

KONSEP MANAGEMEN MEMORI

PURBOYO ADI HARTONO NPM 195120024

Fakultas Komputer purboyoadi@umitra.ac.id

RESUME:

A. Manajemen Memori

Konsep Dasar Memori

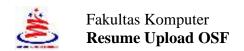
Memori adalah pusat dari operasi pada sistem komputer modern, berfungsi sebagai tempat penyimpanan informasi yang harus diatur dan dijaga sebaik-baiknya. Memori adalah array besar dari word atau byte, yang disebut alamat. CPU mengambil instruksi dari memory berdasarkan nilai dari program counter.

Sedangkan manajemen memori adalah suatu kegiatan untuk mengelola memori komputer. Proses ini menyediakan cara mengalokasikan memori untuk proses atas permintaan mereka, membebaskan untuk digunakan kembali ketika tidak lagi diperlukan serta menjaga alokasi ruang memori bagi proses. Pengelolaan memori utama sangat penting untuk sistem komputer, penting untuk memproses dan fasilitas masukan/keluaran secara efisien, sehingga memori dapat menampung sebanyak mungkin proses dan sebagai upaya agar pemogram atau proses tidak dibatasi kapasitas memori fisik di sistem komputer (Eko, 2009).

Memory manager merupakan salah satu bagian sistem operasi yang mempengaruhi dalam menentukan proses mana yang diletakkan pada antrian.

Jenis Memori

- a) Memori Kerja
- b) ROM/PROM/EPROM/EEPROM
- c) RAM



- d) Cache memory
- e) Memori Dukung
- f) Floppy, harddisk, CD, dll.

Alamat Memori

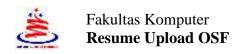
- a) Alamat memori mutlak (alamat fisik)
- b) Alamat memori relatif (alamat logika)
- c) Hubungan antara alamat multak dan alamat relatif
- d) Jenis memori dan alamat memori

Isi Memori

- a) Sistem bahasa penataolahan
- b) Sistem Utilitas
- c) Inti Sistem Operasi
- d) Sistem Operasi
- e) Pengendali alat (device drivers)
- f) File pemakai

Fungsi Manajemen Memori

- a) Mengelola informasi yang dipakai dan tidak dipakai.
- b) Mengalokasikan memori ke proses yang memerlukan.
- c) Mendealokasikan memori dari proses telah selesai.
- d) Mengelola swapping atau paging antara memori utama dan disk.



Sistem operasi memberikan tanggapan terhadap manajemen memori utama untuk aktivitas-aktivitas sebagai berikut:

- a) Menjaga dan memelihara bagian-bagian memori yang sedang digunakan dan dari yang menggunakan.
- b) Memutuskan proses-proses mana saja yang harus dipanggil kememori jika masih ada ruang di memori.
- c) Mengalokasikan dan mendelokasikan ruang memori jika diperlukan

Jenis-Jenis Manajemen Memory

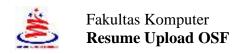
Manajemen Memory Untuk Monoprogramming

Bila program komputer yang dijalankan hanya satu jenis selama proses berlangsung maka dikatakan mode kerja komputer itu adalah monoprogramming. Selama komputer itu bekerja maka memory RAM seluruhnya di kuasai oleh program tersebut. Jadi RAM tidak dapat di masuki oleh program lain. Mode serupa ini di temui pada komputer berbasis DOS.

Penempatan program di memory diatur sedemikain rupa sehingga (Eko, 2009):

- a) BIOS selalu di ROM (BIOS)
- b) Sistem Operasi di RAM bawah (alamat rendah)
- c) Program Aplikasi di RAM tengah (alamat sesudah OS terakhir)
- d) Data Sementara di RAM atas (alamat sesudah Aplikasi terakhi).

