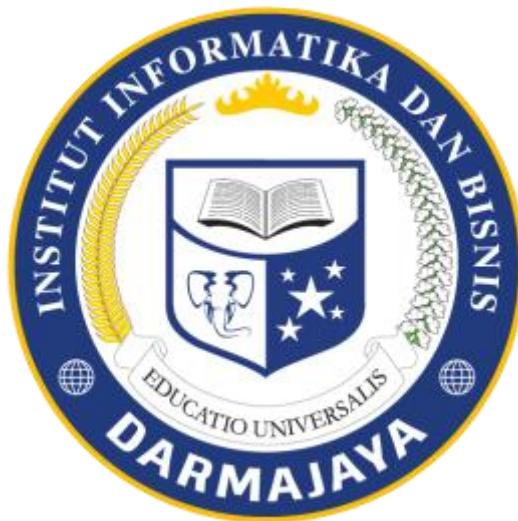


**APLIKASI PENENTUAN KREDIT BERJANGKA *ONLINE* MENGGUNAKAN
ALGORITMA *K-NEAREST NEIGHBOR* PADA PT. COLOMBUS TANJUNG
KARANG PUSAT BANDAR LAMPUNG**

SKRIPSI

Diajukan Sebagai Salah Satu Syarat Untuk Mencapai Gelar
SARJANA KOMPUTER
Pada Jurusan Teknik Informatika
Institut Informatika dan Bisnis Darmajaya



Oleh :

Nama : Mega Lestari

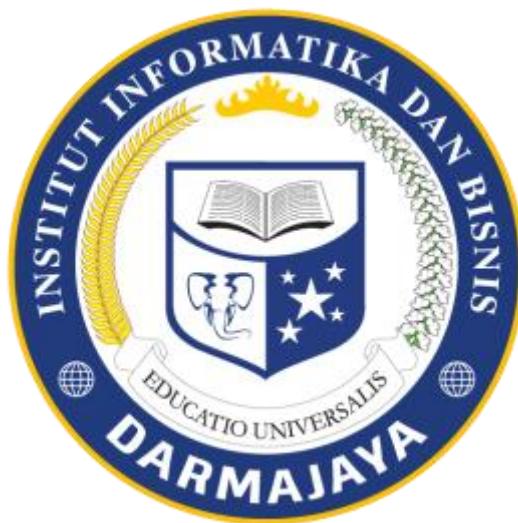
NPM : 1411010008

**JURUSAN TEKNIK INFORMATIKA FAKULTAS ILMU KOMPUTER
IIB DARMAJAYA BANDAR LAMPUNG
2018**

**APLIKASI PENENTUAN KREDIT BERJANGKA *ONLINE* MENGGUNAKAN
ALGORITMA *K-NEAREST NEIGHBOR* PADA PT. COLOMBUS TANJUNG
KARANG PUSAT BANDAR LAMPUNG**

SKRIPSI

Diajukan Sebagai Salah Satu Syarat Untuk Mencapai Gelar
SARJANA KOMPUTER
Pada Jurusan Teknik Informatika
Institut Informatika dan Bisnis Darmajaya



Oleh :

Nama : Mega Lestari

NPM : 1411010008

**JURUSAN TEKNIK INFORMATIKA FAKULTAS ILMU KOMPUTER
IIB DARMAJAYA BANDAR LAMPUNG
2018**



PERNYATAAN

Saya yang bertanda tangan dibawah ini, menyatakan bahwa skripsi yang saya buat ini adalah hasil karya saya sendiri, tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi atau karya yang pernah ditulis atau diterbitkan orang lain kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka. Karya ini adalah milik saya dan pertanggung jawaban sepenuhnya berada di pundak saya.

Bandar Lampung, 2018



Mega Lestari
NPM. 1411010008

HALAMAN PERSETUJUAN

Judul Skripsi : **APLIKASI PENENTUAN KREDIT BERJANGKA
ONLINE MENGGUNAKAN ALGORITMA
K-NEAREST NEIGHBOR PADA PT.
COLOMBUS TANJUNG KARANG PUSAT
BANDAR LAMPUNG**

Nama Mahasiswa : **Mega Testari**

No. Pokok Mahasiswa : **1411010008**

Jurusan : **Teknik Informatika**



Dosen Pembimbing

Rio Kurniawan, M.Cs
NIK. 13010313

Ketua Jurusan

Yuni Arkhiansyah, S.Kom., M.Kom
NIK. 00480802

HALAMAN PENGESAHAN

Telah diuji dan dipertahankan di depan Tim Penguji Ujian Skripsi
Jurusan Teknik Informatika Institut Informatika dan Bisnis Darmajaya
Bandar Lampung dan dinyatakan diterima untuk memenuhi syarat
guna memperoleh gelar Sarjana Komputer

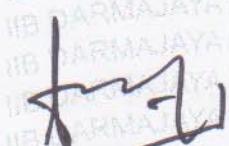
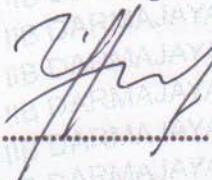
Mengesahkan,

1. Tim Penguji

Ketua : **Yuni Arkhiansyah, M.Kom**

Anggota : **Isnandar Agus, S.Pd., M.Kom**

Tanda Tangan



2. Dekan Fakultas Ilmu Komputer


Sriyanto, S.Kom., M.M

NIK. 00210800

Tanggal Lulus Ujian Skripsi :

RIWAYAT HIDUP

Penulis dilahirkan di kota Palembang pada tanggal 07 Mei 1996, sebagai anak ke dua dari empat bersaudara, dari anak Bapak Sunoto dan Ibu Yuli Siti Julaikah.

Nama : Mega Lestari

Email : megalestari070596@gmail.com

No. Hp : 0853-5706-0120

Penulis menyelesaikan pendidikan :

Sekolah dasar di SDN 1 Labuhan Ratu Satu Way Jepara Lampung Timur lulus pada tahun 2008.

Sekolah Menengah Pertama di SMP Muhammadiyah 1 Way Jepara Lampung Timur lulus pada tahun 2011.

Sekolah Menengah Atas di Sma Teladan Way Jepara Lampung Timur lulus pada tahun 2014.

Terdaftar sebagai mahasiswa jurusan teknik informatika IIB Darmajaya pada tahun 2014. Penulis melaksanakan Praktek Kerja Pengabdian Masyarakat (PKPM) di Pekon Patoman, Kecamatan Pagelaran, Kab. Pringsewu pada tahun 2018.

HALAMAN PERSEMBAHAN

Terimakasih untuk keluarga tercinta Kupersembahkan skripsi ini kepada :

- Ibunda Yuli Siti Julaika sembah sujud dan hormat adinda, terima kasih atas cinta dan kasih sayangnya, serta do'a dan semangat yang telah ibunda berikan kepada adinda.
- Erik Pratama Putra calon suamiku tercinta yang sabar untuk mendengarkan keluh kesahku, selalu memberikan dukungan dan semangat yang tiada henti, sehingga dapat menyelesaikan skripsi ini.
- Kakakku Tercinta Jujuk Sunarko, Adik-adikku Tersayang Triwahyono, Belinda Gunanda Putri. Serta Pakde Sumadi beserta istri Bude Sukarti selaku Pengganti Orang Tuaku Dirumah dan seluruh saudara yang telah memberi semangat dan do'a sehingga laporan skripsi ini dapat terselesaikan dengan baik.
- Keluarga kance pance Irawanto,S.Kom, Noji Tuseno,S.Kom, Oky Prabowo,S.Kom, Ahmad Roff'i, S.Kom, M. Dani Ramanda,S.Kom, Rangga Bagus Brata,S.Kom, Dewi Anggraini,S.Kom, Selvi Novitasari,S.Kom, Muji Rahayu,S.Kom yang selalu ada dan memberi semangat dari semester awal sampai akhir kuliah dan sama-sama berjuang untuk menyelesaikan skripsi ini.
- Teman-teman seperjuangan yang membantu selama proses skripsi ini Ajeng Pramekso Dewi, S.Kom, Feven Indriani, S.Kom, Hafsa Mukaromah, S.Kom.
- Keluarga PKPM 4 Patoman Jaka Saputra, Dimas Lintang Aryadita, Dedy Suwardi, Nyata Nopiyansyah, Nopri Yandi Pratama, Reni Setiawati, dan Selvi Paramita terimakasih sudah menjadi keluarga dan semangat buat skripsinya.

MOTTO

“Skripsi Tanpa Air Mata, Tidak Akan Memberimu Sebuah Pelajaran Serta Pengalaman”

-Mega Lestari

“Jangan Menyia-nyiakan Hidupmu untuk menunggu datangnya sayap. Yakinlah Bahwa Kau Mampu Terbang Tinggi Sendiri”

-Audrey Gene

“Kau Tak Dapat Meraih Sesuatu dalam Hidup Tanpa Pengorbanan Sekecil Apapun”

-Shakira

“Sukses Adalah Guru Yang Buruk. Sukses Menggoda Orang Yang Tekun Kedalam Pemikiran Bahwa Mereka Tidak Dapat Gagal”

-Bill Gates

“Agar Sukses Kemauan Mu Untuk Berhasil Harus Lebih Besar Dari Kemauan Mu Untuk Gagal”

-Bill Cosby

ABSTRAK

APLIKASI PENENTUAN KREDIT BERJANGKA ONLINE MENGGUNAKAN ALGORITMA K-NEAREST NEIGHBOR PADA PT.COLOMBUS TANJUNG KARRANG PUSAT BANDAR LAMPUNG

Oleh

**Mega Lestari
1411010008**

PT. Columbus merupakan salah satu perusahaan retail yang melayani penjualan elektronik dan furniture rumah tangga yang terdapat diberbagai kota di Indonesia. PT. Columbus menyediakan produk-produk elektronik dan furniture yang dapat dibeli secara *cash* dan juga kredit. Pembelian secara kredit belum memiliki sistem yang dapat dijadikan tolak ukur oleh konsumen ketika ingin melakukan pembelian elektronik maupun furniture dengan sistem kredit sehingga pengajuan kredit menjadi lama.

K-NN (K-Nearest Neighbor) merupakan sebuah metode yang digunakan untuk melihat kedekatan kasus baru dengan kasus lama dengan mengolah data yang sudah pernah ada. Pengambilan keputusan diambil berdasarkan kriteria-kriteria seperti penghasilan, jumlah tanggungan, jenis pekerjaan, pendidikan, lama bekerja serta berkas pengajuan. *K-NN (K-Nearest Neighbor)* memprediksi probabilitas di masa depan berdasarkan pengalaman di masa sebelumnya berdasarkan data-data.

Aplikasi kredit berbasis *android* dibuat memiliki kelebihan dari sistem ini yaitu pengambilan keputusan akan lebih cepat dan tepat karena melihat data-data pengajuan yang memiliki ciri yang sama dengan kasus baru tersebut. Kekurangan dari sistem ini adalah keputusan dari kasus baru sangat tergantung dengan sebanyak data lama yang diolah.

Kata kunci : PT. Columbus, Kredit, K-NN (*K-Nearest Neighbor*).

ABSTRACT

ONLINE TIME CREDIT APPLICATION USING K-NEAREST NEIGHBOR ALGORITHM IN PT. COLOMBUS TANJUNG KARANG PUSAT BANDAR LAMPUNG

By

**Mega Lestari
1411010008**

PT. Columbus is one of the retail companies that serve electronic sales and household furniture in various cities in Indonesia. PT. Columbus provides electronic and furniture products that can be purchased on cash and credit. Purchases on credit do not yet have a system that can be used as a benchmark by consumers when they want to make electronic and furniture purchases with a credit system so that credit applications are long.

K-NN (K-Nearest Neighbor) is a method used to see the closeness of a new case with an old case by processing data that has already been there. Decision making is taken based on criteria such as income, number of dependents, type of work, education, length of work and file submission. K-NN (K-Nearest Neighbor) predicts probabilities in the future based on previous experience based on data.

Android-based credit applications are made to have advantages of this system, namely decision making will be faster and more appropriate because it looks at submission data that has the same characteristics as the new case. The disadvantage of this system is that the decision of the new case is very dependent on as much old data as processed.

Keywords: PT. Columbus, Credit, K-NN (*K-Nearest Neighbor*).



PRAKATA

Segala puji syukur saya panjatkan kepada Allah SWT berkat-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi ini. Penulis menyadari bahwa dalam penyusunan skripsi ini tidak lepas dari bantuan semua pihak. Oleh karena itu dalam kesempatan ini penulis menyampaikan terimakasih yang sebesar-besarnya kepada:

1. Bapak DR. Andi Desfiandi, S.E., M.A. Selaku Ketua Yayasan Alfian Husin.
2. Bapak Ir. Firmansyah Y. Alfian, MBA., MSc. Selaku Rektor IBI Darmajaya.
3. Bapak Dr. RZ. Abdul Aziz, ST., M.T. Selaku Wakil Rektor I Bidang Akademik dan Riset Insitut Informatika and Bisnis (IIB) Darmajaya.
4. Bapak Sriyanto, S.Kom, M.M selaku Dekan Fakultas Ilmu Komputer Institut Informatika and Bisnis (IIB) Darmajaya.
5. Bapak Yuni Arkhiansyah, M.Kom. Selaku Ketua Jurusan Teknik Informatika.
6. Bapak Rio Kurniawan,M.Cs Selaku Dosen Pembimbing yang telah membantu membimbing dan mengarahkan serta memberikan petunjuk sehingga skripsi ini dapat diselesaikan dengan baik.
7. Almamaterku tercinta.

Demikian banyaknya bantuan dari berbagai pihak kepada penulis, tentunya tidak menutup kemungkinan bahwa hasil dari laporan ini masih ada kekurangan dan masih jauh dari taraf sempurna. Oleh karena itu, kritik dan saran guna perbaikan di masa depan adalah mutlak sangat penulis perlukan. Semoga Laporan Skripsi ini bermanfaat bagi setiap pembacanya.

Bandar Lampung, 2018

**Mega Lestari
NPM. 1411010008**

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN.....	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN.....	iv
HALAMAN PERSEMBAHAN	v
MOTTO	vi
PRAKATA	vii
RIWAYAT HIDUP.....	viii
ABSTRAK	ix
ABSTRAC.....	x
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xii
DAFTAR ISI.....	xiii

BAB I PENDAHULUAN

1.1 Latar Belakang Masalah.....	1
1.2 Perumusan Masalah.....	2
1.3 Ruang Lingkup Penelitian.....	2
1.4 Tujuan.....	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan	3

BAB II LANDASAN TEORI

2.1 Aplikasi	5
2.2 Pengertian Aplikasi <i>Mobile</i>	5
2.3 <i>Data Mining</i>	6
2.4 Kredit	6
2.5 (<i>K-NN</i>) <i>K-Nearset Neighbor</i>	7
2.5.1 Karakteristik Klasifikasi dengan <i>K-Nearset Neighbor</i>	8
2.6 <i>Android</i>	9
2.7 <i>Web Server</i>	10
2.8 <i>Data Base</i>	11
2.9 <i>Xampp</i>	12
2.10 <i>MySql</i>	12
2.11 <i>HTML</i>	12

2.12 <i>PHP</i>	12
2.13 <i>Web Browser</i>	13
2.14 <i>Web Hosting</i>	14
2.15 <i>Android Studio</i>	14
2.16 <i>ADT (Android Development Tools)</i>	15
2.17 Metode Pengembangan Sistem	15
2.18 Perancangan Sistem <i>UML</i>	16

BAB III METODE PENELITIAN

3.1 Metode Pengumpulan Data	24
3.2.1 <i>Communication</i>	25
3.2.2 <i>Quik Plan</i>	26
3.2.2.1 Analisis dan Definisi Persyaratan.....	26
3.2.2.2 Skema Representasi.....	27
A. Kebutuhan Data.....	27
B. Penentuan Bobot	28
C. Penentuan Sub Atribut.....	28
D. Penentuan Bobot Sub Atribut.....	29
E. Penentuan Nilai Jarak Sub Atribut	30
F. Testing Kasus Baru	30
3.2.3 <i>Modeling Quik Design</i>	36
3.2.4 <i>Construction of Prototype</i>	47
3.2.5 <i>Deployment, Delivery dan Feedback</i>	47

BAB IV PEMBAHASAN

4.1 HasilRancangan <i>Program</i>	55
4.1.1 Tampilan Halaman Awal.....	55
4.1.2 Tampilan Halaman Produk	56
4.1.3 Tampilan Halaman <i>Blog</i>	56
4.1.4 Tampilan Halaman <i>Detail Produk</i>	57
4.1.5 Tampilan Halaman <i>Detail Blog</i>	57
4.1.6 Tampilan Halaman <i>Test Pengajuan Kredit</i>	58
4.1.7 Tampilan Halaman Pengisian <i>Test Pengajuan Kredit</i>	58

4.1.8 Tampilan Halaman Hasil Keputusan <i>Test Kredit</i>	58
4.1.9 Tampilan Halaman <i>Dashboard Admin</i>	59
4.1.10 Tampilan Halaman <i>Manage Produk</i>	60
4.1.11 Tampilan Halaman <i>Manage Kategori</i>	60
4.2 Pengujian	60

DAFTAR TABEL

Tabel 3.1 Data <i>Training</i>	28
Tabel 3.2 Bobot <i>Atribut</i>	28
Tabel 3.3 Sub <i>Atribut</i>	29
Tabel 3.4 Bobot Sub <i>Atribut</i>	30
Tabel 3.5 Jarak Sub <i>Atribut Penghasilan</i>	31
Tabel 3.6 Jarak Sub <i>Atribut Jumlah Tanggungan</i>	32
Tabel 3.7 Jarak Sub <i>Atribut Jenis Pekerjaan</i>	32
Tabel 3.8 Jarak Sub <i>Atribut Pendidikan</i>	33
Tabel 3.9 Jarak Sub <i>Atribut Lama Kerja</i>	33
Tabel 3.10 Jarak Sub <i>Atribut Berkas</i>	34
Tabel 3.11 Kasus Baru	35
Tabel 3.12 Nilai Jarak Berdasarkan Data <i>Training</i>	36
Tabel 3.13 Hasil Nilai antara Kasus Baru dengan Kasus Lama	37
Tabel 3.14 Tabel <i>Admin</i>	44
Tabel 3.15 Tabel <i>Atribut</i>	44
Tabel 3.16 Tabel <i>Blog</i>	45
Tabel 3.17 Tabel <i>Produk</i>	45
Tabel 3.18 Tabel <i>Hasil</i>	46
Tabel 3.19 Tabel <i>Kasus</i>	46
Tabel 3.20 Tabel <i>Kategori</i>	47
Tabel 3.21 Tabel <i>Kedekatan</i>	47
Tabel 3.21 Tabel <i>Produk</i>	48
Tabel 3.21 Tabel Sub <i>Atribut</i>	48
Tabel 3.21 Tabel Data <i>Training</i>	49

