualitas dan efektivitas platform secara keseluruhan. Selain itu, halaman ini juga dapat digunakan untuk menyajikan hasil evaluasi dan perbaikan yang telah dilakukan berdasarkan umpan balik sebelumnya, menyajikan platform ini lebih responsif terhadap kebutuhan pengguna. Secara keseluruhan, sistem informasi beasiswa berbasis web bertujuan untuk mendukung akses pendidikan bagi individu atau kelompok yang membutuhkan bantuan keuangan. Dengan menyediakan informasi yang terperinci, panduan yang jelas, dan layanan dukungan yang efisien, platform ini memudahkan pengguna dalam setiap langkah dari proses pencarian, aplikasi, pengelolaan beasiswa, dan memberikan kesempatan untuk meraih pendidikan yang lebih baik dan mewujudkan impian mereka. Dengan menyediakan halaman-halaman tambahan ini, sistem informasi beasiswa berbasis *web* akan menjadi lebih lengkap dan beragam dalam memberikan dukungan kepada pengguna. Hal ini tidak hanya memudahkan mereka dalam menjalani proses pencarian dan aplikasi beasiswa, tetapi juga memberikan kesempatan untuk belajar, berbagi, dan terhubung dengan sesama anggota komunitas untuk mencapai tujuan pendidikan mereka.

3.4 Bahasa Pemrograman

Bahasa pemrograman adalah sebuah *syntax* atau perangkat lunak yang digunakan untuk membuat program komputer. Menurut Dipraja (2014), bahasa pemrograman memungkinkan seorang *programmer* untuk membuat suatu program aplikasi seperti *Borland Delphi*. Hal ini menunjukkan bahwa bahasa pemrograman adalah alat yang penting dalam proses pengembangan aplikasi dan memungkinkan *programmer* untuk mengomunikasikan instruksi kepada komputer dengan cara yang mudah dipahami oleh komputer. (Sumarno, 2019) Menurut Indrajani (2007), bahasa pemrograman adalah perangkat lunak atau *software* yang digunakan dalam proses pembuatan program melalui beberapa tahapan penyelesaian masalah. Dalam hal ini, bahasa pemrograman dilihat sebagai sebuah alat yang membantu *programmer* dalam memecahkan masalah dan mengembangkan solusi yang tepat. Sehingga, bahasa pemrograman tidak hanya menjadi sebuah *syntax*, tetapi juga menjadi sebuah proses dalam pengembangan aplikasi. (Sidik & Anton Suryoprayogo, 2015)

Suyanto (2006) menjelaskan bahwa bahasa pemrograman adalah bahasa-bahasa yang dipakai *programmer* untuk menuliskan kumpulan-kumpulan instruksi (program) dalam komputer. Definisi ini menunjukkan bahwa bahasa pemrograman adalah sebuah bahasa yang memiliki aturan dan struktur yang harus diikuti oleh *programmer* dalam menulis instruksi atau program. Oleh karena itu, memahami bahasa pemrograman secara mendalam sangatlah penting bagi *programmer*. Menurut Kusriani dan Koniyo (2007), bahasa pemrograman adalah perintah-perintah yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Definisi ini menunjukkan bahwa bahasa pemrograman dilihat sebagai sebuah kumpulan perintah yang harus dipahami oleh komputer dalam menjalankan program. Sehingga, bahasa pemrograman harus ditulis dengan benar dan sesuai dengan aturan dan struktur yang berlaku agar dapat dipahami oleh komputer. (Admin Media, 2022)

Berdasarkan pengertian para ahli di atas, dapat disimpulkan bahwa bahasa pemrograman adalah sebuah *syntax* atau perangkat lunak yang digunakan untuk membuat program komputer dengan menulis sekumpulan instruksi atau perintah yang dapat dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Bahasa pemrograman juga dapat dilihat sebagai proses dalam pengembangan aplikasi, di mana *programmer* menggunakan bahasa pemrograman untuk memecahkan masalah dan mengembangkan solusi yang tepat, sehingga pemahaman yang mendalam mengenai bahasa pemrograman sangat penting bagi *programmer*.

3.4.1 PHP Menggunakan Konsep OOP (*Objected Oriented Programming*)

Menurut Rasmus Lerdorf (1994), mengerti pentingnya pengembangan yang terstruktur dan mudah dipelihara. Dalam penggunaan OOP dengan PHP, ia menekankan bahwa OOP memungkinkan pengembang untuk membagi kode ke dalam unit-unit yang lebih kecil dan terorganisir, memfasilitasi pemeliharaan dan pengembangan berkelanjutan. Matt Zandstra (2004), seorang penulis dan pakar dalam pengembangan PHP, mengakui peran penting OOP dalam meningkatkan kualitas dan berkelanjutan kode. Dalam karyanya, ia menyoroti bagaimana konsep OOP seperti enkapsulasi, pewarisan, dan polimorfisme dapat digunakan dalam PHP untuk menciptakan aplikasi yang skalabel dan mudah diubah. Larry Garfield (2003), seorang pakar dalam pengembangan PHP dan kontributor utama *framework symphony*, menyoroti nilai strategis OOP dalam pengembangan aplikasi *website*. Menurutnya, penggunaan OOP dengan PHP memungkinkan pengembang untuk membuat arsitektur yang lebih fleksibel dan modular, memfasilitasi pengembangan kolaboratif dan peningkatan skala aplikasi dengan mudah. Larry Garfield telah berpartisipasi aktif dalam komunitas PHP, memberikan kontribusi dalam proyek-proyek sumber terbuka, dan berbagi pemikirannya tentang praktik pengembangan perangkat lunak. Dengan pemahaman dan penerapan konsep OOP yang tepat, pengembang PHP dapat membangun aplikasi *website* yang lebih kuat, lebih mudah dipelihara, dan dapat berkembang seiring waktu.

Berikut adalah beberapa jenis konsep OOP yang umum digunakan dalam pengembangan PHP.

1. Kelas

Kelas adalah *blueprint* untuk objek yang mendefinisikan atribut (properti) dan perilaku (metode) dari objek tersebut. Dalam PHP, kelas dideklarasikan menggunakan kata kunci '*class*', diikuti oleh nama kelas dan blok kode yang berisi properti dan metode.

2. Objek

Objek adalah instansi dari suatu kelas yang diciptakan dalam program. Setiap objek memiliki properti yang mewakili data yang dimilikinya, serta metode yang mendefinisikan perilaku objek tersebut. Objek dibuat menggunakan kata kunci '*new*' diikuti dengan nama kelas.

3. Enkapsulasi

Enkapsulasi adalah konsep yang memungkinkan properti dan metode suatu kelas untuk dibungkus bersama dalam satu unit yang disebut objek. Dengan enkapsulasi, properti kelas dapat diakses dan dimodifikasi hanya melalui metode yang ditentukan, sehingga meningkatkan keamanan dan integritas data.

4. Pewarisan (*Inheritance*)

Pewarisan adalah konsep yang memungkinkan kelas untuk mewarisi properti dan metode dari kelas lain yang disebut kelas induk atau superclass.

5. Polimorfisme

Polimorfisme adalah konsep objek untuk berperilaku secara berbeda tergantung pada konteks penggunaannya. Dalam PHP, polimorfisme dapat dicapai melalui penggunaan metode yang sama dengan implementasi yang berbeda dalam setiap kelas turunan.

6. Abstraksi

Konsep yang memungkinkan kelas untuk menyembunyikan detail internal implementasinya dan hanya mengekspos fungsionalitas yang diperlukan kepada pengguna.

Penggunaan konsep OOP (*Object Oriented Programming*) dalam PHP sering dipilih karena beberapa alasan yang mendasar.

Berikut adalah beberapa alasan mengapa PHP dengan konsep OOP sering digunakan.

1. Modularitas

Konsep OOP memungkinkan kode PHP untuk dibagi menjadi bagian-bagian yang lebih kecil atau modul, yang membuat pengembangan dan pemeliharaan lebih mudah karena kode terorganisir dengan baik.

2. Reusabilitas

Objek dan kelas dapat digunakan kembali di berbagai bagian aplikasi PHP, mengurangi redundansi kode dan meningkatkan efisiensi pengembangan.

3. Pewarisan (*Inheritance*)

PHP mendukung pewarisan, yang memungkinkan kelas baru untuk mewarisi sifat-sifat dan perilaku dari kelas yang sudah ada. Ini memungkinkan pengembang untuk membuat hierarki kelas yang terstruktur dan mudah dimengerti.

4. Pengelolaan kompleksitas

Dengan menggunakan OOP, kompleksitas aplikasi PHP dapat dikelola dengan lebih baik melalui pembagian menjadi objek-objek yang lebih kecil dan terpisah.

5. Polimorfisme

Konsep polimorfisme memungkinkan pengembang untuk menggunakan objek dari kelas yang berbeda dalam cara yang seragam, sehingga meningkatkan fleksibilitas dan modularitas kode.

6. Enkapsulasi (*Encapsulation*)

OOP memungkinkan enkapsulasi data dan fungsi ke dalam sebuah objek, yang menjaga integritas data dan mencegah akses yang tidak sah, sehingga meningkatkan keamanan dan keandalan aplikasi.

