BARISAN BILANGAN HAPPY TERURUT

SKRIPSI

DIAN PERMATA SARI
0810432017

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS ANDALAS
PADANG 2012

UCAPAN TERIMA KASIH

Syukur Alhamdulillah ya Allah, Engkau telah memberikan begitu banyak nikmat kepada hamba sehingga hamba dapat menyelesaikan tugas akhir ini pada waktunya. Shalawat dan salam teruntuk baginda Rasulullah saw, sang tauladan sepanjang masa yang telah membawa risalah agung ke dunia ini.

Terimakasih kepada kedua orangtuaku yang telah memberikan begitu banyak perhatian dan kasih sayang yang tulus serta pengorbanan yang tak akan tergantikan oleh apapun dan siapapun serta adik-adik ku tercinta (Ayu, Pita, Dedek) yang telah memberikan dukungan dan semangat kepadaku.

Dosen pembimbing, Bapak Dr. Syafrizal Sy dan Bapak Dr. Admi Nazra, terima kasih banyak atas bimbingan, saran, perhatian, dan waktu luang yang telah bapak berikan sehingga sari bisa menyelesaikan skripsi ini.

Dosen penguji, Ibu Hazmira Yozza, M.Si, Ibu Dr. Lyra Yulianti, dan Bapak Dr. Dodi Devianto, terima kasih atas pengarahan dan saran untuk perbaikan skripsi ini.

Dosen-dosen jurusan Matematika, Pak Budi, Pak Efendi, Pak Narwen, Pak Werman, Pak Made, Pak Iqbal, Pak Yudi, Pak Jon, Pak Zulakmal, Pak Muhafzan, Pak Syafruddin, Pak Ginting, Bu Gema, Bu Ayu, Bu Iza, Bu Monik, Bu Sil, Bu Nova, Bu Riri, dan Bu Maiyastri. Terima kasih banyak atas segala ilmu dan motivasi yang diberikan.

Staf dan pegawai jurusan Matematika, Mama Cun, Bu Eli, Pak Syamsir, Pak Yatim, Bu Dona, Bu Opi, Bu Nang, Bu Debi dan Bu Nun, terima kasih karena telah membantu selama penulis melaksanakan studi di jurusan Matematika Universitas Andalas.

Sohib-sohibku, Siska (Makasii motivasinya, makasii doanya, dan makasii untuk semuanya), Feby (Makasii atas perhatiaan dan kasih sayang nya), Eris (Semangat cinto, insyaallah suatu saat kita akan bareng kuliah lagi), Iin (Makasii banyak in atas segala pengorbanannya, sar ga' bakalan lupa kalo kita pernah bersama-sama), Voenid (Makasii Fu karna telah mengajarkan kedewasaan kepadaku dengan caramu. Sekali lagi Makasii), Dita (Makasii telah menjadi orang pertama yang mengajarkanku arti islam yang syamil wa mutakammil), Ecy, Ade, KK Su (Makasii atas kebersamaannya selama ini). "*Sahabat bukan mereka yang lama berada di samping mu, tetapi mereka yang telah lama berada di hatimu*".

Shofiyah members, KK Ul, KK Laila, KK IL, Elsa, Eel, Asa, Tutut, Rina, Dibah, widi, Yosi, Dahlia, Nia, Widi. Terimakasih atas semua perhatian dan kebersamaan selama ini. Semoga kelak kita bisa berkumpul lagi.

Keluarga besar FSI dan HIMATIKA FMIPA UNAND, semangat dan perjuangan bersama-sama akan menjadi kenangan di kemudian hari.

Teman-teman angkatan 2008 O'Laplace, Ivone, Liza, Via, Virza, Nurma, Vebby, Elza, Iin, Tika Y, Eed, Ica, Rere, Elin, Mimi, Tika K, Manda, Kak Su, Anggi, Oji, Ichel, Cinta, Erik, Shanda, Ade, Tere, Hasan, Welly , Desi, Helcy, Lindo, Yuli, Opa, Ana, Mia, Mezi, Wili, Enid, Elvi, Cesa, Dina Irawati, Ririn, Meta, Rika, my aunty(Dina Yelni) , Uthe, Wiwiek, Kak Nini, Eris, Ciphie, Sarti, Bayu, Oni, Tama, Neli, Putri, Rara, Lia, dan Yolwi. Terimakasih untuk semua pengalaman yang telah kita lewati sejak tahun pertama.

Uda-uda dan Uni-uni senior Matematika dan untuk semua yang tidak bisa disebutkan satu per satu. Semoga Allah membalas semua kebaikan yang telah diberikan.

Dian Permata Sari

KATA PENGANTAR

Syukur alhamdulillah, segala puji Penulis haturkan atas kehadirat Allah SWT, yang telah melimpahkan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini dengan judul "Barisan Bilangan *Happy* Terurut ". Shalawat dan salam semoga selalu tercurahkan kepada Baginda Rasulullah SAW yang telah menebarkan ilmu dan iman dalam cahaya Islam yang beliau bawa. Penulis menyampaikan ungkapan terima kasih dan penghargaan yang tulus kepada yang terhormat:

1. Bapak Dr. Syafrizal Sy sebagai ketua jurusan pada jurusan Matematika FMIPA Universitas Andalas sekaligus pembimbing 1 yang telah bersedia meluangkan waktu untuk memberikan bimbingan, pengarahan, dan saran kepada penulis sampai selesainya skripsi ini.

2. Bapak Dr. Admi Nazra sebagai pembimbing II yang telah memberikan bimbingan, saran, dan inspirasi kepada penulis dalam menyelesaikan skripsi ini.

3. Ibu Ir. Hazmira Yozza, M.Si, Ibu Dr. Lyra Yulianti, dan Bapak Dr. Dodi Devianto sebagai penguji yang telah memberikan pengarahan dan saran untuk perbaikan penulisan skripsi ini.

4. Bapak Narwen, M.Si selaku pembimbing akademis yang telah memberikan nasehat dan motivasi kepada penulis.

5. Seluruh staf pengajar Jurusan Matematika Universitas Andalas yang telah banyak memberikan ilmunya kepada penulis dan seluruh staf tata usaha Jurusan Matematika yang telah banyak membantu selama penulis melaksanakan studi di jurusan Matematika Universitas Andalas.

6. Seluruh teman-teman yang telah mendukung dan memberikan semangat kepada penulis terutama teman-teman angkatan 2008 (O'Laplace). Buat

teman seperjuangan: Eris, Iin, Enid, Kak su, Helcy, Ade yang telah menemani dalam suka dan duka serta kakak-kakak senior dan adik-adik junior yang tidak bisa disebutkan satu persatu di Jurusan Matematika Universitas Andalas.

7. Semua pihak yang telah membantu penulis yang tidak dapat disebutkan satu persatu.

Selesainya skripsi ini tidak lepas dari do'a yang tulus, motivasi, dorongan semangat, dan bantuan yang senantiasa diberikan oleh kedua orang tua ayahanda Danil Kenedi dan ibunda Yulinur, Adinda Ayu, Pita, Wini serta seluruh keluarga besar penulis.

Penulis menyadari penulisan skripsi ini masih jauh dari kesempurnaan. Oleh karena itu, dengan kerendahan hati penulis mengharapkan kritik dan saran agar kedepannya diperoleh hasil yang lebih baik. Penulis berharap agar skripsi ini dapat bermanfaat bagi semua pihak yang membutuhkannya. Aamiin.

Padang, April 2012

Dian Permata Sari

ABSTRAK

Misalkan n adalah suatu bilangan asli dengan k digit, sedemikian sehingga n dapat dinyatakan dalam bentuk $n = a_1a_2 \dots a_k$ dengan $a_1 \neq 0$, $k \in \mathbb{N}$ dan a_k adalah digit ke-k dari n. Misalkan $T : \mathbb{N} \rightarrow \mathbb{N}$ merupakan suatu fungsi yang didefinisikan sebagai berikut

$$T(n) = {a_1}^2 + {a_2}^2 + \dots + {a_k}^2.$$

Suatu bilangan *happy* n adalah suatu bilangan yang memenuhi kondisi $T^r(n) = 1$ untuk suatu $r \geq 0$ dimana

$$T^r(n) = \underbrace{T(T(\cdots T}_{sebanyak\ r}(n)\cdots)).$$

Sebaliknya, jika $T^r(n) \in \{4,\ 16,\ 37,\ 58,\ 89,\ 145,\ 42,\ 20\}$ maka n dikatakan bilangan *unhappy*. Pada skripsi ini dikaji eksistensi dari barisan bilangan *happy* terurut dengan panjang sebarang yaitu untuk setiap $m \in \mathbb{N}$ terdapat sebuah bilangan asli l_0 sedemikian sehingga setiap anggota dari barisan terbatas $l_0 + 1,\ l_0 + 2,\ l_0 + 3,\ \dots,\ l_0 + m$ adalah bilangan *happy*. Pada skripsi ini juga diperoleh bahwa bilangan terkecil dari barisan bilangan *happy* terurut dengan panjang enam adalah 78999999999999959999999996.

Kata Kunci: *bilangan happy, bilangan happy terurut.*

DAFTAR ISI

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pada tahun 1994, Richard K. Guy dalam bukunya yang berjudul "*Unsolved Problems in Number Theory*" mengutarakan beberapa masalah pada teori bilangan, salah satunya mengenai bilangan *happy*. Misalkan n adalah suatu bilangan asli dengan k digit, sedemikian sehingga n dapat dinyatakan dalam bentuk $n = a_1 a_2 \ldots a_k$ dengan $a_1 \neq 0$, $k \in \mathbb{N}$ dan a_k adalah digit ke-k dari n. Misalkan $T : \mathbb{N} \rightarrow \mathbb{N}$ merupakan suatu fungsi yang didefinisikan sebagai berikut

$$T(n) = {a_1}^2 + {a_2}^2 + \ldots + {a_k}^2.$$

Suatu bilangan *happy* n adalah suatu bilangan yang memenuhi kondisi $T^r(n) = 1$ untuk suatu $r \geq 0$ dimana

$$T^r(n) = \underbrace{T(T(\cdots T}_{sebanyak\ r}(n) \cdots)).$$

Sebaliknya, jika $T^r(n) \in \{4,\ 16,\ 37,\ 58,\ 89,\ 145,\ 42,\ 20\}$ untuk suatu $r \geq 0$ maka n dikatakan bilangan *unhappy*.

Adapun barisan 100 bilangan *happy* pertama dapat dilihat sebagai berikut: 1, 7, 10, 13, 19, 23, 28, 31, 32, 44, 49, 68, 70, 79, 82, 86, 91, 94, 97, 100, 103, 109, 129, 130, 133, 139, 167, 176, 188, 190, 192, 193, 203, 208, 219, 226, 230, 236, 239, 262, 263, 280, 291, 293, 301, 302, 310, 313, 319, 320, 326, 329, 331, 338, 356, 362, 365, 367, 368, 376, 379, 383, 386, 391, 392, 397, 404, 409, 440, 446, 464, 469, 478, 487, 490, 496, 536, 556, 563, 565, 566, 608, 617, 622, 623, 632, 635, 637, 638, 644, 649, 653, 655, 656, 665, 671, 673, 680, 683, 694.

Salah satu permasalahan yang menarik untuk dikaji pada topik bilangan *happy* ini adalah mengenai barisan bilangan *happy* terurut. Suatu barisan bilangan terurut adalah suatu barisan aritmatika dengan selisih d untuk suatu $d \in \mathbb{Z}^+$. Jika $d = 1$ maka barisan tersebut dapat dinyatakan dalam bentuk $n,\ n+1,\ n+2,\ \ldots,\ n+l$ untuk suatu $n \in \mathbb{Z}$ dan $l \in \mathbb{N}$. Barisan bilangan terurut $n,\ n+1,\ n+2,\ \ldots,\ n+l$ yang semua anggotanya adalah bilangan *happy* dinamakan barisan bilangan *happy* terurut dengan panjang $m = l+1$.

Jika panjang barisan bilangan *happy* terurut adalah dua, maka pasangan terkecil dari barisan tersebut adalah 31 dan 32. Sementara itu, jika panjang barisan bilangan *happy* terurut adalah tiga, empat, atau lima maka bilangan-bilangan terkecil dari barisan tersebut diberikan pada Tabel 1.1.1.

Tabel 1.1.1. Bilangan terkecil dari barisan bilangan *happy* terurut.

m	Bilangan terkecil dari barisan bilangan *happy* terurut dengan panjang m	Barisan bilangan *happy* terurut dengan panjang m
2	31	31, 32
3	1880	1880, 1881, 1882
4	7839	7839, 7840, 7841, 7842
5	44488	44488, 44489, 44490, 44491,44492

Pada skripsi ini akan dikaji mengenai eksistensi barisan bilangan *happy* terurut dengan panjang sebarang, sehingga dapat dijamin adanya barisan bilangan *happy* terurut dengan panjang $2, 3, 4, \dots m$. Bilangan terkecil dari masing-masing barisan tersebut dapat ditentukan dengan menggunakan berbagai *software*. Seorang matematikawan bernama Jud Mc Cranie telah menghitung bahwa hingga bilangan 1×10^{20} tidak ditemukan suatu barisan bilangan *happy* terurut dengan panjang enam. Oleh karena itu, pada skripsi ini juga akan ditentukan bilangan terkecil dari barisan bilangan *happy* terurut yang panjangnya enam dengan menggunakan Program *Maple* 14.