Bila sistem operasi telah selasai dimuat maka tampillah prompt di layar monitor, dan itu adalah tanda bahwa komputer siap menerima program aplikasi. Letakkan disk yang berisi program aplikasi pada diskdrive yang aktif lalu eksekusi , sehingga program itu termuat seluruhnya ke RAM. Dengan demikian program aplikasi siap digunakan menurut semestinya.Kita lihat ketika komputer mula-mula dinyalakan maka proses yang dibaca pertama kali adalah apa yang tertulis di dalam ROM. Setelah semua perintah di adalam ROM BIOS selesai dibaca maka komputer



meminta kita memasukkan DOS ke dalam RAM-nya.Ketika DOS dibaca maka diletakkan sebagian dari program DOS yang terpenting saja ke dalam RAM, seperti : COMMAND.COM dan INTERNAL COMMAND. Sedangkan program DOS yang lain masih tetap di dalam disk dan apabila kita perlukan dapat di eksekusi. Hal itu berguna untuk mrnjaga agar RAM tidak penuh oleh Sistem Operasi saja.

Ketika kita bekerja dengan program aplikasi tasdi maka kita akan menghasilkan data. Data itu akan di simpan sementara di RAM yang masih tersisa. Data yang disimpan di RAM bersifat voletile, artinya data hanya bisa bertahan selama catudaya komputer masih ON. Untuk berjaga-jaga biasakan menyimpan data ke disk dalam jangka waktu yang tidak terlalu lama, misalnya setiap 5 menit sekali. Selain menjaga data agar tidak amblas menyimpan ke disk bertujuan juga untuk mengosongkan RAM agar tidak cepat penuh.

Didalam sistem juga dapat kita lihat bahwa sistem operasi terletak berdekatan dengan program lain di RAM sehingga kemungkinan sistem operasi ter ganggu atau terubah oleh proses yang sedang berjalan sangat besar .Hal itu tidak boleh terjadi.Untuk mencegah terganggu sitem operasi tersebut maka alamat tertinggi dari sistem operasi dletakkan pada register batas dalam CPU. Jika ada proses yang mengacu ke alamat itu atau yang lebih rendah dari itu maka proses di hentikan dan program akan menampilkan pesan kesalahan.

Manajemen Memory Untuk Multiprogramming

Untuk sistem komputer yang berukuran besar (bukan small computers), membutuhkan pengaturan memori, karena dalam multiprogramming akan melibatkan banyak pemakai secara simultan sehingga di memori akan terdapat lebih dari satu proses bersamaan. Oleh karena itu dibutuhkan sistem operasi yang mampu mendukung dua kebutuhan tersebut, meskipun hal tersebut saling bertentangan, yaitu (Ama, 2003):

- a) Pemisahan ruang-ruang alamat.
- b) Pemakaian bersama memori.

Fakultas Komputer
Resume Upload OSF

Manajer memori harus memaksakan isolasi ruang-ruang alamat tiap proses agar mencegah proses aktif atau proses yang ingin berlaku jahat mengakses dan merusak ruang alamat proses lain. Manajer memori di lingkungan multiprogramming sekalipun melakukan dua hal, yaitu:

- a) Proteksi memori dengan isolasi ruang-ruang alamat secara dis-joint.
- b) Pemakaian bersama memori.

Memungkinkan proses-proses bekerja sama mengakses daerah memori bersama. Ketika konsep multiprogramming digunakan, pemakaian CPU dapat ditingkatkan. Sebuah model untuk mengamati pemakaian CPU secara probabilistic :

CPU utilization = 1 - p n

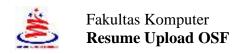
Dengan:

- a) N menunjukkan banyaknya proses pada suatu saat, sehingga kemungkinan bahwa semua n proses akan menunggu menggunakan I/O (masalah CPU menganggur) adalah sebesar pn. Fungsi dari n disebut sebagai degree of multiprogramming.
- b) P menunjukkan besarnya waktu yang digunakan sebuah proses

B. Strategi Manajemen Memori

Strategi yang dikenal untuk mengatasi hal tersebut adalah memori maya. Memori maya menyebabkan sistem seolah-olah memiliki banyak memori dibandingkan dengan keadaan memori fisik yang sebenarnya. Memori maya tidak saja memberikan peningkatan komputasi, akan tetapi memori maya juga memiliki bberapa keuntungan seperti :

Large Address Space



Membuat sistem operasi seakan-akan memiliki jumlah memori melebihi kapasitas memori fisik yang ada. Dalam hal ini memori maya memiliki ukuran yang lebih besar daripada ukuran memori fisik.

Proteksi.