7. Pemeliharaan yang mudah

Kode PHP yang ditulis dengan konsep OOP cenderung lebih mudah dipahami, dimodifikasi, dan diperbaiki karena struktur yang terorganisir dengan baik.

8. Pengembangan kolaboratif

Dengan OOP, tim pengembang dapat bekerja secara paralel dan kolaboratif dalam pengembangan aplikasi PHP, karena setiap anggota tim dapat fokus pada bagian-bagian tertentu dari kode.

9. Skalabilitas

Struktur OOP memungkinkan aplikasi PHP untuk berkembang seiring waktu tanpa kehilangan keterbacaan dan kestabilan, sehingga lebih mudah untuk menangani perubahan dan peningkatan kebutuhan.

10. Kode yang bersih dan terstruktur

Dengan OOP, kode PHP cenderung lebih bersih dan terstruktur, memungkinkan pengembang untuk memahami fungsionalitas aplikasi dengan lebih baik dan memfasilitasi pemeliharaan jangka panjang.

Kesimpulan dari beberapa fungsi diatas adalah modularitas memungkinkan kode dipisahkan menjadi unit-unit kecil, yang membuat pengembangan menjadi lebih mudah. Reusabilitas objek dan kelas mengurangi duplikasi kode dan meningkatkan efisiensi. Pewarisan memungkinkan pembuatan hierarki kelas yang terstruktur dan mudah dimengerti, sedangkan polimorfisme meningkatkan fleksibilitas. Enkapsulasi memastikan keamanan dan integritas data, sementara pemeliharaan yang mudah dan pengembangan kolaboratif memperkuat efisiensi tim. Struktur OOP juga memastikan skalabilitas aplikasi PHP seiring waktu, sambil mempertahankan keterbacaan dan kestabilan kode. Kode yang bersih dan terstruktur mempermudah pemahaman dan pemeliharaan aplikasi, meningkatkan produktivitas tim, mendukung pengembangan aplikasi yang kuat dan berskala besar.

Berikut ini adalah penjelasan mengenai beberapa versi PHP yang signifikan dalam konteks OOP (*Object Oriented Programming*).

1. PHP 3

Pada PHP 3, dukungan untuk konsep OOP diperkenalkan. Namun, pendekatan OOP saat itu masih sederhana dan tidak begitu kuat.

2. PHP 4

PHP 4 menandai langkah besar dalam pengembangan OOP di PHP. dukungan untuk kelas dan objek yang sebenarnya diperkenalkan. Namun, implementasinya masih terbatas, dan paradigma OOP tidak seluruhnya terintegrasi dengan bahasa.

3. PHP 5

PHP 5 adalah titik balik dalam sejarah PHP dengan konsep OOP. Pada versi ini, fitur-fitur OOP diperluas secara signifikan. Diperkenalkan konsep seperti visibilitas (*public*, *private*, *protected*), *static methods* dan *properties*, *namespaces*, *interfaces*, *traits*, dan banyak lagi. Dengan PHP 5, pengembang dapat mengadopsi pendekatan OOP yang lebih kuat dan lebih terstruktur dalam pengembangan aplikasi mereka.

4. PHP 7

PHP 7 menghadirkan peningkatan kinerja yang signifikan dibandingkan dengan versi sebelumnya. Meskipun bukan secara eksklusif berkaitan dengan OOP, peningkatan kinerja ini sangat memengaruhi cara pengembang membangun aplikasi berorientasi objek, karena memungkinkan aplikasi yang lebih kompleks dan lebih besar untuk dijalankan dengan lebih efisien.

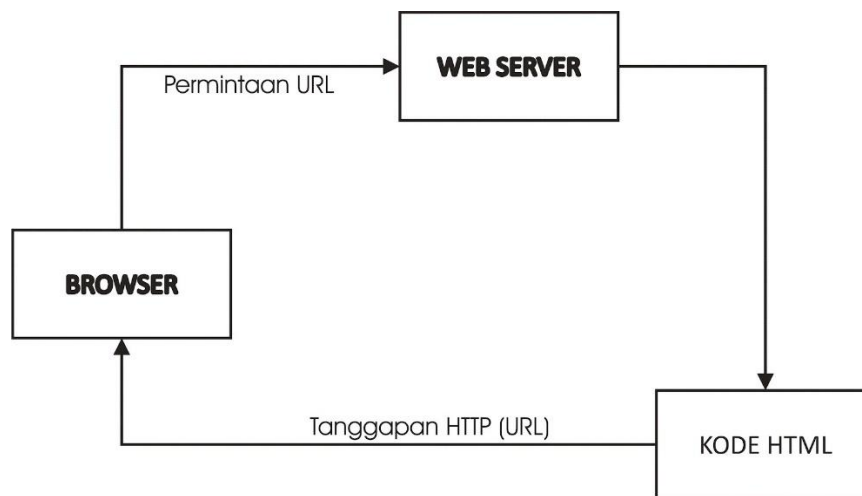
5. PHP 8

PHP 8 merupakan versi terbaru PHP (pada saat penulisan), dan membawa banyak peningkatan yang signifikan dalam hal OOP termasuk fitur-fitur seperti *union types*, *constructor property promotion*, *named arguments*, dan banyak peningkatan sintaksis lainnya.

3.4.2 HTML (*HyperText Markup Language*)

HTML adalah singkatan dari *Hypertext Markup Language*, yaitu sebuah standar untuk membuat halaman *web*. Menurut Lia Kuswayatno, HTML adalah standar untuk membuat halaman *web* yang memungkinkan pengguna untuk menavigasi dari satu halaman ke halaman lain dengan mengklik *link* teks. Dalam dunia *web*, HTML sering disebut sebagai *markup language* yang berfungsi untuk mendefinisikan struktur, *layout*, dan konten dari sebuah halaman *web*. Dengan HTML, *web developer* dapat membuat teks, gambar, *video*, dan elemen-elemen multimedia lainnya untuk ditampilkan di halaman *web*. HTML juga memungkinkan pengembang untuk membuat formulir, tombol, dan interaksi lainnya di halaman *web*. Menurut Nugroho (2006), HTML adalah bahasa pemformatan teks yang digunakan dalam dokumen yang dibagikan melalui jaringan komputer, atau yang biasa dikenal sebagai WWW (*World Wide Web*). HTML memungkinkan pengguna untuk memformat dokumen dengan berbagai tampilan dan konten, seperti teks, gambar, *video*, tabel, dan *link*. Bahasa ini memungkinkan pengguna untuk membuat tautan dan navigasi antara halaman *web*, dan untuk menyajikan informasi dalam berbagai format yang mudah dibaca dan dipahami.

Menurut Oktavian (2010), HTML merupakan suatu bahasa yang dikenal oleh *web browser*, agar mendapatkan informasi bahasa yang digunakan untuk membuat konten yang menarik secara visual sebagai lawan dari teks biasa. HTML memungkinkan pengembang *web* untuk menentukan tampilan dan format teks, gambar, dan elemen multimedia lainnya pada halaman *web*, serta menambahkan fungsi interaktif seperti formulir dan tombol. Bahasa ini juga memungkinkan pengembang untuk menambahkan *metadata*, seperti deskripsi halaman *web* dan kata kunci, untuk membantu mesin pencari memahami isi dari halaman *web* tersebut. Menurut Arief (2011), HTML adalah salah satu tipe *file* yang digunakan untuk membuat dokumen dan program berbasis *web*. Dengan menggunakan HTML, pengembang *web* dapat membuat struktur dan konten halaman *web* dengan mudah, dan menambahkan format dan tampilan yang menarik untuk meningkatkan pengalaman pengguna. (Lovata Andrean, 2022)



Gambar 3.4.2 Cara Kerja HTML

Kesimpulan dari beberapa penjelasan diatas adalah menyoroti peran HTML (*Hyper Text Markup Language*) sebagai fondasi utama dalam pembuatan halaman *web*. Dalam ekosistem *web*, HTML menjadi fondasi yang memungkinkan pengembang untuk mendefinisikan struktur, layout, dan konten dari sebuah halaman *web*. Melalui penggunaan HTML, para pengembang dapat dengan mudah menampilkan berbagai elemen seperti teks, gambar, video, dan elemen multimedia lainnya secara interaktif. Lebih dari sekadar itu, HTML memungkinkan penggunaan metadata untuk membantu mesin pencari dalam memahami konten halaman *web* tersebut.