1.2 Perumusan Masalah

Adapun masalah yang dibahas dalam skripsi ini adalah bagaimana mengkaji eksistensi barisan bilangan *happy* terurut dengan panjang sebarang serta menentukan bilangan terkecil dari barisan bilangan *happy* terurut dengan panjang enam.

1.3 Pembatasan Masalah

Agar penulisan skripsi ini terarah, maka penulis akan memfokuskan membahas mengenai eksistensi barisan bilangan *happy* terurut dengan panjang sebarang pada basis 10.

1.4 Tujuan

Tujuan dari penulisan skripsi ini adalah untuk mengkaji eksistensi barisan bilangan *happy* terurut dengan panjang sebarang serta menentukan bilangan terkecil dari barisan bilangan *happy* terurut dengan panjang enam.

1.5 Sistematika Penulisan

Skripsi ini dibagi menjadi empat bab. Bab I terdiri dari latar belakang masalah, perumusan masalah, pembatasan masalah, tujuan dan sistematika penulisan. Pada Bab II dijelaskan mengenai himpunan, relasi, pemetaan, induksi matematika, representasi bilangan asli, dan bilangan *happy*. Bab III memuat pembahasan mengenai bilangan *happy* pada basis 10, eksistensi barisan bilangan *happy* terurut dan bilangan terkecil dari barisan bilangan *happy* terurut dengan panjang enam. Kesimpulan dan saran dari hasil pembahasan terdapat pada Bab IV.

BAB II

LANDASAN TEORI

2.1 Himpunan, Relasi, dan Fungsi

Konsep himpunan merupakan konsep mendasar dalam semua cabang ilmu matematika. Secara intuitif, sebuah himpunan adalah suatu daftar, kumpulan atau kelas dari objek-objek yang didefinisikan secara jelas. Objek-objek tersebut dinamakan *elemen-elemen* atau *anggota-anggota* dari himpunan. Himpunan dapat didefinisikan dengan dua cara, yaitu:

1. Enumerasi, yaitu dengan cara mendaftarkan semua anggota himpunannya. Jika banyaknya anggota terlalu besar, tetapi mengikuti suatu pola tertentu maka dapat digunakan penulisan elipsis yaitu "..." .

   Contoh 2.1.1. $A = \{a, b, c, \ldots, y, z\}$.

   Contoh 2.1.2. $B = \{2^0, 2^1, 2^2, 2^3, \ldots\}$.

2. Pembangun himpunan, yaitu dengan cara mendeskripsikan sifat-sifat yang harus dipenuhi oleh setiap elemen himpunannya.

   Contoh 2.1.3. $C = \{x |\ x \in \mathbb{Z} \text{ dan } x \bmod 2 = 0\}$.

   Contoh 2.1.4. $D = \{p|\ p$ adalah orang yang pernah menjabat sebagai Presiden Republik Indonesia$\}$.

Himpunan semesta adalah himpunan yang anggotanya merupakan semua objek yang menjadi topik pembicaraan. Biasanya himpunan semesta dilambangkan dengan huruf S atau U.

Contoh 2.1.5. Dalam studi kependudukan di suatu negara, himpunan semesta terdiri atas semua orang di negara tersebut.

Himpunan kosong adalah himpunan yang tidak memiliki anggota dan dilambangkan dengan $\varnothing$ atau $\{\ \}$.

Contoh 2.1.6. $A = \{x | x^2 < 0 \text{ dan } x \in \mathbb{R}\}$.

Himpunan A dikatakan **himpunan bagian** dari himpunan B, jika dan hanya jika setiap anggota A merupakan anggota B. Sebaliknya, jika terdapat anggota A yang bukan merupakan anggota B, maka A bukan himpunan bagian dari B. Dengan kata lain, $A \subseteq B$ jika dan hanya jika

$$x \in A \Longrightarrow x \in B.$$

Contoh 2.1.7. Misalkan $A = \{1,\ 3,\ 5\}$ dan $B = \{0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6\}$. Maka diperoleh $A \subset B$.

Definisi 2.1.1. [8] *Misalkan diberikan dua himpunan yaitu himpunan A dan B. Perkalian kartesian (cartesian products) dari himpunan A dan B yang ditulis dengan $A \times B$ adalah himpunan semua pasangan terurut yang terbentuk dengan komponen pertama dari himpunan A dan komponen kedua dari himpunan B. Dengan kata lain,*

$$A \times B = \{(a, b) |\ a \in A \ dan\ b \in B\}.$$

Contoh 2.1.8. Misalkan $C = \{1,\ 2,\ 3\}$ dan $D = \{a,\ b\}$, maka perkalian kartesian dari C dan D adalah $C \times D = \{(1,a),(1,b),(2,a),(2,b),(3,a),(3,b)\}$

Definisi 2.1.2. [8] *Relasi biner R antara himpunan A dan B adalah himpunan bagian dari perkalian kartesian antara A dan B.*

Contoh 2.1.9. Misalkan $P = \{2,\ 3,\ 4\}$ dan $Q = \{2,\ 4,\ 8,\ 9,\ 15\}$. Jika didefinisikan relasi R dari P ke Q dengan

$$(p,q) \in R \text{ jika } p \text{ habis membagi } q,$$

maka diperoleh $R = \{(2,2),\ (2,4),\ (2,8),\ (3,9),\ (3,15),\ (4,4),\ (4,8)\}$.

Definisi 2.1.3. [8] *Relasi pada himpunan A adalah relasi antara A dan A.*

Contoh 2.1.10. Misalkan $A = \{2,\ 3,\ 4,\ 8,\ 9\}$. Jika didefinisikan relasi R dari A ke A dengan

$$(x,y) \in R \text{ jika } x \text{ adalah faktor prima dari } y,$$

maka diperoleh $R = \{(2,2),\ (2,4),\ (2,8),\ (3,3),\ (3,9)\}$.

Definisi 2.1.4. [8] *Relasi R pada pada himpunan A dikatakan sebagai suatu relasi ekivalen pada A, jika untuk setiap a, b, c di A berlaku*

1. $(a,a) \in R$.
2. *Jika* $(a,b) \in R$ *maka* $(b,a) \in R$.
3. *Jika* $(a,b) \in R$ *dan* $(b,c) \in R$ *maka* $(a,c) \in R$.

Contoh 2.1.11. Misalkan $a, b \in \mathbb{Z}$ dengan $\mathbb{Z}$ adalah himpunan bilangan bulat. Didefinisikan $(a, b) \in R$ apabila $a-b$ adalah bilangan genap. Berdasarkan Definisi 2.1.4, akan ditunjukkan bahwa R adalah relasi ekivalen.

1. Karena $a - a = 0$ adalah bilangan genap, maka $(a, a) \in R$.

2. Misalkan $(a, b) \in R$, akan ditunjukkan bahwa $(b, a) \in R$. Karena $a - b$ adalah bilangan genap, maka $b-a = -(a-b)$ juga bilangan genap sehingga $(b, a) \in R$.

3. Misalkan $(a, b) \in R$ dan $(b, c) \in R$, akan ditunjukkan bahwa $(a, c) \in R$. Karena $a-b$ dan $b-c$ adalah bilangan genap, maka $a-c = (a-b)+(b-c)$ juga bilangan genap sehingga $(a, c) \in R$.

Karena syarat 1 - 3 pada Definisi 2.1.4 terpenuhi, maka diperoleh bahwa R adalah relasi ekivalen.

Definisi 2.1.5. [8] *Misalkan A dan B adalah himpunan tak kosong. Relasi biner f dari A ke B merupakan suatu fungsi jika setiap elemen di A dihubungkan dengan tepat satu elemen di B. Jika f adalah fungsi dari A ke B maka ditulis*

$$f : A \to B$$

yang artinya f ***memetakan*** *A ke B.*

Himpunan A pada fungsi f di atas disebut **daerah asal** (*domain*), sedangkan himpunan B disebut **daerah kawan** (*codomain*). Jika $f(a) = b$, maka b disebut bayangan (*image*) dari a, sedangkan a disebut pra-bayangan (*pre-image*) dari b.

Himpunan yang berisi semua nilai pemetaan f disebut **daerah hasil** (*range*) dari f dan merupakan himpunan bagian dari B.

Contoh 2.1.12. Relasi f dimana $f = \{(1,u),(2,u),(3,w)\}$ dari $A = \{1,\ 2,\ 3\}$ ke $B = \{u,\ v,\ w\}$ adalah suatu fungsi, meskipun u merupakan bayangan dari dua elemen pada A.

Definisi 2.1.6. [8] *Fungsi f dikatakan* ***fungsi satu-satu*** *(one-one function) atau injektif jika tidak ada dua elemen pada himpunan A yang memiliki bayangan yang sama. Dengan kata lain, jika $x_1, x_2 \in A$ dan $x_1 \neq x_2$ maka $f(x_1) \neq f(x_2)$.*

Contoh 2.1.13. Misalkan $A = \{x \in \mathbb{R} : x \neq 1\}$ dan $f : A \to \mathbb{R}$ didefinisikan oleh $f(x) := x/(x-1)$. Akan ditunjukkan bahwa f adalah fungsi satu-satu. Misalkan $x_1, x_2 \in A$ dan $f(x_1) = f(x_2)$. Akan ditunjukkan $x_1 = x_2$. Perhatikan bahwa

$$\begin{aligned}
f(x_1) &= f(x_2) \\
\frac{x_1}{x_1-1} &= \frac{x_2}{x_2-1} \\
\frac{x_1}{x_1-1} - \frac{x_2}{x_2-1} &= 0 \\
\frac{x_1(x_2-1) - x_2(x_1-1)}{(x_1-1)(x_2-1)} &= 0 \\
\frac{x_1x_2 - x_1 - x_1x_2 + x_2}{x_1x_2 - x_1 - x_2 + 1} &= 0 \\
-x_1 + x_2 &= 0 \\
x_1 &= x_2.
\end{aligned}$$

Jadi terbukti bahwa f adalah fungsi satu-satu.

Definisi 2.1.7. [8] *Fungsi f dikatakan* ***fungsi pada*** *(onto function) atau surjektif jika setiap elemen pada himpunan B merupakan bayangan dari satu atau lebih* *elemen pada himpunan A. Dengan kata lain, untuk setiap $y \in B$ terdapat $x \in A$ sedemikian sehingga $f(x) = y$.*

Contoh 2.1.14. Misalkan $f : \mathbb{Z} \to \mathbb{Z}$ didefinisikan oleh $f(x) = x - 1$. Akan ditunjukkan bahwa f adalah fungsi pada. Ambil $y \in \mathbb{Z}$ sebarang. Pilih $x = y + 1$ dimana $x \in \mathbb{Z}$, sehingga

$$f(x) = f(y+1) = (y+1) - 1 = y.$$

Jadi terbukti bahwa f adalah fungsi pada.

Definisi 2.1.8. [8] *Fungsi f dikatakan **berkoresponden satu-satu** atau **bijektif** jika f adalah fungsi satu-satu dan pada.*

Contoh 2.1.15. Fungsi f pada Contoh 2.1.14 di atas juga merupakan fungsi satu-satu. Misalkan $x_1, x_2 \in \mathbb{Z}$ dan $f(x_1) = f(x_2)$. Akan ditunjukkan bahwa f adalah fungsi satu-satu. Perhatikan bahwa

$$\begin{aligned} x_1 - 1 &= x_2 - 1 \\ x_1 &= x_2. \end{aligned}$$

Jadi terbukti bahwa f juga merupakan fungsi satu-satu, sehingga f dikatakan berkorespondensi satu-satu.

2.2 Induksi Matematika

Induksi matematika adalah suatu metode pembuktiaan yang digunakan untuk memeriksa validasi suatu pernyataan mengenai bilangan asli. Himpunan bilangan asli dinotasikan dengan notasi khusus yaitu $\mathbb{N}$ dengan $\mathbb{N} = \{1, 2, 3, \ldots\}$.

Selanjutnya, bilangan-bilangan asli ini memiliki sifat terurut sempurna (*well-ordering property*) yaitu setiap himpunan bagian yang tak kosong dari $\mathbb{N}$ mempunyai elemen terkecil [10]. Sifat ini mengatakan bahwa jika S adalah himpunan bagian dari $\mathbb{N}$ dan S bukan himpunan kosong, maka terdapat suatu $m \in S$ sedemikian sehingga $m \leq k$ untuk setiap $k \in S$.

Teorema 2.2.9. [1] ***Prinsip Induksi Matematika.*** *Misalkan S adalah himpunan bagian dari $\mathbb{N}$ sedemikian sehingga*

- $1 \in S$
- *jika $k \in S$ maka $k+1 \in S$.*

Hal ini menunjukkan bahwa $S = N$.

Prinsip induksi matematika ini dapat dinyatakan sebagai berikut [1]:

Misalkan $\mathrm{P}(n)$ adalah suatu pernyataan mengenai bilangan asli n untuk setiap $n \in \mathbb{N}$. Asumsikan bahwa:

1. $\mathrm{P}(1)$ benar, dan
2. jika $\mathrm{P}(k)$ benar maka $\mathrm{P}(k+1)$ juga benar.