DAFTAR GAMBAR

Gambar 2.1 Rumus <i>K-Nearest Neighbors</i>	7
Gambar 2.2 Model Prototype.....	17
Gambar 2.3 Diagram <i>Unified Modelling Language</i>	18
Gambar 2.4 Simbol-simbol <i>Class Diagram</i>	20
Gambar 2.5 Simbol-simbol <i>Use Case Class Diagram</i>	22
Gambar 2.6 Simbol-simbol <i>Use Case Class Diagram</i>	23
Gambar 3.7 <i>Use Case Sistem</i>	38
Gambar 3.8 <i>Activity Diagram Login Admin</i>	39
Gambar 3.9 <i>Activity Diagram User Melihat Produk</i>	40
Gambar 3.10 <i>Activity Diagram User Melakukan Uji Keputusan Kredit</i>	40
Gambar 3.11 <i>Sequence Diagram Home</i>	41
Gambar 3.12 <i>Sequence Diagram Produk</i>	42
Gambar 3.13 <i>Sequence Diagram Blog</i>	42
Gambar 3.14 <i>Sequence Diagram Uji Keputusan Kredit</i>	43
Gambar 3.14 <i>Sequence Diagram Login</i>	43
Gambar 3.15 <i>Class Diagram</i>	49
Gambar 3.16 Rancangan <i>Interface Program Tampilan Utama</i>	50
Gambar 3.17 Rancangan <i>Interface Halaman Produk</i>	51
Gambar 3.18 Rancangan <i>Interface Program Tampilan Blog</i>	51
Gambar 3.19 Rancangan <i>Interface Program Tampilan Uji Keputusan Kredit</i>	52
Gambar 3.20 Rancangan <i>Interface Program Tampilan Contact</i>	52
Gambar 4.21 Halaman Awal Aplikasi	55
Gambar 4.22 Halaman <i>Beranda</i>	56
Gambar 4.23 Halaman Data <i>Blog</i>	56
Gambar 4.24 Halaman <i>Detail Produk</i>	57
Gambar 4.25 Halaman <i>Detail Blog</i>	57
Gambar 4.26 Halaman <i>Test Pengajuan Kredit</i>	58
Gambar 4.27 Halaman Pengisian <i>Test Pengajuan Kredit</i>	58
Gambar 4.28 Halaman Hasil Keputusan <i>Test Pengajuan Kredit</i>	59
Gambar 4.29 Halaman <i>Beranda Admin</i>	59

Gambar 3.30 Halaman <i>Manage</i> Produk.....	60
Gambar 3.31 Halaman <i>Manage</i> Kategori.....	60

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi saat ini dimanfaatkan dengan baik oleh perusahaan dalam rangka meningkatkan target penjualan untuk memperoleh lebih banyak keuntungan. Teknologi informasi yang biasanya dimanfaatkan oleh perusahaan tersebut biasanya berupa *website* maupun aplikasi *android* yang banyak diakses atau digunakan oleh konsumen. Melalui *website* maupun aplikasi android perusahaan-perusahaan mempromosikan produk-produknya serta menyampaikan berbagai informasi terkait perusahaan tersebut.

PT. Columbus merupakan salah satu perusahaan retail yang melayani penjualan elektronik dan furniture rumah tangga yang terdapat diberbagai kota di Indonesia. Salah satu nya terdapat di Bandar Lampung tepatnya di Tj. Karang Pusat, Bandar Lampung. PT. Columbus menyediakan produk-produk elektronik dan furniture yang dapat dibeli secara *cash* dan juga kredit. Untuk pembelian secara kredit terdapat syarat-syarat tertentu yang harus dipenuhi untuk dapat diterima pengajuan kreditnya.

Saat ini pada PT. Columbus Tj. Karang Pusat, Bandar Lampung belum memiliki sebuah aplikasi yang dapat dijadikan sebagai media untuk mempromosikan atau memberikan informasi terkait produk-produk yang dijual. Sistem yang sebelumnya dianggap kurang cepat dan mudah untuk menyampaikan informasi tersebut kepada konsumen. Para konsumen saat ini lebih sering menggunakan *smartphone* nya untuk melihat informasi ataupun memenuhi kebutuhan sehari-hari mereka. Untuk mencapai penjualan serta mempromosikan produk-produk nya PT.Columbus harus dapat menyaediakan sebuah aplikasi berbasis android sehingga konsumen dapat dengan mudah memperoleh informasi dengan membuka aplikasi tersebut. Selain itu belum adanya sistem yang dapat dijadikan

tolak ukur oleh konsumen ketika ingin melakukan pembelian elektronik maupun furniture dengan sistem kredit, sehingga konsumen bisa mengetahui apakah dengan persyaratan yang ada mereka dapat melakukan pembelian secara kredit atau tidak.

Dengan pengolahan data histori konsumen yang pernah melakukan kredit dapat dengan benar data tersebut dijadikan dasar tolak ukur untuk menantukan pengajuan kredit. Dengan pemanfaatan teknologi pengolahan data yang mengoptimasi proses pencarian informasi prediksi dalam data yang besar. *K-Nearest Neighbor* memprediksi probabilitas di masa depan berdasarkan pengalaman di masa sebelumnya berdasarkan data-data.

Dari uraian di atas, penulis ingin membangun sebuah aplikasi yang dapat dijadikan sebagai media penyampain informasi dan juga mengetahui keputusan kredit berbasis *mobile android* yang mana aplikasi ini nantinya akan memberikan manfaat dalam hal penjualan dan promosi dengan media yang sering digunakan oleh konsumen serta memberikan keputusan terhadap pengajuan kredit yang akan dilakukan oleh konsumen. Penulis mengangkat masalah ini dalam penyusunan skripsi yang berjudul **“Aplikasi Penentuan Kredit Berjangka Online Menggunakan Algoritma K-Nearst Neighbor Pada PT. Columbus Tanjung Karang Pusat Bandar Lampung”**.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, diperoleh perumusan masalah sebagai berikut : “Bagaimana Merancang dan Membangun Aplikasi Penentuan Kredit Berjangka *Online* menggunakan Algoritma K-NN Berbasis *Mobile Android* yang dapat bermanfaat bagi PT. Columbus serta konsumen dalam hal melakukan pembelian kredit.

1.3 Ruang Lingkup Penelitian

Ruang lingkup permasalahan atau yang menjadi batasan dalam penelitian ini adalah:

1. Penelitian dilakukan di cabang PT. Columbus Tj. Karang Pusat Bandar Lampung.
2. Aplikasi ini dibangun pada perangkat *mobile* yang bersistem operasi *Android* minimal *platform* versi 4.2.1 (*jelly bean*).
3. Aplikasi ini ditujukan untuk konsumen PT. Columbus Tj. Karang Pusat Bandar Lampung.

1.4 Tujuan

Tujuan dari penelitian yang ingin dicapai adalah sebagai berikut:

1. Membangun sebuah Aplikasi penentuan kredit elektronik dan forniture berjangka pada PT. Columbus TJ. Karang Pusat Bandar Lampung dengan menggunakan Algoritma K-NN bedasarkan data kredit yang pernah ada sebelumnya.

1.5 Manfaat Penelitian

Manfaat penelitian yang dilakukan adalah sebagai berikut:

1. Memudahkan PT. Columbus Tj. Karang Pusat Bandar Lampung dalam memberikan informasi terkait produk-produknya serta agenda-agenda yang akan dilakukan.
2. Memudahkan konsumen dalam melihat apakah kredit akan diterima serta berjalan tanpa ada kendala hingga kredit selesai.

1.6 Sistematika Penulisan

Pembahasan dalam penulisan ini tersusun sebagai berikut:

BAB I PENDAHULUAN

Menguraikan Latar Belakang Masalah, Perumusan Masalah, Tujuan Penelitian, Manfaat Penelitian, Batasan Masalah, Waktu dan Tempat Penelitian, dan Sistematika Penulisan.

BAB II LANDASAN TEORI

Menguraikan tentang hal-hal yang berhubungan dengan teori sebagai landasan pembahasan.

BAB III METODELOGI PENELITIAN

Membahas mengenai analisis permasalahan, pengumpulan fakta-fakta, proses penerimaan pengetahuan, proses kerja mesin inferensi, perancangan database dan rancangan program.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

Membahas pembuatan program aplikasi penentuan pengajuan kredit dan pengujian program.

BAB V KESIMPULAN DAN SARAN

Berisi kesimpulan dan saran yang diambil berdasarkan hasil penelitian yang dilakukan.

DAFTAR PUSTAKA**LAMPIRAN**

BAB II

LANDASAN TEORI

2.1 Aplikasi

Menurut Safaat H (2012,p.9) aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media. Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (*application suite*). Contohnya adalah *Microsoft Office* dan *Open Office.org*, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan setiap aplikasi. Aplikasi ini memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna.

2.2 Pengertian Aplikasi *Mobile*

Menurut Buyens (2011) aplikasi *mobile* berasal dari kata *application* dan *mobile*. *Application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju sedangkan *mobile* dapat diartikan sebagai perpindahan dari suatu tempat ke tempat yang lain. Kata *mobile* mempunyai arti bergerak atau berpindah, sehingga aplikasi *mobile* menurut Rangsang Purnama (2010) adalah sebutan untuk aplikasi yang berjalan di *mobile device*. Dengan menggunakan aplikasi *mobile*, dapat dengan mudah melakukan berbagai macam aktifitas mulai dari hiburan, berjualan, belajar mengerjakan pekerjaan kantor, *browsing* dan lain sebagainya.

Pemanfaatan aplikasi *mobile* untuk hiburan paling banyak digemari oleh pengguna telepon seluler, karena dengan memanfaatkan adanya fitur *game*, *music player*, sampai *video player* membuat kita menjadi semakin mudah menikmati hiburan kapan saja dan dimanapun. Perangkat *mobile* memiliki banyak jenis dalam hal ukuran, *desain layout*, tetapi memreka memiliki kesamaan karakteristik yang sanagt berbeda dari *dekstop* sistem. Perangkat *mobile* memiliki *memory* yang kecil.

2.3 Data Mining

Data Mining (Witten, 2011) didefinisikan sebagai proses penemuan pola dalam data. Berdasarkan tugasnya. Menurut Prasetyo (2014), *data mining* bertujuan untuk memanfaatkan data dalam basis data dengan mengolahnya sehingga menghasilkan informasi baru yang berguna. Dengan kondisi adanya sisi data yang belum dieksplorasi, dan di sisi lain, kemajuan teknik komputasi dan ilmu komputer sudah tumbuh pesat, keterbutuhan akan ekplorasi informasi baru semakin meningkat. *Data mining* menjadi solusi penyelesaian pencarian informasi yang sebelumnya tidak dapat dideteksi secara tradisional dan hanya menggunakan kemampuan analisis manusia.

2.4 Kredit

Kredit adalah penyerahan barang, jasa, atau uang dari satu pihak (kreditor/pemberi pinjaman) atas dasar kepercayaan kepada pihak lain (nasabah atau pengutang/*borrower*) dengan janji membayar dari penerima kredit kepada pemberi kredit pada tanggal yang telah disepakati kedua belah pihak. Memperoleh kredit berarti memperoleh kepercayaan. Atas dasar kepercayaan kepada seseorang yang memerlukannya maka di berikan uang, barang atau jasa dengan syarat membayar kembali atau memberikan penggantinya dalam satu jangka waktu yang telah diperjanjikan (Linof ,2011).

2.5 K-Nearest Neighbour (K-NN)

Algoritma K-Nearest Neighbour (K-NN) adalah algoritma pengklasifikasian data sederhana dimana penghitungan jarak terpendek dijadikan ukuran untuk mengklasifikasikan suatu kasus baru berdasarkan ukuran kemiripan. Algoritma ini telah digunakan dalam estimasi statistik dan pengenalan pola sejak awal tahun 1970-an. Algoritma K-NN tergolong dalam algoritma *supervised* yaitu proses pembentukan algoritma diperoleh melalui proses pembelajaran (learning) pada *record-record* lama yang sudah terkласifikasi dan hasil pembelajaran tersebut dipakai untuk mengklasifikasikan *record* baru dengan *output* yang belum diketahui.

Dalam algoritma K-NN sebuah data baru diklasifikasikan berdasarkan jarak data baru tersebut dengan tingkat kemiripan data baru terdekat terhadap data pola. Jumlah data tetangga terdekat ditentukan dan dinyatakan dengan K. Misalkan ditentukan K=1, maka kasus ini hanya diklasifikasikan untuk satu data dari tetangga terdekat.

Jika nilai K didefinisikan berbeda oleh *user*, misal K=5 maka kasus dengan 5 jarak terpendek dipilih, kemudian diklasifikasi berdasarkan instance kelas target dimana kasus dengan jumlah mayoritas instance kelas target ditentukan sebagai klasifikasi untuk kasus baru. Secara umum untuk mendefinisikan jarak antara dua objek x dan y, digunakan rumus jarak *Euclidean*, seperti terlihat pada persamaan 1

$$D_{xy} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Gambar 2.1 Rumus K-Nearest Neighbour

Jarak antara objek x dan y didefinisikan sebagai D_{xy}, dimana x_i merupakan *record* yang akan diprediksi dan y_i merupakan *record* data pola sedangkan nilai n didefinisikan sebagai jumlah *atribut*. Nilai i merujuk kepada *record* ke-i. Persamaan (1) merupakan rumus jarak untuk tipe data kontinu sehingga data nominal harus dikonversi ke bentuk numerik. Proses penghitungan jarak dapat digambarkan dalam

tahapan algortima K-NN.

Tahapan algoritma K-NN berdasarkan persamaan (1)

1. Baca data pola terstandarisasi
2. Baca data yang akan diprediksi terstandarisasi
3. Masukkan nilai k
4. Tentukan $J = \text{Jumlah Atribut}$
5. Tentukan $N = \text{jumlah record data pola}$
6. Tentukan $L = 1$
7. *While not EOF*
8. $I = 1$
9. $Y = 0$
10. *While $I < J + 1$, Do*
11. $A(L,I) = \text{instance data baris ke-L kolom ke-I}$
12. $B(I) = \text{instance data prediksi kolom ke - I}$
13. $X = (B(I) - A(L,I))^2$
14. $Y = Y + X$
15. $I = I + 1$
16. *Loop (Step 10)*
17. $Z(L) = \text{SQRT}(Y)$
18. *Replace jarak with Z(L)*
19. $L = L + 1$
20. Next 1
21. *Loop (Step 7)*
22. *Sort data ascending by jarak*
23. *Filter data 1 to k*
24. Hasil prediksi = nilai mayoritas

2.5.1 Karakteristik Klasifikasi dengan *K-Nearest Neighbor*

Beberapa karakteristik K-NN adalah sebagai berikut:

1. K-NN merupakan algoritma yang menggambarkan seluruh data *training* untuk melakukan proses klasifikasi (*complete storage*). Hal ini mengakibatkan proses prediksi yang sangat lama untuk data dalam jumlah yang sangat besar. Tetapi semakin banyak data *training*, maka akan dapat dibuat batas keputusan yang semakin halus.
2. Algoritma K-NN tidak membedakan setiap fitur dengan suatu bobot, semua bobot untuk fiturnya bernilai sama untuk satu sama lainnya.
3. Karena K-NN masuk kategori *lazy learning* yang menyimpan sebagian atau semua data dan hampir tidak ada proses pelatihan, K-NN sangat cepat dalam proses pelatihan (karena memang tidak ada), tetapi sangat lambat dalam proses prediksi.
4. Hal yang rumit adalah menentukan nilai k yang paling sesuai karena K-NN pada prinsipnya memilih tetangga terdekat, parameter jarak juga penting untuk dipertimbangkan sesuai dengan kasus datanya. Jarak *Ecluidian* sangat cocok untuk menggunakan jarak terdekat (lurus) antara dua data.

2.6 *Android*

Android menurut Nazaruddin (2012,p.1) merupakan sistem operasi untuk telepon seluler yang berbasis *Linux*. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. *Android* umum digunakan di *smartphone* dan juga *tablet PC*. Fungsinya sama seperti sistem operasi *Symbian* di *Nokia*, *iOS* di *Apple* dan *BlackBerry OS*.

Android tidak terikat ke satu merek *Handphone* saja, beberapa vendor terkenal yang sudah memakai *Android* antara lain *Samsung*, *Sony Ericsson*, *HTC*, *Nexus*, *Motorolla*, dan lain-lain Pada Juli 2000, *Google* bekerjasama dengan

Android Inc, perusahaan yang berada di *Palo Alto, California* Amerika Serikat.

Para pendiri *Android Inc*. bekerja pada *Google*, diantaranya Andy Rubin, *Rich Miner*, *Nick Sears*, dan *Chris White*. Saat itu banyak yang menganggap fungsi *Android Inc*. Hanyalah sebagai perangkat lunak pada telepon *seluler*. Sejak saat itu muncul rumor bahwa *Google* hendak memasuki pasar telepon *seluler*. Di perusahaan *Google*, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel *Linux*. Hal ini menunjukkan indikasi bahwa *Google* sedang bersiap menghadapi persaingan dalam pasar telepon *seluler*. Versi *android* terbaru yaitu versi 4.0. (*Ice Cream Sandwich*).

Android juga sudah bergabung dengan beberapa *smart mobile* seperti *LG*, *Samsung*, *Sony Ericsson*, dan lainnya. Sekitar September 2007 sebuah studi melaporkan bahwa *Google* mengajukan hak paten aplikasi telepon seluler (akhirnya *Google* mengenalkan *Nexus One*, salah satu jenis telepon pintar *GSM* yang menggunakan *Android* pada sistem operasinya. Telepon seluler ini diproduksi oleh *HTC Corporation* dan tersedia di pasaran pada (5 Januari 2010). Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja *Android ARM Holdings*, *Atheros Communications*, diproduksi oleh *Asustek Computer Inc*, *Garmin Ltd*, *Softbank*, *Sony Ericsson*, *Toshiba Corp*, dan *Vodafone Group Plc*. Seiring pembentukan *Open Handset Alliance*, *OHA* mengumumkan produk perdana mereka, *Android*, perangkat bergerak (*Mobile*) yang merupakan modifikasi kernel *Linux 2.6*. Sejak *Android* dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru.