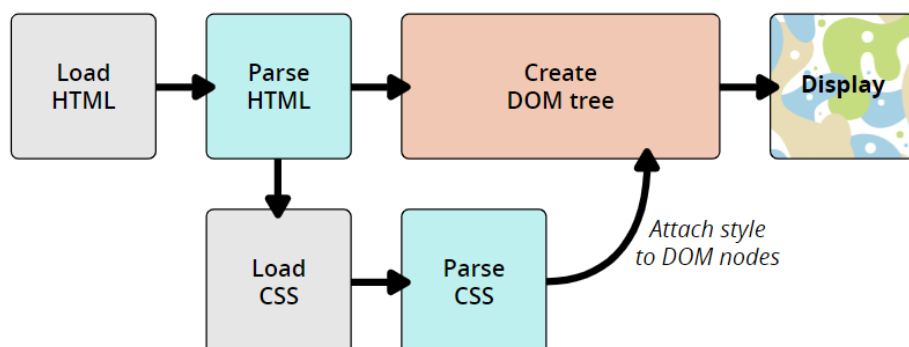
Memanfaatkan kemampuan HTML, pengembang dapat menciptakan pengalaman *web* yang menarik dan informatif bagi pengguna. Melalui pemformatan teks, gambar, dan elemen lainnya, HTML memungkinkan pembuatan halaman *web* yang menarik secara visual. Selain itu, penambahan *metadata* seperti deskripsi halaman dan kata kunci memperkuat visibilitas halaman tersebut di mesin pencari. Dengan demikian, HTML bukan hanya sebuah Bahasa *markup*, tetapi juga fondasi yang menggerakkan interaksi dan pengalaman pengguna dalam menjelajahi konten *web* secara halus dan mudah dipahami. Dengan menggunakan HTML, pengembang *web* dapat membuat konten yang menarik secara visual.

Bahasa ini juga memungkinkan pengembang untuk menambahkan *metadata* dan navigasi antara halaman *web*, sehingga memudahkan pengguna untuk menavigasi di antara konten halaman *web*. Secara keseluruhan, HTML menjadi fondasi yang memungkinkan pengembang untuk menciptakan pengalaman web yang menarik dan inovatif bagi pengguna. Dengan memanfaatkan kemampuan dasar HTML, pengembang dapat menciptakan tampilan visual yang menarik dan memperkaya pengalaman pengguna dalam menjelajahi konten *web*.

Selain itu, HTML juga memungkinkan pengembang untuk menyediakan konten yang dapat diakses oleh berbagai perangkat dan platform, tanpa memandang perangkat keras atau perangkat lunak yang digunakan pengguna. Dengan menggunakan standar HTML, pengembang dapat memastikan bahwa halaman *web* dan perangkat yang berbeda, mulai dari *computer desktop* hingga perangkat *mobile*. Ini menghasilkan pengalaman pengguna yang seragam dan dapat diandalkan, yang merupakan salah satu manfaat utama dari penggunaan HTML dalam pengembangan *web*. Tujuan utama dari HTML adalah menyediakan struktur dasar yang diperlukan untuk membangun sebuah halaman *web* yang fungsional dan mudah digunakan. Dengan menyediakan elemen-elemen dasar seperti teks, gambar, dan *hyperlink*, HTML memungkinkan pengembang untuk membuat konten yang dapat dinavigasi dan diakses dengan mudah oleh pengguna. Selain itu, HTML juga bertujuan untuk memungkinkan pembangunan halaman *web* yang ramah terhadap mesin pencari, dengan menyediakan *metadata* dan struktur yang memungkinkan mesin pencari untuk mengindeks dan memahami konten halaman *web* secara efisien. HTML berperan sebagai fondasi yang memungkinkan pembangunan halaman *web* yang terstruktur dan terorganisir dengan baik. Secara keseluruhan, fungsi dan manfaat HTML sangatlah luas, mulai dari menyediakan struktur dasar untuk halaman *web* hingga memungkinkan pembangunan konten *web* yang dapat diakses dan diindeks dengan mudah oleh pengguna dan mesin pencari untuk meningkatkan visibilitas dan aksesibilitas konten *web* secara keseluruhan.

3.4.3 CSS (*Cascading Style Sheet*)

CSS, atau *Cascading Style Sheet*, adalah bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu dokumen *web*. Menurut Sulistiyawan (2008), CSS digunakan untuk mengatur *style* dokumen. Sementara menurut Sya'ban (2010), CSS merupakan bahasa yang dikhususkan untuk mengatur gaya atau *layout* halaman *web*. CSS memungkinkan pengembang untuk memperindah tampilan *web* dengan mengatur warna, *font*, *margin*, *padding*, dan berbagai aspek tampilan lainnya. Menurut Saputra (2012), CSS didesain dengan tujuan khusus untuk mengontrol dan membangun komponen-komponen dalam halaman *web* sehingga tampilannya lebih teratur, terstruktur, dan konsisten. Sebagai pelengkap pada elemen HTML, CSS memungkinkan pengontrolan semua elemen tersebut melalui penggunaan *script* CSS, serta memperluas kemampuan HTML dalam memformat dokumen *web*. CSS juga memperluas kemampuan HTML dalam memformat dokumen *web*. Dengan penggunaan CSS, pengembang dapat membuat tampilan *web* yang lebih menarik dan lebih mudah dipahami oleh pengguna. Bunafit Nugroho (2014) menjelaskan bahwa penulisan kode CSS dilakukan dengan menyisipkannya pada *tag* HTML. Dalam penggunaannya, CSS memiliki aturan *cascading*, yang memungkinkan pengembang untuk menentukan prioritas pada tampilan elemen-elemen pada halaman *web*. Hal ini memungkinkan pengembang untuk membuat tampilan yang lebih terstruktur, seragam, dan mudah diakses oleh pengguna. (Admin Media, 2022)



Gambar 3.4.3 Cara Kerja CSS

CSS bekerja saat sebuah *browser* menampilkan dokumen, karena disaat itu *browser* harus menggabungkan konten dokumen dengan gaya informasi atau gaya presentasi dalam sebuah halaman. Untuk membangun sebuah halaman dan membuat presentasinya semenarik mungkin..

CSS memiliki beberapa langkah cara kerja sebagai berikut.

1. *Load HTML*

Ditahap ini *browser* akan memuat HTML dari jaringan.

2. *Parse HTML*

Pada tahap ini, *browser* akan memuat sebagian besar sumber daya yang akan ditautkan oleh dokumen HTML seperti gambar atau video sematan. Dalam hal ini CSS akan ditautkan.

3. *Parse CSS*

Sama halnya seperti HTML, CSS akan diurai dengan mengurutkan aturan yang berbeda berdasarkan jenis pemilihnya ke dalam *bucket* yang berbeda, seperti elemen, kelas, ID, dan sebagainya. Penyeleksian ini akan menentukan aturan mana yang harus diterapkan ke simpul-simpul DOM (*Document Object Mode*).

4. *Render Tree*

Tahap penyeleksian dalam setiap elemen-elemen CSS yang akan diterapkan ke simpul DOM sesuai kebutuhannya yang biasanya disebut sebagai *render tree*. *Render tree* ini diletakkan dalam struktur yang seharusnya muncul setelah aturan dan penyesuaian simpul dan elemen telah diterapkan.

5. *Display*

Setelah penyesuaian tahap-tahap sebelumnya, maka CSS dan DOM akan membentuk tampilan visual sebuah halaman situs yang tertampil dilayar.

Dari langkah-langkah diatas, dapat disimpulkan bahwa proses pengaplikasian CSS pada sebuah halaman *web* melibatkan beberapa tahapan yang terstruktur. Browser memuat HTML dari jaringan dan membangun model struktur dokumen. Selanjutnya, CSS akan diparsing dan diurutkan berdasarkan aturannya, kemudian dipasang dengan elemen-elemen DOM sesuai seleksi yang telah ditentukan.

Berikut ini adalah beberapa fungsi dari CSS yaitu.

1. Pengaturan Tampilan (*Styling*)

Fungsi utama CSS adalah mengatur tampilan elemen-elemen HTML dalam halaman *web*. Ini mencakup properti-properti seperti warna, ukuran, jenis font, dan berbagai properti lainnya yang memungkinkan pengembang untuk menyesuaikan penampilan halaman *web* secara keseluruhan. Contohnya adalah menggunakan CSS agar dapat menentukan teks yang berwarna biru tua, font sans-serif, dan ukuran font 16px. Dengan cara ini, CSS memberikan kendali yang kuat terhadap aspek visual dari halaman *web*.

2. Penataan *Layout* (Layouting)

CSS memungkinkan pengaturan tata letak (*layout*) dari elemen-elemen dalam halaman *web*. Ini termasuk menentukan posisi, dimensi, dan jarak antara elemen-elemen tersebut. Misalnya, pengembang dapat menggunakan CSS untuk menentukan bahwa suatu elemen harus diletakkan di tengah halaman, atau dua elemen berada dalam satu baris. Dengan kemampuan ini, CSS memungkinkan pembuatan tata letak halaman *web* yang sesuai dengan kebutuhan desain.

3. Responsif (*Responsiveness*)

Fungsi responsif CSS memungkinkan pengembang untuk membuat desain halaman *web* yang responsif terhadap berbagai ukuran layar dan perangkat. Dengan menggunakan media *queries*, pengembang dapat menyesuaikan gaya tampilan berdasarkan lebar layar perangkat, sehingga halaman *web* tetap terlihat baik di perangkat desktop, tablet, atau ponsel.

4. Animasi (*Animation*)

CSS memungkinkan pembuatan animasi pada elemen halaman *web* tanpa menggunakan *JavaScript*. Ini mencakup pengaturan perubahan properti.

5. Transisi (*Transition*)

Fungsi ini memungkinkan penambahan efek transisi yang halus saat perubahan properti terjadi pada elemen. Ini memberikan efek yang lebih menarik dan halus ketika elemen berubah, seperti perubahan warna atau perubahan ukuran.