Maka $\mathrm{P}(n)$ benar untuk setiap $n \in \mathbb{N}$.

Contoh 2.2.16. Tunjukkan bahwa semua bilangan yang berbentuk $7^n - 2^n$ dapat dibagi oleh 5 untuk setiap $n \in \mathbb{N}$. Misalkan $\mathrm{P}(n)$: $7^n - 2^n$ dapat dibagi oleh 5 untuk setiap $n \in \mathbb{N}$.

1. Untuk $n = 1$ maka $7^1 - 2^1 = 5$. Karena 5 dapat dibagi oleh 5, maka $\mathrm{P}(1)$ bernilai benar.

2. Asumsikan $\mathrm{P}(k)$ benar untuk setiap $k \in \mathbb{N}$, yaitu

$$\mathrm{P}(k) : 7^k - 2^k \text{ dapat dibagi oleh 5 untuk setiap } k \in \mathbb{N}.$$

Akan ditunjukkan $\mathrm{P}(k+1)$ juga benar untuk setiap $k \in \mathbb{N}$.

Perhatikan bahwa:

$$\begin{aligned} 7^{k+1} - 2^{k+1} &= 7 \cdot 7^k - 7 \cdot 2^k + 7 \cdot 2^k - 2 \cdot 2^k \\ &= 7(7^k - 2^k) + 5 \cdot 2^k \\ &= 7 \cdot 5m + 5 \cdot 2^k \text{ untuk suatu } m \in \mathbb{N} \\ &= 5(7m + 2^k). \end{aligned}$$

Karena $m, k \in \mathbb{N}$ maka $7m + 2^k \in \mathbb{N}$ sehingga $7^{k+1} - 2^{k+1}$ dapat dibagi 5.

Dari 1 dan 2, terbukti bahwa $P(n) : 7^n - 2^n$ dapat dibagi oleh 5 untuk setiap $n \in \mathbb{N}$.

Teorema 2.2.10. [1] ***Prinsip Induksi Kuat.*** *Misalkan S adalah suatu himpunan bagian dari $\mathbb{N}$ sedemikian sehingga*

- $1 \in S$
- *jika* $\{1, 2, \ldots, k\} \subseteq S$ *maka* $k + 1 \in S$.

Hal ini menunjukkan bahwa $S = N$.

Prinsip induksi kuat ini dapat dinyatakan sebagai berikut:

Misalkan $\mathrm{P}(n)$ adalah suatu pernyataan mengenai bilangan asli n untuk setiap $n \in \mathbb{N}$. Asumsikan bahwa:

1. P(1) benar, dan
2. jika P(1), P(2), ... , P(k) benar maka P($k+1$) juga benar.

Maka P(n) benar untuk setiap $n \in \mathbb{N}$.

Contoh 2.2.17. Tunjukkan bahwa setiap bilangan asli n dengan $n \geq 2$ mempunyai satu atau lebih faktor prima. Misalkan P(n) : untuk setiap $n \geq 2$ dan $n \in \mathbb{N}$, n mempunyai satu atau lebih faktor prima.

1. Jika $n = 2$, maka 2 adalah bilangan prima. Hal ini menunjukkan bahwa 2 mempunyai satu buah faktor prima yaitu dirinya sendiri.
2. Asumsikan P(k) benar untuk $k \geq 2$ sehingga 2, 3, 4, $\ldots k$ mempunyai satu atau lebih faktor prima . Akan ditunjukkan bahwa P($k+1$) juga benar untuk setiap $k \geq 2$. Jika $k+1$ adalah bilangan prima, maka $k+1$ mempunyai satu atau lebih faktor prima. Jika $k+1$ bukan bilangan prima, maka terdapat suatu bilangan asli a sedemikian sehingga

   $$\frac{k+1}{a} = b \text{ atau } (k+1) = a \cdot b \text{ untuk suatu } 2 \leq a \leq b \leq k.$$

   Karena a dan b mempunyai satu atau lebih faktor prima, maka $a \cdot b$ juga mempunyai satu atau lebih faktor prima. Hal ini menunjukkan bahwa jika $k+1$ bukan bilangan prima maka $k+1$ juga mempunyai satu atau lebih faktor prima.

Dari 1 dan 2, terbukti bahwa untuk setiap $n \geq 2$ dan $n \in \mathbb{N}$, n mempunyai satu atau lebih faktor prima.

2.3 Representasi Bilangan Asli

Teorema 2.3.11. [10] *Misalkan b adalah suatu bilangan asli dengan $b \geq 2$, maka setiap bilangan asli n dapat ditulis secara tunggal dalam bentuk*

$$n = a_k b^k + a_{k-1} b^{k-1} + \ldots + a_1 b + a_0$$

dimana k adalah suatu bilangan bulat tak negatif, $a_j \in \mathbb{Z}$ dengan $0 \leq a_j \leq b-1$ untuk $j = 0, 1, \ldots, k$ dan koefisien awal $a_k \neq 0$.

Contoh 2.3.18. Misalkan $(a_k a_{k-1} a_{k-2} \ldots a_1 a_0)_b$ adalah bentuk lain dari $a_k b^k + a_{k-1} b^{k-1} + \ldots + a_1 b + a_0$, maka

$$(236)_7 = 2 \cdot 7^2 + 3 \cdot 7^1 + 6 \cdot 7^0 = 125$$

$$(100010011)_2 = 1 \cdot 2^7 + 1 \cdot 2^4 + 1 \cdot 2^1 + 1 \cdot 2^0 = 147$$

2.4 Bilangan *Happy*

Misalkan n adalah suatu bilangan asli dengan k digit, sedemikian sehingga n dapat dinyatakan dalam bentuk $n = a_1 a_2 \ldots a_k$ dengan $a_1 \neq 0$, $k \in \mathbb{N}$ dan a_k adalah digit ke-k dari n. Misalkan $T : \mathbb{N} \rightarrow \mathbb{N}$ merupakan suatu fungsi yang didefinisikan sebagai berikut

$$T(n) = {a_1}^2 + {a_2}^2 + \ldots + {a_k}^2.$$

Suatu bilangan *happy* n adalah suatu bilangan yang memenuhi kondisi $T^r(n) = 1$ untuk suatu $r \geq 0$ dimana

$$T^r(n) = \underbrace{T(T(\cdots T}_{sebanyak\ r}(n) \cdots)).$$

Sebaliknya, jika $T^r(n) \in \{4,\ 16,\ 37,\ 58,\ 89,\ 145,\ 42,\ 20\}$ untuk suatu $r \geq 0$ maka n dikatakan bilangan *unhappy*.

Contoh 2.4.19. Tunjukkan bahwa 694 adalah bilangan *happy*.

Perhatikan bahwa:

$T^1(694) = 6^2 + 9^2 + 4^2 = 133$

$T^2(694) = T^1(133) = 1^2 + 3^2 + 3^2 = 19$

$T^3(694) = T^2(133) = T^1(19) = 1^2 + 9^2 = 82$

$T^4(694) = T^3(133) = T^2(19) = T^1(82) = 8^2 + 2^2 = 68$

$T^5(694) = T^4(133) = T^3(19) = T^2(82) = T^1(68) = 6^2 + 8^2 = 100$

$T^6(694) = T^5(133) = T^4(19) = T^3(82) = T^2(68) = T^1(100) = 1^2 + 0^2 + 0^2 = 1.$

karena $T^6(694) = 1$, maka 694 adalah bilangan *happy*.

BAB III

BARISAN BILANGAN *HAPPY* TERURUT

3.1 Bilangan *Happy* pada Basis 10

Misalkan b adalah suatu bilangan asli dengan $b \geq 2$, maka setiap bilangan asli n dapat ditulis secara tunggal dalam bentuk

$$\begin{aligned} n &= a_k b^k + a_{k-1} b^{k-1} + \ldots + a_1 b + a_0 \\ &= \sum_{i=0}^{k} a_i b^i \end{aligned}$$

dimana k adalah suatu bilangan bulat tak negatif, $a_j \in \mathbb{Z}$ dengan $0 \leq a_j \leq b - 1$ untuk $j = 0, 1, \ldots, k$ dan koefisien awal $a_k \neq 0$.

Misalkan $T_{e,b} : \mathbb{N} \rightarrow \mathbb{N}$ dengan $e \in \mathbb{N}$ dan $b \in \mathbb{N}$. Jika $e \geq 2$, $b \geq 2$ dan $0 \leq a_i \leq b - 1$, maka fungsi $T_{e,b}$ didefinisikan sebagai berikut

$$T_{e,b}(n) = T_{e,b}(\sum_{i=0}^{k} a_i b^i) = \sum_{i=0}^{k} a_i^e.$$

Pada basis 10, fungsi $T_{e,b}$ didefinisikan sebagai berikut

$$T_{e,10}(n) = T_{e,10}(\sum_{i=0}^{k} a_i 10^i) = \sum_{i=0}^{k} a_i^e.$$

Pada kajian selanjutnya, akan dibahas mengenai fungsi $T_{2,10}$ dan ditulis dengan T.

Contoh 3.1.1. Misalkan $n = 20590$ maka $T_{2,10}(20590) = T(20590) = 2^2 + 0^2 + 5^2 + 9^2 = 110$.

Misalkan T^r adalah iterasi ke r dari fungsi T, yaitu

$$T^r(n) = \underbrace{T(T(\cdots T}_{sebanyak\ r}(n)\cdots)).$$

Jika $T(n)$ diiterasi sebanyak r kali dengan $r \geq 0$, maka terdapat dua himpunan penting dimana setiap anggotanya adalah $T^r(n)$ untuk suatu $r \geq 0$. Himpunan tersebut adalah $H = \{1\}$ dan $S = \{4,\ 16,\ 37,\ 58,\ 89,\ 145,\ 42,\ 20\}$.

Contoh 3.1.2. Misalkan $n = 35$ maka

$T^0(35) = 35$,

$T^1(35) = T(35) = 3^2 + 5^2 = 34$,

$T^2(35) = T(T(35)) = 3^2 + 4^2 = 25$,

$T^3(35) = T(T(T(35))) = 2^2 + 5^2 = 29$,

$T^4(35) = T(T(T(T(35)))) = 2^2 + 9^2 = 85$,

$T^5(35) = T(T(T(T(T(35))))) = 8^2 + 5^2 = 89$, dan seterusnya.

Jika $T(35)$ diiterasi sebanyak r kali dengan $r \geq 0$, maka terdapat sebuah himpunan $S = \{4,\ 16,\ 37,\ 58,\ 89,\ 145,\ 42,\ 20\}$.

Definisi 3.1.1. *Suatu bilangan asli n dikatakan titik tetap dari T jika $T(n) = n$.*

Lema 3.1.2. 1 *adalah titik tetap dari fungsi T.*

Bukti. Jika $n = 1$ maka $T(n) = T(1) = 1^2 = 1$. Karena $T(1) = 1$, maka 1 adalah titik tetap dari T. □

Lema 3.1.3. *Untuk setiap $s \in S$ dan $r \geq 0$, berlaku $T^r(s) \in S$.*

Bukti. Ambil $r \geq 0$ sebarang. Akan ditunjukkan untuk setiap $r \geq 0$, berlaku $T^r(s) \in S$.

Perhatikan matriks berikut

$$A = \begin{pmatrix} T^0(4) & T^0(16) & T^0(37) & T^0(58) & T^0(89) & T^0(145) & T^0(42) & T^0(20) \\ T^1(4) & T^1(16) & T^1(37) & T^1(58) & T^1(89) & T^1(145) & T^1(42) & T^1(20) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ T^r(4) & T^r(16) & T^r(37) & T^r(58) & T^r(89) & T^r(145) & T^r(42) & T^r(20) \end{pmatrix}$$

$$= \begin{pmatrix} 4 & 16 & 37 & 58 & 89 & 145 & 42 & 20 \\ 16 & 37 & 58 & 89 & 145 & 42 & 20 & 4 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ T^r(4) & T^r(16) & T^r(37) & T^r(58) & T^r(89) & T^r(145) & T^r(42) & T^r(20) \end{pmatrix}$$

Dari matriks A di atas, terlihat bahwa $T^r(4) \in S$, $T^r(16) \in S$, $T^r(37) \in S$, $T^r(58) \in S$, $T^r(89) \in S$, $T^r(145) \in S$, $T^r(42) \in S$ dan $T^r(20) \in S$ untuk setiap $r \geq 0$. Jadi terbukti bahwa untuk setiap $s \in S$ dan $r \geq 0$, berlaku $T^r(s) \in S$. □

Teorema 3.1.4. *Untuk setiap $n \in \mathbb{N}$, terdapat suatu $r \geq 0$ sedemikian sehingga $T^r(n) \in H$ atau $T^r(n) \in S$.*

Untuk membuktikan Teorema 3.1.4 ini, digunakan lema berikut

Lema 3.1.5. *Untuk setiap $k \geq 4$, berlaku $81k < 10^{k-1}$.*

Bukti. Pembuktian Lema 3.1.5 ini dilakukan dengan menggunakan induksi terhadap k untuk setiap $k \geq 4$. Akan ditunjukkan bahwa untuk setiap $k \geq 4$, berlaku $81k < 10^{k-1}$. Misalkan $\mathrm{P}(n)$: $81n < 10^{n-1}$ untuk setiap $n \geq 4$.