Setiap proses di dalam sistem memiliki virtual address space. Virtual address space tiap proses berbeda dengan proses yang lainnya lagi, sehingga apapun yang terjadi pada sebuah proses tidak akan berpengaruh secara langsung pada proses lainnya

Memory Mapping

Memory mapping digunakan untuk melakukan pemetaan image dan file-file data ke dalam alamat proses. Pada pemetaan memori, isi dari file akan di link secara langsung ke dalam virtual address space dari proses.

Fair Physical Memory Allocation

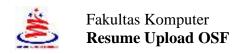
Digunakan oleh Manajemen Memori untuk membagi penggunaan memori fisik secara "adil" ke setiap proses yang berjalan pada sistem.

Shared Virtual Memory.

Meskipun tiap proses menggunakan address space yang berbeda dari memori maya, ada kalanya sebuah proses dihadapkan untuk saling berbagi penggunaan memori.

C. Ruang Alamat Logika dan Fisik

Alamat Logika adalah alamat ya dibentuk di CPU, disebut juga alamat virtual. Alamat fisik adalah alamat yang terlihat oleh memori. Untuk mengubah dari alamat logika ke alamat fisik diperlukan suatu perangkat keras yang bernama MMU (Memory Management Unit). Pengubahan dari alamat logika ke alamat fisik adalah pusat dari manajemen memori. Alamat yang dibangkitkan oleh CPU disebut alamat logika (logical address) dimana alamat terlihat sebagai uni memory yang disebut alamat fisik (physical address). Tujuan utama manajemen memori adalah konsep meletakkan ruang alamat logika ke ruang alamat fisik (Ama, 2003).



Hasil skema waktu kompilasi dan waktu pengikatan alamat pada alamat logika dan alamat memori adalah sama. Tetapi hasil skema waktu pengikatan alamat waktu eksekusi berbeda. dalam hal ini, alamat logika disebut dengan alamat maya (virtual address). Himpunan dari semua alamat logika yang dibangkitkan oleh program disebut dengan ruang alamat logika (logical address space); himpunan dari semua alamat fisik yang berhubungan dengan alamat logika disebut dengan ruang alamat fisik (physical address space).

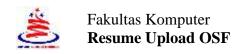
Memory Manajement Unit (MMU) adalah perangkat keras yang memetakan alamat virtual ke alamat fisik. Pada skema MMU, nilai register relokasi ditambahkan ke setiap alamat yang dibangkitkan oleh proses user pada waktu dikirim ke memori.

Register basis disebut register relokasi. Nilai dari register relokasi ditambahkan ke setiap alamat yang dibangkitkan oleh proses user pada waktu dikirim ke memori, sebagai contoh, apabila basis 14000, maka user mencoba menempatkan ke alamat lokasi 0 dan secara dinamis direlokasi ke lokasi 14000. Pengaksesan ke lokasi logika 346, maka akan dipetakan ke lokasi 14346. Sistem operasi MS-DOS yang masih keluarga intel 80X86 menggunakan empat register relokasi ketika proses loading dan running.

User program tidak pernah melihat alamat fisik secara real. Program dapat membuat sebuah penunjuk ke lokasi 346, mengirimkan ke memory, memanipulasinya, membandingkan dengan alamat lain, semua menggunakan alamat 346. Hanya ketika digunakan sebagai alamat memory akan direlokasi secara relatif ke register basis.

D. Swapping

Sebuah proses, sebagaimana telah diterangkan di atas, harus berada di memori sebelum dieksekusi. Proses swapping menukarkan sebuah proses keluar dari memori untuk sementara waktu ke sebuah penyimpanan sementara dengan sebuah proses lain yang sedang membutuhkan sejumlah alokasi memori untuk dieksekusi. Tempat penyimpanan sementara ini biasanya berupa sebuah fast disk dengan kapasitas yang dapat menampung semua salinan dari semua gambaran memori serta menyediakan akses langsung ke gambaran tersebut. Jika eksekusi proses yang dikeluarkan tadi

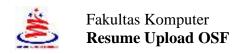


akan dilanjutkan beberapa saat kemudian, maka ia akan dibawa kembali ke memori dari tempat penyimpanan sementara tadi. Bagaimana sistem mengetahui proses mana saja yang akan dieksekusi? Hal ini dapat dilakukan dengan ready queue. Ready queue berisikan semua proses yang terletak baik di penyimpanan sementara maupun memori yang siap untuk dieksekusi. Ketika penjadwal CPU akan mengeksekusi sebuah proses, ia lalu memeriksa apakah proses bersangkutan sudah ada di memori ataukah masih berada dalam penyimpanan sementara. Jika proses tersebut belum berada di memori maka proses swapping akan dilakukan seperti yang telah dijelaskan di atas.