3.5 Sistem Keamanan Informasi

Menurut G.J. Simons (2018), keamanan sistem informasi berkaitan dengan upaya yang dilakukan untuk mencegah atau minimal mendeteksi kecurangan dalam sebuah sistem berbasis informasi, di mana informasi tersebut tidak memiliki bentuk fisik. Menurut Whitman & Mattord (2011), keamanan sistem informasi adalah suatu konsep yang mengacu pada perlindungan informasi sebagai salah satu aset penting. Dalam konteks perusahaan, penting bagi mereka untuk menjaga keamanan aset informasi mereka. Kerugian finansial dan penurunan produktivitas dapat terjadi akibat kebocoran informasi atau kegagalan sistem. (Shinta Nurul, 2022)

Menurut standar ISO/IEC 27002 tahun 2013 tentang keamanan sistem informasi, keberadaan kontrol keamanan sangat penting dalam melindungi informasi dari berbagai ancaman. Kontrol keamanan tersebut berperan dalam memastikan kelangsungan bisnis, mengurangi risiko bisnis, serta meningkatkan investasi dan peluang bisnis. Menurut Chaeikar (2012), keamanan sistem informasi adalah suatu upaya perlindungan yang dilakukan untuk menjaga keamanan informasi dari berbagai ancaman. Tujuannya adalah untuk memastikan kelangsungan proses bisnis, mengurangi risiko bisnis, serta meningkatkan *return of investment* dan peluang bisnis. Penerapan CSS dalam desain *web* membawa dampak signifikan dalam mengatur tampilan dan interaksi pengguna secara efektif.

Dari proses awal memuat HTML hingga pembentukan tampilan visual, CSS memberikan kontrol yang kuat terhadap aspek visual dan fungsional dari halaman *web*. Melalui parsing HTML, CSS diintegrasikan dengan elemen-elemen halaman, memungkinkan penyesuaian yang terstruktur dan efisien terhadap gaya dan *layout*. Fungsi utamanya, seperti pengaturan tampilan dan penataan *layout*, memungkinkan pengembang untuk menciptakan halaman *web* yang menarik dan mudah dinavigasi. Responsif CSS memastikan bahwa konsistensi tampilan di berbagai perangkat, menjadikan pengalaman pengguna terjamin tanpa kehilangan kualitas desain. kemampuan animasi dan transisi memperkaya interaksi pengguna, menambah dimensi dinamis dan menarik animasi dalam pengalaman *browsing*.

Penggunaannya yang cermat dan terstruktur, dari tahap parsing hingga pembentukan *render tree*, memastikan bahwa tampilan halaman *web* bersifat terstruktur, seragam, dan mudah diakses. Melalui aturan cascadingnya, CSS memberikan prioritas yang jelas pada tampilan elemen-elemen, memastikan konsistensi dan kelancaran dalam presentasi visual. Keseluruhan, CSS berperan krusial dalam memungkinkan pengembang untuk menciptakan halaman *web* yang menarik, responsif, dan dapat meningkatkan pengalaman pengguna dengan berbagai fitur interaktifnya. Selain itu, CSS juga memungkinkan pengembang untuk menciptakan pengalaman pengguna yang lebih konsisten di seluruh platform dan perangkat, meningkatkan daya tarik dan efektivitas situs *web* dalam mencapai tujuan bisnisnya.

Dari keempat pengertian para ahli di atas, dapat ditarik kesimpulan bahwa keamanan sistem informasi didefinisikan sebagai upaya perlindungan yang dilakukan untuk mencegah kecurangan dan melindungi informasi sebagai aset penting dalam sebuah sistem berbasis informasi. Kontrol keamanan menjadi faktor penting dalam melindungi informasi dari ancaman, menjaga kelangsungan bisnis, mengurangi risiko bisnis, serta meningkatkan investasi dan peluang bisnis. Keamanan sistem informasi juga berperan dalam mencegah kerugian finansial dan penurunan produktivitas akibat kebocoran informasi atau kegagalan sistem. Dengan demikian, penting bagi perusahaan untuk menjaga keamanan aset informasi mereka guna menjaga keberlangsungan dan keberhasilan bisnis. Aspek penting yang harus ada dalam keamanan sistem informasi seperti:

1. Privasi atau kerahasiaan informasi: Memastikan bahwa data tidak dapat diakses oleh pihak yang tidak berwenang.
2. Otorisasi: Memberikan hak akses kepada pengguna dalam sistem informasi berdasarkan peran dan tanggung jawab mereka.
3. Validasi: Memastikan keabsahan data dengan melakukan pengecekan dan verifikasi. (mmsi.binus.ac.id, 2017)

3.6 Data Flow Diagram (DFD)

Menurut Saputra (2018), Diagram Alir Data (DFD) adalah representasi grafis yang menggambarkan aliran data dalam suatu entitas ke sistem atau entitas lainnya. DFD juga digunakan untuk menggambarkan transformasi data yang terjadi dari *input* atau masukan hingga *output* atau keluaran. Menurut Rosa A.S M. Shalahudin (2016), DFD atau Diagram Alir Data adalah diagram grafis yang menggambarkan aliran informasi dan transformasi data dari *input* ke *output*. Menurut Yuniar Supardi (2015:5), DFD merupakan gambaran sistem secara lokal yang tidak bergantung pada perangkat keras, perangkat lunak, struktur data, atau organisasi *file*. DFD digunakan untuk menggambarkan arus data dalam sistem secara terstruktur dan jelas. Menurut James A. Hall, DFD adalah diagram yang menggunakan simbol-simbol untuk merepresentasikan proses, sumber data, aliran data, dan entitas dalam sebuah sistem. (Coki Siadari, 2015)

Kesimpulan yang didapat dari pendapat ahli di atas adalah *Data Flow Diagram* (DFD) adalah representasi grafis yang menggambarkan aliran data dalam suatu sistem atau entitas. DFD digunakan untuk menggambarkan transformasi data dari *input* ke *output*, serta menggambarkan arus informasi dan transformasi informasi dalam sistem secara terstruktur dan jelas. DFD tidak bergantung pada perangkat keras, perangkat lunak, struktur data, atau organisasi *file*, sehingga dapat digunakan untuk menggambarkan sistem secara lokal. DFD merupakan alat yang populer dalam analisis dan perancangan sistem karena dapat membantu dalam pemahaman dan dokumentasi arus data dan proses dalam suatu sistem dengan menggunakan simbol-simbol yang khas. Dengan menggunakan simbol-simbol yang khas, DFD membantu dalam memvisualisasikan dan memahami secara lebih baik, bagaimana informasi mengalir dan diolah dalam suatu lingkungan kerja. DFD juga membantu mengidentifikasi potensi perbaikan atau penyempurnaan suatu system yang sedang dianalisis. Kesimpulannya, DFD merupakan alat yang efektif dalam analisis dan desain sistem informasi, serta membantu dalam pengambilan keputusan yang lebih baik terkait dengan perancangan dan pengembangan sistem.

3.7 *Unified Modelling Language (UML)*

Menurut Nugroho (2009), UML adalah bahasa pemodelan khusus yang digunakan dalam sistem atau perangkat lunak dengan paradigma berorientasi objek. Tujuan dari pemodelan adalah untuk menyederhanakan kompleksitas kendala-kendala yang ada, sehingga lebih mudah dipahami dan dipelajari. Menurut Joomla. (2007), UML adalah bahasa pemodelan standar yang terdiri dari berbagai diagram, yang dikembangkan untuk membantu pengembang sistem dan perangkat lunak dalam menyelesaikan berbagai tugas seperti spesifikasi, visualisasi, desain arsitektur, konstruksi, simulasi, pengujian, dan dokumentasi.

Menurut Mulyani (2015), UML adalah kumpulan praktik yang digunakan dalam penyederhanaan bahasa pemrograman perangkat lunak. UML memungkinkan kita untuk menyajikan data atau informasi yang beragam dalam satu kerangka dengan konsep berorientasi objek. Herlawati (2011) menyatakan bahwa UML memiliki 9 jenis diagram, meskipun beberapa sumber hanya menyebutkan 8. Hal ini karena beberapa diagram digabungkan menjadi satu, seperti penggabungan diagram urutan, diagram komunikasi, dan diagram pewaktuan menjadi satu diagram interaksi. (LamanIT, 2022)

Berdasarkan pendapat para ahli di atas, dapat disimpulkan bahwa UML (*Unified Modeling Language*) adalah bahasa pemodelan khusus yang digunakan dalam sistem atau perangkat lunak dengan paradigma berorientasi objek. Tujuannya adalah untuk menyederhanakan kompleksitas kendala dalam pengembangan sistem, sehingga lebih mudah dipahami dan dipelajari. UML memungkinkan penyajian data atau informasi yang beragam dalam satu kerangka dengan konsep berorientasi objek. Dalam pemodelan sistem menggunakan UML (*Unified Modeling Language*), terdapat beberapa jenis diagram yang digunakan untuk menggambarkan berbagai aspek dan interaksi dalam sistem. Dengan beragam jenis diagram yang tersedia, UML memberikan fleksibilitas dalam mewakili berbagai aspek dan detail dalam pengembangan sistem atau perangkat lunak.