Banyak *smartphone* dan *PC Tablet* menggunakan sistem operasi dengan versi yang berbeda. Semakin tinggi versi, fiturnya semakin canggih dan banyak. Telepon pertama yang memakai sistem operasi *Android* adalah *HTC Dream* yang dirilis pada tanggal 22 oktober 2008.

2.7 Web Server

Web server merupakan *software* yang memberikan layanan data, berfungsi menerima permintaan *HTTP* atau *HTTPS* dari client yang dikenal dengan *browser* web dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman *web* yang umumnya berbentuk dokumen *HTML*, konsep *web server* antara lain:

- a) *Web server* merupakan mesin aplikasi atau software yang beroperasi dalam medistribusikan *web page* ke *user*, tentu saja sesuai dengan permintaan user.
- b) Hubungan antara *web server* dan *browser* internet merupakan gabungan atau jaringan komputer yang berada diseluruh dunia. Setelah terhubung secara fisik, protocol *TCP/IP* (*networking protocol*) yang memungkinkan semua komputer dapat berkomunikasi antar satu dengan lainnya.

Pada saat aplikasi *browser* meminta data *web page* ke *server* maka instruksi permintaan data oleh *browser* tersebut dikemas dalam *TCP* yang merupakan protocol transport dan dikirim ke alamat yang merupakan *protocol* berikutnya yaitu *hyper text transfer protocol (HTTP)*. Data yang diparsing dari *browser* ke *web server* disebut sebagai *HTTP request* yang meminta halaman *web* dan kemudian *web server* akan mencari data *HTML* yang dibutuhkan dan dikemas dalam *TCP protocol* kemudian dikirim kembali ke *browser*. Data yang dikirim dari *server* ke *browser* disebut sebagai *HTTP response*. Jika data yang diminta oleh *browser* tidak ditemukan pada *web server* maka akan menampilkan *error* pada *web page* yaitu Error: 404 *Page Not Found*.

2.8 *Database*

Beberapa pengertian *database* menurut para ahli:

1. Menurut Kustyaningsih (2011,p.146), “*Database* adalah Struktur penyimpanan data. Untuk menambah, mengakses dan memperoses data yang disimpan dalam sebuah *database* komputer, diperlukan *sistem manajemen database* seperti *MySQL Server*”.
2. Menurut Anhar (2010,p.45), “*Database* adalah sekumpulan tabel-tabel yang berisi data dan merupakan kumpulan dari *field* atau kolom. Struktur file yang menyusun sebuah *database* adalah Data *Record* dan *Field*”.

3. Menurut Martono, (2009,p.32), “*Database* adalah sebagai kumpulan data dari penempatan tenaga kerja yang saling terkait dan mempengaruhi sesuai dengan tingkat kepentingannya sehingga data tersebut terintegrasi dan *independence*”.

Berdasarkan beberapa pendapat para ahli yang dikemukakan di atas dapat ditarik kesimpulan bahwa *database* adalah sekelompok data yang mempunyai ciri-ciri khusus dan dapat dikelola sedemikian rupa sehingga bisa menghasilkan sebuah format data yang baru.

2.9 Xampp

XAMPP merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, *MySQL*, *PHP*, *Perl*. *XAMPP* adalah *tool* yang menyediakan paket perangkat lunak dalam satu buah paket. Dalam paket *XAMPP* sudah terdapat *Apache* (*webserver*), *MySQL* (*database*), *PHP* (*server side scripting*), *Perl*, *FTP server*, *PHPMyAdmin* (Prasetyo, 2011).

2.10 MySQL

Beberapa pengertian *MySQL* menurut para ahli :

1. Menurut Raharjo (2011,p.21), “*MySQL* merupakan *RDBMS* (atau *server database*) yang mengelola *database* dengan cepat menampung dalam jumlah sangat besar dan dapat di akses oleh banyak *user*”.
2. Menurut Kadir (2008,p.2), “*MySQL* adalah sebuah *software open source* yang digunakan untuk membuat sebuah *database*”.

Berdasarkan pendapat yang dikemukakan di atas dapat ditarik kesimpulan bahwa *MySQL* adalah suatu *software* atau program yang digunakan untuk membuat sebuah *database* yang bersifat *open source*.

2.11 HTML

HTML kependekan dari *Hyper Text Markup Language*. Dokumen *HTML* adalah file *teks* murni yang dapat dibuat dengan *editor teks* sembarang. Dokumen ini dikenal sebagai *web page*. Dokumen *HTML* merupakan dokumen yang disajikan dalam browser *web surfer*. Dokumen ini umumnya berisi informasi atau *interface* sistem di dalam internet (Sidik & Pohan, 2010).

2.12 PHP

PHP adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah web dan bisa digunakan pada *HTML*. *PHP* merupakan singkatan dari “*PHP : Hypertext Preprocessor*”, dan merupakan bahasa yang disertakan dalam dokumen *HTML* sekaligus bekerja disisi server (*server-side HTML-embedded scripting*) (West, 2013). *PHP* adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru atau *up to date*. Semua script *PHP* dieksekusi pada *server* di mana *script* tersebut dijalankan. Berikut kelebihan dari *PHP* :

1. *PHP* mudah dibuat dan dijalankan karena *PHP* dapat berjalan dalam *web server* dan dalam sistem operasi yang berbeda pula.
2. *PHP* adalah *software open source* yang gratis dan bebas didistribusikan kembali dibawah dengan lisensi *GPL (GNU Public Licence)* jadi *user* dapat mengunduh *source code PHP* tanpa harus mengeluarkan uang atau khawatir dituntut oleh pihak pencipta *PHP*.
3. *PHP* bisa dioperasikan pada platform *linux* ataupun *windows*.
4. Banyak *web server* yang mendukung *PHP*, antara lain: *Apache*, *PWS*, *IIS*, dan lain-lain.
5. *PHP* juga didukung oleh banyak *database*, seperti *MySQL*, *PostgreSQL*, *Interbase*, *SQL*, dan lain-lain.
6. Bahasa pemrograman *PHP* sintaknya sederhana, singkat dan mudah untuk dipahami.

7. PHP merupakan bahasa yang dapat ditulis dengan menempelkan pada sinak-sintak *HTML*.

2.13 Web Browser

Menurut Winarno dan Utomo (2010,p.31) *web browser* adalah alat yang digunakan untuk melihat halaman web. *Web browser* adalah *program* untuk menampilkan halaman yang berbentuk kode *HTML*. Semua halaman *web* ditulis dengan *bahasa HTML (Hypertext Mark Up Language)*. Walaupun beberapa *file* mempunyai ekstensi yang berbeda (contoh : *html*, *.php*, *.php3*), *output file-file* tersebut tetap *HTML*. *HTML* adalah medium yang selalu dikirimkan ke *web browser* baik halaman itu berupa halaman statis, sebuah script (seperti **PHP**), ataupun yang dibuat oleh *program CGI (Common Gateaway Interface)*.

2.14 Web Hosting

Web Hosting atau dapat juga diartikan sebagai tempat penyimpanan data berupa file dengan ukuran *megabytes* (MB) hingga ukuran besar yaitu terabytes (TB) yang memiliki koneksi ke internet sehingga data dalam web hosting tersebut dapat di request (di minta) atau di akses oleh pengguna internet dari semua penjuru atau global. Hal inilah yang menyebabkan *website* dapat di akses secara bersamaan dalam dekade satu waktu.

Pada dasarnya server *web hosting* menggunakan sebuah komputer biasa yang sudah kita kenal, namun komputer untuk *web hosting* menggunakan beberapa komponen dengan jumlah daya tamung luar biasa dan *program* dasar sebuah *server hosting* harus mampu untuk *online* dalam waktu 24 jam setiap hari dan tanpa harus dimatikan atau di *refresh* dalam jangka waktu cukup lebih lama daripada komputer biasa yang sudah kita kenal.

Sekarang kita sedikit tahu tentang apa itu *Web Hosting* ? untuk sekarang ini banyak sekali layanan web hosting yang memungkinkan anda untuk memilih dari berbagai tempat. misalnya di indonesia, *singapore*, amerika dan banyak lagi. pemilihan tempat ini sangat terpengaruh pada kecepatan akses ketika seseorang

ingin mengunjungi *website* anda. untuk menentukan serta menangani pengguna sebaiknya anda memilih *web hosting* yang terdekat dengan anda. ini salah satu pilihan dan salah satu trik tentang *seo*.

Banyak yang menyarankan untuk memilih beberapa layanan *web hosting*. untuk itu tidak anda salahnya kalau anda baca halaman yang pernah saya kupas sebelumnya yaitu menentukan *web hosting* yang handal. demikian pengertian sebenarnya semoga artikel ini bermanfaat untuk anda untuk menentukan *website* anda dimana seharunya.

2.15 *Android Studio*

Android Studio adalah lingkungan pengembangan Android berdasarkan *IntelliJ IDEA*. Mirip dengan *Eclipse* dengan *ADT Plugin*, *Android Studio* menyediakan alat pengembang *Android* terintegrasi untuk pengembangan dan *debugging*. *Android Lab* merupakan platform *editor* yang berbasis *Gradle*, dengan *Android* lab proses *refactoring* dan perbaikan berlangsung dengan cepat, hal tersebut dapat dilakukan karena ada *tool* bernama *Lint* untuk menangkap kinerja, kegunaan, kompatibilitas versi *Android* dan masalah lainnya. *Android Lab* juga merupakan sebuah *layout editor* yang kaya akan fitur yang memungkinkan untuk *drag-and-drop* komponen *UI*.

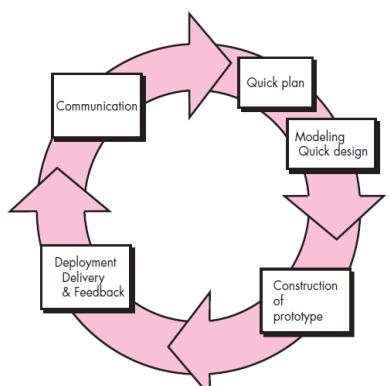
2.16 *ADT (Android Development Tools)*

Android Development Tools (ADT) menurut Nazarudin Safaat (2010,p.6) adalah *plugin* untuk *Eclipse Intergrated Development Environment (IDE)* yang dirancang untuk memberikan lingkungan yang terpadu di mana untuk membangun aplikasi *Android*. *ADT* memperluas kemampuan *Eclipse* untuk membiarkan para *developer* lebih cepat dalam membuat proyek baru *Android*, membuat aplikasi *UI*, menambahkan komponen berdasarkan *Android Framework API*, *debug aplikasi* dalam pengunaan *Android SDK*, dan membuat *file APK* untuk mendistribusikan aplikasi.

Mengembangkan aplikasi di *Eclipse* dengan *ADT* sangat dianjurkan dan merupakan cara tercepat untuk memulai membuat aplikasi *android*, karena banyak kemudahan-kemudahan sebagai *tools* yang terintegrasi seperti, *custom XML editor*, dan debug panel *output*. Selain itu *ADT* memberikan dorongan luar biasa dalam mengembangkan aplikasi.

2.17 Metode Pengembangan Sistem

Metodelogi yang digunakan untuk membangun sistem ini adalah Model *Prototype*. Model ini merupakan sebuah pendekatan terhadap pengembangan perangkat lunak yang sistematik, dengan beberapa tahapan, yaitu: *System Engineering, Analysis, Design, Coding, Testing dan Maintenance*. Pressman (2010,p.51) mengutarakan bahwa prototype adalah pengembangan yang cepat dan pengujian terhadap model kerja (*prototype*) dari aplikasi baru melalui proses interaksi dan berulang-ulang yang biasa digunakan ahli sistem informasi dan ahli bisnis. *Prototype* disebut juga desain aplikasi cepat (*rapid application design/RAD*) karena menyederhanakan dan mempercepat desain sistem. Bagian user kesulitan mengungkapkan keinginannya untuk mendapatkan aplikasi yang sesuai dengan kebutuhannya. Kesulitan ini yang perlu diselesaikan oleh analis dengan memahami kebutuhan user dan menerjemahkannya ke dalam bentuk model (*prototype*). Model ini selanjutnya diperbaiki secara terus menerus sampai sesuai dengan kebutuhan user.



Gambar 2.2 Model Prototype

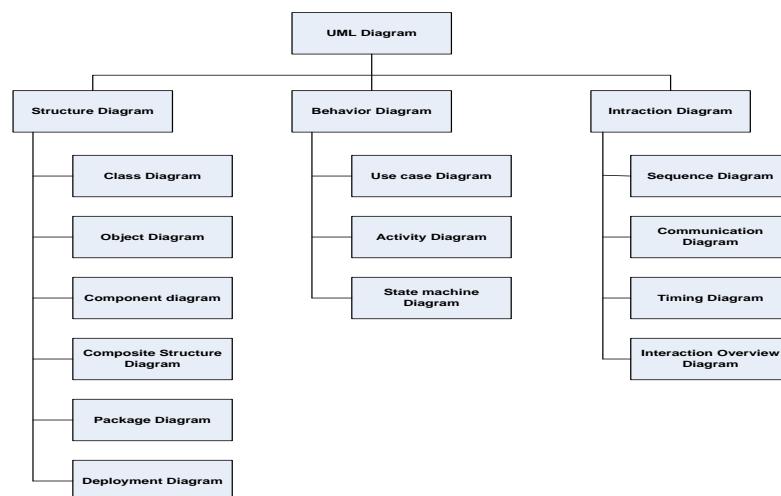
Penjelasan dari gambar 2.2 diatas adalah sebagai berikut :

Tahap pertama adalah *communication* dan pengumpulan data awal yaitu tahap suatu perencanaan yang di lakukan, mulai dari menciptakan dan melaksanakan proses untuk memastikan bahwa perencanaan tersebut berkualitas tinggi, terpercaya, *efisiensi* biaya. Tahap kedua adalah *quick plan* yaitu *analisis* terhadap kebutuhan pengguna. Tahap ketiga adalah *modelling quick design* yaitu pembuatan desain secara umum untuk selanjutnya dikembangkan kembali. Tahap keempat adalah *construction of prototype* adalah pembuatan perangkat *prototype* termasuk pengujian dan penyempurnaan. Tahap kelima adalah *deployment, delivery, and feedback* adalah tahap penyerahan sistem ke pengguna dan umpan balik.

2.18 Perancangan Sistem UML (*Unified Modelling Language*)

Menurut Rosa A.S dan M. Shalahuddin (2011,p.113):

“UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.” Pada *UML* terdiri dari 13 macam diagram yang dikelompokkan dalam tiga kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini:



Gambar 2.3 Diagram UML (*Unified Modelling Language*)

Sumber: Rosa A.S – M. Shalahuddin (2011)

Berikut ini merupakan penjelasan singkat dari pembagian kategori tersebut :

1. *Structure diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antarsubsistem pada suatu sistem.

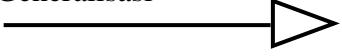
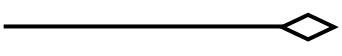
Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada UML (*Unified Modeling Language*):

1. *Class Diagram*

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut maupun metode atau operasi. *Atribut* merupakan variable-variabel yang dimiliki oleh suatu kelas. Sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Berikut adalah simbol-simbol yang ada pada diagram kelas:

Simbol	Deskripsi
Kelas	Kelas pada struktur sistem

nama_kelas
+atribut
+operasi()

Antarmuka/ <i>Interface</i>		Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
nama_interface		
Asosiasi/ <i>asociation</i>		Relasi antar kelas dengan makna umum, asosiasi biasanya disertai dengan <i>multiplicity</i>
Asosiasi berarah/ <i>directed association</i>		Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i> .
Generalisasi		Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan/ <i>dependency</i>		Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi/ <i>aggregation</i>		Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

Gambar 2.4 Simbol-simbol Class Diagram

Sumber: Rosa, A.S (2011,p.123-124)

4. *Object Diagram*

Object Diagram menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. Pada diagram objek harus dipastikan semua kelas yang sudah didefinisikan pada diagram kelas harus dipakai objeknya, karena jika tidak maka pendefinisian kelas itu tidak dapat dipertanggung jawabkan.

5. Component Diagram

Component Diagram dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. *Diagram* komponen fokus pada komponen sistem yang dibutuhkan dan ada didalam sistem.

6. Composite Structure Diagram

Composite Structure Diagram baru mulai ada pada *UML* versi 2.0, pada versi 1.x *diagram* ini belum muncul. *Diagram* ini dapat digunakan untuk menggambarkan struktur dari bagian-bagian yang saling terhubung maupun mendeskripsikan struktur pada saat berjalan dari *instance* yang saling terhubung.

7. Package diagram

Package diagram menyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam diagram *UML*. Hampir semua diagram dalam *UML* dapat dikelompokkan menggunakan *Package diagram*.

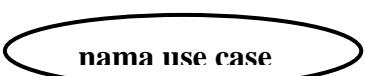
8. Deployment Diagram

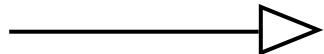
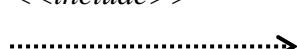
Deployment Diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi.

9. Use Case Diagram

Use Case Diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Berikut adalah simbol-simbol yang ada pada diagram *use case*:

Simbol	Deskripsi
<i>Use Case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar

		pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
Aktor/ <i>actor</i>		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang
Asosiasi/ <i>asociation</i>	<hr/>	Komunikasi antara aktor dan <i>use case</i> yang berpatisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
Ekstensi/ <i>extend</i>		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan
Generalisasi/ <i>generalization</i>		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya
Menggunakan/ <i>Include/uses</i>		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai

syarat dijalankan *use case* ini.