Sebuah contoh untuk menggambarkan teknik swapping ini adalah sebagai berikut: Algoritma Round-Robin yang digunakan pada multiprogramming environment menggunakan waktu kuantum (satuan waktu CPU) dalam pengeksekusian prosesprosesnya. Ketika waktu kuantum berakhir, memory manager akan mengeluarkan (swap out) proses yang telah selesai menjalani waktu kuantumnya pada suatu saat serta memasukkan (swap in) proses lain ke dalam memori yang telah bebas tersebut. Pada saat yang bersamaan penjadwal CPU akan mengalokasikan waktu untuk proses lain dalam memori. Hal yang menjadi perhatian adalah, waktu kuantum harus cukup lama sehingga waktu penggunaan CPU dapat lebih optimal jika dibandingkan dengan proses penukaran yang terjadi antara memori dan disk.

Teknik swapping roll out, roll in menggunakan algoritma berbasis prioritas dimana ketika proses dengan prioritas lebih tinggi tiba maka memory manager akan mengeluarkan proses dengan prioritas yang lebih rendah serta me-load proses dengan prioritas yang lebih tinggi tersebut. Saat proses dengan prioritas yang lebih tinggi telah selesai dieksekusi maka proses yang memiliki prioritas lebih rendah dapat dimasukkan kembali ke dalam memori dan kembali dieksekusi.

Sebagian besar waktu swapping adalah waktu transfer. Sebagai contoh kita lihat ilustrasi berikut ini: sebuah proses pengguna memiliki ukuran 5 MB, sedangkan tempat penyimpanan sementara yang berupa harddisk memiliki kecepatan transfer data sebesar 20 MB per detiknya. Maka waktu yang dibutuhkan untuk mentransfer



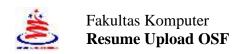
proses sebesar 5 MB tersebut dari atau ke dalam memori adalah sebesar 5000 KB / 20000 KBps = 250 ms.

Perhitungan di atas belum termasuk waktu latensi, sehingga jika kita asumsikan waktu latensi sebesar 2 ms maka waktu swap adalah sebesar 252 ms. Oleh karena terdapat dua kejadian dimana satu adalah proses pengeluaran sebuah proses dan satu lagi adalah proses pemasukan proses ke dalam memori, maka total waktu swap menjadi 252 + 252 = 504 ms.

Agar teknik swapping dapat lebih efisien, sebaiknya proses-proses yang di- swap hanyalah proses-proses yang benar-benar dibutuhkan sehingga dapat mengurangi waktu swap. Oleh karena itulah, sistem harus selalu mengetahui perubahan apapun yang terjadi pada pemenuhan kebutuhan terhadap memori. Disinilah sebuah proses memerlukan fungsi system call, yaitu untuk memberitahukan sistem operasi kapan ia meminta memori dan kapan membebaskan ruang memori tersebut.

Jika kita hendak melakukan swap, ada beberapa hal yang harus diperhatikan. Kita harus menghindari menukar proses dengan M/K yang ditunda (asumsinya operasi M/K tersebut juga sedang mengantri di antrian karena peralatan M/Knya sedang sibuk). Contohnya seperti ini, jika proses P1dikeluarkan dari memori dan kita hendak memasukkan proses P2, maka operasi M/K yang juga berada di antrian akan mengambil jatah ruang memori yang dibebaskan P1 tersebut. Masalah ini dapat diatasi jika kita tidak melakukan swap dengan operasi M/K yang ditunda. Selain itu, pengeksekusian operasi M/K hendaknya dilakukan pada buffer sistem operasi.