1. Jika $n = 4$ maka

$$81 \cdot 4 = 324 < 10^{4-1} = 1000.$$

Karena $81 \cdot 4 < 1000$, maka P(4) berlaku.

2. Asumsikan bahwa P(k) berlaku untuk setiap $k \geq 4$ yaitu $81k < 10^{k-1}$. Akan ditunjukkan bahwa P($k+1$) juga berlaku, yaitu $81(k+1) < 10^k$.

   Perhatikan bahwa

$$\begin{aligned} 81k + 81 &< 10^{k-1} + 81 \\ 81(k+1) &< 10^{k-1} + 8.1 \cdot 10 < 10^{k-1} + 10 \cdot 10 \\ 81(k+1) &< \frac{1}{10}(10^k + 10^3) \end{aligned}$$

Klaim. *Untuk setiap $k \geq 4$, berlaku $\frac{1}{10}(10^k + 10^3) < 10^k$.*

Bukti klaim. Misalkan Q(n) : $\frac{1}{10}(10^n + 10^3) < 10^n$ untuk setiap $n \geq 4$.

(a) Jika $n = 4$ maka

$$\frac{1}{10}(10^4 + 10^3) = 1100 < 10^4 = 10000.$$

Karena $\frac{1}{10}(10^4 + 10^3) < 10000$ maka Q(4) berlaku.

(b) Asumsikan bahwa Q(k) berlaku untuk setiap $k \geq 4$ yaitu

$$\frac{1}{10}(10^k + 10^3) < 10^k.$$

Akan ditunjukkan bahwa Q($k+1$) juga berlaku yaitu

$$\frac{1}{10}(10^{k+1} + 10^3) < 10^{k+1}.$$

Perhatikan bahwa

$$\begin{aligned} 10.\frac{1}{10}(10^k + 10^3) = \frac{1}{10}(10^{k+1} + 10^4) &< 10.10^k \\ \frac{1}{10}(10^{k+1} + 10^3) < \frac{1}{10}(10^{k+1} + 10^4) &< 10^{k+1} \end{aligned}$$

Karena $\frac{1}{10}(10^{k+1}+10^3) < 10^{k+1}$ maka Q$(k+1)$ juga berlaku. Dari (a) dan (b), terbukti bahwa untuk setiap $k \geq 4$ maka $\frac{1}{10}(10^k+10^3) < 10^k$.

Karena $81(k+1) < \frac{1}{10}(10^k+10^3) < 10^k$ maka diperoleh $81(k+1) < 10^k$, sehingga P$(k+1)$ juga berlaku. Dari 1 dan 2, terbukti bahwa untuk setiap $k \geq 4$, berlaku $81k < 10^{k-1}$. $\square$

Bukti Teorema 3.1.4. Ambil $n \in \mathbb{N}$ sebarang. Misalkan $n = a_1a_2a_3...a_k$ dengan $a_1 \neq 0$, $k \in \mathbb{N}$ dan a_k adalah digit ke k dari n. Perhatikan bahwa

$$1 \leq a_1 \leq 9$$

$$0 \leq a_k \leq 9 \quad \forall\, k \geq 2, k \in \mathbb{N}.$$

Nilai maksimum dari $T(n)$ diperoleh jika $a_k = 9$ untuk setiap $k \in \mathbb{N}$, sehingga

$$T(n) \leq 9^2 \cdot k = 81 \cdot k. \tag{3.1.1}$$

Karena $a_1 \neq 0$, maka

$$10^{k-1} \leq n \leq 10^k. \tag{3.1.2}$$

Secara jelas terlihat bahwa untuk setiap $k \leq 3$, berlaku $81k > 10^{k-1}$. Selanjutnya, berdasarkan Lema 3.1.5 diperoleh bahwa untuk setiap $k \geq 4$, berlaku $81k < 10^{k-1}$. Dari (3.1.1) , (3.1.2) dan Lema 3.1.5 diperoleh

$$T(n) \leq 81 \cdot k < 10^{k-1} \leq n$$

sehingga untuk setiap $k \geq 4$, berlaku $T(n) < n$ sedangkan untuk $k \leq 3$ terdapat $T(n)$ yang lebih besar dari n. Nilai n terbesar untuk $k \leq 3$ adalah 999, sehingga $T(999) = 9^2 + 9^2 + 9^2 = 243$. Jika $k \geq 4$ maka $T(n) < n$, sehingga untuk sebarang $n > 243$ terdapat suatu r sedemikian sehingga $T^r(n)$ berada pada interval

[1,243]. Jadi, untuk membuktikan bahwa teorema ini berlaku untuk setiap $n \in \mathbb{N}$, cukup ditunjukkan $T^r(n) \in H$ atau $T^r(n) \in S$ pada interval [1,243]. Dengan menggunakan Program Borland C++ (Lampiran a), diperoleh bahwa untuk setiap $n \in [1, 243] \cap \mathbb{Z}$, berlaku $T^r(n) \in H$ atau $T^r(n) \in S$. Jadi terbukti bahwa untuk setiap $n \in \mathbb{N}$ terdapat suatu $r \geq 0$ sedemikian sehingga $T^r(n) \in H$ atau $T^r(n) \in S$. $\square$

Definisi 3.1.6. *Suatu bilangan asli n dikatakan bilangan happy jika $T^r(n) \in H$ untuk suatu $r \geq 0$.*

Definisi 3.1.7. *Suatu bilangan asli n dikatakan bilangan unhappy jika $T^r(n) \in S$ untuk suatu $r \geq 0$.*

Akibat 3.1.8. *Untuk setiap $n \in \mathbb{N}$, $T(n) = n$ jika dan hanya jika $n = 1$.*

Bukti.

($\Rightarrow$) Ambil $n \in \mathbb{N}$ sebarang. Jika $T(n) = n$ maka $T^2(n) = n$, $T^3(n) = n$, dan seterusnya sehingga $T^r(n) = n$ untuk suatu $r \geq 0$. Dari Hasil yang diperoleh pada Teorema 3.1.4, nilai $T(n) = n$ hanya dipenuhi oleh $n = 1$. Jadi terbukti, jika $T(n) = n$ maka $n = 1$.

($\Leftarrow$) Jika $n = 1$ maka $T(1) = 1^2 = 1$. Jadi terbukti, jika $n = 1$ maka $T(n) = n$. Jadi terbukti bahwa untuk setiap $n \in \mathbb{N}$, $T(n) = n$ jika dan hanya jika $n = 1$. $\square$

3.2 Barisan Bilangan *Happy* Terurut

Pada subbab ini, akan dikaji eksistensi dari barisan bilangan *happy* terurut dengan panjang sebarang. Untuk mengkaji eksistensi dari barisan tersebut, maka digunakanlah lema berikut

Lema 3.2.9. *Untuk setiap $y \in \mathbb{N}$, terdapat takhingga banyaknya bilangan asli n sedemikian sehingga $T(n) = y$.*

Bukti. Ambil $y \in \mathbb{N}$ sebarang. Pilih $n = \sum_{i=0}^{y-1} 10^i$ dimana n memiliki digit sebanyak y, sehingga

$$\begin{aligned} T(n) &= T(\sum_{i=0}^{y-1} 10^i) \\ &= T(\underbrace{11111\ldots1}_{sebanyak\ y}) \\ &= \underbrace{1^2+1^2+1^2+\ldots+1^2}_{sebanyak\ y} \\ &= y. \end{aligned}$$

Karena y dipilih sebarang, maka untuk setiap $y \in \mathbb{N}$ terdapat suatu bilangan asli n sedemikian sehingga $T(n) = y$. Misalkan $A = \{n | n \in \mathbb{N} \text{ dan } T(n) = y\}$, maka A bukan merupakan himpunan kosong dan memiliki takhingga banyaknya anggota.

□

Lema 3.2.10. *Jika x, y, dan z adalah bilangan asli sedemikian sehingga $10^z > y$ maka $T(10^z + y) = T(x) + T(y)$.*

Bukti. Misalkan x, y, dan z adalah bilangan asli sedemikian sehingga $10^z > y$. Perhatikan dua buah kemungkinan berikut:

- Misalkan $n = 10^z x + y$, maka digit-digit dari n adalah digit-digit dari x dan y sehingga

$$\begin{aligned} T(10^z x + y) &= T(xy) \\ &= T(x) + T(y) \end{aligned}$$

- Misalkan $n = 10^z x + y$, maka digit-digit dari n adalah digit-digit dari x dan y serta 0 diantara keduanya sehingga

$$\begin{aligned} T(10^z x + y) &= T(x\underbrace{0000\ldots0}_{sebanyak\ z}y) \\ &= T(x) + \underbrace{T(0) + T(0) + T(0) + \ldots + T(0)}_{sebanyak\ z} + T(y) \\ &= T(x) + T(y) \end{aligned}$$

Jadi terbukti bahwa jika x, y, dan z adalah bilangan asli sedemikian sehingga $10^z > y$ maka $T(10^z x + y) = T(x) + T(y)$. □

Misalkan $D = H \cup S = \{1,\ 4,\ 16,\ 37,\ 58,\ 89,\ 145,\ 42,\ 20\}$.

Lema 3.2.11. *Untuk setiap $x \in \mathbb{N}$ terdapat $r_x \geq 0$ sedemikian sehingga $T^r(x) \in D$ untuk setiap $r \geq r_x$.*

Lema ini merupakan bentuk lain dari Teorema 3.1.4.

Lema 3.2.12. *Jika $T^r(y) = x$ dimana $r \geq 0$ dan x adalah bilangan happy maka y juga bilangan happy.*

Bukti. Misalkan x adalah bilangan *happy* dan $T^r(y) = x$. Karena x adalah bilangan *happy*, maka terdapat $r' \geq 0$ sedemikian sehingga $T^{r'}(x) = 1$. Selanjutnya,

$$T^{r+r'}(y) = T^{r'}(x) = 1.$$

Hal ini menunjukkan bahwa y juga bilangan *happy*. Jadi terbukti bahwa jika $T^r(y) = x$ dimana $r \geq 0$ dan x adalah bilangan *happy* maka y juga bilangan *happy*. □

Lema 3.2.13. *Terdapat suatu bilangan asli l sedemikian sehingga l+u adalah happy untuk setiap $u \in D$.*

Bukti. Misalkan D' adalah himpunan bagian dari D dimana $D' = \{1, 4, 16, 20, 37, 42, 58\}$. Selanjutnya, akan ditentukan nilai l' sedemikian sehingga $l' + u'$ adalah *happy* untuk setiap $u' \in D'$. Dengan menggunakan Program Borland C++, diperoleh nilai l' sedemikian sehingga $l' + u'$ adalah *happy* untuk setiap $u' \in D'$ (Lampiran b). Adapun 20 nilai l' pertama yang diperoleh adalah sebagai berikut

8880958, 18809958, 47809958, 48709958, 74809958, 78409958, 80880958, 81809958, 84709958, 87409958, 88080958, 88109958, 88800958, 108809958, 125099958, 125990958, 129590958, 129950958, 152099958, 152990958.

Perhatikan bahwa, semua nilai l' di atas berakhir dengan digit 958. Misalkan $l = 10^4x + 958$ untuk setiap $x \leq 4 \cdot 10^9$, maka

$$l + u = 10^4x + 958 + u = 10^4x + (u + 958).$$

Karena $u + 958 < 10^4 \ \forall \ u \in D$, maka berdasarkan Lema 3.2.10 diperoleh:

$$T(l + u) = T(10^4x + (u + 958)) = T(x) + T(u + 958).$$

Misalkan $E = \{T(u + 958) : u \in D\} = \{1, 11, 38, 66, 121, 146, 187, 194\}$. Selanjutnya, akan dicari nilai y sedemikian sehingga $y + v$ adalah *happy* untuk setiap $v \in E$. Jika terdapat suatu $y \in \mathbb{N}$ maka berdasarkan Lema 3.2.9, terdapat nilai x sedemikian sehingga $T(x) = y$. Dengan menggunakan Program Borland C++ (Lampiran c), diperoleh nilai y sedemikian sehingga $y + v$ adalah *happy* untuk setiap $v \in E$. Adapun 20 nilai y pertama yang diperoleh adalah sebagai berikut:

18888556, 47888556, 48788556, 48878556, 48887556, 74888556, 78488556, 78848556, 78884556, 81888556, 84788556, 84878556, 84887556, 87488556, 87848556, 87884556, 88188556, 88478556, 88487556, 88748556.

Perhatikan dua buah nilai y berikut:

- Misalkan $y = 18888556 = 233192 \text{ x } 81 + 4$ maka $x = (\sum_{r=1}^{233192} 9 \text{ x } 10^r) + 2$.