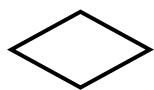
Gambar 2.5 Simbol-Simbol Use Case Diagram

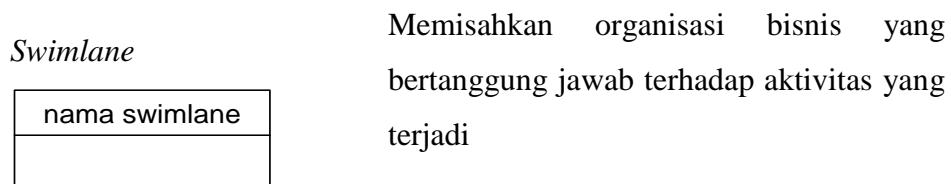
Sumber: Rosa, A.S (2011,p.131-133)

10. *Activity Diagram*

Activity Diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas.

Berikut adalah simbol-simbol yang ada pada diagram *use case*:

Simbol	Deskripsi
Status awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
	
Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
	
Percabangan/ <i>decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
	
Penggabungan/ <i>join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
	
Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	



Gambar 2.6 Simbol-Simbol Use Case Diagram

Sumber: Rosa, A.S (2011,p.134-135)

11. *State Machine Diagram*

State Machine Diagram digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem. Perubahan tersebut digambarkan dalam suatu *graf* berarah. *Diagram* ini cocok digunakan untuk menggambarkan alur interaksi pengguna dengan sistem.

12. *Sequence Diagram*

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek.

13. *Communication Diagram*

Communication Diagram pada UML versi 2.x menyederhanakan dari diagram kolaborasi pada UML 1.x. jadi diagram kolaborasi tidak muncul lagi pada UML versi 2.x. diagram komunikasi menggambarkan interaksi antar objek dalam bentuk urutan pengiriman pesan. *Diagram* komunikasi merepresentasikan informasi yang diperoleh dari diagram kelas, *diagram* sekuen, dan *diagram use case* untuk mendeskripsikan gabungan antara struktur statis dan tingkah laku dinamis dari suatu sistem.

14. *Timing Diagram*

Timing Diagram merupakan *diagram* yang fokus pada penggambaran terkait batasan waktu. *Diagram* ini digunakan untuk menggambarkan tingkah laku sistem dalam periode waktu tertentu. *Timing diagram* biasanya digunakan untuk

mendeskripsikan operasi dari alat digital karena penggambaran secara visual akan lebih mudah dipahami dari pada kata-kata.

15. *Interaction Overview Diagram*

Interaction Overview Diagram mirip dengan diagram aktivitas yang berfungsi untuk menggambarkan sekumpulan urutan aktivitas. *Interaction Overview Diagram* adalah bentuk aktivitas diagram yang setiap titik merepresentasikan diagram interaksi.

BAB III

METODE PENELITIAN

3.1 Metode Pengumpulan Data

Pengumpulan data yang akan dilakukan pada penelitian ini adalah sebagai berikut:

1. Wawancara (*Interview*)

Pengumpulan data dengan metode *interview* yaitu metode pengumpulan data dengan cara tanya jawab secara langsung dengan orang-orang yang terkait yaitu Karywan dan Pimpinan PT. Colombus yang ada di Tanjung Karang Pusat Bandar Lampung. Melakukan pertanyaan seputar sistem yang berjalan mengenai penyampain informasi kepada konsumen serta proses pembelian elektronik dan forniture secara kredit.

2. Pengamatan (*Observation*)

Pengumpulan data dengan mengamati atau *observation* yaitu metode pengumpulan data dengan cara pengamatan dan pencatatan secara langsung. Mempelajari segala sesuatu yang berhubungan dengan sistem yang akan dibangun. Mengamati secara langsung seputar sistem yang berjalan mengenai informasi penjualan elektronik dan forniture serta pemebelian secara *kredit*.

3. Dokumentasi (*Documment*)

Merupakan *metode* pengumpulan data dengan cara membaca, mencatat, mengutip, dan mengumpulkan data-data secara teoritis dari buku-buku dan Internet sebagai landasan penyusunan penelitian. Peneliti meminjam buku diperpustakaan, mencari data dari internet juga dilakukan untuk referensi laporan ini, dimana teori tersebut diletakkan pada landasan teori.

4. Tinjauan Pustaka (*Library Research*)

Metode mempelajari kumpulan buku-buku yang dilakukan dengan cara membaca literatur-literatur dan tata bahasa baik yang ada di perpustakaan maupun lainnya

yang terkait dengan data yang dibutuhkan, sehingga dapat menunjang proses penelitian.

A. Metode Pengembangan Perangkat Lunak

Pada metode pengembangan perangkat lunak penulisan memerlukan bantuan untuk menghasilkan suatu rangcangan dalam membuat sebuah Rancang Bangun Aplikasi Penentuan Kredit Berjangka *Online* Menggunakan Algoritma *K-Nearst Neighbor*. Pada PT. Columbus Tanjung Karang Pusat Bandar Lampung. Metode yang digunakan adalah model *prototype* yang memiliki lima tahapan yaitu sebagai berikut :

3.2.1 *Communication*

Tahap *communication* pada penelitian ini yaitu suatu perencanaan yang di lakukan, mulai dari menciptakan dan melaksanakan proses untuk memastikan bahwa perencanaan tersebut berkualitas tinggi, terpercaya, *efisiensi* biaya dan terjadwalkan data-data yang didapat saat penelitian di PT.Columbus Tanjung Karang Bandar Lampung. Adapun secara ringkas langkah-langkah metode ilmiah adalah sebagai berikut:

a. Merumuskan Masalah

Tahapan ini merupakan langkah pertama metode ilmiah. Merumuskan masalah bertujuan untuk memperjelas masalah dengan mengajukan beberapa atau serangkaian pertanyaan terhadap masalah yang ada.

b. Melakukan Penyusunan Rencana Penelitian

Langkah kedua dalam metode ilmiah adalah penyusunan rencana. Rencana penelitian dibuat dengan membuat tujuan penelitian agar rencana penelitian lebih jelas. Hal ini tentu saja dilakukan dengan membuat tinjauan pustaka sehingga diperoleh data-data yang berhubungan dan metode penelitian yang akan dilakukan.

c. Melakukan Penelitian

Ini merupakan langkah metode ilmiah yang dilakukan setelah rencana penelitian atau proposal telah diterima. Penelitian sendiri tergantung pada langkah penelitian atau *metodologi* penelitian yang akan digunakan.

3.2.2 Quick Plan

3.2.2.1 Analisis dan Definisi Persyaratan

Tahapan *quick plan* dilakukan untuk menetapkan bagaimana perangkat lunak akan di operasikan. Hal ini berkaitan untuk menentukan perangkat keras, perangkat lunak, tampilan program dan *form-form* yang akan dipakai dalam pembuatan *prototype*.

Data dari kebutuhan *software* yang akan diperoleh pada tahap sebelumnya, kemudian dianalisis dan menghasilkan sebuah data kebutuhan dari pengguna aplikasi. Adapun analisis kebutuhan *software* yang telah diperoleh adalah sebuah kebutuhan perangkat lunak yang digunakan untuk membangun sebuah Aplikasi Penentuan Kredit Berjangka *Online* Menggunakan Algoritma *K-Nearst Neighbor* Pada PT. Columbus Tanjung Karang Pusat Bandar Lampung adalah sebagai berikut :

a. Analisis *software*

Software yang digunakan merupakan perangkat lunak yang akan digunakan sebagai media pembuatan dan menjalankan perintah pada aplikasi yang akan dibuat. Adapun spesifikasi *software* yang diperlukan adalah :

1. Sistem operasi *Microsoft Windows 7 Professional*.
2. *Software* pendukung dalam pembuatan aplikasi antara lain, sebagai berikut :
 - a) *Adobe Photoshop* digunakan untuk membuat desain logo dan icon aplikasi.
 - b) *Xampp*, digunakan sebagai server.
 - c) *Atom*, digunakan sebagai pengkodingan system.
 - d) *Android Studio*
 - e) Analisis *hardware*

Hardware berfungsi sebagai perangkat keras yang mendukung jalannya sebuah mengolahan data serta memberikan *output* pada aplikasi yang ada pada perangkat *mobile* maupun *smartphone*. Spesifikasi *hardware* diperlukan adalah :

1. *Processor* Intel Core™ i3.
2. RAM 4 GB RAM DDR3.
3. Monitor 14 in.
4. *Harddisk* 500 GB.

3.2.2.2 Skema Representasi

Pangkalan pengetahuan yang dibangun menggunakan suatu skema representasi pengetahuan. Skema representasi yang digunakan adalah sebagai berikut :

A. Kebutuhan Data

Data set yang diperoleh dari hasil observasi yang dilakukan pada PT.Columbus akan dijadikan sebagai data training untuk menentukan kelayakan kredit. Atribut-atribut yang diperlukan dalam perencanaan sistem pengajuan kredit adalah sebagai berikut :

1. Jenis Pekerjaan
2. Jumlah Tanggungan
3. Penghasilan
4. Pendidikan
5. Lama Kerja
6. Berkas

Dibawah ini tabel 3.1 merupakan data training yang diperoleh dari hasil penelitian :

Tabel 3.1 Tabel Data Training

Nama	Penghasilan	Jumlah Tanggungan	Jenis Pekerjaan	Pendidikan	Lama bekerja	Berka s	Keputu san
Anita, A.md	2-3 jta	2	Karyawan Tetap	D3	>1 Thn	KTP & KK	ACC

Kadir, S.Pd	4-9 jta	3	PNS	S1	>1 Thn	KTP & KK	ACC
Mujab	2-3 jta	1	Wiraswasta	SMA	>1 Thn	KTP & KK	ACC
Sugiri	<2 jta	3	Wiraswasta	SMA	<3 Bulan	KTP/ KK Saja	Tidak ACC
Wari	2-3 jta	1	Wiraswasta	SMA	>1 Thn	KTP &KK	ACC

B. Penentuan Bobot Atribut

Setelah menentukan atribut, selanjutnya menentukan bobot untuk setiap atribut sesuai dengan tingkat kepentingan atau seberapa pengaruhnya atribut terhadap hasil keputusan pengajuan kredit. Dalam pemberian bobot ini diberikan berdasarkan tingkat kepentingan dari setiap atribut dengan total persentase 100%. Bobot penentuan kredit berdasarkan tingkat kepentingan dapat dilihat pada tabel 3.2 :

Tabel 3.2 Bobot Atribut

No	Nama Atribut	Tingkat Kepentingan	Bobot
1	Penghasilan	1	0.35
2	Jumlah Tanggungan	2	0.25
3	Jenis Pekerjaan	3	0.20
4	Pendidikan	4	0.10
5	Lama Kerja	5	0.8
6	Berkas	6	0.2

C. Penentuan Sub Atribut

Setelah menentukan atribut dan bobot , selanjutnya menentukan sub atribut untuk setiap atribut sesuai dengan atribut. Penentuan sub atribut ini akan digunakan dalam menentukan nilai Jarak untuk setiap sub atribut dalam menentukan hasil keputusan berdasarkan data yang lama. Penentuan sub atribut dapat dilihat pada tabel 3.3 :

Tabel 3.3 Sub Atribut

No	Nama Atribut	Sub Atribut
1	Penghasilan	1. < 2 Juta 2. 2 – 3 Juta 3. 4-9 Juta 4. >10 Juta
2	Jumlah Tanggungan	1. 0 (Tidak Ada) 2. 1 3. 2-3 4. >3
3	Jenis Pekerjaan	1. Horoner 2. Wiraswasta 3. Karyawan Tetap 4. PNS
4	Pendidikan	1. SMP 2. SMA 3. D3 4. S1-S3
5	Lama Kerja	1. < 3 bulan 2. 3 – 6 bulan 3. 6 – 12 bulan 4. > 1 tahun
6	Berkas	1. KTP & KK 2. KTP / KK Saja 3. Tidak Ada

D. Penentuan Bobot Sub Atribut

Setelah menentukan atribut ,bobot serta atribut , selanjutnya menentukan bobot sub atribut untuk setiap atribut sesuai dengan atribut. Penentuan bobot sub atribut ini akan digunakan dalam menentukan nilai Jarak untuk setiap sub atribut dalam menentukan hasil keputusan berdasarkan data yang lama. Penentuan sub atribut dapat dilihat pada tabel 3.4 :

Tabel 3.4 Bobot Sub Atribut

No	Nama Atribut	Sub Atribut	Bobot
1	Penghasilan	>10 Juta	1
		4- 9 Juta	0.75
		2 – 3 Juta	0.50
		< 2Juta	0.25
2	Jumlah Tanggungan	0 (Tidak Ada)	1
		1	0.75
		2-3	0.50
		>3	0.25
3	Jenis Pekerjaan	PNS	1
		Karyawan Tetap	0.75
		Wiraswasta	0.50
		Honorer	0.25
4	Pendidikan	S1-S3	1
		D3	0.75
		SMA	0.50
		SMP	0.25
5	Lama Kerja	> 1 Tahun	1
		6 - 12 Bulan	0.75
		3 – 6 Bulan	0.50
		< 3 Bulan	0.25
6	Berkas	KTP & KK	1
		KTP/KK Saja	0.75
		Tidak Ada	0.50

E. Penentuan Nilai Jarak Sub Atribut

Setelah menentukan *atribut*, bobot, sub *atribut* serta bobot sub *atribut* , selanjutnya menentukan nilai jarak untuk setiap sub *atribut* dengan sub *atribut* lainnya. Penentuan nilai Jarak untuk setiap sub atribut dapat dilihat pada tabel 3.5 dibawah ini :

Tabel 3.5 Jarak sub *atribut* pada *atribut penghasilan*

No	Sub Atribut	Sub Atribut	(x-y)	$x - y^2$
1	>10 Juta	< 2 Juta	0,75	0,5625
2	>10 Juta	2- 3 Juta	0,5	0,25
3	>10 Juta	4 - 9 Juta	0,25	0,0625
4	>10 Juta	> 10 Juta	0	0
5	4 – 9 Juta	< 2 Juta	0,5	0,25
6	4 – 9 Juta	2- 3 Juta	0,25	0,0625
7	4 – 9 Juta	4 - 9 Juta	0	0
8	4 – 9 Juta	> 10 Juta	-0,25	0,0625
9	2 - 3 Juta	< 2 Juta	0,25	0,0625
10	2 - 3 Juta	2- 3 Juta	0	0
11	2 - 3 Juta	4 - 9 Juta	-0,25	0,0625
12	2 - 3 Juta	> 10 Juta	-0,5	0,25
13	< 2 Juta	< 2 Juta	0	0
14	< 2 Juta	2- 3 Juta	-0,25	0,0625
15	< 2 Juta	4 - 9 Juta	-0,50	0,25
16	< 2 Juta	> 10 Juta	-0,75	0,5625

Dari tabel dibawah menjelaskan rincian perhitungan pada *atribut jumlah tanggungan*, yang mana pada setiap nilai memiliki bobot tertentu dan kemudian dihitung dengan data base setelah itu dikuadratkan sehingga hasilnya seperti tabel 3.6 dibawah ini :

Tabel 3.6 Jarak sub *atribut* pada *atribut jumlah tanggungan*

No	Sub Atribut	Sub Atribut	(x-y)	$x - y^2$
1	0 (Tidak Ada)	0 (Tidak Ada)	0,00	0,00
2	0 (Tidak Ada)	1	0,25	0,0625
3	0 (Tidak Ada)	2-3 Juta	0,50	0,25
4	0 (Tidak Ada)	>3	0,75	0,5625
5	1	0 (Tidak Ada)	-0,25	0,0625
6	1	1	0,00	0,00
7	1	2—3	0,25	0,0625
8	1	>3	0,50	0,25
9	2-3	0 (Tidak Ada)	-0,50	0,25

10	2-3	1	-0,25	0,0625
11	2-3	2—3	0,00	0,00
12	2-3	>3	0,25	0,0625
13	>3	0 (Tidak Ada)	-0,75	0,5625
14	>3	1	-0,50	0,25
15	>3	2--3	-0,25	0,0625
16	>3	>3	0,00	0,00

Dari tabel dibawah menjelaskan rincian perhitungan pada atribut jenis pekerjaan, yang mana pada setiap nilai memiliki bobot tertentu dan kemudian dihitung dengan *database* setelah itu dikuadratkan sehingga hasilnya seperti tabel 3.7 dibawah ini :

Tabel 3.7 Jarak sub *atribut* pada *atribut* jenis pekerjaan

No	Sub Atribut	Sub Atribut	$(x-y)^2$	x	y^2
1	PNS	Honorer	0,75		0,5625
2	PNS	Wiraswasta	0,50		0,25
3	PNS	Karyawan Tetap	0,25		0,0625
4	PNS	PNS	0,00		0,00
5	Karyawan Tetap	Honorer	0,50		0,25
6	Karyawan Tetap	Wiraswasta	0,25		0,0625
7	Karyawan Tetap	Karyawan Tetap	0,00		0,00
8	Karyawan Tetap	PNS	-0,25		0,0625
9	Wiraswasta	Honorer	0,25		0,0625
10	Wiraswasta	Wiraswasta	0,00		0,00
11	Wiraswasta	Karyawan Tetap	-0,25		0,0625
12	Wiraswasta	PNS	-0,50		0,25
13	Honorer	Honorer	0,00		0,00
14	Honorer	Wiraswasta	-0,25		0,0625
15	Honorer	Karyawan Tetap	-0,50		0,25
16	Honorer	PNS	-0,75		0,5625

Dari tabel dibawah menjelaskan rincian perhitungan pada *atribut* pendidikan, yang mana pada setiap nilai memiliki bobot tertentu dan kemudian dihitung dengan *data base* setelah itu dikuadratkan sehingga hasilnya seperti tabel 3.8 dibawah ini :

Tabel 3.8 Jarak sub *atribut* pada *atribut* pendidikan

No	Sub Atribut	Sub Atribut	(x-y)	$x - y^2$
1	S1-S3	SMP	0,75	0,5625
2	S1-S3	SMA	0,50	0,25
3	S1-S3	D3	0,25	0,0625
4	S1-S3	S1-S3	0,00	0,00
5	D3	SMP	0,50	0,25
6	D3	SMA	0,25	0,0625
7	D3	D3	0,00	0,00
8	D3	S1-S3	-0,25	0,0625
9	SMA	SMP	0,25	0,0625
10	SMA	SMA	0,00	0,00
11	SMA	D3	-0,25	0,0625
12	SMA	S1-S3	-0,50	0,25
13	SMP	SMP	0,00	0,00
14	SMP	SMA	-0,25	0,0625
15	SMP	D3	-0,50	0,25
16	SMP	S1-S3	-0,75	0,5625