Tiap sistem operasi memiliki versi masing-masing pada teknik swapping yang digunakannya. Sebagai contoh pada UNIX, swapping pada dasarnya tidak diaktifkan, namun akan dimulai jika banyak proses yang membutuhkan alokasi memori yang banyak. Swapping akan dinonaktifkan kembali jika jumlah proses yang dimasukkan berkurang. Pada sistem operasi Microsoft Windows 3.1, jika sebuah proses baru dimasukkan dan ternyata tidak ada cukup ruang di memori untuk menampungnya, proses yang lebih dulu ada di memori akan dipindahkan ke disk. Sistem operasi ini pada dasarnya tidak menerapkan teknik swapping secara penuh, hal ini disebabkan



pengguna lebih berperan dalam menentukan proses mana yang akan ditukar daripada penjadwal CPU. Dengan ketentuan seperti ini proses-proses yang telah dikeluarkan tidak akan kembali lagi ke memori hingga pengguna memilih proses tersebut untuk dijalankan.

Manajemen Memori Berdasarkan Keberadaan Swapping Atau Paging

- a) Manajemen tanpa swapping atau paging
- b) Manajemen dengan swapping atau paging

Memori Tanpa Swapping Or Paging

Merupakan manajemen memori tanpa pemindahan citra proses antara memori utama dan disk selama eksekusi. Manajemen ini terdiri dari :

Monoprogramming

Ciri-ciri:

- a) Hanya satu proses pada satu saat
- b) Hanya satu proses menggunakan semua memori
- c) Pemakai memuatkan program ke seluruh memori dari disk atau tape
- d) Program mengambil kendali seluruh mesin

Multiprogramming Dengan Pemartisian Statis

- a) Pemartisian menjadi partisi-partisi berukuran sama, yaitu ukuran semua partisi memori adalah sama
- b) Pemartisian menjadi partisi-partisi berukuran berbeda, yaitu ukuran semua partisi memori adalah berbeda.

Strategi Penempatan Program Ke Partisi



Satu Antrian Tunggal Untuk Semua Partisi Keuntungan: Lebih fleksibel serta implementasi dan operasi lebih minimal karena hanya mengelola satu antrian. Kelemahan: Proses dapat ditempatkan di partisi yang banyak diboroskan, yaitu proses kecil ditempatkan di partisi sangat besar.

Tetap dengan Satu Antrian Satu Antrian Untuk Tiap Partisi (banyak antrian Untuk Seluruh Partisi). Keuntungan : Meminimalkan pemborosan memori. Kelemahan : Dapat terjadi antrian panjang di suatu partisi sementara antrian partisi – partisi lain kosong

Multiprogramming Dengan Swapping

Merupakan manajemen memori dengan pemindahan citra proses antara memori utama dan disk selama eksekusi, atau dengan kata lain merupakan manajemen pemindahan proses dari memori utama ke disk dan kembali lagi (swapping). Manajemen ini terdiri dari :

Multiprogramming Dengan Pemartisisan Dinamis

Jumlah, lokasi dan ukuran proses di memori dapat beragam sepanjang waktu secara dinamis. Kelemahan: a) Dapat terjadi lubang-lubang kecil memori di antara partisipartisi yang dipakai; b) Merumitkan alokasi dan dealokasi memori.

Solusi:

Lubang-lubang kecil di antara blok-blok memori yang digunakan dapat diatasi dengan pemadatan memori yaitu menggabungkan semua lubang kecil menjadi satu lubang besar dengan memindahkan semua proses agar saling berdekatan.

Strategi Alokasi Memori

a) First fit algorithm: memory manager men-scan list untuk menemukan hole yg cukup untuk menampung proses yg baru. Proses akan menempati hole pertama yg ditemuinya yg cukup untuk dirinya.

Fakultas Komputer
Resume Upload OSF

b) Next fit algorithm : sama dengan first fit, tetapi pencarian hole dimulai dari

hole ditemuinya dari scan sebelumnya.

c) Best fit algorithm : dicari hole yang akan menghasilkan sisa paling sedikit

setelah dimasuki proses.

d) Worst fit algorithm: kebalikan dari best fit.

e) Quick fit algorithm: mengelompokkan hole-hole dan membuat listnya sendiri.

Misalnya, ada list untuk hole 4K, satu list untuk 8K, dst.

f) Sistem Buddy: Memori di susun dalm senari blok-blok bebas berukuran

1,2,4,8,16 byte dst, sampai kapasitas memori.