  Sehingga diperoleh,

$$\begin{aligned} l &= 10^4 x + 958 \\ &= 10^4((\sum_{r=1}^{233192} 9 \cdot 10^r) + 2) + 958 \\ &= (\sum_{r=1}^{233192} 9 \cdot 10^{r+4}) + 20958. \end{aligned}$$

  Karena $D = \{1,\ 4,\ 16,\ 37,\ 58,\ 89,\ 145,\ 42,\ 20\}$ maka

$$\begin{aligned} T(l+u) &= T(10^4((\sum_{r=1}^{233192} 9 \cdot 10^r) + 2) + (958 + u)) \\ &= T((\sum_{r=1}^{233192} 9 \cdot 10^r) + 2) + T(958 + u) \\ &= T(10 \cdot (\sum_{r=1}^{233192} 9 \cdot 10^{r-1}) + 2) + T(958 + u) \\ &= T(\sum_{r=1}^{233192} 9 \cdot 10^{r-1}) + T(2) + T(958 + u) \\ &= T(\underbrace{9999\ldots999}_{sebanyak\ 233192}) + T(2) + T(958 + u) \\ &= 9^2 \cdot 233192 + 2^2 + T(958 + u) \\ &= 18888556 + T(958 + u). \end{aligned}$$

  Dengan menggunakan Program Borland C++ (Lampiran d), diperoleh bahwa $18888556 + T(958 + u)$ adalah *happy* untuk setiap $u \in D$. Jadi terbukti bahwa terdapat suatu $l \in \mathbb{N}$ dengan

$$l = (\sum_{r=1}^{233192} 9 \cdot 10^{r+4}) + 20958$$

  sedemikian sehingga $l + u$ adalah *happy* untuk setiap $u \in D$.

- Misalkan $y = 88188556 = 1088747 \times 81 + 49$ maka $x = (\sum_{r=1}^{1088747} 9 \times 10^r) + 7$.

Sehingga diperoleh,

$$\begin{aligned} l &= 10^4 x + 958 \\ &= 10^4((\sum_{r=1}^{1088747} 9 \cdot 10^r) + 7) + 958 \\ &= (\sum_{r=1}^{1088747} 9 \cdot 10^{r+4}) + 70958. \end{aligned}$$

Karena $D = \{1, 4, 16, 37, 58, 89, 145, 42, 20\}$ maka

$$\begin{aligned} T(l+u) &= T(10^4((\sum_{r=1}^{1088747} 9 \cdot 10^r) + 7) + (958 + u)) \\ &= T((\sum_{r=1}^{1088747} 9 \cdot 10^r) + 7) + T(958 + u) \\ &= T(10 \cdot (\sum_{r=1}^{1088747} 9 \cdot 10^{r-1}) + 7) + T(958 + u) \\ &= T(\sum_{r=1}^{1088747} 9 \cdot 10^{r-1}) + T(7) + T(958 + u) \\ &= T(\underbrace{9999\ldots999}_{sebanyak\ 1088747}) + T(7) + T(958 + u) \\ &= 9^2 \cdot 1088747 + 7^2 + T(958 + u) \\ &= 88188556 + T(958 + u). \end{aligned}$$

Dengan menggunakan Program Borland C++ (Lampiran d), diperoleh bahwa $88188556 + T(958 + u)$ adalah *happy* untuk setiap $u \in D$. Jadi terbukti bahwa terdapat suatu $l \in \mathbb{N}$ dengan

$$l = (\sum_{r=1}^{1088747} 9 \cdot 10^{r+4}) + 70958$$

sedemikian sehingga $l + u$ adalah *happy* untuk setiap $u \in D$. $\square$

Teorema 3.2.14. *Terdapat barisan bilangan happy terurut dengan panjang sebarang yaitu untuk setiap $m \in \mathbb{N}$ terdapat suatu bilangan asli l_0 sedemikian sehingga* *setiap anggota dari barisan terbatas* $l_0+1, l_0+2, l_0+3, \dots, l_0+m$ *adalah bilangan happy.*

Bukti. Akan ditunjukkan untuk setiap $m \in \mathbb{N}$, terdapat suatu barisan bilangan *happy* terurut dengan panjang m. Berdasarkan Lema 3.2.11, terdapat suatu $r \geq 0$ sedemikian sehingga $T^r(i) \in D$ untuk setiap $i = 1, 2, \dots, m$. Himpunan $\{T^j(i) : i = 1, 2, \dots, m; j = 0, 1, \dots, r\}$ adalah himpunan terbatas, sehingga terdapat suatu z sedemikian sehingga

$$10^z > T^j(i) \text{ untuk setiap } i = 1, 2, \dots, m \text{ dan } j = 0, 1, \dots, r$$

Selanjutnya, l_0, l_1, l_2, $\dots$, l_r didefinisikan sebagai berikut:

- diberikan suatu nilai l_r
- untuk setiap $0 \leq j \leq r-1$, nilai l_j bergantung pada l_{j+1}

Misalkan $l_r = l$ dimana l adalah suatu nilai yang eksistensinya dijamin oleh Lema 3.2.13. Untuk setiap $0 \leq j \leq r-1$, nilai l_j yang bergantung pada l_{j+1} didefinisikan sebagai berikut:

Berdasarkan Lema 3.2.9, terdapat suatu k_{j+1} sedemikian sehingga $T(k_{j+1}) = l_{j+1}$. Misalkan $l_j = 10^z k_{j+1}$. Perhatikan bahwa:

- Untuk setiap $0 \leq j \leq r-1$ berlaku $T(l_j) = T(10^z k_{j+1}) = T(k_{j+1}) = l_{j+1}$.
- Jika $0 \leq j \leq r-1$ dan $y < 10^z$, maka berdasarkan Lema 3.2.10 diperoleh

$$T(l_j + y) = T(10^z k_{j+1} + y) = T(k_{j+1}) + T(y) = l_{j+1} + T(y).$$

Oleh karena itu, untuk setiap $i = 1, 2, \ldots, m$ dapat dilihat bahwa

$$\begin{aligned} T^r(l_0 + i) &= T^{r-1}(T(l_0 + i)) = T^{r-1}(l_1 + T(i)) \\ &= T^{r-2}(T^2(l_0 + i)) = T^{r-2}(T(l_1) + T^2(i)) = T^{r-2}(l_2 + T^2(i)) \\ &= T^{r-3}(T^3(l_0 + i)) = T^{r-3}(T(l_2) + T^3(i)) = T^{r-3}(l_3 + T^3(i)) \\ &\vdots \\ &= T^{r-r}(T^r(l_0 + i)) = \ldots = T^{r-r}(l_r + T^r(i)) = l_r + T^r(i). \end{aligned}$$

Karena $l_r = l$ maka $T^r(l_0+i) = l_r+T^r(i) = l+T^r(i)$ untuk setiap $i = 1, 2, \ldots, m$. Selanjutnya, karena telah dipilih r sedemikian sehingga $T^r(i) \in D$ untuk setiap $i = 1, 2, ..., m$ dan l sedemikian sehingga $l + u$ adalah bilangan *happy* untuk setiap $u \in D$, maka $T^r(l_0 + i) = l + T^r(i)$ adalah bilangan *happy* untuk setiap $i = 1, 2, \ldots, m$. Berdasarkan Lema 3.2.12, $l_0 + i$ adalah bilangan *happy* untuk setiap $i = 1, 2, \ldots, m$. Jadi terbukti bahwa terdapat barisan bilangan *happy* terurut dengan panjang sebarang. □

3.3 Bilangan Terkecil dari Barisan Bilangan *Happy* Terurut dengan Panjang Enam

Seperti yang telah dikemukakan pada Bab I bahwa bilangan terkecil dari barisan bilangan *happy* terurut dengan panjang 2, 3, 4 dan 5 telah diketahui dan dapat dilihat pada Tabel 1.1.1. Seorang matematikawan bernama Jud Mc Cranie telah menghitung bahwa hingga bilangan 1×10^{20} tidak ditemukan suatu barisan bilangan *happy* terurut dengan panjang enam. Berdasarkan Teorema 3.2.14 telah terbukti adanya barisan bilangan *happy* terurut dengan panjang m, sehingga dapat dijamin adanya bilangan terkecil dari barisan-barisan tersebut.

Dengan menggunakan Program *Maple* 14 (Lampiran e dan f) diperoleh bahwa bilangan terkecil dari barisan bilangan *happy* terurut dengan panjang enam adalah

7899999999999959999999996,

sehingga

7899999999999959999999996, 7899999999999959999999997,

7899999999999959999999998, 7899999999999959999999999,

7899999999999960000000000, 7899999999999960000000001

merupakan barisan terkecil dari barisan bilangan *happy* terurut dengan panjang enam.

BAB IV

PENUTUP

4.1 Kesimpulan

Suatu bilangan *happy* n adalah suatu bilangan yang memenuhi kondisi $T^r(n) = 1$ untuk suatu $r \geq 0$, sedangkan jika $T^r(n) \in \{4, 16, 37, 58, 89, 145, 42, 20\}$ untuk suatu $r \geq 0$ maka n dikatakan bilangan *unhappy*. Pada barisan bilangan *happy* ini, terdapat barisan bilangan *happy* terurut dengan panjang sebarang yaitu untuk setiap $m \in \mathbb{N}$ terdapat sebuah bilangan asli l_0 sedemikian sehingga setiap anggota dari barisan terbatas $l_0+1, l_0+2, l_0+3, \ldots, l_0+m$ adalah bilangan *happy*.

Bilangan terkecil dari barisan bilangan *happy* terurut ini dapat ditentukan dengan menggunakan berbagai *software*. Dengan menggunakan *Maple* 14 diperoleh bahwa bilangan terkecil dari barisan bilangan *happy* terurut dengan panjang enam adalah 78999999999999959999999996.

4.2 Saran

Untuk penelitian selanjutnya, penulis menyarankan untuk mengkaji eksistensi barisan bilangan *happy* terurut pada basis lainnya.

DAFTAR PUSTAKA

[1] Bartle, R. G. dan Donald R. S. 2000. *Introduction to Real Analysis.* John Wiley and Sons, New York.

[2] Elsedi, E. dan Samir S. 2000. On Happy Numbers. *Rocky Mountain J. Math.* **30** : 565-570

[3] Grundman, H. G. dan Elizabeth A. T. 2007. Sequences of Generalized Happy Numbers with Small Bases. *Journal of Integer Sequences*, **10** : Article 07.1.8

[4] Grundman, H. G. dan Elizabeth A. T. 2003. Heights of Happy Numbers and Cubic Happy Numbers. *Fibonacci Quart*, **41** : 301-306

[5] Grundman, H. G. dan Elizabeth A. T. 2001. Generalized Happy Numbers. *Fibonacci Quart*, **39** : 462-466

[6] Guy, R. K. 1994. *Unsolved Problems in Number Theory.* Springer Velag, New York.

[7] Honsberger, R. 1970. *Ingenuity in Mathematics.* Mathematical Association of America, Washington.

[8] Munir, R. 2003. *Matematika Diskrit.* Informatika Bandung, Bandung.

[9] Mutter, S. A. 2010. Happy Number: An Exploration of An Iterated Function in Different Bases. *Senior Project*, tidak diterbitkan.

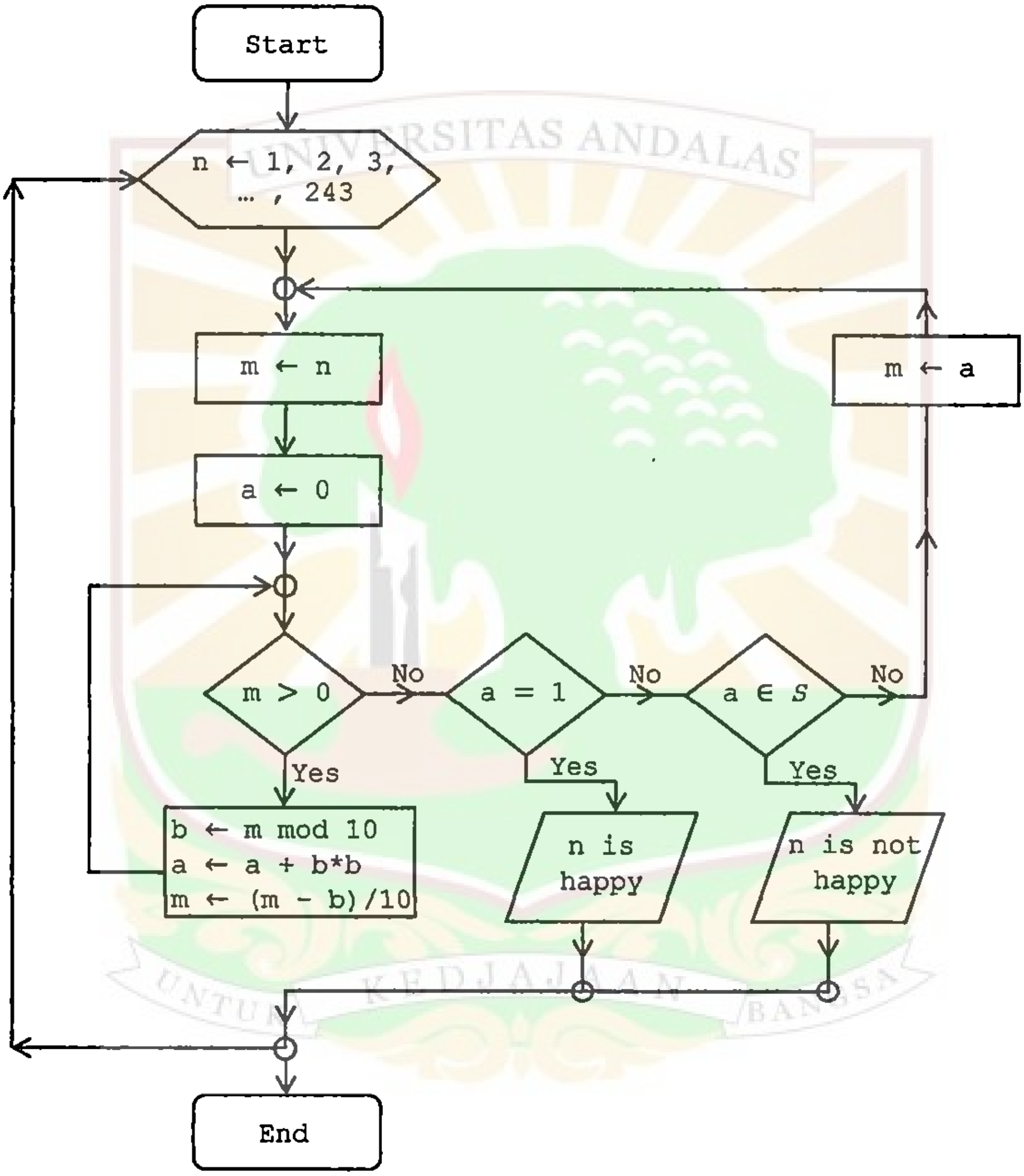
[10] Rosen, K. H. 2005. *Elementary Number Theory and Its Applications.* Addison Wesley Publishing Company, New York.