Dari tabel dibawah menjelaskan rincian perhitungan pada *atribut* lama kerja, yang mana pada setiap nilai memiliki bobot tertentu dan kemudian dihitung dengan data base setelah itu dikuadratkan sehingga hasilnya seperti tabel 3.9 dibawah ini :

Tabel 3.9 Jarak sub *atribut* pada *atribut* lama kerja

No	Sub Atribut	Sub Atribut	(x-y)	$x - y^2$
1	>1Tahun	< 3 Bulan	0,75	0,5625
2	>1Tahun	3-6 Bulan	0,50	0,25
3	>1Tahun	6 – 12 Bulan	0,25	0,0625
4	>1Tahun	>1Tahun	0,00	0,00
5	6 – 12 Bulan	< 3 Bulan	0,50	0,25
6	6 – 12 Bulan	3-6 Bulan	0,25	0,0625
7	6 – 12 Bulan	6 – 12 Bulan	0,00	0,00
8	6 – 12 Bulan	>1Tahun	-0,25	0,0625
9	3-6 Bulan	< 3 Bulan	0,25	0,0625
10	3-6 Bulan	3-6 Bulan	0,00	0,00

11	3-6 Bulan	6 – 12 Bulan	-0,25	0,0625
12	3-6 Bulan	>1Tahun	-0,50	0,25
13	< 3 Bulan	< 3 Bulan	0,00	0,00
14	< 3 Bulan	3-6 Bulan	-0,25	0,0625
15	< 3 Bulan	6 – 12 Bulan	-0,50	0,25
16	< 3 Bulan	>1Tahun	-0,75	0,5625

Dari tabel dibawah menjelaskan rincian perhitungan pada *atribut* berkas, yang mana pada setiap nilai memiliki bobot tertentu dan kemudian dihitung dengan data base setelah itu dikuadratkan sehingga hasilnya seperti tabel 3.10 dibawah ini :

Tabel 3.10 Jarak sub *atribut* pada *atribut* berkas

No	Sub Atribut	Sub Atribut	(x-y)	$x - y^2$
1	KTP & KK	KTP & KK	0,00	0,00
2	KTP & KK	KTP/KK Saja	0,25	0,0625
3	KTP & KK	Tidak Ada	0,50	0,25
4	KTP/KK Saja	KTP & KK	-0,25	0,0625
5	KTP/KK Saja	KTP/KK Saja	0,00	0,00
6	KTP/KK Saja	Tidak Ada	0,25	0,0625
7	Tidak Ada	KTP & KK	-0,50	0,25
8	Tidak Ada	KTP/KK Saja	-0,25	0,0625
9	Tidak Ada	Tidak Ada	0,00	0,00

F. Testing Kasus Baru

Setelah menentukan *atribut*, bobot, sub *atribut* serta bobot sub *atribut* , serta menentukan nilai Jarak untuk setiap sub *atribut* dengan sub *atribut* lainnya, Selanjutkan menghitung kasus baru berdasarkan data *training* yang telah ada:

Tabel 3.11 Tabel Kasus Baru

Nama	Penghasilan	Jumlah Tanggungan	Jenis Pekerjaan	Pendidikan	Lama bekerja	Berkas
Ridho Yan Fadli	2-3 jta	0	Karyawan Tetap	SMA	>1 Thn	KTP&KK

Selanjutnya membandingkan nilai Jarak dengan data *training* yang telah ada sebelumnya :

1. Ridho Yan Fadli dengan Mujab
 - a. Nilai Jarak Penghasilan 2-3Jta dengan 2-3 Jta = 0
 - b. Bobot Penghasilan = 0.35
 - c. Nilai Jarak jumlah tanggungan 0 dengan 1 = 0.0625
 - d. Bobot Jumlah tanggungan =0.25
 - e. Nilai Jarak jenis pekerjaan Wiraswasta dengan karyawan tetap =0,0625
 - f. Bobot Jenis Pekerjaan =0.20
 - g. Nilai Jarak pendidikan SMA dengan SMA =0
 - h. Bobot Pendidikan =0.10
 - i. Nilai Jarak Lama Kerja > 1 Tahun dengan >1 Tahun = 0
 - j. Bobot Jarak Lama Kerja = 0.8
 - k. Nilai Jarak berkas KTP& KK dengan KTP& KK = 0
 - l. Bobot berkas = 0.02

Kemudian mencari nilai Jarak dengan antara kasus baru dengan kasus lama:

$$D_{mujab} = \sqrt{(0,35 * 0) + (0,25 * 0,0625) + (0,20 * 0,0625) + (0,10 * 0) + (0,08 * 0) + (0,02 * 0)} \\ = 0,182002747$$

Tabel 3.12 Tabel Nilai Jarak Berdasarkan Data Training

Nama	Penghasilan	Jumlah Tanggungan	Jenis Pekerjaan	Pendidikan	Lama bekerja	Berkas
Ridho Yan Fadli	2-3 jta	0	Karyawan Tetap	SMA	>1 Thn	KTP&KK

Nama	Penghasilan	Jumlah Tanggungan	Jenis Pekerjaan	Pendidikan	Lama bekerja	Berkas
	n		n	n		
Anita, A.md	0	0,25	0	0,0625	0	0
Kadir, S.Pd	0,0625	0,25	0,0625	0,25	0	0
Mujab	0	0,0625	0,0625	0	0,5625	0
Sugiri	0,0625	0,25	0,0625	0	0	0,0625
Wari	0	0,0625	0,0625	0,0625	0	0

Selanjutnya melakukan perhitungan yang sama seperti yang dilakukan untuk mencari nilai D pada anita diatas

$$D_{\text{Kadir}} = \sqrt{(0,35 * 0,0625) + (0,25 * 0,25) + (0,20 * 0,0625) + (0,10 * 0,25) + (0,08 * 0) + (0,02 * 0)} \\ = 0,340036763$$

$$D_{\text{Sugiri}} = \sqrt{(0,35 * 0,0625) + (0,25 * 0,25) + (0,20 * 0,0625) + (0,10 * 0) + (0,08 * 0) + (0,02 * 0,0625)} \\ = 0,287228132$$

$$D_{\text{Wari}} = \sqrt{(0,35 * 0) + (0,25 * 0,0625) + (0,20 * 0,0625) + (0,10 * 0,0625) + (0,08 * 0) + (0,02 * 0)} \\ = 0,185404962$$

Danita

$$= \sqrt{(0,35 * 0) + (0,25 * 0,25) + (0,20 * 0) + (0,10 * 0,0625) + (0,08 * 0) + (0,02 * 0)} \\ = 0,2622022$$

Hasil perhitungan nilai Jarak setalah diurutkan berdasarkan yang paling dekat atau yang memiliki nilai terkecil pada tabel 3.13 :

Tabel 3.13 Tabel Hasil Nilai antara kasus baru dengan kasus lama

Nama	Nilai Jarak
Anita, A.md	0,262202212
Kadir, S.Pd	0,340036763
Mujab	0,182002747
Sugiri	0,287228132
Wari	0,185404962

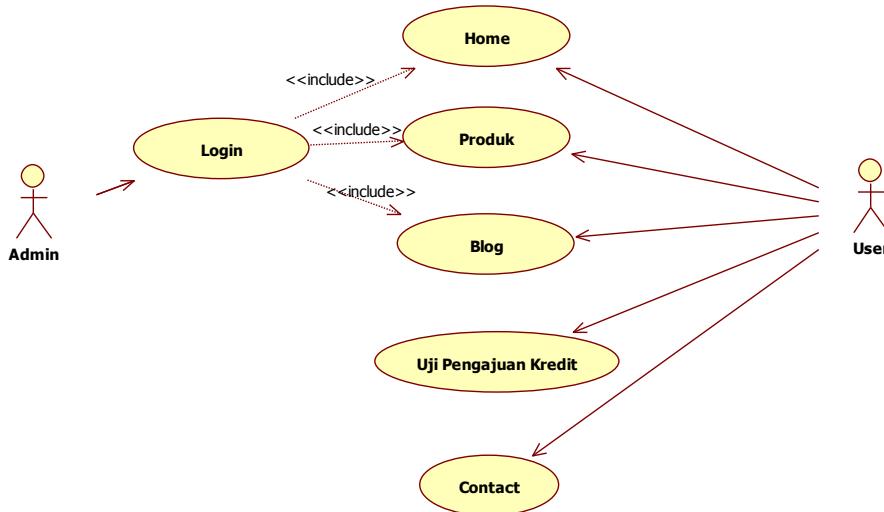
Berdasarkan nilai Jarak diatas maka kasus baru tersebut penyejuan kredit diterima dikarenakan yang memiliki nilai Jarak dengan kasus baru tersebut adalah kasus atas nama Mujab, dan Mujab memiliki history pengajuan kredit yang ACC.

3.2.3 Modelling Quick Design

Adapun penerapan *Unified Modeling Language* (UML) untuk perancangan desain interface tampilan Aplikasi Penentuan Kredit Berjangka *Online* Menggunakan Algoritma *K-Nearst Neighbor* Pada PT. Colombus Tanjung Karang Pusat Bandar Lampung, sebagai berikut :

1. Use Case Sistem

Diagram dibawah ini menunjukkan fungsi sebuah sistem atau kelas, bagaimana sistem tersebut dapat berinteraksi dengan pengguna (*User*). Adapaun use case pada aplikasi ini sebagai berikut :



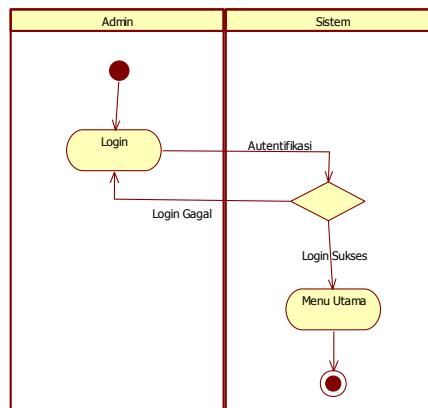
Gambar 3.7 Use Case Sistem

- Nama use case** : Menu Home
Actor : User & Admin
Tujuan : Melihat informasi menu awal aplikasi
Deskripsi : Pada menu ini memnampilkan beberapa menu yang dapat digunakan oleh user.
- Nama use case** : Menu Produk
Actor : User & Admin
Tujuan : Menampilkan data produk yang ada
Deskripsi : Pada menu ini user dapat melihat produk-produk yang tersedia dan untuk admin dapat menambahkan produk baru.
- Nama use case** : Menu Blog
Actor : User & Admin
Tujuan : Menampilkan informasi-informasi.
Deskripsi : Pada menu ini user dapat melihat informasi-informasi bary yang tersedia dan untuk admin dapat menambahkan informasi baru.
- Nama use case** : Menu Uji Pengajuan Kredit
Actor : User

- Tujuan : Untuk menguji apakah user dapat mengajukan kredit atau tidak.
- Deskripsi : Pada menu ini user dapat mengisi form yang telah disediakan dan dapat mengetahui hasil dari kredit yang ingin diajukan.
- e. Nama use case : Menu Contact
 Actor : User
 Tujuan : Untuk memberikan informasi kontak admin atau customer service.
 Deskripsi : Pada menu ini user dapat melihat nomor customer service yang dapat dihubungi serta alamat tempat PT.Columbus.

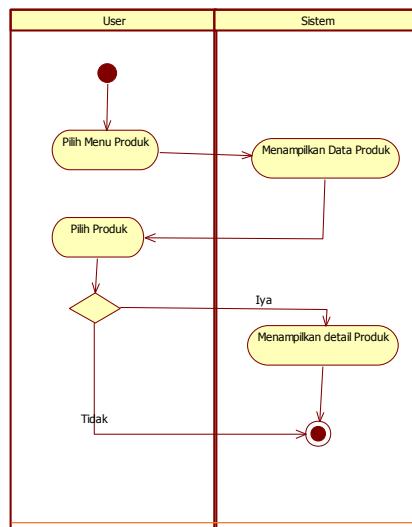
2. *Activity Diagram* pada Sistem

Activity Diagram merupakan bentuk khusus dari state machine yang bertujuan memodelkan komputasi-komputasi dan aliran-aliran kerja yang terjadi dalam sistem/perangkat lunak yang sedang dikembangkan. *Activity diagram* dapat dilihat pada gambar-gambar di berikut ini. Gambar 3.8 berikut merupakan *activity diagram* ketika admin melakukan login ke sistem admin :



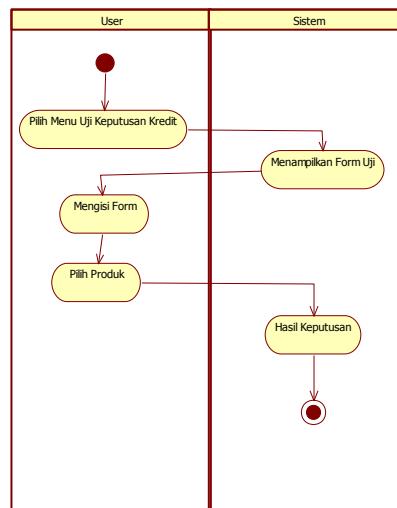
Gambar 3.8 Activity Diagram Login Admin

Gambar 3.9 berikut merupakan activity diagram ketika user akan melihat dan memilih produk :



Gambar 3.9 Activity Diagram User Melihat Produk

Gambar 3.10 berikut merupakan activity diagram ketika user akan melakukan pengujian keputusan kredit :



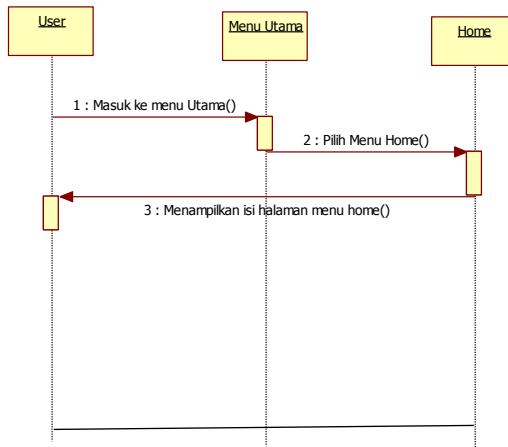
Gambar 3.10 Activity Diagram User Melakukan Uji Keputusan Kredit

3. Sequence Diagram

Pada sequence diagram akan menjelaskan interaksi antar objek dan bagaimana alur yang akan dijalankan pada aplikasi sistem tersebut. Adapun sequence diagram sebagai berikut :

a. Sequence diagram Home

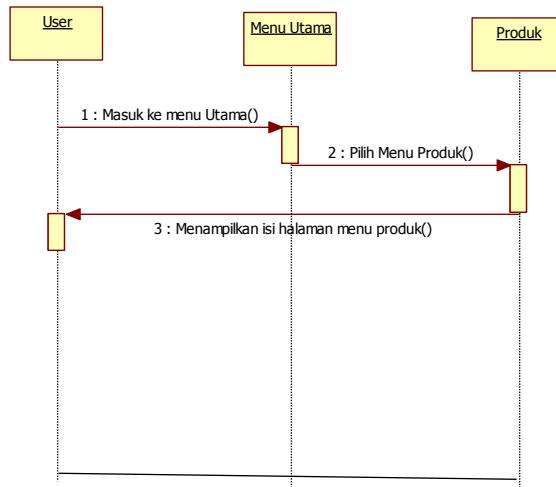
Pengguna (*User*) akan masuk pada halaman utama *website* dan akan muncul beberapa tombol menu seperti Home, Produk, *Blog*, Uji Keputusan Kredit dan login. Gambar 3.11 menjelaskan sequence diagram menu *home* :



Gambar 3.11 Sequence Diagram Home

b. Sequence Diagram Produk

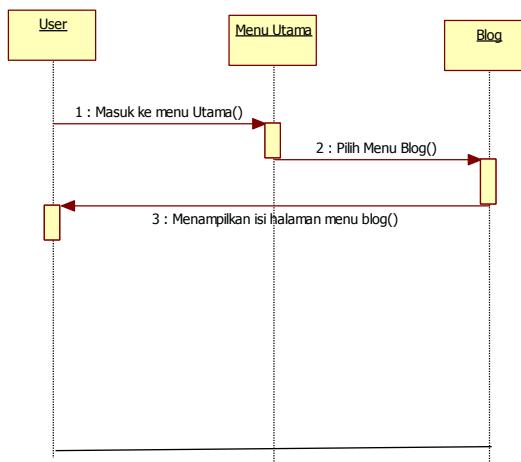
Pengguna (*User*) akan masuk pada halaman utama aplikasi dan akan muncul beberapa *menu item*, dan *user* memilih *menu produk*. Pada menu ini *user* akan mendapatkan informasi tentang produk-produk pada PT.Columbus. Gambar 3.12 menjelaskan sequence diagram *menu produk* :



Gambar 3.12 Sequence Diagram Produk

c. Sequence Diagram Blog

Pengguna (*User*) akan masuk pada halaman utama aplikasi dan akan muncul beberapa *menu item*, dan *user* memilih *menu blog*. Pada menu ini memuat tentang informasi tentang PT.Columbus. Gambar 3.13 menjelaskan *sequence diagram menu blog*.

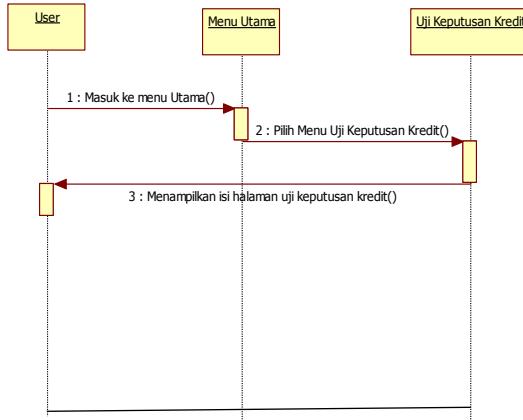


Gambar 3.13 Sequence Diagram Blog

d. Sequence Diagram Uji Keputusan Kredit

Pengguna (*User*) akan masuk pada halaman utama aplikasi dan akan muncul beberapa *menu item*, dan *user* memilih *menu uji keputusan kredit*. Pada menu

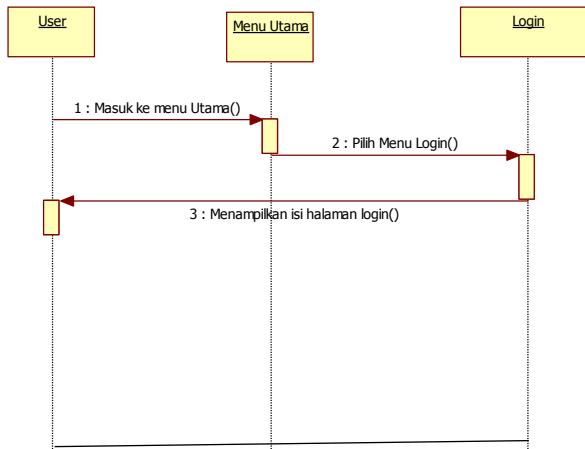
ini memuat tentang *form uji keputusan kredit*. Gambar 3.14 menjelaskan *sequence diagram* menu uji keputusan kredit.