Dari berbagai cara alokasi tsb. Di atas, sebuah hole yg ditempati proses akan terbagi

menjadi bagian yang dipakai proses dan memori yang tidak terpakai (fragmen).

Timbulnya memori yang tidak terpakai disebut fragmentasi. Ada dua macam fragmen

:

a) Internal: sisa hole yang tidak terpakai setelah terisi proses.

b) Eksternal : hole yang secara utuh terlalu kecil untuk dipakai oleh proses

manapun.

Alokasi Ruang Swap pada Disk (Penempatan proses pada disk setelah di-swap-

out dari memori)

a) Ruang disk tempat swap dialokasikan begitu diperlukan

b) Ruang disk tempat swap dialokasikan lebih dahulu.

Algoritma untuk pengaturan ruang swap pada disk sama dengan untuk memori

utama. Perbedaannya adalah ruang pada disk harus dialokasikan sebagai kelipatan

bilangan bulat dari disk block.

BAB III: PENUTUP

A. Kesimpulan

28



Memori dan Manajemen Memori

Memori adalah pusat dari operasi pada sistem komputer modern, berfungsi sebagai tempat penyimpanan informasi yang harus diatur dan dijaga sebaik-baiknya. Memori adalah array besar dari word atau byte, yang disebut alamat. CPU mengambil instruksi dari memory berdasarkan nilai dari program counter (Ama, 2003).

Sedangkan manajemen memori adalah suatu kegiatan untuk mengelola memori komputer. Proses ini menyediakan cara mengalokasikan memori untuk proses atas permintaan mereka, membebaskan untuk digunakan kembali ketika tidak lagi diperlukan serta menjaga alokasi ruang memori bagi proses. Manajemen Memori merupakan salah satu bagian terpenting pada sistem operasi. Sejak awal komputer digunakan untuk keperluan komputasi, kebutuhan akan memori yang lebih besar dibandingkan dengan keadaan fisik memori di dalam sistem terus meningkat. Berbagai perhitungan dan strategi terus dilakukan untuk mengatasi keterbatasan ukuran memori fisik (Ama, 2003).

Strategi Manjemen Memori

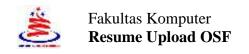
Strategi yang dikenal untuk mengatasi hal tersebut adalah memori maya. Memori maya menyebabkan sistem seolah-olah memiliki banyak memori dibandingkan dengan keadaan memori fisik yang sebenarnya. Memori maya tidak saja memberikan peningkatan komputasi.

Ruang Alamat Logika dan Fisik

Tujuan utama manajemen memori adalah konsep meletakkan ruang alamat logika ke ruang alamat fisik. Alamat Logika adalah alamat yg dibentuk di CPU, disebut juga alamat virtual. Alamat fisik adalah alamat yang terlihat oleh memori.

Swapping

Sebuah proses yang harus berada di memori sebelum dieksekusi. Proses swapping menukarkan sebuah proses keluar dari memori untuk sementara waktu ke sebuah penyimpanan sementara dengan sebuah proses lain yang sedang membutuhkan sejumlah alokasi memori untuk dieksekusi. Tempat penyimpanan sementara ini biasanya berupa sebuah fast disk dengan kapasitas yang dapat menampung semua salinan dari semua gambaran memori serta menyediakan akses langsung ke gambaran tersebut.