Berikut adalah beberapa contoh diagram UML yang umum digunakan:

1. *Use Case Diagram*

Use Case Diagram adalah salah satu jenis diagram UML yang menggambarkan hubungan interaksi antara sistem dengan aktor. Diagram ini digunakan untuk mendeskripsikan tipe interaksi antara pengguna sistem dengan sistem itu sendiri. *Use Case Diagram* sangat berguna dalam memodelkan aksi-aksi yang dilakukan oleh aktor dan sistem, serta membantu dalam pemahaman tentang fungsi dan skenario penggunaan sistem.

2. *Activity Diagram*

Activity Diagram merupakan diagram yang digunakan untuk memodelkan serangkaian proses atau aktivitas yang terjadi dalam sistem. Diagram ini menggambarkan urutan langkah-langkah proses secara vertikal, mirip dengan alur jalannya sistem. *Activity Diagram* adalah contoh lain dari diagram UML yang berkaitan dengan *Use Case Diagram*, karena dapat menggambarkan proses-proses yang terkait dengan skenario penggunaan sistem.

3. *Class Diagram*

Class Diagram atau diagram kelas digunakan untuk menampilkan struktur kelas dan hubungan antara kelas-kelas dalam suatu sistem. Diagram ini menggambarkan kelas-kelas dan paket-paket yang digunakan dalam sistem untuk memenuhi kebutuhan perangkat lunak.

4. *Sequence Diagram*

Sequence Diagram adalah diagram yang menggambarkan interaksi antara objek-objek dalam sistem berdasarkan urutan waktu. Diagram ini dapat menggambarkan urutan atau tahapan yang harus dilakukan oleh objek-objek dalam sistem untuk menghasilkan suatu hasil. *Sequence Diagram* juga berhubungan dengan *Use Case Diagram* dalam hal menggambarkan skenario interaksi antara aktor dan sistem.

Dengan menggunakan berbagai jenis diagram UML seperti *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram*, para pengembang sistem dapat memodelkan dan menggambarkan berbagai aspek penting dari sistem yang sedang dikembangkan, sehingga memudahkan perancangan sistem secara efektif. (Dicoding Intern, 2021)

Dalam praktik pengembangan perangkat lunak, penggunaan berbagai jenis diagram UML memainkan peran krusial dan memahami, merencanakan, mengkomunikasikan desain secara efektif. Pengguna UML juga memfasilitasi pembaruan dan perbaikan sistem yang lebih efisien. Dengan visualisasi yang jelas tentang berbagai aspek sistem, pengembang dapat dengan cepat mengidentifikasi area-area yang memerlukan perubahan dan peningkatan. *Use Case Diagram*, memungkinkan pengembang untuk mengidentifikasi dan mengeksplorasi interaksi antara aktor dan sistem, membantu dalam menetapkan kebutuhan fungsional sistem dengan jelas. Sementara itu *Activity Diagram*, membantu dalam memodelkan alur kerja atau proses bisnis dalam sistem, memfasilitasi pemahaman yang lebih baik tentang bagaimana informasi dan tugas bergerak dalam sistem yang sedang dikembangkan. *Class Diagram* disisi lain, memungkinkan pengembang untuk menggambarkan struktur statis dari sistem, termasuk entitas-entitas utama, atribut-atributnya, dan hubungan antara kelas-kelas tersebut. Dengan demikian, memungkinkan pengembang untuk memvisualisasikan hubungan objek dan hierarki kelas dengan lebih baik. Sedangkan *Sequence Diagram* memberikan pandangan yang jelas tentang bagaimana objek-objek berinteraksi dalam urutan waktu tertentu, membantu dalam memahami aliran kontrol dan komunikasi antara objek-objek. Dengan memanfaatkan berbagai jenis diagram ini, pengembang dapat menggambarkan sistem secara menyeluruh dari berbagai sudut pandang, memungkinkan pemahaman yang baik, komunikasi yang lebih efektif, serta perancangan yang lebih terstruktur dan terukur. Hal ini mendorong terciptanya sistem yang lebih handal, mudah dimengerti, dan sesuai dengan kebutuhan pengguna.

3.8 *Entity Relationship Diagram (ERD)*

Menurut Brady dan Loonam (2010), *Entity Relationship Diagram (ERD)* digunakan sebagai teknik untuk memodelkan kebutuhan data dari suatu organisasi dalam tahap analisis persyaratan proyek pengembangan sistem. ERD membantu dalam desain *database* relasional yang menjadi dasar sistem informasi yang dikembangkan. ERD bersama dengan detail pendukungnya menjadi model data yang digunakan sebagai spesifikasi untuk *database*. Menurut Whitten dan Bentley (2007), ERD adalah model konseptual yang menggambarkan hubungan antar penyimpanan dalam *Data Flow Diagram (DFD)*. ERD digunakan untuk memodelkan struktur data dan hubungan antar data, tanpa memperhatikan proses yang harus dilakukan. ERD membantu menjawab pertanyaan mengenai data yang diperlukan dan bagaimana data tersebut saling berhubungan.

Menurut Dalimunthe (2020), *Entity Relationship Diagram (ERD)* adalah dokumen yang mengidentifikasi *entity data* dan menampilkan hubungan antara entitas-entitas tersebut. Menurut Yanto (2016), ERD adalah diagram yang menggambarkan desain konseptual dari model basis data relasional. ERD merelasikan objek-objek dari dunia nyata yang dikenal sebagai entitas dan menggambarkan hubungan antar entitas tersebut. (admin, 2020)

Dari keempat pengertian ahli di atas, dapat disimpulkan *Entity Relationship Diagram (ERD)* merupakan teknik yang digunakan dalam pemodelan kebutuhan data suatu organisasi. ERD membantu dalam tahap analisis persyaratan proyek pengembangan sistem dengan memodelkan hubungan antar entitas dan struktur data. Dokumen ERD ini mengidentifikasi *entity data* dan menggambarkan hubungan antar entitas secara konseptual. ERD juga digunakan sebagai spesifikasi untuk desain basis data relasional yang menjadi dasar sistem informasi yang dikembangkan. Dengan ERD, kita dapat memahami data yang diperlukan, bagaimana data tersebut berhubungan, dan merencanakan desain basis data yang efektif.

3.9 Normalisasi

Menurut Yanto (2016), normalisasi adalah tahapan pengelompokan komponen data menjadi tabel-tabel yang menggambarkan entitas suatu relasi. Menurut Kursini dan Koniyo (2007) menjelaskan bahwa normalisasi merupakan teknik dalam desain logika sebuah *database* yang melibatkan pengelompokan atribut dari suatu relasi untuk membentuk struktur relasi yang baik tanpa redundansi. Menurut Yuhefizard (2008) menyebutkan bahwa normalisasi adalah proses yang bertujuan mengurangi redundansi data dan meningkatkan stabilitas *database* dengan menciptakan struktur-struktur *entity*. Sedangkan menurut Sutabri (2011) menjelaskan bahwa normalisasi merupakan proses pengelompokkan elemen data menjadi tabel-tabel yang menunjukkan entitas dan relasinya, dan proses ini diuji pada beberapa kondisi. Sebelum mempelajari normalisasi, penting untuk memahami konsep-konsep seperti *field* atau atribut kunci dan kebergantungan kunci. (carayayang.blogspot.com, 2017)

Berdasarkan pendapat para ahli di atas, dapat disimpulkan bahwa normalisasi merupakan tahapan penting dalam desain logika sebuah *database*. Normalisasi melibatkan pengelompokan atribut atau komponen data menjadi tabel-tabel yang mewakili entitas dan relasi dalam sebuah relasi. Tujuan dari normalisasi adalah menciptakan struktur relasi yang baik, mengurangi redundansi data, dan meningkatkan stabilitas *database*.

3.10 Database

Pengertian *database* menurut Connolly dan Begg (2010) adalah kumpulan data yang secara logis terkait dan dirancang untuk memenuhi kebutuhan informasi suatu organisasi. Menurut Indrajani (2015), *database* adalah kumpulan data yang saling berhubungan secara logis. Menurut Enterprise (2017), *database* adalah aplikasi yang menyimpan sekumpulan data yang memiliki perintah khusus untuk membuat, mengakses, mengatur, mencari, dan menyalin data. (Laudia Tysara, 2023)

Berdasarkan pendapat ahli di atas, dapat disimpulkan bahwa *database* adalah kumpulan data yang terhubung secara logis dan dirancang untuk memenuhi kebutuhan informasi suatu organisasi. *Database* disimpan secara sistematis dalam komputer dan dapat diakses melalui program komputer untuk mendapatkan informasi yang dibutuhkan. Dalam konteks ini, *database* berfungsi sebagai solusi untuk menyimpan, mengelola, dan mengorganisir data dengan tujuan mendukung aktivitas dan kebutuhan organisasi.

3.15.1 Contoh – contoh *Database*

Berikut ini adalah contoh-contoh *database* yang sering digunakan oleh pengembang web.

1. MySQL

MySQL adalah sistem pengelolaan basis data relasional yang sering digunakan dalam pengembangan *website* individu maupun perusahaan. MySQL memungkinkan definisi hubungan antara data yang lebih baik, sehingga dapat meningkatkan kecepatan *website*. MySQL dapat digunakan secara gratis, menjadikannya pilihan yang populer bagi para pengembang yang ingin mempelajari lebih dalam tentang basis data.