[11] Styer, R. 2010. Smallest Examples of Strings of Consecutive Happy Numbers. *Journal of Integer Sequences.* **13** : Article 10.6.3

LAMPIRAN

Lampiran a. Program untuk menentukan apakah untuk setiap $n \in \mathbb{N}$ dimana $1 \leq n \leq 243$, berlaku $T^r(n) \in H$ atau $T^r(n) \in S$ untuk suatu $r \geq 0$ dan $r \in \mathbb{Z}$. Adapun himpunan $H = \{1\}$ dan $S = \{4, 16, 37, 58, 89, 145, 42, 30\}$.

Flowchart programnya adalah:



Kode programnya adalah:

```
#include <iostream>
int main ()
{

   unsigned long int m,n;
   int a,b;

   for (n=1; n<244; n++) {
      m = n;
      while (m > 0) {
         a = 0;
         while (m > 0) {
            b = m % 10;
            a = a + b*b;
            m = (m - b)/10;
         }

      if (a==1) {
         cout << n << " is happy" << endl;
      }  else if (a == 4 || a==16 || a==37 || a==58 ||
         a==89 || a==145 || a==42 || a==20) {
         cout << n << " is not happy" << endl;
      }  else {
         m = a;
      }
      }
   }
   return 0;

}
```

Output programnya adalah:

```
1 is happy
2 is not happy
3 is not happy
4 is not happy
5 is not happy
6 is not happy
7 is happy
8 is not happy
9 is not happy
10 is happy
11 is not happy
12 is not happy
```

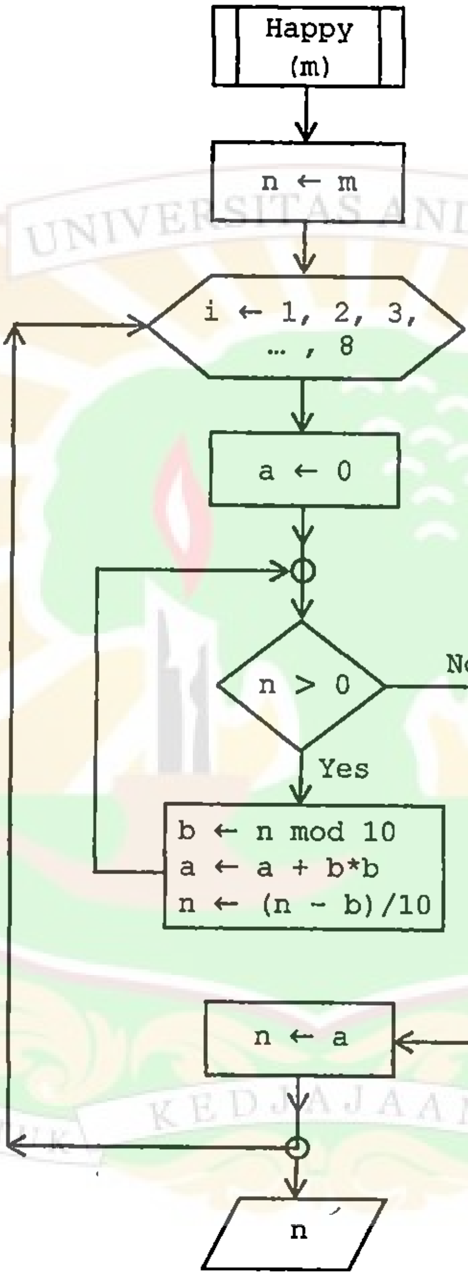
13 is happy
14 is not happy
15 is not happy
16 is not happy
17 is not happy
18 is not happy
19 is happy
20 is not happy
21 is not happy
22 is not happy
23 is happy
24 is not happy
25 is not happy
26 is not happy
27 is not happy
28 is happy
29 is not happy
30 is not happy
31 is happy
32 is happy
33 is not happy
34 is not happy
35 is not happy
36 is not happy
37 is not happy
38 is not happy
39 is not happy
40 is not happy
41 is not happy
42 is not happy
43 is not happy
44 is happy
45 is not happy
46 is not happy
47 is not happy
48 is not happy
49 is happy
50 is not happy
51 is not happy
52 is not happy
53 is not happy
54 is not happy
55 is not happy
56 is not happy
57 is not happy
58 is not happy
59 is not happy
60 is not happy
61 is not happy
62 is not happy
63 is not happy
64 is not happy
65 is not happy
66 is not happy
67 is not happy
68 is happy
69 is not happy
70 is happy
71 is not happy
72 is not happy
73 is not happy
74 is not happy
75 is not happy
76 is not happy
77 is not happy
78 is not happy
79 is happy
80 is not happy
81 is not happy
82 is happy
83 is not happy
84 is not happy
85 is not happy
86 is happy
87 is not happy
88 is not happy
89 is not happy
90 is not happy
91 is happy
92 is not happy
93 is not happy
94 is happy
95 is not happy
96 is not happy
97 is happy
98 is not happy

```
99 is not happy
100 is happy
101 is not happy
102 is not happy
103 is happy
104 is not happy
105 is not happy
106 is not happy
107 is not happy
108 is not happy
109 is happy
110 is not happy
111 is not happy
112 is not happy
113 is not happy
114 is not happy
115 is not happy
116 is not happy
117 is not happy
118 is not happy
119 is not happy
120 is not happy
121 is not happy
122 is not happy
123 is not happy
124 is not happy
125 is not happy
126 is not happy
127 is not happy
128 is not happy
129 is happy
130 is happy
131 is not happy
132 is not happy
133 is happy
134 is not happy
135 is not happy
136 is not happy
137 is not happy
138 is not happy
139 is happy
140 is not happy
141 is not happy
142 is not happy
143 is not happy
144 is not happy
145 is not happy
146 is not happy
147 is not happy
148 is not happy
149 is not happy
150 is not happy
151 is not happy
152 is not happy
153 is not happy
154 is not happy
155 is not happy
156 is not happy
157 is not happy
158 is not happy
159 is not happy
160 is not happy
161 is not happy
162 is not happy
163 is not happy
164 is not happy
165 is not happy
166 is not happy
167 is happy
168 is not happy
169 is not happy
170 is not happy
171 is not happy
172 is not happy
173 is not happy
174 is not happy
175 is not happy
176 is happy
177 is not happy
178 is not happy
179 is not happy
180 is not happy
181 is not happy
182 is not happy
183 is not happy
184 is not happy
```

```
185 is not happy
186 is not happy
187 is not happy
188 is happy
189 is not happy
190 is happy
191 is not happy
192 is happy
193 is happy
194 is not happy
195 is not happy
196 is not happy
197 is not happy
198 is not happy
199 is not happy
200 is not happy
201 is not happy
202 is not happy
203 is happy
204 is not happy
205 is not happy
206 is not happy
207 is not happy
208 is happy
209 is not happy
210 is not happy
211 is not happy
212 is not happy
213 is not happy
214 is not happy
215 is not happy
216 is not happy
217 is not happy
218 is not happy
219 is happy
220 is not happy
221 is not happy
222 is not happy
223 is not happy
224 is not happy
225 is not happy
226 is happy
227 is not happy
228 is not happy
229 is not happy
230 is happy
231 is not happy
232 is not happy
233 is not happy
234 is not happy
235 is not happy
236 is happy
237 is not happy
238 is not happy
239 is happy
240 is not happy
241 is not happy
242 is not happy
243 is not happy
```

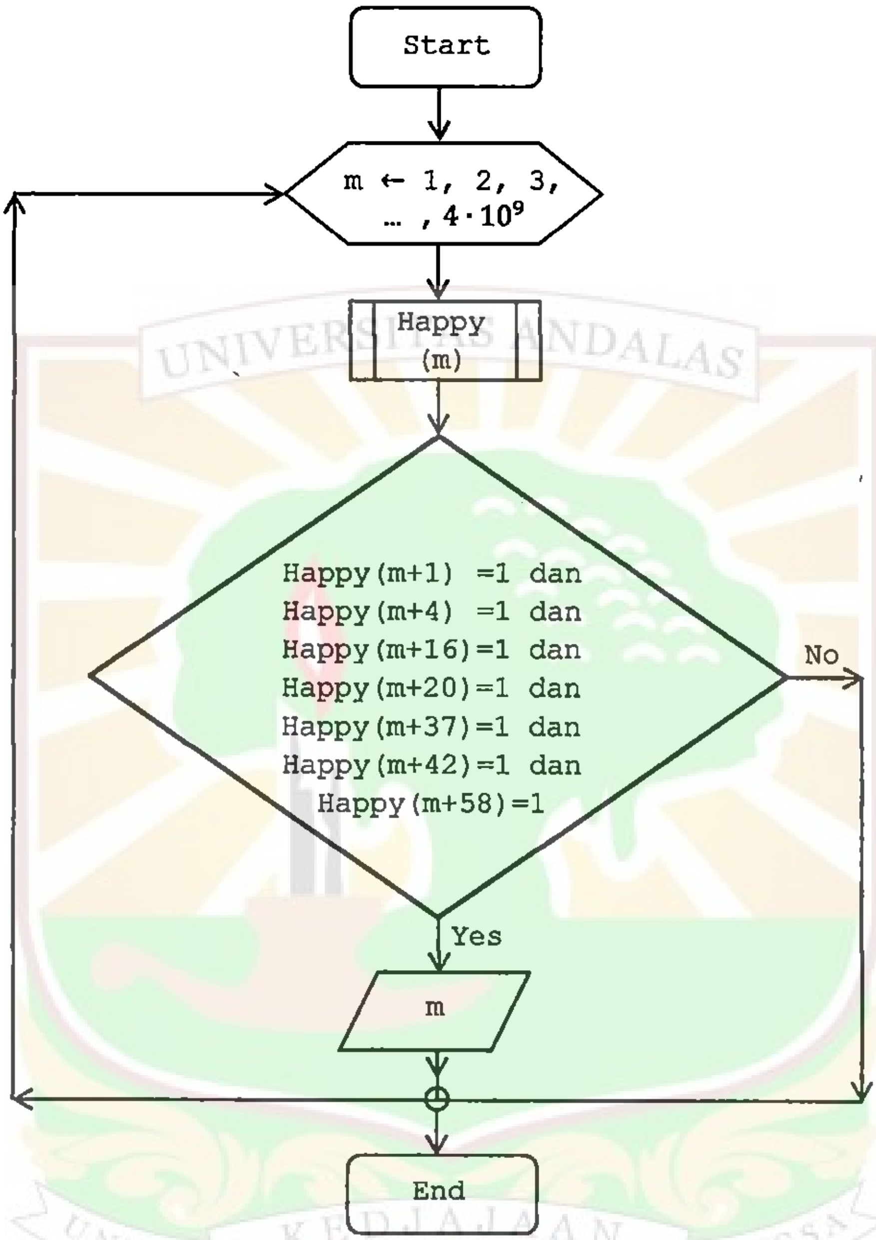
Lampiran b. Program untuk mencari nilai l' sedemikian sehingga $l' + u'$ adalah bilangan *happy* untuk setiap $u' \in \{1, 4, 16, 20, 37, 42, 58\}$.

Flowchart programnya adalah:



Keterangan: Fungsi Happy(m) ini akan digunakan pada program Lampiran c dan d.