Gambar 3.14 Sequence Diagram Uji Keputusan Kredit

e. Sequence Diagram Login

Admin akan masuk pada halaman utama aplikasi dan akan muncul beberapa menu item, dan *user* memilih *menu login*. Pada menu ini *admin* yang akan melakukan *manage* data produk dan *blog* harus melakukan *login* terlebih dahulu. Gambar 3.15 menjelaskan *sequence diagram* menu login :



Gambar 3.15 Sequence Diagram Login

4. Struktur Database

Struktur database dari sistem pakar dalam mendiagnosa kelainan *conduct disorder* pada anak adalah sebagai berikut :

- a. Nama database : db_colombus
- Nama tabel : tbl_admin
- Fungsi : menyimpan dan mengelola data admin
- Primary Key : id_admin

Struktur *database* tabel *admin* dapat dilihat pada tabel 3.14 :

Tabel 3.14 Tabel Admin

No	Field	Type	Length	Constraint
1.	Id_admin	Int	11	<i>Primary key</i>
2.	Username	Varchar	40	
3.	Password	Varchar	40	

- b. Nama database : db_colombus
- Nama tabel : tbl_atribut
- Fungsi : menyimpan dan mengelola data admin
- Primary Key : id_atribut

Struktur *database* tabel *atribut* dapat dilihat pada tabel 3.15 :

Tabel 3.15 Tabel Atribut

No	Field	Type	Length	Constraint
1.	Id_atribut	Int	11	<i>Primary key</i>
2.	nm_atribut	Varchar	50	
3.	bobot	Float		

- c. Nama database : db_colombus
 Nama tabel : tbl_blog
 Fungsi : menyimpan dan mengelola data admin
 Primary Key : id_blog

Struktur *database* tabel *blog* dapat dilihat pada tabel 3.16 :

Tabel 3.16 Tabel Blog

No	Field	Type	Length	Constraint
1.	Id_blog	Int	11	<i>Primary key</i>
2.	judul_blog	Varchar	100	
3.	deskripsi	Text		
4.	gambar	Varchar	100	
5.	tgl_post	Datetime		

- d. Nama database : db_colombus
 Nama tabel : tbl_gambar_produk
 Fungsi : menyimpan dan mengelola data admin
 Primary Key : id_gambar_produk

Struktur *database* tabel produk dapat dilihat pada tabel 3.17.

Tabel 3.17 Tabel Produk

No	Field	Type	Length	Constraint
1.	id_gambar_produk	Int	11	<i>Primary key</i>
2.	id_produk	Int	11	
3.	gambar	Text		

- e. Nama database : db_colombus
 Nama tabel : tbl_hasil
 Fungsi : menyimpan dan mengelola data admin
 Primary Key : id_gambar_produk

Struktur *database* tabel *tabel hasil* dapat dilihat pada tabel 3.18 :

Tabel 3.18 Tabel Hasil

No	Field	Type	Length	Constraint
1.	id_hasil	Int	11	<i>Primary key</i>
2.	id_kasus	Int	11	
3.	id_training	Int	11	
4.	Hasil	Float		

- f. Nama database : db_colombus
 Nama tabel : tbl_kasus
 Fungsi : menyimpan dan mengelola data admin
 Primary Key : id_kasus

Struktur *database* tabel kasus dapat dilihat pada tabel 3.19.

Tabel 3.19 Tabel Kasus

No	Field	Type	Length	Constraint
1.	id_kasus	Int	11	<i>Primary key</i>
2.	Nik	Char	16	
3.	nama	Varchar	50	
4.	Penghasilan	Int	11	
5.	Jt	Int	11	
6.	Jp	Int	11	
7.	pend	Int	11	
8.	Lk	Int	11	
9.	Berkas	Int	11	

- g. Nama database : db_colombus
 Nama tabel : tbl_kategori
 Fungsi : menyimpan dan mengelola data kategori produk
 Primary Key : id_kategori

Struktur *database* tabel kategori dapat dilihat pada tabel 3.20 :

Tabel 3.20 Tabel Kategori

No	Field	Type	Length	Constraint
1.	Id_produk	Int	11	<i>Primary key</i>
2.	Nm_kategori	Varchar	50	

- h. Nama database : db_colombus
 Nama tabel : tbl_kedekatan
 Fungsi : menyimpan dan mengelola data kedekatan
 Primary key : id_kedekatan

Struktur *database* tabel kedekatan dapat dilihat pada tabel 3.21 :

Tabel 3.21 Tabel kedekatan

No	Field	Type	Length	Constraint
1.	Id_kedekatan	Int	11	<i>Primary key</i>
2.	Id_subatribut1	Int	11	
3	Id_subatribut2	Int	11	Foreign Key
4	xy	Float		

- i. Nama database : db_colombus
 Nama tabel : tbl_produk
 Fungsi : menyimpan dan mengelola data produk
 Primary Key : id_produk

Struktur *database* tabel produk dapat dilihat pada tabel 3.22 :

Tabel 3.22 Tabel produk

No	Field	Type	Length	Constraint
1.	Id_produk	Int	11	<i>Primary key</i>
2.	Nama_produk	Varchar	50	
3	Id_kategori	Int	11	Foreign Key
4	harga	Int	11	
5	deskripsi	text		
6	Rating_produk	varchar	25	

- j. Nama database : db_colombus
 Nama tabel : tbl_subatribut
 Fungsi : menyimpan dan mengelola data subatribut
 Primary Key : id_subatribut

Struktur *database* tabel sub *atribut* dapat dilihat pada tabel 3.23 ;

Tabel 3.23 Tabel sub atribut

No	Field	Type	Length	Constraint
1.	Id_subatribut	Int	11	<i>Primary key</i>
2.	Id_atribut	Int	11	
3	Nm_subatribut	Varchar	50	
4	Bobot	Float		

- k. Nama database : db_colombus
 Nama tabel : tbl_training
 Fungsi : menyimpan dan mengelola data training
 Primary Key : id_training

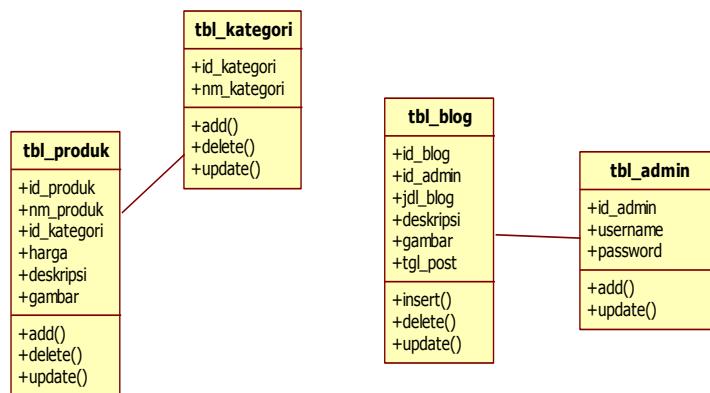
Struktur *database* tabel data *training* dapat dilihat pada tabel 3.24 :

Tabel 3.24 Tabel Data Training

No	Field	Type	Length	Constraint
1.	Id_training	Int	11	<i>Primary key</i>
2.	nama	varchar	50	
3	Penghasilan	Int	11	
4	Jt	Int	11	
5	jp	int	11	
6	Pend	Int	11	
7	Lb	Int	11	
8	Berkas	Int	11	
9	Status	Char	20	

5. Class Diagram

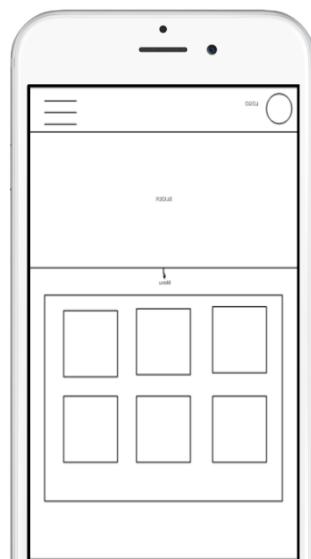
Class diagram menggambarkan struktur statis dari kelas dalam sistem dan menggambarkan atribut, operasi dan hubungan antar kelas. *Class diagram* dapat dilihat pada gambar 3.15 di bawah ini :



Gambar 3.15 Class Diagram

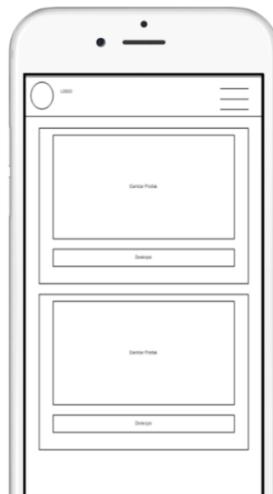
6. Rancangan *Input/Output*

Proses perancangan ini pengembang dapat membagi kebutuhan-kebutuhan menjadi perangkat lunak. Proses ini menghasilkan sebuah arsitektur perangkat lunak sehingga dapat diterjemahkan kedalam kode-kode program. Perancangan antar muka dari aplikasi penentuan kredit ditunjukan pada dibawah ini :



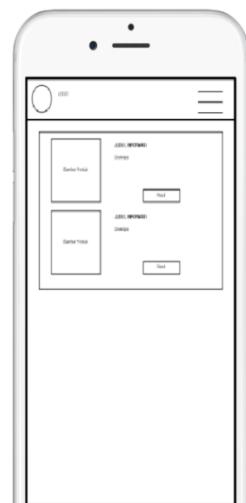
Gambar 3.16 Rancangan *Interface Program* Tampilan Utama

Pada rancangan *interface* program tampilan utama, halaman pertama pada saat user mengakses program. Pada halaman ini terdapat logo dari aplikasi yaitu : Colombus dan deretan *menu Button* lainnya. Tombol *menu* ini memiliki fungsi masing-masing yang akan menampilkan informasi yang berbeda-beda. Perancangan antar muka halaman produk ditunjukan pada gambar 3.16 diatas.



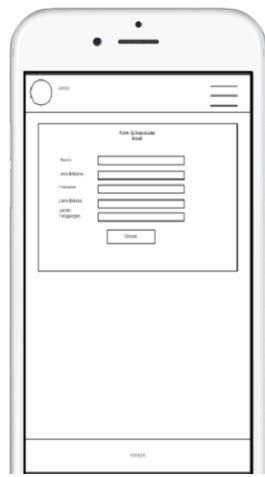
Gambar 3.17 Rancangan *Interface* Halaman Produk

Pada rancangan *interface* program tampilan produk, halaman ini berisi tentang produk-produk yang tersedia pada PT.Columbus. Perancangan antar muka tampilan produk ditunjukan pada gambar 3.17 diatas.



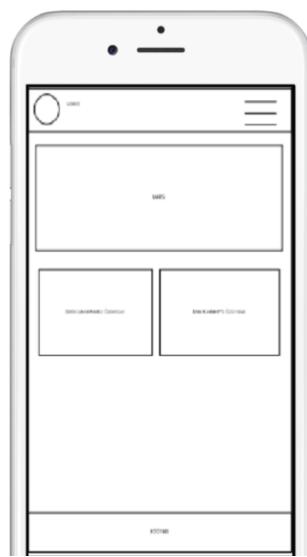
Gambar 3.18 Rancangan *Interface Program* Tampilan *blog*

Pada rancangan *interface* program tampilan *blog*, halaman ini berisi tentang informasi-informasi terkait PT.Columbus yang *diinput* oleh *admin* pada gambar 3.18 diatas.



Gambar 3.19 Rancangan Interface Program Tampilan Uji Keputusan Kredit

Pada rancangan *interface* program tampilan tipe uji keputusan kredit, halaman ini berisi *form* yang harus diisi sebelum memilih barang yang ingin dilakukan untuk mengetahui apakah kredit tersebut dapat diterima dapat dilihat pada gambar 3.19 diatas.



Gambar 3.20 Rancangan Interface Program Tampilan Contact

Pada rancangan *interface* program tampilan *contact*, halaman ini berisi *menu* informasi kontak dan lokasi PT.Columbus. Tombol menu *contact* akan

memberikan informasi nomor yang dapat dihubungi terkait produk-produk yang ada di PT.Columbus. gambar 3.20 diatas.

3.2.4 Construction of Prototype

Tahapan *construction of prototype* pada penelitian ini yaitu pembuatan *script coding*. Mulai dari kerangka aplikasi sampai dengan perhitungan menggunakan metode K-NN. Hal ini berkaitan untuk penghitung kemungkinan-kemungkinan dari kepuutusan pengejuan kredit diterima atau ditolak.

3.2.5 Deployment, Delivery & Feedback

Tahapan *deployment & delivery feedback* dilakukan setelah semua tahapan dari *communication, quick plan, modelling quick design, dan construction of prototype* yang sudah sesuai dengan keinginan PT.Columbus. Pada tahap ini, sistem identifikasi akan ditest semua fungsi-fungsi tombol dan perhitungan K-NN.

BAB IV

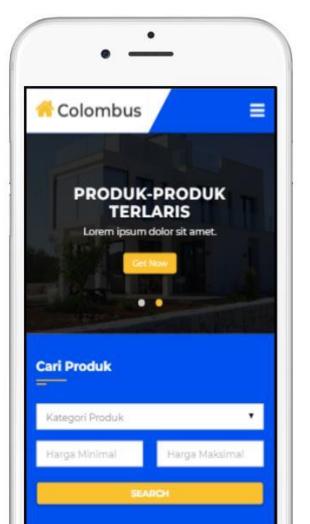
HASIL PENELITIAN DAN PEMBAHASAN

4.1 Hasil Rancangan Program

Hasil rancangan program merupakan tahap mewujudkan perancangan menjadi sebuah aplikasi. Berikut ini akan dijelaskan mengenai hasil program Aplikasi Penentuan Kredit Berjangka *Online* Menggunakan Algoritma *K-Nearst Neighbor*.

4.1.1 Tampilan Halaman Awal

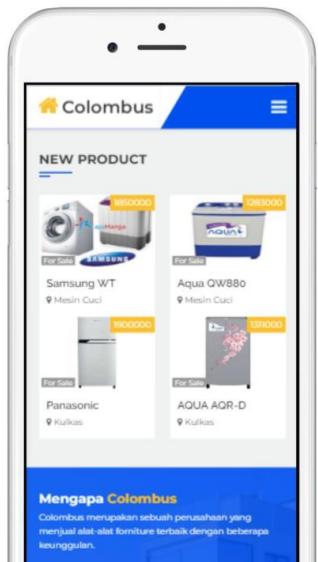
Berikut ini merupakan tampilan halaman awal aplikasi penentuan kredit berjangka pada PT.Columbus. Tampilan halaman *login* dapat dilihat pada gambar 4.21 :



Gambar 4.21 Halaman Awal Aplikasi

4.1.2 Tampilan Halaman Produk

Berikut ini merupakan tampilan halaman produk untuk *user*. Dapat dilihat pada gambar 4.22 :



Gambar 4.22 Halaman Beranda

4.1.3 Tampilan Halaman *Blog*

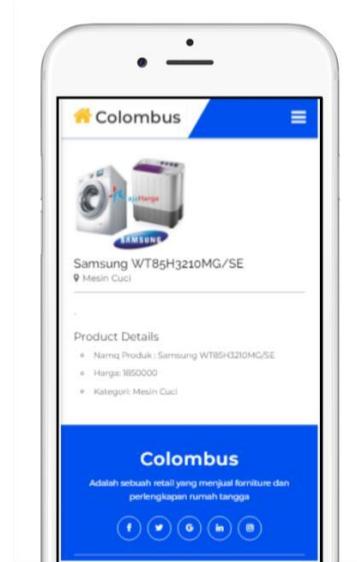
Berikut ini merupakan tampilan halaman mengenai *blog* informasi. Dapat dilihat pada gambar 4.23 :



Gambar 4.23 Halaman Data *Blog*

4.1.4 Tampilan Halaman *Detail Produk*

Dibawah ini merupakan tampilan dari halaman *detail* produk. Dapat dilihat pada gambar 4.24 :



Gambar 4.24 Halaman *Detail Produk*

4.1.5 Tampilan Halaman *Detail Blog*

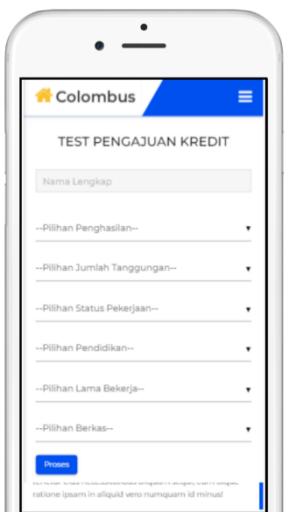
Berikut ini merupakan tampilan dari halaman *detail blog*. Dapat dilihat pada gambar 4.25:



Gambar 4.25 Halaman *Detail Blog*

4.1.6 Tampilan Halaman *Test Pengajuan Kredit*

Berikut ini merupakan tampilan dari halaman *test pengajuan kredit*. Dapat dilihat pada gambar 4.26 :



Gambar 4.26 Halaman *Test Pengajuan Kredit*

4.1.7 Tampilan Halaman Pengisian *Test Pengajuan Kredit*

Dibawah ini merupakan tampilan dari pengisian *test* pengajuan kredit. Dapat dilihat pada gamabar 4.27 :