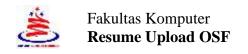


A. ID SECURITY

QWTD44112377-ASP-524414475

B. REFERENCE

- [1] O. M. Febriani and A. S. Putra, "Sistem Informasi Monitoring Inventori Barang Pada Balai Riset Standardisasi Industri Bandar Lampung," *J. Inform.*, vol. 13, no. 1, pp. 90–98, 2014.
- [2] A. S. Putra, "Paperplain: Execution Fundamental Create Application With Borland Delphi 7.0 University Of Mitra Indonesia," 2018.
- [3] A. S. Putra, "2018 Artikel Struktur Data, Audit Dan Jaringan Komputer," 2018.
- [4] A. S. Putra, "ALIAS MANAGER USED IN DATABASE DESKTOP STUDI CASE DB DEMOS."
- [5] A. S. Putra, "COMPREHENSIVE SET OF PROFESSIONAL FOR DISTRIBUTE COMPUTING."
- [6] A. S. Putra, "DATA ORIENTED RECOGNITION IN BORLAND DELPHI 7.0."
- [7] A. S. Putra, "EMBARCADERO DELPHI XE 2 IN GPU-POWERED FIREMONKEY APPLICATION."
- [8] A. S. Putra, "HAK ATAS KEKAYAAN INTELEKTUAL DALAM DUNIA TEKNOLOGY BERBASIS REVOLUSI INDUSTRI 4.0."
- [9] A. S. Putra, "IMPLEMENTASI PERATURAN PERUNDANGAN UU. NO 31 TAHUN 2000 TENTANG DESAIN INDUSTRI BERBASIS INFORMATION TECHNOLOGY."
- [10] A. S. Putra, "IMPLEMENTATION OF PARADOX DBASE."
- [11] A. S. Putra, "IMPLEMENTATION OF TRADE SECRET CASE STUDY SAMSUNG MOBILE PHONE."
- [12] A. S. Putra, "IMPLEMENTATION PATENT FOR APPLICATION WEB BASED CASE STUDI WWW. PUBLIKLAMPUNG. COM."
- [13] A. S. Putra, "IMPLEMENTATION SYSTEM FIRST TO INVENT IN DIGITALLY INDUSTRY."
- [14] A. S. Putra, "MANUAL REPORT & INTEGRATED DEVELOPMENT ENVIRONMENT BORLAND DELPHI 7.0."
- [15] A. S. Putra, "PATENT AS RELEVAN SUPPORT RESEARCH."
- [16] A. S. Putra, "PATENT FOR RESEARCH STUDY CASE OF APPLE. Inc."
- [17] A. S. Putra, "PATENT PROTECTION FOR APPLICATION INVENT."
- [18] A. S. Putra, "QUICK REPORT IN PROPERTY PROGRAMMING."
- [19] A. S. Putra, "REVIEW CIRCUIT LAYOUT COMPONENT REQUIREMENT ON ASUS NOTEBOOK."
- [20] A. S. Putra, "REVIEW TRADEMARK PATENT FOR INDUSTRIAL TECHNOLOGY BASED 4.0."
- [21] A. S. Putra, "TOOLBAR COMPONENT PALLETTE IN OBJECT ORIENTED PROGRAMMING."



- [22] A. S. Putra, "WORKING DIRECTORY SET FOR PARADOX 7."
- [23] A. S. Putra, "ZQUERY CONNECTION IMPLEMENTED PROGRAMMING STUDI CASE PT. BANK BCA Tbk."
 - [24] A. S. Putra, D. R. Aryanti, and I. Hartati, "Metode SAW (Simple Additive Weighting) sebagai Sistem Pendukung Keputusan Guru Berprestasi (Studi Kasus: SMK Global Surya)," in *Prosiding Seminar Nasional Darmajaya*, 2018, vol. 1, no. 1, pp. 85–97.
- [25] A. S. Putra and O. M. Febriani, "Knowledge Management Online Application in PDAM Lampung Province," in *Prosiding International conference on Information Technology and Business (ICITB)*, 2018, pp. 181–187.
- [26] A. S. Putra, O. M. Febriani, and B. Bachry, "Implementasi Genetic Fuzzy System Untuk Mengidentifikasi Hasil Curian Kendaraan Bermotor Di Polda Lampung," *SIMADA (Jurnal Sist. Inf. dan Manaj. Basis Data)*, vol. 1, no. 1, pp. 21–30, 2018.
- [27] A. S. Putra, H. Sukri, and K. Zuhri, "Sistem Monitoring Realtime Jaringan Irigasi Desa (JIDES) Dengan Konsep Jaringan Sensor Nirkabel," *IJEIS* (*Indonesian J. Electron. Instrum. Syst.*, vol. 8, no. 2, pp. 221–232.
- [28] D. P. Sari, O. M. Febriani, and A. S. Putra, "Perancangan Sistem Informasi SDM Berprestasi pada SD Global Surya," in *Prosiding Seminar Nasional Darmajaya*, 2018, vol. 1, no. 1, pp. 289–294.