PROGRAM UTAMA



Kode programnya adalah:

```
#include <iostream>

int happy (unsigned long int m)
{

    int i;
    int a,b;
    unsigned long int n;

    n = m;
    for (i=1 ; i<9 ; i++) {
        a = 0;
        while (n > 0) {
            b = n % 10;
            a = a + b*b;
            n = (n - b)/10;
        }
        n = a;
    }

    return n;
}

int main ()
{
    unsigned long int m;

    for (m=1 ; m<4000000000 ; m++) {
        if (happy(m+1) ==1 && happy(m+4) ==1 &&
            happy(m+16)==1 && happy(m+20)==1 &&
            happy(m+37)==1 && happy(m+42)==1 &&
            happy(m+58)==1 ) {
            cout <<  m << endl;
        }
    }

    return 0;
}
```

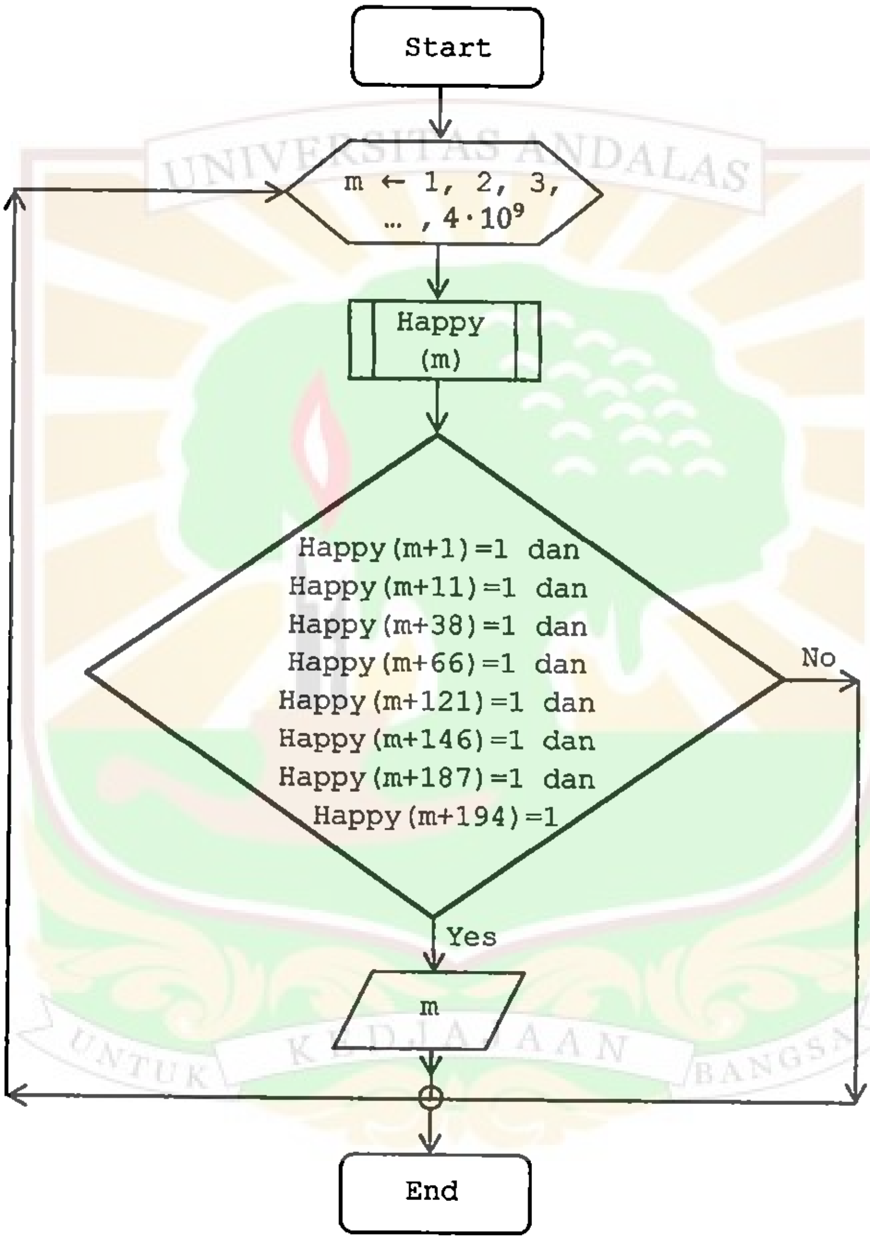
Output programnya adalah:

```
1)    8880958     18809958    47809958    48709958    74809958
6)    78409958    80880958    81809958    84709958    87409958
11)   88080958    88109958    88800958    108809958   125099958
16)   125990958   129590958   129950958   152099958   152990958
21)   156790958   156970958   157690958   157960958   159290958
26)   159670958   159760958   159920958   165790958   165970958
31)   167590958   167950958   169570958   169750958   175690958
36)   175960958   176590958   176950958   179560958   179650958
41)   180809958   188009958   192590958   192950958   195290958
46)   195670958   195760958   195920958   196570958   196750958
51)   197560958   197650958   199250958   199520958   215099958
56)   215990958   219590958   219950958   237790958   237970958
61)   239770958   251099958   251990958   256809958   258609958
66)   259190958   259910958   265809958   268509958   273790958
71)   273970958   277390958   277930958   279370958   279730958
76)   285609958   286509958   291590958   291950958   293770958
81)   295190958   295910958   297370958   297730958   299150958
86)   299510958   327790958   327970958   329770958   367770958
91)   372790958   372970958   376770958   377290958   377670958
96)   377760958   377920958   379270958   379720958   392770958
101)  397270958   397720958   407809958   408709958   444099958
106)  444909958   449409958   470809958   478009958   480709958
111)  487009958   494409958   512099958   512990958   516790958
116)  516970958   517690958   517960958   519290958   519670958
121)  519760958   519920958   521099958   521990958   526809958
126)  528609958   529190958   529910958   555690958   555960958
131)  556590958   556950958   559560958   559650958   561790958
136)  561970958   562809958   565590958   565950958   567190958
141)  567910958   568209958   569170958   569550958   569710958
146)  571690958   571960958   576190958   576910958   579160958
```

dan seterusnya.

Lampiran c. Program untuk mencari nilai y sedemikian sehingga $y + v$ adalah bilangan *happy* untuk setiap $v \in \{1, 11, 38, 66, 121, 146, 187, 194)$.

Flowchart programnya adalah:



Kode programnya adalah:

```
#include <iostream>
int happy (unsigned long int m)
{
    int i;
    int a,b;
    unsigned long int n;

    n = m;
    for (i=1 ; i<9 ; i++) {
       a = 0;
       while (n > 0) {
           b = n % 10;
           a = a + b*b;
           n = (n - b)/10;
       }
       n = a;
    }
    return n;
}

int main ()
{
    unsigned long int m;

    for (m=1 ; m<4000000000 ; m++) {
       if (happy(m+1)  ==1 && happy(m+11) ==1 &&
           happy(m+38) ==1 && happy(m+66) ==1 &&
           happy(m+121)==1 && happy(m+146)==1 &&
           happy(m+187)==1 && happy(m+194)==1) {
           cout <<  m << endl;
       }
    }

   return 0;

}
```

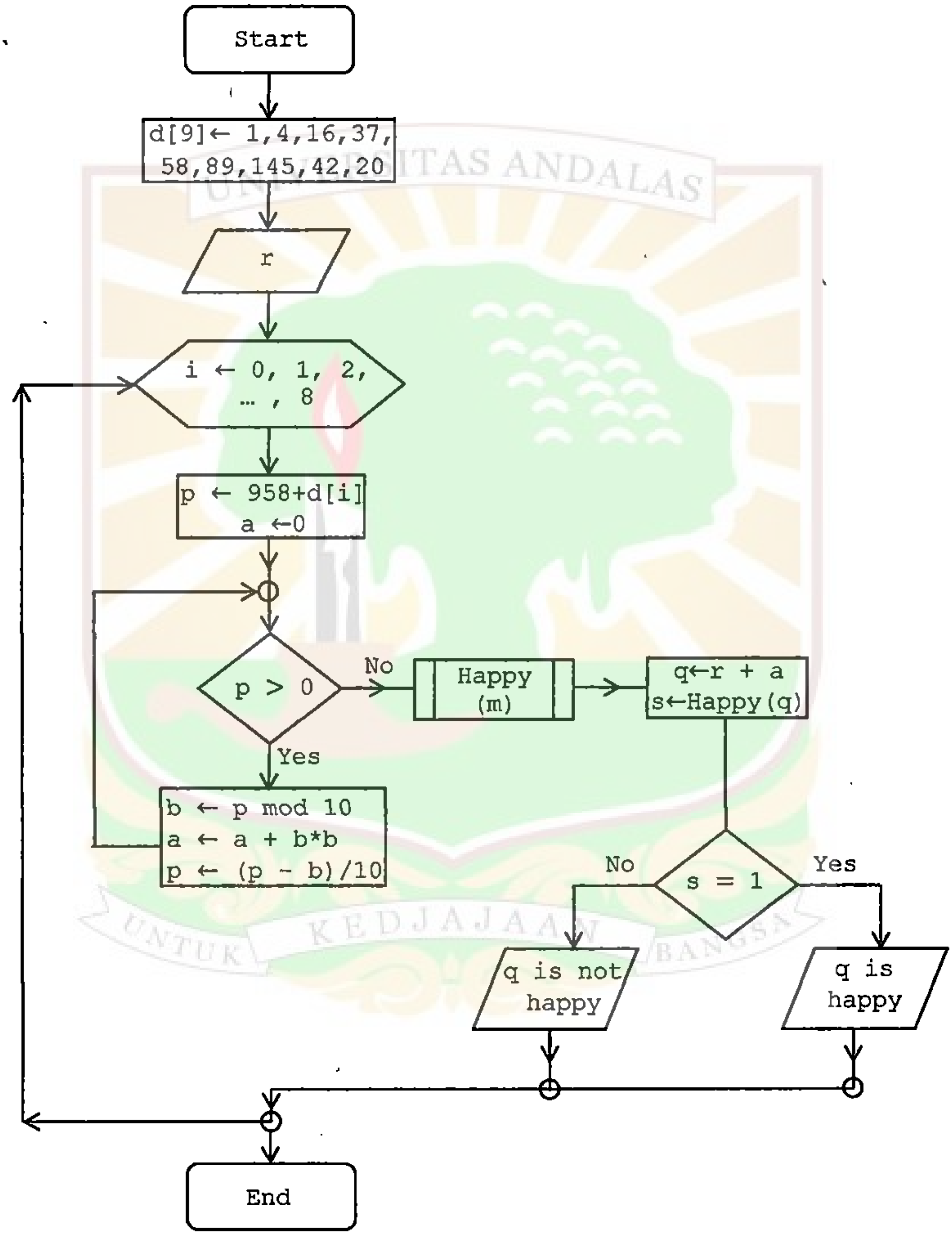
Output programnya adalah:

```
1)     18888556    47888556    48788556    48878556    48887556
6)     74888556    78488556    78848556    78884556    81888556
11)    84788556    84878556    84887556    87488556    87848556
16)    87884556    88188556    88478556    88487556    88748556
21)    88784556    88818556    88847556    88874556    88881556
26)    108888556   123999556   129399556   129939556   129993556
31)    132999556   136799556   136979556   136997556   137399856
36)    137699556   137969556   137996556   139299556   139679556
41)    139697556   139769556   139796556   139929556   139967556
46)    139976556   139992556   163799556   163979556   163997556
51)    167399556   167939556   167993556   169379556   169397556
56)    169739556   169793556   169937556   169973556   173399856
61)    173699556   173969556   173996556   176399556   176939556
66)    176993556   179369556   179396556   179639556   179693556
71)    179936556   179963556   180888556   188088556   188808556
76)    188880556   192399556   192939556   192993556   193299556
81)    193679556   193697556   193769556   193796556   193929556
86)    193967556   193976556   193992556   196379556   196397556
91)    196739556   196793556   196937556   196973556   197369556
96)    197396556   197639556   197693556   197936556   197963556
101)   199239556   199293556   199329556   199367556   199376556
106)   199392556   199637556   199673556   199736556   199763556
111)   199923556   199932556   213999556   219399556   219939556
116)   219993556   231999556   239199556   239919556   239991556
121)   256888556   257779556   257797556   257977556   258688556
126)   258868556   258886556   259777556   265888556   266689556
131)   266698556   266869556   266896556   266968556   266986556
136)   268588556   268669556   268696556   268858556   268885556
141)   268966556   269668556   269686556   269866556   275779556
146)   275797556   275977556   277579556   277597556   277759556
```

dan seterusnya.

Lampiran d. Program untuk menentukan apakah $n + T(958 + u)$ adalah bilangan *happy* untuk setiap $u \in \{1, 4, 16, 20, 37, 42, 58, 89, 145)$.