Gambar 4.27 Halaman Pengisian *Test Pengajuan Kredit*

4.1.8 Tampilan Halaman Hasil Keputusan *Test Pengajuan Kredit*

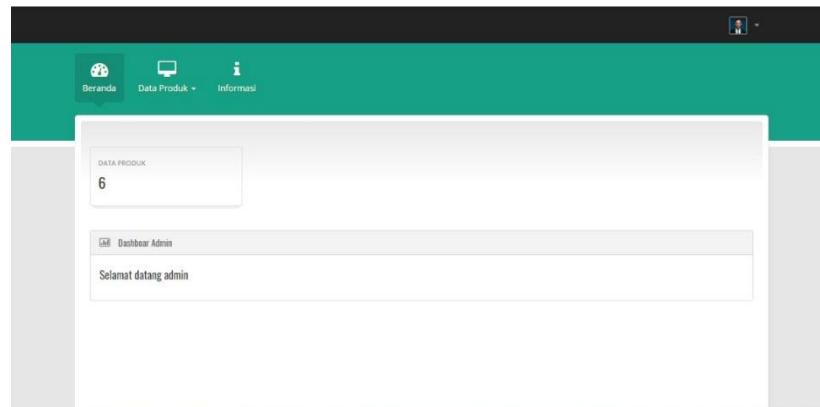
Dibawah ini merupakan tampilan dari hasil keputusan *test* pengajuan kredit. Dapat dilihat pada gamabar 4.28 :



Gambar 4.28 Halaman Hasil Keputusan *Test Pengajuan Kredit*

4.1.9 Tampilan Halaman *Beranda Admin*

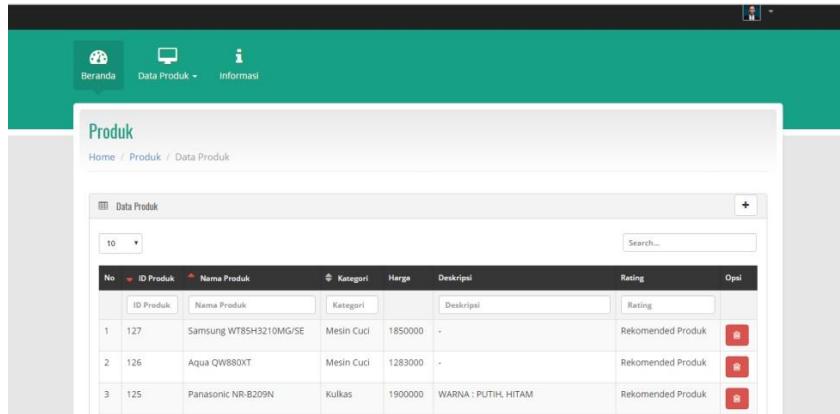
Dibawah ini merupakan tampilan *menu* untuk *dashboard admin*. Dapat dilihat pada gambar 4.29 :



Gambar 4.29 Halaman *Beranda Admin*

4.1.10 Tampilan Halaman *Manage Produk*

Dibawah ini merupakan tampilan halaman *manage* produk. Dapat dilihat pada gambar 4.30 :

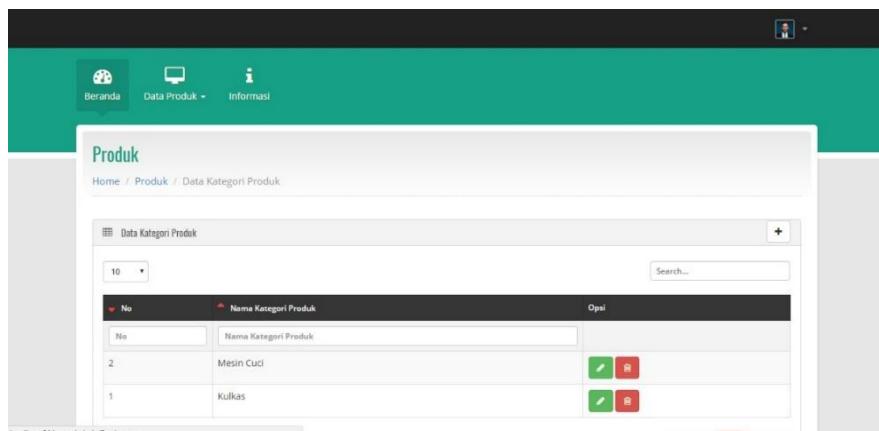


No	ID Produk	Nama Produk	Kategori	Harga	Deskripsi	Rating	Opsi
1	127	Samsung W785H3210MG/SE	Mesin Cuci	1850000	-	Rekomended Produk	
2	126	Aqua QW880XT	Mesin Cuci	1283000	-	Rekomended Produk	
3	125	Panasonic NR-B209N	Kulkas	1900000	WARNA : PUTIH, HITAM	Rekomended Produk	

Gambar 4.30 Halaman *Manage Produk*

4.1.11 Tampilan Halaman *Manage Kategori*

Dibawah ini merupakan tampilan halaman *manage* kategori produk pada admin. Dapat dilihat pada gambar 4.31 :



No	Nama Kategori Produk	Opsi
2	Mesin Cuci	
1	Kulkas	

Gambar 4.31 Halaman *Manage Kategori*

4.2 Pengujian

Pengujian perangkat lunak berfungsi untuk menentukan pencapaian tujuan penetuan keputusan kredit diterima atau tidak. Dalam hal ini pengujian dilakukan dengan menghubungkan setiap entitas dari sistem sesuai dengan spesifikasi *hardware* dan *software*. Perngujian kinerja aplikasi

hanya dilakukan terhadap aplikasi yang berkaitan dengan pemerosesan *database* dalam program *Notepad++* dan *database mysql*. Adapun pengujian dilakukan dengan variasi banyaknya database yang harus di eksekusi. Selain itu kinerja dari perangkat lunak yang sangat terkait dengan kondisi konektivitas *database* yang digunakan. Setelah *Xampp* terinstal untuk sementara *server* yang digunakan adalah *localhost*. Pengaturan *database* dapat dilakukan melalui *phpmyadmin* yang berfungsi untuk membuat, merubah, dan menghapus *database*. Dengan fasilitas ini akan memudahkan dalam pembuatan *database mysql* karena tidak menggunakan perintah (*syntax*) manual *sql*.

4.3 Pembahasan

Perangkat lunak yang *diimplementasikan* telah sesuai dengan *spesifikasi* kebutuhan pada bagian *analisis* dan perancangan. Hal ini dibuktikan dengan keberhasilan masing-masing *subsistem* melakukan apa yang menjadi *spesifikasi* seperti telah ditanyakan dibagian hasil pengujian diatas, sehingga proses yang terjadi telah dengan prosedur yang *dispesifikasikan*.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini menghasilkan suatu aplikasi promosi produk dan penentuan pengajuan kredit berjangka *online*. Sehingga dapat membantu masyarakat untuk melihat informasi produk dan mengetahui pengajuan kredit diterima atau tidak dengan cepat. Dengan pemanfaatan aplikasi ini, masyarakat dengan mudah mendapat informasi kredit dengan cepat dan singkat.

5.1 Saran

Sesuai dengan permasalahan yang ada pada skripsi ini, maka saran yang dapat diberikan adalah sebagai berikut : Sistem ini menerapakan 1 algoritma dalam menyelesaikan masalah kredit dan diharapkan dapat dikembangkan dengan beberapa algoritma agar hasil yang diperoleh menjadi lebih sempurna. Sistem Aplikasi penentuan kredit ini bisa berubah sesuai dengan aturan kredit yang baru, sehingga perlu adanya pengembangan dengan membuat atribut/kriteria menjadi dimanis.

DAFTAR PUSTAKA

- A..S Rosa, Salahudin, M. (2011), Modul Pembelajaran Rekayasa Perangkat Lunak (terstruktur dan Berorientasi objek). Bandung : Modula.
- Anhar, (2010). Panduan Menguasai *PHP & MySQL* Secara otodidak, Mediakita Jakarta.
- Astuti Puji. Komprasi Penerapan Algoritma C45, K-NN dan *Neural Network* Dalam Proses Kelayakan Penerimaan Kredit Kendaraan Bermotor , Vol 9, No. 1 2016, Halaman 87-101.
- Kadir Abdul. 2008. Tuntunan Praktis Belajar *Database* Menggunakan *MySQL*, C.V Andi Offset. Yogyakarta.
- Kustianingsih, Yeni, R.A. Devie. (2011), Pemrograman Basis Data Berbasis Web Menggunakan *PHP & MySQL*. Yogyakarta: Graha Ilmu.
- Ledyana Henny. Penerapan Algoritma *K-Nearest Neighbor* untuk Penentuan Resiko Kredit Kepemilikan Kendaraan Bermotor. Jurnal Penelitian Ilmu Komputer, *System Embedded & Logic*, Vol 1, N0 1, : Halaman 65-76 2013.
- Moch Abdul Mukid, Rusgiyono Agus, dan Fatimah. Analisa *Credit Scoring* Menggunakan Metode *Bagging K-Nearest Neighbor*. Jurnal Gaussian, Vol 6, No 1, 2017, Halaman 161-170.
- R.S. Pressman (2010), *Software Engineering : a approach*, McGraw-Hill, New York.
- Safaat Nazaruddin H. 2012 (Edisi Revisi). Pemrograman Aplikasi *Mobile* Smartphone dan Tablet PC Berbasis *Android*, Informatika, Bandung.



**SURAT KEPUTUSAN
REKTOR IIB DARMAJAYA
NOMOR : SK.180/DMJ/DFIK/BAAK/III-18**
Tentang
Dosen Pembimbing Skripsi
Program Studi S1 Teknik Informatika

REKTOR IIB DARMAJAYA

Memperhatikan : 1. Bahwa dalam rangka usaha peningkatan mutu dan peranan IIB Darmajaya dalam melaksanakan Pendidikan Nasional perlu ditingkatkan kemampuan mahasiswa dalam Skripsi.

Menimbang : 2. Laporan dan usulan Ketua Program Studi **S1 Teknik Informatika**.
1. Bahwa untuk mengefektifkan tenaga pengajar dalam Skripsi mahasiswa perlu ditetapkan **Dosen Pembimbing Skripsi**.
2. Bahwa untuk maksud tersebut dipandang perlu menerbitkan Surat Keputusan Rektor.

Mengingat : 1. UU No.20 Tahun 2003 Tentang Sistem Pendidikan Nasional.
2. Peraturan Pemerintah No.60 Tahun 2010 tentang Pendidikan Sekolah Tinggi
3. Surat Keputusan Menteri Pendidikan Nasional Republik Indonesia No.165/D/0/2008 tertanggal 20 Agustus 2008 tentang Perubahan Status STMIK-STIE Darmajaya menjadi Informatics and Business Institute (IBI) Darmajaya
4. STATUTA IBI Darmajaya
5. Surat Ketua Yayasan Pendidikan Alfian Husin No. IM.003/YP-AH/X-08 tentang Persetujuan Perubahan Struktur Organisasi
6. Surat Keputusan Rektor 0383/DMJ/REK/X-08 tentang Struktur Organisasi.

Menetapkan

Pertama : Mengangkat nama-nama seperti tersebut dalam lampiran Surat Keputusan ini sebagai Dosen Pembimbing Skripsi mahasiswa Program Studi S1 Teknik Informatika.

Kedua : Pembimbing Skripsi berkewajiban melaksanakan tugasnya sesuai dengan jadwal yang telah ditetapkan.

Ketiga : Pembimbing Skripsi yang ditunjuk akan diberikan honorarium yang besarnya sesuai dengan ketentuan peraturan dan norma penggajian dan honorarium IBI Darmajaya.

Keempat : Surat Keputusan ini berlaku sejak tanggal ditetapkan dan apabila dikemudian hari terdapat kekeliruan dalam keputusan ini, maka keputusan ini akan ditinjau kembali.

Ditetapkan di : Bandar Lampung
Pada tanggal : 12 Maret 2018
oleh Rektor IIB Darmajaya,
Dekan Fakultas Ilmu Komputer


Sriyanto, S.Kom., M.M.
NIK. 00210801

1. Kabiro. SDM
2. Ketua Jurusan S1 Teknik Informatika
3. Yang bersangkutan
4. Arsip

Lampiran : Surat Keputusan Rektor IIB Darmajaya
 Nomor : SK. 180/DMI/DFIK/BAAK/III-18
 Tanggal : 12 Maret 2018
 Perihal : Pembimbing Penulisan Skripsi
 Program Studi Strata Satu (S1) Teknik Informatika

JUDUL SKRIPSI DAN DOSEN PEMBIMBING

PROGRAM STUDI STRATA SATU (S1) TEKNIK INFORMATIKA

No	NAMA	NPM	JUDUL	PEMBIMBING
15	1411010003	Kevin Moniaga	Penerapan Algoritma Naïve Bayes Classifier Untuk Mengetahui Minat Beli Petangan Terhadap Sofa (Studi Kasus di Mebel Kelumer Bayau)	Ketut Artaye, S.Kom, M.T.I
16	1411010001	Frank Wijaya	Implementasi Algoritma Naïve Bayes Dalam Klasifikasi Produk Ban Terlaris Pada Mitra Mekar Mandiri	
17	Thiu Aan Rian Hermawan	1411010098	Perangkat Lunak E-Learning Elemen - Elemen Seni Teater Berbasis Android	Nisar, S.Kom, M.T
18	Desma Fauzi Suseno	1211010032	Rancang Bangun Sistem Diagnosis Penyakit Pada Sapi Menggunakan Metode Case Based Reasoning	Puput Budi Wintoro, S.Kom, M.T.I
19	1411010102	Ilham Bawazier	Peningkatan Vocabulary Bahasa Inggris Untuk Siswa Sekolah Menengah Pertama Berbasis Android (Studi Kasus Sekolah Menengah Pertama Pajajaran)	
20	Ajeng Prameksa Dewi	14110110044	Perancangan Perangkat Lunak Pendekripsi Penyakit Gigi dan Mulut Menggunakan Metode Naïve Bayes Classifier Berbasis Android	Rio Kurniawan, M.Cs
21	Muji Rahayu	1411010032	Perangkat Lunak Sosial Media Sebagai Sarana Komunikasi Jurusan Web Mobile (Studi Kasus Jurusan Teknik Informatika)	
22	Edi Susanto	1211010213	Aplikasi Panduan Pengolahan Rotan Sebagai Bahan Untuk Mebelier Berbasis Mobile	Rionaldi Ali, S.Kom, M.T.I
23	Mega Lestari	1411010008	Aplikasi Penentuan Kredit Berjangka On Line Menggunakan Algoritma K-Nearest Neighbor pada PT. Colombus Tanjung Karang Pusat Bandar Lampung	
24	1411010043	Septiani Dwi Lestari	Membangun Aplikasi Untuk Menentukan Ayan Segar atau Siren Berdasarkan Template dan Segmentasi Warna	
25	I Wayan Merta Anggara	1411010057	Rancang Bangun Notifikasi Imunitasi Bagi Balita Berbasis Android	
26	Bayu Pandu Putra Pratama	1411010068	Implementasi Location Based Service Berbasis Android untuk Pencarian Akomodasi Pariwisata Di Pesisir Barat	Septilia Afida, S.Kom, M.T.I
27	Rudi Yulianto	1111010196	Rancang Bangun Aplikasi Media Pembelajaran Menu Sehat Untuk Balita Berbasis Multimedia	
28	Ashbilly Satria Guitom	1311010084	Rancang Bangun Aplikasi Visualisasi 3D Bendera Negara Asia Tenggara Untuk Anak SD Berbasis Augmented Reality	Tri Wahyuni, S.Kom, M.T.I
29	Bagus Gibran Agung Prayuky	1311010091	Rancang Bangun Aplikasi Syarat Pembuatan Kartu Keluarga Pada Kantor Catanan Sipil Tanggamus Berbasis Android	



Institut Informatika & Bisnis

DARMAJAYA

Yayasan Alfian Husin

Jl. Zainal Abidin Pagar Alam No. 93 Bandar Lampung 35142 Telp 787214 Fax. 700261 http://darmajaya.ac.id

FORMULIR

BIRO ADMINISTRASI AKADEMIK KEMAHASISWAAN (BAAK)

FORM KONSULTASI/BIMBINGAN SKRIPSI/TUGAS AKHIR *

NAMA : MEGA LESTARI
N P M : 1411010008
PEMBIMBING I : Rio Kurniawan MCS
PEMBIMBING II :
JUDUL LAPORAN : Aplikasi Perencanaan Kredit Berjangka Online Menggunakan Algoritma k-nearest neighbor Pada PT Colombrus Tanjung Karang Pusat Bandar Lampung s.d (6+2 bulan)
TANGGAL SK :

No	HARI/TANGGAL	HASIL KONSULTASI	PARAF
1	27 maret 2018	ACC Judul	f.
2	2 April 2018	Acc Seminar	f.
3	2 Juli 2018	Revisi Bab 1, Latar belakang kurang lengkap Permasalahan Bab 2, penulisan nama Pengarang bvtv, Judul buku, tahun dan halaman. Spasi kebanyaknya	f.
4	10 Agustus 2018	Revisi Bab 3, Perhitungan kurang detail, kata kedekatan diganti dengan jarak.	f.
5	20 Agustus 2018	Revisi Bab 5 Daftar pustaka harus di lembar beritutnya	f.
6	28 Agustus 2018	Revisi Bab 1 Spasi, Bab 2 huruf cekak miring, Bab 3 penomoran tabel, Bab 4 namagambar	f.
7	4 Sep 2018	Revisi Deskripsi tabel dan gambar, Penulisan Daftar Pustaka dicatatkan.	f.
8	5. Sep 2018	Acc urw surpir	f.
9			
10			

*) Coret yang tidak perlu



b - 9 - 2018

```
<?php
include"koneksi.php";
$id = $_GET['id'];

$delete = mysqli_query($connect,"delete from tbl_atribut where
id_atribut='$id'");

if($delete){
header("location:../?p=data_atribut&pesan=sukses_delete");
}

?>

<?php
include"koneksi.php";
$id = $_GET['id'];

$delete = mysqli_query($connect,"delete from tbl_kedekatan where
id_kedekatan='$id'");

if($delete){
header("location:../?p=data_kedekatan&pesan=sukses_delete");
}

?>

<div class="content-container">

<div class="content-header">
<h2 class="content-header-title">Atribut</h2>
<ol class="breadcrumb">
<li><a href="?p=home">Home</a></li>
<li><a href="">Atribut</a></li>
<li class="active">Data Atribut</li>
```

```

</ol>
</div> <!-- /.content-header -->

<div class="row">
<div class="col-md-12">
<div class="portlet">
<div class="portlet-header">
<h3>
<i class="fa fa-table"></i>
Data Atribut
</h3>
<a href="?p=form_atribut"><button class="btn btn-default"
style="float:right"><i class="fa fa-plus"></i></button></a>
</div> <!-- /.portlet-header -->

<div class="portlet-content">
<?php
if(!empty($_GET['pesan']) && $_GET['pesan']=='sukses_simpan'){
echo '<div class="alert alert-success alert-dismissible">
<button type="button" class="close" data-dismiss="alert" aria-
hidden="true">&times;</button> Data Berhasil Disimpan.
</div>';
}else if(!empty($_GET['pesan']) &&
$_GET['pesan']=='sukses_delete'){
echo '<div class="alert alert-success alert-dismissible">