2. MongoDB

MongoDB merupakan salah satu *database* yang menggunakan pendekatan NoSQL, berbeda dengan sebagian besar *database* lain yang menggunakan *Structured Query Language* (SQL). MongoDB tidak menggunakan tabel, namun menggunakan format dokumen yang mirip dengan JSON.

3. PostgreSQL

PostgreSQL adalah jenis *Relational Database Management System* (RDBMS) yang bersifat *open-source* dan banyak digunakan oleh para pengembang. Kelebihan PostgreSQL sebagai *database open-source* adalah kemampuannya untuk disesuaikan sesuai kebutuhan pengguna. PostgreSQL memiliki empat bahasa procedural (Elly Santi, 2023)

3.15.2 MySQL

Menurut Agustini (2017), MySQL adalah sebuah *database multiuser* yang menggunakan bahasa *structured query language* (SQL). MySQL termasuk dalam kategori DBMS (*Database Management System*) yang bersifat *open-source*. Menurut Budi Raharjo, MySQL adalah RDBMS yang dapat mengelola *database* dengan cepat, mampu menampung data dalam jumlah besar, dan dapat diakses oleh banyak pengguna.

Arief (2011) menjelaskan bahwa MySQL adalah jenis *database* server yang populer dan banyak digunakan dalam pembangunan aplikasi web yang menggunakan *database* sebagai sumber dan pengelolaan data. MySQL bersifat *open source* dan menggunakan SQL (*Structured Query Language*). Sedangkan Fatmawati (2016) menyebutkan bahwa MySQL digunakan untuk mengelola *database* dalam sebuah *website* dinamis, memungkinkan pengunjung memberikan komentar, saran, dan masukan. (pelayananpublik.id, 2022)

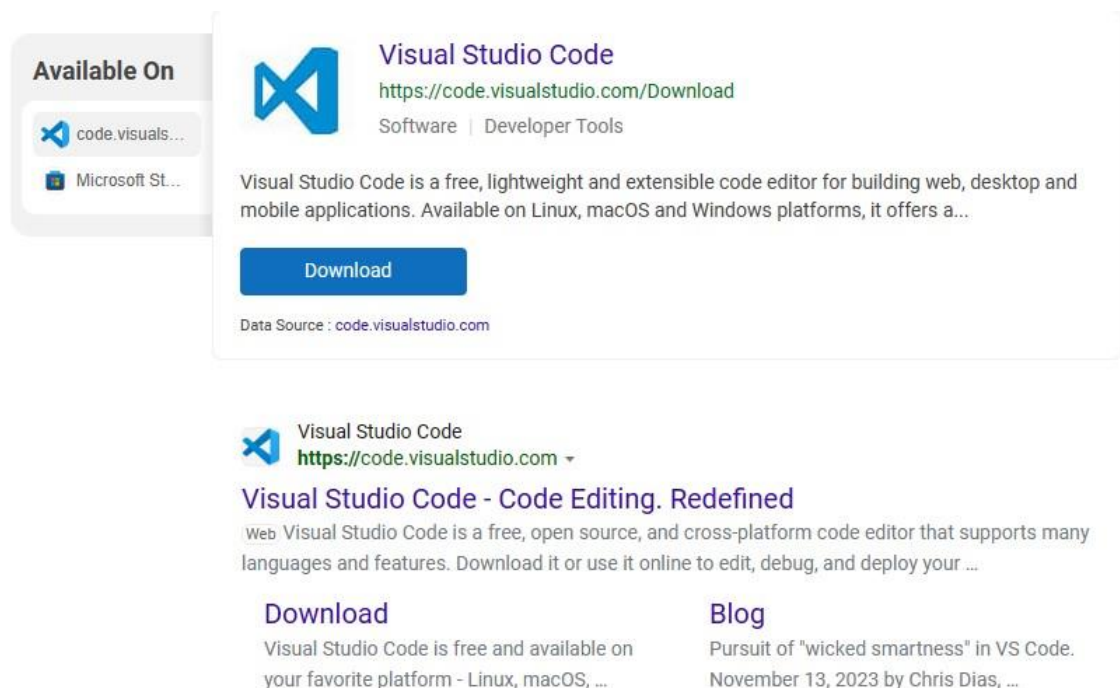
Berdasarkan penjelasan dari beberapa sumber, dapat disimpulkan bahwa MySQL adalah sebuah *database multiuser* yang menggunakan bahasa *structured query language* (SQL). MySQL merupakan DBMS (*Database Management System*) yang populer, cepat, dan mampu menampung data dalam jumlah besar. MySQL banyak digunakan dalam pembangunan aplikasi web yang membutuhkan pengelolaan dan penyimpanan data. Salah satu kelebihan MySQL adalah bersifat *open-source*, sehingga dapat diakses secara gratis. MYSQL juga dikenal karena kemampuannya untuk berintegrasi dengan berbagai Bahasa pemrograman dan platform, termasuk *PHP*, *Phyton*, *Java*, dan banyak lainnya. Hal ini membuat MYSQL menjadi pilihan utama bagi para pengembang *web* yang ingin membangun aplikasi yang dinamis dan responsif. Dengan kombinasi antara kehandalan kinerja, fleksibilitas, dan dukungan yang luas, MYSQL tetap menjadi pilihan yang sangat populer bagi para pengembang dalam membangun dan mengelola basis data untuk berbagai aplikasi *web* dan bisnis.

BAB IV PEMBAHASAN

4.1 Instalasi

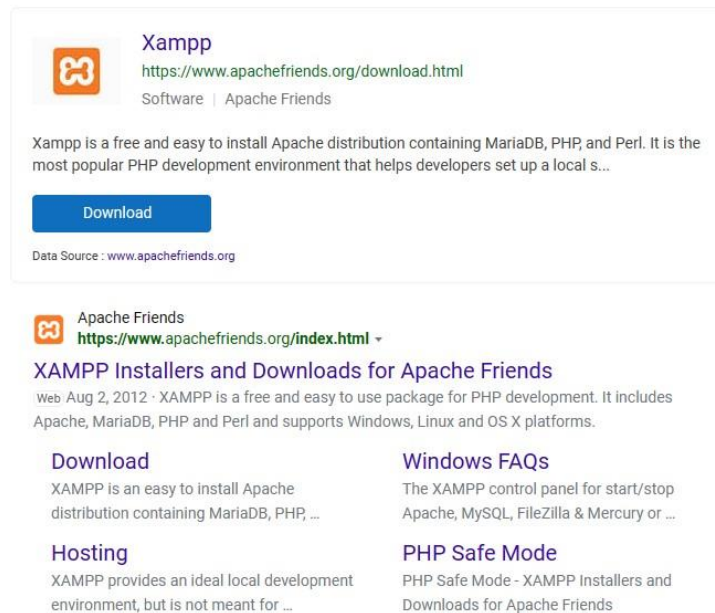
Pembuatan aplikasi sistem informasi beasiswa berbasis web ini diawali dengan melakukan beberapa instalasi terlebih dahulu. Instalasi yang dilakukan dapat dilakukan di berbagai platform seperti Windows, MacOS, ataupun Linux. Berikut ini adalah beberapa instalasi yang harus dilakukan.

1. Meng-install Visual Studio Code sebagai text editor. Visual Studio Code bisa di-install pada browser seperti Chrome, Firefox, dan sebagainya. Langkah-langkah instalasi Visual Studio Code dapat diikuti dengan arahan yang sudah diberikan pada situs website instalasinya.



Gambar 4.1 Install Visual Studi Code

2. Meng-install XAMPP sebagai database connector. Langkah-langkah untuk instalasi XAMPP bisa diikuti pada website instalasi XAMPP.



Xampp
<https://www.apachefriends.org/download.html>
 Software | Apache Friends

Xampp is a free and easy to install Apache distribution containing MariaDB, PHP, and Perl. It is the most popular PHP development environment that helps developers set up a local s...

[Download](#)

Data Source : www.apachefriends.org

Apache Friends
<https://www.apachefriends.org/index.html>

XAMPP Installers and Downloads for Apache Friends
 Web Aug 2, 2012 · XAMPP is a free and easy to use package for PHP development. It includes Apache, MariaDB, PHP and Perl and supports Windows, Linux and OS X platforms.

Download
 XAMPP is an easy to install Apache distribution containing MariaDB, PHP, ...