```

```

        <button type="button" class="close" data-dismiss="alert" aria-
hidden="true">&times;</button> Data Berhasil Dihapus.

    </div>';

} else if(!empty($_GET['pesan']) &&
$_GET['pesan']=='sukses_update'){

    echo '<div class="alert alert-success alert-dismissible">

        <button type="button" class="close" data-dismiss="alert" aria-
hidden="true">&times;</button> Data Berhasil Diupdate.

    </div>';

}

?>

<div class="table-responsive">

    <table
        class="table table-striped table-bordered table-hover table-highlight
table-checkable"
        data-provide="datatable"
        data-display-rows="10"
        data-info="true"
        data-search="true"
        data-length-change="true"
        data-paginate="true"
    >

        <thead>
            <tr>
                <th data-filterable="true" data-sortable="true" data-
direction="desc">No</th>
                <th data-direction="asc" data-filterable="true" data-
sortable="true">Nama Atribut</th>
                <th data-direction="asc" data-filterable="true" data-
sortable="true">Bobot</th>
                <th>Opsi</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>1</td>
                <td>Nama</td>
                <td>0.5</td>
                <td><button type="button" class="close" data-
dismiss="alert" aria-hidden="true">&times;</button> <button type="button" class="btn btn-primary" data-
button="button">Edit</button> <button type="button" class="btn btn-danger" data-
button="button">Hapus</button></td>
            </tr>
        </tbody>
    </table>
</div>

```

```

        </tr>
    </thead>
    <tbody>
        <?php
            include"aksi/koneksi.php";
            $no =1;
            $sql = mysqli_query($connect,"select * from tbl_atribut order by id_atribut ASC");
            while($dt = mysqli_fetch_assoc($sql)){
                ?>
                <tr>
                    <td><?php echo $no;?></td>
                    <td><?php echo $dt['nm_atribut'];?></td>
                    <td><?php echo $dt['bobot'];?></td>
                    <td>
                        <a href=""><button class="btn btn-success"><i class="fa fa-pencil"></i></button></a>
                        <a href="aksi/delete_atribut.php?id=<?php echo $dt['id_atribut'];?>"><button class="btn btn-danger"><i class="fa fa-trash-o"></i></button></a>
                    </td>
                </tr>
            <?php
                $no++;
            } ?>
        </tbody>
    </table>
</div> <!-- /.table-responsive -->

</div> <!-- /.portlet-content -->

```

```
</div> <!-- /.portlet -->

</div> <!-- /.col -->

</div> <!-- /.row -->

</div> <!-- /.content-container -->

<div class="content-container">

    <div class="content-header">
        <h2 class="content-header-title">Form Report</h2>
        <ol class="breadcrumb">
            <li><a href="?p=home">Home</a></li>
            <li class="active">Form Report Mingguan/Bulanan</li>
        </ol>
    </div> <!-- /.content-header -->

    <div class="portlet">

        <div class="portlet-header">

            <h3>
                <i class="fa fa-tasks"></i>
                Form Report Mingguan/Bulanan
            </h3>
        </div>
    </div>
</div>
```

```
</div> <!-- /.portlet-header -->

<div class="portlet-content">

<div class="row">
    <div class="col-sm-6 col-sm-offset-3">
        <form method="post" action="laporan/lihat_all.php?p=bulan">
            <div class="col-md-6">
                <div class="form-group">
                    <label for="select-input">Awal</label>
                    <input type="date" class="form-control" name="tgl">
                </div>
            </div>
            <div class="col-md-6">
                <div class="form-group">
                    <label for="select-input">Akhir</label>
                    <input type="date" class="form-control" name="tgl1">
                </div>
            </div>
            <div class="col-md-12">
                <div class="form-group">
                    <button type="submit" class="btn btn-success" ><i class ="fa fa-save"></i> Cetak</button>
                </div>
            </div>
        </form>
    </div> <!-- /.col -->
</div> <!-- /.row -->
</div> <!-- /.portlet-content -->
```

```
</div> <!-- /.portlet -->
</div> <!-- /.content-container -->

<div class="content-container">

<div class="content-header">
    <h2 class="content-header-title">Produk</h2>
    <ol class="breadcrumb">
        <li><a href="?p=home">Home</a></li>
        <li><a href="">Produk</a></li>
        <li class="active">Data Produk</li>
    </ol>
</div> <!-- /.content-header -->

<div class="row">

<div class="col-md-12">

<div class="portlet">

<div class="portlet-header">

    <h3>
        <i class="fa fa-table"></i>
        Data Produk
    </h3>
    <a href="?p=form_produk"><button class="btn btn-default"
style="float:right"><i class="fa fa-plus"></i></button></a>
</div> <!-- /.portlet-header -->
```

```

<div class="portlet-content">
<?php
if(!empty($_GET['pesan']) && $_GET['pesan']=='sukses_simpan'){
echo '<div class="alert alert-success alert-dismissible">
<button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button> Data Berhasil Disimpan.
</div>';
}
else if(!empty($_GET['pesan']) && $_GET['pesan']=='sukses_delete'){
echo '<div class="alert alert-success alert-dismissible">
<button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button> Data Berhasil Dihapus.
</div>';
}
else if(!empty($_GET['pesan']) && $_GET['pesan']=='sukses_update'){
echo '<div class="alert alert-success alert-dismissible">
<button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button> Data Berhasil Diupdate.
</div>';
}
?>

<div class="table-responsive">

<table
class="table table-striped table-bordered table-hover table-highlight
table-checkable"
data-provide="datatable"
data-display-rows="10"
data-info="true"
data-search="true"
data-length-change="true"

```

```

data-paginate="true"
>
<thead>
<tr>

<th >No</th>
<th data-filterable="true" data-sortable="true" data-direction="desc">ID Produk</th>
<th data-direction="asc" data-filterable="true" data-sortable="true">Nama Produk</th>
<th data-filterable="true" data-sortable="true">Kategori</th>
<th data-filterable="false" class="hidden-xs hidden-sm">Harga</th>
<th data-filterable="true" class="hidden-xs hidden-sm">Deskripsi</th>
<th data-filterable="true" class="hidden-xs hidden-sm">Rating</th>
<th>Opsi</th>
</tr>
</thead>
<tbody>
<?php
    include"aksi/koneksi.php";
    $no = 1;
    $sql = mysqli_query($connect,"select * from tbl_produk inner join
tbl_kategori on tbl_produk.id_kategori=tbl_kategori.id_kategori order by
id_produk DESC");
    while($dt = mysqli_fetch_assoc($sql)){
?
<tr>
<td><?php echo $no;?></td>
<td><?php echo $dt['id_produk'];?> </td>
<td><?php echo $dt['nama_produk'];?></td>
<td><?php echo $dt['nama_kategori'];?></td>

```

```
<td><?php echo $dt['harga'];?></td>
<td><?php echo $dt['deskripsi'];?></td>
<td><?php echo $dt['rating_produk'];?></td>

<td>
    <a href="aksi/delete_produk.php?id=<?php echo
$dt['id_produk'];?>"<button class="btn btn-danger"><i class="fa fa-trash-
o"></i></button></a>
</td>
</tr>
<?php
$no++;
} ?>
</tbody>
</table>
</div> <!-- /.table-responsive -->

</div> <!-- /.portlet-content -->

</div> <!-- /.portlet -->

</div> <!-- /.col -->

</div> <!-- /.row -->

</div> <!-- /.content-container -->

<div class="content-container">
```

```
<div class="content-header">
    <h2 class="content-header-title">Atribut</h2>
    <ol class="breadcrumb">
        <li><a href="?p=home">Home</a></li>
        <li><a href="">Atribut</a></li>
        <li class="active">Data Atribut</li>
    </ol>
</div> <!-- /.content-header -->

<div class="row">
    <div class="col-md-12">
        <div class="portlet">
            <div class="portlet-header">
                <h3>
                    <i class="fa fa-table"></i>
                    Data Atribut
                </h3>
                <a href="?p=form_subatribut"><button class="btn btn-default"
style="float:right"><i class="fa fa-plus"></i></button></a>
            </div> <!-- /.portlet-header -->
            <div class="portlet-content">
                <?php
                    if(!empty($_GET['pesan']) && $_GET['pesan']=='sukses_simpan'){

```

```

echo '<div class="alert alert-success alert-dismissible">
    <button type="button" class="close" data-dismiss="alert" aria-
hidden="true">&times;</button> Data Berhasil Disimpan.

</div>';

} else if(!empty($_GET['pesan']) &&
$_GET['pesan']=='sukses_delete'){
    echo '<div class="alert alert-success alert-dismissible">
        <button type="button" class="close" data-dismiss="alert" aria-
hidden="true">&times;</button> Data Berhasil Dihapus.

</div>';

} else if(!empty($_GET['pesan']) &&
$_GET['pesan']=='sukses_update'){
    echo '<div class="alert alert-success alert-dismissible">
        <button type="button" class="close" data-dismiss="alert" aria-
hidden="true">&times;</button> Data Berhasil Diupdate.

</div>';

}
?>

<div class="table-responsive">

<table
    class="table table-striped table-bordered table-hover table-highlight
table-checkable"
    data-provide="datatable"
    data-display-rows="10"
    data-info="true"
    data-search="true"
    data-length-change="true"
    data-paginate="true"
>
    <thead>
        <tr>

```

```

<th data-filterable="true" data-sortable="true" data-
direction="desc">No</th>

<th data-direction="asc" data-filterable="true" data-
sortable="true">>Nama Atribut</th>

<th data-direction="asc" data-filterable="true" data-
sortable="true">>Nama Sub Atribut</th>

<th data-direction="asc" data-filterable="true" data-
sortable="true">Bobot</th>

<th>Opsi</th>

</tr>

</thead>

<tbody>

<?php

    include"aksi/koneksi.php";

    $no =1;

    $sql = mysqli_query($connect,"select
tbl_subatribut.* ,tbl_atribut.nm_atribut from tbl_subatribut inner join tbl_atribut
on tbl_subatribut.id_atribut=tbl_atribut.id_atribut order by id_subatribut ASC");

    while($dt = mysqli_fetch_assoc($sql)) {

?>

<tr>

<td><?php echo $no;?></td>

<td><?php echo $dt['nm_atribut'];?></td>

<td><?php echo $dt['nm_subatribut'];?></td>

<td><?php echo $dt['bobot'];?></td>

<td>

    <a href=""><button class="btn btn-success"><i class="fa fa-
pencil"></i></button></a>

    <a href="aksi/delete_subatribut.php?id=<?php echo
$dt['id_subatribut'];?>"><button class="btn btn-danger"><i class="fa fa-trash-
o"></i></button></a>

</td>

</tr>

<?php

```

```
$no++;  
} ?>  
</tbody>  
</table>  
</div> <!-- /.table-responsive -->  
  
</div> <!-- /.portlet-content -->  
  
</div> <!-- /.portlet -->  
  
</div> <!-- /.col -->  
  
</div> <!-- /.row -->  
  
</div> <!-- /.content-container -->  
  
<div class="content-container">  
  <div class="content-header">  
    <h2 class="content-header-title">Atribut</h2>  
    <ol class="breadcrumb">  
      <li><a href="?p=home">Home</a></li>  
      <li><a href="">Data Training</a></li>  
      <li class="active">Data Training</li>  
    </ol>  
  </div> <!-- /.content-header -->
```

```

<div class="row">

    <div class="col-md-12">

        <div class="portlet">

            <div class="portlet-header">

                <h3>
                    <i class="fa fa-table"></i>
                    Data Training
                </h3>

                <a href="?p=form_training"><button class="btn btn-default"
style="float:right"><i class="fa fa-plus"></i></button></a>
            </div> <!-- /.portlet-header -->

            <div class="portlet-content">
                <?php
                    if(!empty($_GET['pesan']) && $_GET['pesan']=='sukses_simpan'){
                        echo '<div class="alert alert-success alert-dismissible">
                            <button type="button" class="close" data-dismiss="alert" aria-
hidden="true">&times;</button> Data Berhasil Disimpan.
                        </div>';
                    }else if(!empty($_GET['pesan']) &&
$_GET['pesan']=='sukses_delete'){
                        echo '<div class="alert alert-success alert-dismissible">
                            <button type="button" class="close" data-dismiss="alert" aria-
hidden="true">&times;</button> Data Berhasil Dihapus.
                        </div>';
                    }else if(!empty($_GET['pesan']) &&
$_GET['pesan']=='sukses_update'){
                        echo '<div class="alert alert-success alert-dismissible">
';
                }
            </div>

```

```

        <button type="button" class="close" data-dismiss="alert" aria-
hidden="true">&times;</button> Data Berhasil Diupdate.

</div>';

}

?>

<div class="table-responsive">

<table
    class="table table-striped table-bordered table-hover table-highlight
table-checkable"
    data-provide="datatable"
    data-display-rows="10"
    data-info="true"
    data-search="true"
    data-length-change="true"
    data-paginate="true"
>

<thead>
    <tr>
        <th data-filterable="true" data-sortable="true" data-
direction="desc">No</th>
        <th data-direction="asc" data-filterable="true" data-
sortable="true">Nama</th>
        <th data-direction="asc" data-filterable="true" data-
sortable="true">Penghasilan</th>
        <th data-direction="asc" data-filterable="true" data-
sortable="true">Jumlah Tanggungan</th>
        <th data-direction="asc" data-filterable="true" data-
sortable="true">Jenis Pekerjaan</th>
        <th data-direction="asc" data-filterable="true" data-
sortable="true">Pendidikan</th>
        <th data-direction="asc" data-filterable="true" data-
sortable="true">Lama Bekerja</th>
    </tr>

```

```

<th data-direction="asc" data-filterable="true" data-
sortable="true">Berkas</th>

<th data-direction="asc" data-filterable="true" data-
sortable="true">Status</th>

<th>Opsi</th>
</tr>
</thead>
<tbody>
<?php
    include"aksi/koneksi.php";
    $no =1;
    $sql = mysqli_query($connect,"select * from tbl_training order by
id_training ASC");
    while($dt = mysqli_fetch_assoc($sql)){
?>
<tr>
<td><?php echo $no;?></td>
<td><?php echo $dt['nama'];?></td>
<td><?php echo $dt['penghasilan'];?></td>
<td><?php echo $dt['jt'];?></td>
<td><?php echo $dt['jp'];?></td>
<td><?php echo $dt['pend'];?></td>
<td><?php echo $dt['lb'];?></td>
<td><?php echo $dt['berkas'];?></td>
<td><?php echo $dt['status'];?></td>
<td>
    <a href=""><button class="btn btn-success"><i class="fa fa-
pencil"></i></button></a>
    <a href="aksi/delete_training.php?id=<?php echo
$dt['id_training'];?>"><button class="btn btn-danger"><i class="fa fa-trash-
o"></i></button></a>
</td>
</tr>

```

```
<?php  
$no++;  
} ?>  
</tbody>  
</table>  
</div> <!-- /.table-responsive -->  
  
</div> <!-- /.portlet-content -->  
  
</div> <!-- /.portlet -->  
  
</div> <!-- /.col -->  
  
</div> <!-- /.row -->  
  
</div> <!-- /.content-container -->  
  
<div class="content-container">  
  
<div class="content-header">  
    <h2 class="content-header-title">Form Produk</h2>  
    <ol class="breadcrumb">  
        <li><a href="?p=home">Home</a></li>  
        <li><a href="">Produk</a></li>  
        <li class="active">Form Produk</li>  
    </ol>  
</div> <!-- /.content-header -->
```

```
<div class="portlet">

<div class="portlet-header">

<h3>
<i class="fa fa-tasks"></i>
Form Produk
</h3>

</div> <!-- /.portlet-header -->

<div class="portlet-content">

<div class="row">
<form method="post" action="aksi/simpan_produk.php"
enctype="multipart/form-data">
<div class="col-sm-6">

<div class="form-group">
<label for="text-input">Nama Produk</label>
<input type="text" id="text-input" class="form-control"
placeholder="Nama Produk" name="nama_produk">
</div>

<div class="form-group">
<label for="textarea-input">Kategori Produk</label>
<select id="select-input" class="form-control" name="kategori">
<?php
include"aksi/koneksi.php";
```

```
$sql = mysqli_query($connect,"select * from tbl_kategori");
while($dt = mysqli_fetch_assoc($sql)){
    ?
<option value=<?php echo $dt['id_kategori'];?>><?php echo
$dt['nama_kategori'];?></option>
<?php
} ?
</select>
</div>
<div class="form-group">
    <label for="select-input">Harga</label>
    <input type="text" class="form-control" name="harga"
placeholder="Harga">
</div>
</div> <!-- /.col -->
<div class="col-sm-6">

    <div class="form-group">
        <label for="textarea-input">Rating Produk</label>
        <select id="select-input" class="form-control" name="rate">
            <option value="Rekomended Produk">Rekomended Produk</option>
            <option value="Produk Baru">Produk Baru</option>
        </select>
    </div>

    <div class="form-group">
        <label for="select-multi-input">Deskripsi Produk</label>
        <textarea class="form-control" rows="6" name="deskripsi"></textarea>
    </div>
    <div class="form-group">
        <label for="select-multi-input">Gambar Produk</label>
```

```
    <input type="file" class="form-control" accept="image/*" multiple  
name="foto[]">>  
    </div>  
    <div class="form-group">  
        <button type="submit" class="btn btn-success" style="float:right"><i  
class ="fa fa-save"></i> Simpan</button>  
    </div>  
    </div> <!-- /.col -->  
    </form>  
    </div> <!-- /.row -->  
    </div> <!-- /.portlet-content -->  
  
    </div> <!-- /.portlet -->  
    </div> <!-- /.content-container -->
```