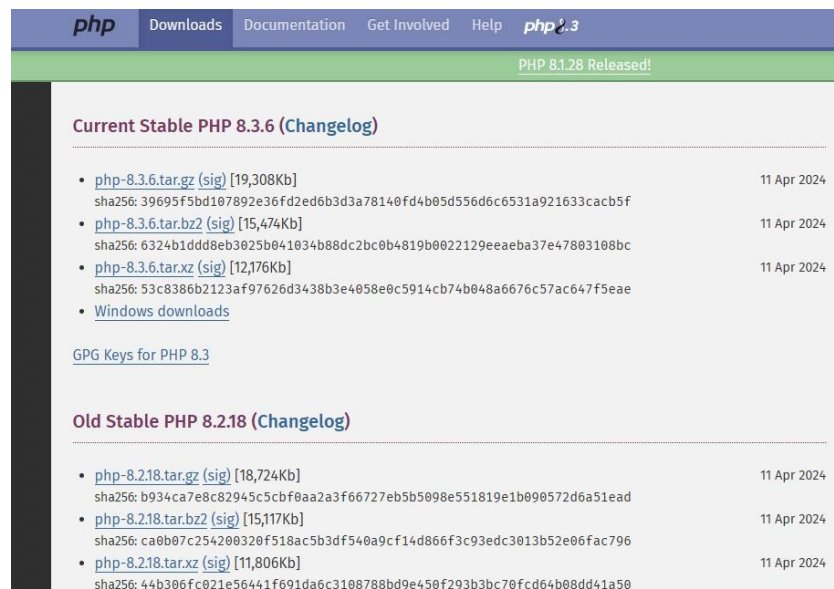
Windows FAQs
 The XAMPP control panel for start/stop Apache, MySQL, FileZilla & Mercury or ...

Hosting
 XAMPP provides an ideal local development environment, but is not meant for ...

PHP Safe Mode
 PHP Safe Mode - XAMPP Installers and Downloads for Apache Friends

Gambar 4.2 Install XAMPP

- Langkah selanjutnya adalah memastikan bahwa bahasa pemrograman PHP sudah terinstalasi dalam device. Jika belum, maka instalasi PHP juga dapat dilakukan pada browser yang dimiliki device. Langkah-langkah instalasi dapat diikuti berdasarkan panduan yang sudah disediakan.



php Downloads Documentation Get Involved Help **php 8.3**

PHP 8.1.28 Released!

Current Stable PHP 8.3.6 (Changelog)

- [php-8.3.6.tar.gz \(sig\)](#) [19,308Kb] 11 Apr 2024
 sha256: 39695f5bd107892e36fd2ed6b3d3a78140fd4b05d556d6c6531a921633cacb5f
- [php-8.3.6.tar.bz2 \(sig\)](#) [15,474Kb] 11 Apr 2024
 sha256: 6324b1ddd8eb3025b041034b88dc2bc0b4819b0022129eeaba37e47803108bc
- [php-8.3.6.tar.xz \(sig\)](#) [12,176Kb] 11 Apr 2024
 sha256: 53c8386b2123af97626d3438b3e4058e0c5914cb74b048a6676c57ac647f5eae
- [Windows downloads](#)

[GPG Keys for PHP 8.3](#)

Old Stable PHP 8.2.18 (Changelog)

- [php-8.2.18.tar.gz \(sig\)](#) [18,724Kb] 11 Apr 2024
 sha256: b934ca7e8c82945c5cbf0aa2a3f66727eb5b5098e551819e1b090572d6a51ead
- [php-8.2.18.tar.bz2 \(sig\)](#) [15,117Kb] 11 Apr 2024
 sha256: ca0b07c254200320f518ac5b3df540a9cf14d866f3c93edc3013b52e06fac796
- [php-8.2.18.tar.xz \(sig\)](#) [11,806Kb] 11 Apr 2024
 sha256: 44b306fc021e56441f691da6c3108788bd9e450f293b3bc70fcd64b08dd41a50

Gambar 4.2 Install PHP

4.2 Pembuatan Aplikasi

Pembuatan aplikasi system informasi beasiswa berbasis web ini dilakukan dengan menggunakan bahasa pemrograman PHP berbasis OOP (Object Oriented Programming). Pembuatan aplikasi menggunakan bahasa pemrograman PHP berbasis OOP melibatkan perencanaan yang matang, desain kelas yang tepat untuk mempresentasikan objek-objek dalam aplikasi, pemisahan kode dengan prinsip OOP seperti enkapsulasi dan pewarisan, serta implementasi fungsionalitas dengan memanfaatkan metode dan fungsi yang telah ditentukan. Selain itu, koneksi ke database perlu dibuat menggunakan PDO atau Mysqli, sementara validasi input diperlukan untuk meningkatkan keamanan. Pengujian menyeluruh diperlukan untuk memastikan kinerja yang baik dan penanganan bug yang efektif. Pengoptimalan, penyempurnaan, dan dokumentasi juga penting untuk meningkatkan kualitas dan pemeliharaan aplikasi.

Berikut ini adalah langkah-langkah pembuatan aplikasi sistem informasi beasiswa.

4.2.1 Terminal Setup

Langkah pertama yang harus dilakukan adalah melakukan terminal setup. Hal-hal ini mencakup komponen penting yang diperlukan ketika membuat aplikasi ini. Tanpa komponen ini, pembuatan aplikasi tidak akan dapat dilakukan.

Berikut ini adalah beberapa hal penting yang perlu dilakukan sebelum membuat aplikasi ini.

1. Membuka terminal pada Visual Studio Code
2. Masuk ke dalam path folder project
3. Ketik “npm init-y” dan enter. Tunggu hingga selesai install.
4. Kemudian, ketik “npm i express cors dotenv” dan enter. Command ini akan meng-install express, cors, dan dotenv pada folder project.
5. Jika sudah, ketik “npm i -D prisma” dan enter. Command ini akan meng-install prisma pada folder project.
6. Ketik “npm i @prisma/client” dan enter. Command ini akan meng-install prisma client agar dapat digunakan

7. Kemudian, ketik “npm prisma init” dan enter. Command ini akan memasukkan folder bernama prisma beserta file schema.prisma di dalamnya.

Jika seluruh langkah-langkah di atas sudah dilakukan, maka pembuatan aplikasi sudah dapat dimulai. Komponen-komponen yang di-install pada langkah-langkah di atas, akan membantu dalam proses pembuatan aplikasi. Tanpa komponen di atas, pembuatan aplikasi tidak dapat dimulai.

4.2.2 Terminal Setup

Sebelum membuat model pada file prisma, tahapan yang perlu dilakukan adalah memasukkan nama database yang akan digunakan pada file “koneksi” yang tersedia. Code yang akan dimasukkan ke dalam file koneksi adalah sebagai berikut.

```
CRUDMaster > config > koneksi.php
1  <?php
2  // Definisikan koneksi ke database
3  $host = "localhost"; // Ganti dengan host database Anda
4  $username = "root"; // Ganti dengan username database Anda
5  $password = ""; // Ganti dengan password database Anda
6  $database = "siswa"; // Ganti dengan nama database Anda
7
8  // Buat koneksi
9  $kon = mysqli_connect($host, $username, $password, $database);
10
11 // Periksa koneksi
12 if (!$kon) {
13     die("Koneksi database gagal: " . mysqli_connect_error());
14 }
15 ?>
```

Gambar 4.2.2 Pengaturan *file* koneksi

Pada gambar 4.2.2 terdapat tampilan codingan pada file koneksi.php yang digunakan untuk mendefinisikan koneksi ke database. Database pada \$host untuk memasukkan nama database yang dibuat pada localhost MySQL. Dalam gambar tersebut, nama database yang digunakan untuk aplikasi ini adalah siswa. Agar koneksi database tidak memiliki kesalahan dan bisa diperiksa kembali, dengan cara menggunakan `mysqli_connect_error` agar pada saat koneksi tidak terhubung maka akan muncul teks “Koneksi database gagal”. Langkah selanjutnya adalah menyalakan XAMPP Control Panel, aktifkan bagian Apache dan MySQL.

4.2.3 Database

Server: 127.0.0.1 > Database: siswa

Struktur SQL Cari Kueri Ekspor Impor Operasi Hak Akses Routine Event Trigger

Filters

Mengandung kata:

Tabel	Tindakan	Baris	Jenis	Penyortiran	Ukuran	Beban
<input type="checkbox"/> gambar1		0	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> guru		3	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> matapelajaran		8	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> nilai		3	InnoDB	utf8mb4_general_ci	32.0 KB	-
<input type="checkbox"/> siswaa		4	InnoDB	utf8mb4_general_ci	16.0 KB	-
5 tabel	Jumlah	18	InnoDB	utf8mb4_general_ci	96.0 KB	0 B

Gambar 4.2.3 Database

Pada gambar di atas, terdapat database dengan nama siswa yang berisi table gambar1, guru, matapelajaran, nilai, siswaa. Isi dari masing-masing table sesuai dengan yang dibuat pada file database.

4.2.4 Pembuatan Beasiswa

ketika pembuatan database telah selesai, tahapan selanjutnya adalah membuat index beasiswa, update beasiswa, delete beasiswa, view beasiswa.

1. Index beasiswa

```

114 <?php
115 while ($userAmbilData = mysqli_fetch_array($ambildata)) {
116     echo "<tr>";
117     echo "<td>" . $id_siswa = $userAmbilData['id_siswa'] . "</td>";
118     echo "<td>" . $nama_siswa = $userAmbilData['nama_siswa'] . "</td>";
119     echo "<td>" . $jurusan = $userAmbilData['jurusan'] . "</td>";
120     echo "<td>" . $jk_siswa = $userAmbilData['jk_siswa'] . "</td>";
121     echo "<td>";
122     $data = mysqli_query($kon, "SELECT * FROM siswaa WHERE id_siswa='{ $userAmbilData['id_siswa'] }'");
123     while ($row = mysqli_fetch_array($data)) {
124         echo "<a href='javascript:void(0);' onclick='\"window.open('public/siswaa/aset/{ $userAmbilData['gambar
125     }
126     echo "</td>";
127     // Menambahkan link untuk view, edit, dan delete
128     echo "<td><a href='public/siswaa/view.php?id_siswa=$userAmbilData[id_siswa]'>View</a> |
129         <a href='public/siswaa/update.php?id_siswa=$userAmbilData[id_siswa]'>Edit</a> |
130         <a href='public/siswaa/delete.php?id_siswa=$userAmbilData[id_siswa]'>Hapus</a> |
131         <a href='public/siswaa/nilai.php?id_siswa=$userAmbilData[id_siswa]'> Pengumuman Nilai</a></td>"
132     echo "</tr>";
133 }
134 ?>

```

Gambar 4.2.4 Index Beasiswa

Pada gambar di atas, codingan tersebut digunakan untuk menampilkan data siswa dari database dalam bentuk tabel. Setiap baris tabel menampilkan informasi seperti id siswa, nama siswa, jurusan, dan jenis kelamin siswa.

Selain itu, setiap Baris juga menyediakan link untuk melihat gambar siswa, melihat detail siswa, mengedit data siswa, menghapus data siswa, dan melihat pengumuman nilai siswa. Loop while digunakan untuk mengambil setiap baris data dari hasil kueri yang dilakukan sebelumnya. Potongan kode di dalam loop while mengekstrak nilai-nilai dari setiap baris data dan menampilkannya dalam sel-sel tabel. Selain itu, terdapat potongan kode yang menampilkan link “lihat file” untuk membuka gambar siswa dalam jendela baru. Setiap link untuk mengelola data siswa mengarah ke halaman terkait yang memproses tindakan yang sesuai. Misalnya, melihat, mengedit, atau menghapus data siswa.

2. Update Beasiswa

```

102 <body>
103   <h1>Edit Data Siswa</h1>
104   <form name="update_data" method="post" action="update.php">
105     <table border="0">
106       <tr>
107         <td>ID Siswa</td>
108         <td><input type="text" name="id_siswa" readonly value="<?php echo htmlspecialchars($id_siswa); ?>"></td>
109       </tr>
110       <tr>
111         <td>Nama Siswa</td>
112         <td><input type="text" name="nama_siswa" value="<?php echo htmlspecialchars($nama_siswa); ?>"></td>
113       </tr>
114       <tr>
115         <td>Jurusan</td>
116         <td><input type="text" name="jurusan" value="<?php echo htmlspecialchars($jurusan); ?>"></td>
117       </tr>
118       <tr>
119         <td>Jenis Kelamin</td>
120         <td><input type="text" name="jk_siswa" value="<?php echo htmlspecialchars($jk_siswa); ?>"></td>
121       </tr>
122       <tr>
123         <td><input type="hidden" name="id_siswa" value="<?php echo $id_siswa; ?>"></td>
124         <td>
125           <input type="submit" name="update" value="Update">
126           <a href="../../index.php" class="home-btn">Home</a>
127         </td>
128       </tr>
129     </table>
130   </form>
131

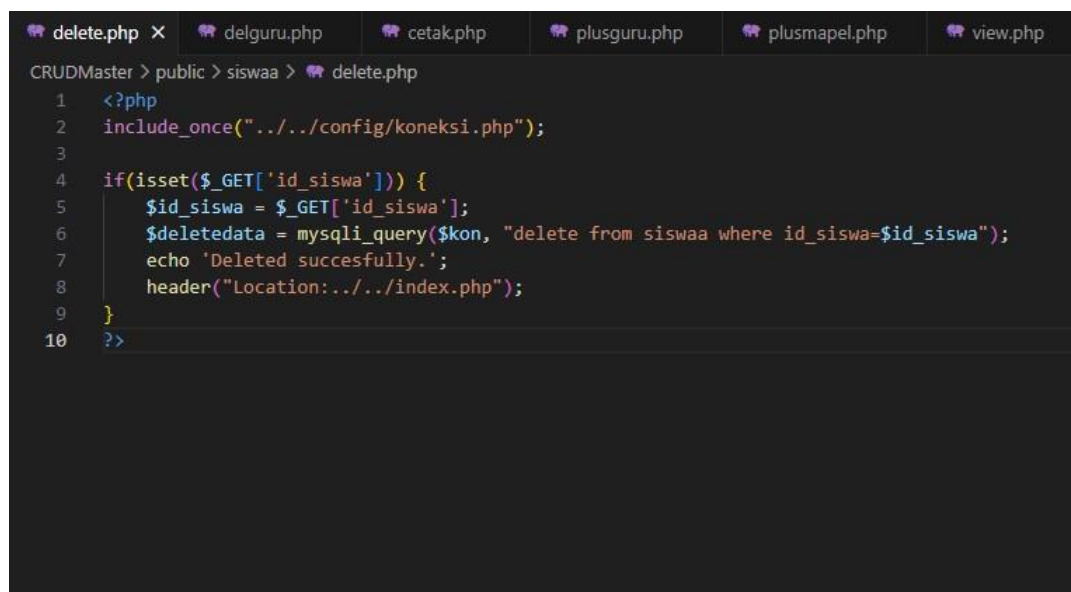
```

Gambar 4.2.4 Index Beasiswa

Pada gambar di atas, codingan tersebut merupakan formulir HTML untuk mengedit data siswa. Formulir ini memungkinkan pengguna untuk mengubah informasi seperti nama siswa, jurusan, dan jenis kelamin. Setiap input field diisi dengan nilai yang sesuai dari data siswa yang akan diubah, yang diperoleh dari variabel PHP seperti \$id_siswa, \$nama_siswa, \$jurusan, \$jk_siswa.

Namun, input field untuk id siswa diberi atribut readonly untuk mencegah pengguna mengubahnya karena umumnya id siswa adalah kunci utama yang tidak boleh diubah. Saat formulir dikirimkan dengan menekan tombol “update”, data akan diproses oleh file “update.php”. selain tombol “update”, terdapat juga link “home” yang mengarahkan kembali ke halaman utama. Dengan demikian, codingan ini memberikan fungsionalitas kepada pengguna untuk mengedit data siswa yang tersimpan dalam database.

3. Delete beasiswa



```

delete.php x delguru.php cetak.php plusguru.php plusmapel.php view.php
CRUDMaster > public > siswaa > delete.php
1  <?php
2  include_once("../config/koneksi.php");
3
4  if(isset($_GET['id_siswa'])) {
5      $id_siswa = $_GET['id_siswa'];
6      $deletedata = mysqli_query($kon, "delete from siswaa where id_siswa=$id_siswa");
7      echo 'Deleted succesfully.';
8      header("Location:../index.php");
9  }
10 ?>

```

Gambar 4.2.4 Delete Beasiswa

Pada gambar diatas, codingan tersebut adalah sebuah file PHP yang berfungsi untuk menghapus data siswa dari database. Pertama, file ini mengimpor file koneksi.php yang diperlukan untuk menghubungkan ke database. Kemudian, ia memeriksa apakah parameter id_siswa telah diterima melalui metode GET. Jika iya, maka dilakukan penghapusan data siswa dari tabel siswaa berdasarkan id siswa yang diterima. Setelah penghapusan berhasil dilakukan, pesan ‘Deleted succesfully.’ Ditampilkan. Dengan demikian, fungsi utama dari codingan ini adalah untuk mengelola penghapusan data siswa dari database sesuai dengan id yang diberikan melalui parameter GET.

4.2.5 Pembuatan bagian mata pelajaran

Pembuatan controller untuk model mapel akan disambungkan ke dalam route model mapel. Sebuah file route berisi seluruh pembuatan link atau path yang akan digunakan untuk menjalankan aksi pada controller.

1. Index mata pelajaran

```
CRUDMaster > public > siswa > indexmapel.php
66  y>
67  <div class="container">
73  <table>
79  <?php
80  if (isset($_GET['cari'])) {
81      $cari = $_GET['cari'];
82      $ambildata = mysqli_query($kon, "SELECT * FROM matapelajaran WHERE id_mapel LIKE '%$cari%' OR nama_m
83  } else {
84      $ambildata = mysqli_query($kon, "SELECT * FROM matapelajaran ORDER BY id_mapel ASC");
85  }
86  while ($userAmbilData = mysqli_fetch_array($ambildata)) {
87      echo "<tr>";
88      echo "<td>" . $userAmbilData['id_mapel'] . "</td>";
89      echo "<td>" . $userAmbilData['nama_mapel'] . "</td>";
90      echo "<td>";
91      echo "<a class='btn btn-delete' href='delmapel.php?id_mapel={$userAmbilData['id_mapel']}'>Hapus</a>";
92      echo "</td>";
93      echo "</tr>";
94  }
95  ?>
96  </table>
97  <a href="plusmapel.php" class="btn">Tambah Data Mata Pelajaran</a>
98  <a href="..." index.php" class="btn">Home</a>
99  <a href="public/siswa/tambah.php" class="btn">Cetak Data Mata Pelajaran</a>
100 </div>
```

Gambar 4.2.4 Index mata pelajaran