Flowchart programnya adalah:



Kode programnya adalah:

```
#include <iostream>

int happy (int m) {
   int i;
   int a,b,n;

   n = m;
   for (i=1 ; i<9 ; i++) {
       a = 0;
      while (n > 0) {
         b = n % 10;
         a = a + b*b;
         n = (n - b)/10;
      }
      n = a;
   }
   return n;
}

int main ()
{
   int d[9] = {1,4,16,37,58,89,145,42,20};
   int a,b;
   int i;
   int p,q,r,s;


   cout << "input n : " ;
   cin  >> r;

   for (i=0 ; i<9 ; i++) {
      p = 958 + d[i];
      a = 0;
      while (p > 0) {
         b = p % 10;
         a = a + b*b;
         p = (p - b)/10;
      }

      q = r + a;
      s = happy(q);

     if (s==1) {
        cout<<r<<"+"<<"T(958 + "<<d[i]<<")"<<" = "<<q<< "
        is happy " << endl;
      } else {
```

```
        cout<<r<<"+"<<"T(958 + "<<d[i]<<")"<<" = "<<q<< "
        is not happy " << endl;
      }

   }

   return 0;

}
```

Output programnya adalah:

```
input n : 18888556
18888556+T(958 + 1)   = 18888743 is happy
18888556+T(958 + 4)   = 18888677 is happy
18888556+T(958 + 16) = 18888702 is happy
18888556+T(958 + 37) = 18888743 is happy
18888556+T(958 + 58) = 18888594 is happy
18888556+T(958 + 89) = 18888622 is happy
18888556+T(958 + 145)= 18888567 is happy
18888556+T(958 + 42) = 18888557 is happy
18888556+T(958 + 20) = 18888750 is happy

input n : 88188556
88188556+T(958 + 1)   = 88188743 is happy
88188556+T(958 + 4)   = 88188677 is happy
88188556+T(958 + 16) = 88188702 is happy
88188556+T(958 + 37) = 88188743 is happy
88188556+T(958 + 58) = 88188594 is happy
88188556+T(958 + 89) = 88188622 is happy
88188556+T(958 + 145)= 88188567 is happy
88188556+T(958 + 42) = 88188557 is happy
88188556+T(958 + 20) = 88188750 is happy
```

Lampiran e. Program mencari nilai *n* terkecil sedemikian sehingga $T(n) = y$ untuk setiap $y \in \mathbb{N}$.

> $f := n \rightarrow n^2$; #*f* adalah suatu fungsi dari *n* ke n^2.

$$f := n \rightarrow n^2 \tag{1}$$

> bs := 10; #bs adalah basis dari *n*.

$$bs := 10 \tag{2}$$

```
> onestep :=proc(n1)
      #onestep adalah suatu prosedur untuk mencari nilai dari happy function
    local ans, n, d;
    n := n1;
    ans := 0;
    while n > 0 do
    d := n mod bs;
    ans := ans + f(d);
    n := (n - d)/bs;
    end do;
    ans;
    end;
```

onestep := **proc**(n1) **(3)**

```
    local ans, n, d;
    n := n1;
    ans := 0;
    while 0 < n do
        d := mod(n, bs); ans := ans + f(d); n := (n - d) / bs
    end do;
    ans
end proc
```

> *onestep*(200590)

$$110 \tag{4}$$

```
> happy :=proc(n)
      #happy adalah suatu prosedur dimana outputnya adalah
      -1 (bilangan unhappy) atau r (bilangan happy) untuk suatu r ∈ N.
    local m, j, height;
    m := n ;
    height := -1 ;
    for j from 1 to 100 while (m > 1 and m ≠ 4) do
      m := onestep(m);
    end do;
    if m = 1 then height := j;
    end if;
    height;
    end;
```

happy := **proc**(*n*) **(5)**

```
    local m, j, height;
    m := n;
    height := -1;
    for j to 100 while 1 < m and m<>4 do m := onestep(m) end do;
    if m = 1 then height := j end if;
    height
end proc
> happy(7)
                                6                                    (6)
> happy(298)
                                -1                                   (7)
> addsortdigit :=proc(a, d)
      #addsortdigit adalah suatu prosedur untuk menambahkan sebuah digit
      d kedalam sebuah bilangan a , kemudian mengurutkan digit
      − digitnya dari yang terkecil hingga yang terbesar.
   local b, e, i;
   b := a;
   e := [ ];
   while b > 0 do e := [b mod bs, op(e) ] ; b := iquo(b − (b mod bs),
       bs); end do;
   e := [op(e), d];
   e := sort(e);
   b := 0;
   for i from 1 to nops(e) do
   b := b·bs + e[i];
   end do;
   b;
   end;
addsortdigit := proc(a, d)                                           (8)
    local b, e, i;
    b := a;
    e := [ ];
    while 0 < b do
        e := [mod(b, bs), op(e)]; b := iquo(b − (mod(b, bs)), bs)
    end do;
    e := [op(e), d];
    e := sort(e);
    b := 0;
    for i to nops(e) do b := b * bs + e[i] end do;
    b
end proc
> addsortdigit(23452, 8)
                              223458                                 (9)
```

Nilai n yang terkecil sedemikian sehingga $T(n) = y$ untuk setiap $1 \le y \le 9$ dan $y \in \mathbb{N}$ dapat dilihat pada barisan A berikut

```
> A := [1, 11, 111, 2, 12, 112, 1112, 22, 3]
```

$$A := [1, 11, 111, 2, 12, 112, 1112, 22, 3] \tag{10}$$

Nilai n yang terkecil sedemikian sehingga $T(n) = y$ untuk setiap $1 \le y \le 16$ dan $y \in \mathbb{N}$ dapat dilihat pada barisan A berikut

```
> for n from 10 to 15 do
  L := addsortdigit(A[n - 1], 1);
  for d from 2 to 3 do
  M := addsortdigit(A[n - onestep(d)], d);
  if M < L then L := M; end if;
  end do;
  A := [op(A), L];
  end do:
  A;
```

$$[1, 11, 111, 2, 12, 112, 1112, 22, 3, 13, 113, 222, 23, 123, 1123] \tag{11}$$

```
> A := [op(A), 4]
```

$$A := [1, 11, 111, 2, 12, 112, 1112, 22, 3, 13, 113, 222, 23, 123, 1123, 4] \tag{12}$$

Nilai n yang terkecil sedemikian sehingga $T(n) = y$ untuk setiap $1 \le y \le 25$ dan $y \in \mathbb{N}$ dapat dilihat pada himpunan A berikut

```
> for n from 17 to 24 do
  L := addsortdigit(A[n - 1], 1);
  for d from 2 to 4 do
  M := addsortdigit(A[n - onestep(d)], d);
  if M < L then L := M; end if;
  end do;
  A := [op(A), L];
  end do:
  A;
```

$$[1, 11, 111, 2, 12, 112, 1112, 22, 3, 13, 113, 222, 23, 123, 1123, 4, 14, 33, 133, 24, 124, 233, 1233, 224] \tag{13}$$

```
> A := [op(A), 5];
```

$$A := [1, 11, 111, 2, 12, 112, 1112, 22, 3, 13, 113, 222, 23, 123, 1123, 4, 14, 33, 133, 24, 124, 233, 1233, 224, 5] \tag{14}$$

Nilai n yang terkecil sedemikian sehingga $T(n) = y$ untuk setiap $1 \le y \le 36$ dan $y \in \mathbb{N}$ dapat dilihat pada barisan A berikut

```
> for n from 26 to 35 do
  L := addsortdigit(A[n - 1], 1);
  for d from 2 to 5 do
  M := addsortdigit(A[n - onestep(d)], d);
  if M < L then L := M; end if;
  end do;
  A := [op(A), L];
```

```
end do:
A;
```

[1, 11, 111, 2, 12, 112, 1112, 22, 3, 13, 113, 222, 23, 123, 1123, 4, 14, 33, 133, 24, 124, 233, 1233, 224, 5, 15, 115, 1115, 25, 125, 1125, 44, 144, 35, 135] **(15)**

```
> A := [op(A), 6];
```

A := [1, 11, 111, 2, 12, 112, 1112, 22, 3, 13, 113, 222, 23, 123, 1123, 4, 14, 33, 133, 24, 124, 233, 1233, 224, 5, 15, 115, 1115, 25, 125, 1125, 44, 144, 35, 135, 6] **(16)**

Nilai n yang terkecil sedemikian sehingga $T(n) = y$ untuk setiap $1 \leq y \leq 49$ dan $y \in \mathbb{N}$ dapat dilihat pada barisan A berikut

```
> for n from 37 to 48 do
  L := addsortdigit(A[n - 1], 1);
  for d from 2 to 6 do
  M := addsortdigit(A[n - onestep(d)], d);
  if M < L then L := M; end if;
  end do;
  A := [op(A), L];
  end do:
  A;
```

[1, 11, 111, 2, 12, 112, 1112, 22, 3, 13, 113, 222, 23, 123, 1123, 4, 14, 33, 133, 24, 124, 233, 1233, 224, 5, 15, 115, 1115, 25, 125, 1125, 44, 144, 35, 135, 6, 16, 116, 1116, 26, 45, 145, 335, 226, 36, 136, 1136, 444] **(17)**

```
> A := [op(A), 7];
```

A := [1, 11, 111, 2, 12, 112, 1112, 22, 3, 13, 113, 222, 23, 123, 1123, 4, 14, 33, 133, 24, 124, 233, 1233, 224, 5, 15, 115, 1115, 25, 125, 1125, 44, 144, 35, 135, 6, 16, 116, 1116, 26, 45, 145, 335, 226, 36, 136, 1136, 444, 7] **(18)**

Nilai n yang terkecil sedemikian sehingga $T(n) = y$ untuk setiap $1 \leq y \leq 64$ dan $y \in \mathbb{N}$ dapat dilihat pada barisan A berikut

```
> for n from 50 to 63 do
  L := addsortdigit(A[n - 1], 1);
  for d from 2 to 7 do
  M := addsortdigit(A[n - onestep(d)], d);
  if M < L then L := M; end if;
  end do;
  A := [op(A), L];
  end do:
  A;
```

[1, 11, 111, 2, 12, 112, 1112, 22, 3, 13, 113, 222, 23, 123, 1123, 4, 14, 33, 133, 24, 124, 233, 1233, 224, 5, 15, 115, 1115, 25, 125, 1125, 44, 144, 35, 135, 6, 16, 116, 1116, 26, 45, 145, 335, 226, 36, 136, 1136, 444, 7, 17, 117, 46, 27, 127, 1127, 246, 227, 37, 137, 1137, 56, 156, 1156] **(19)**

```
> A := [op(A), 8];
```

$A := [1, 11, 111, 2, 12, 112, 1112, 22, 3, 13, 113, 222, 23, 123, 1123, 4, 14, 33, 133, 24, 124, 233, 1233, 224, 5, 15, 115, 1115, 25, 125, 1125, 44, 144, 35, 135, 6, 16, 116, 1116, 26, 45, 145, 335, 226, 36, 136, 1136, 444, 7, 17, 117, 46, 27, 127, 1127, 246, 227, 37, 137, 1137, 56, 156, 1156, 8]$ **(20)**

Nilai n yang terkecil sedemikian sehingga $T(n) = y$ untuk setiap $1 \leq y \leq 81$ dan $y \in \mathbb{N}$ dapat dilihat pada barisan A berikut

```
> for n from 65 to 80 do
  L := addsortdigit(A[n - 1], 1);
  for d from 2 to 8 do
  M := addsortdigit(A[n - onestep(d)], d);
  if M < L then L := M; end if;
  end do;
  A := [op(A), L];
  end do:
  A;
```

$[1, 11, 111, 2, 12, 112, 1112, 22, 3, 13, 113, 222, 23, 123, 1123, 4, 14, 33, 133, 24, 124, 233, 1233, 224, 5, 15, 115, 1115, 25, 125, 1125, 44, 144, 35, 135, 6, 16, 116, 1116, 26, 45, 145, 335, 226, 36, 136, 1136, 444, 7, 17, 117, 46, 27, 127, 1127, 246, 227, 37, 137, 1137, 56, 156, 1156, 8, 18, 118, 337, 28, 128, 356, 1356, 66, 38, 57, 157, 266, 238, 257, 1257, 48]$ **(21)**

```
> A := [op(A), 9];
```

$A := [1, 11, 111, 2, 12, 112, 1112, 22, 3, 13, 113, 222, 23, 123, 1123, 4, 14, 33, 133, 24, 124, 233, 1233, 224, 5, 15, 115, 1115, 25, 125, 1125, 44, 144, 35, 135, 6, 16, 116, 1116, 26, 45, 145, 335, 226, 36, 136, 1136, 444, 7, 17, 117, 46, 27, 127, 1127, 246, 227, 37, 137, 1137, 56, 156, 1156, 8, 18, 118, 337, 28, 128, 356, 1356, 66, 38, 57, 157, 266, 238, 257, 1257, 48, 9]$ **(22)**

Nilai n yang terkecil sedemikian sehingga $T(n) = y$ untuk setiap $1 \leq y \leq 1100$ dan $y \in \mathbb{N}$ dapat dilihat pada barisan A berikut

```
> for n from 82 to 900 do
  L := addsortdigit(A[n - 1], 1);
  for d from 2 to 9 do
  M := addsortdigit(A[n - onestep(d)], d);
  if M < L then L := M; end if;
  end do;
  A := [op(A), L];
  end do:
  A;
```

$[1, 11, 111, 2, 12, 112, 1112, 22, 3, 13, 113, 222, 23, 123, 1123, 4, 14, 33,$ **(23)**

133, 24, 124, 233, 1233, 224, 5, 15, 115, 1115, 25, 125, 1125, 44, 144, 35, 135, 6, 16, 116, 1116, 26, 45, 145, 335, 226, 36, 136, 1136, 444, 7, 17, 117, 46, 27, 127, 1127, 246, 227, 37, 137, 1137, 56, 156, 1156, 8, 18, 118, 337, 28, 128, 356, 1356, 66, 38, 57, 157, 266, 238, 257, 1257, 48, 9, 19, 119, 248, 29, 129, 1129, 466, 58, 39, 139, 1139, 258, 239, 1239, 448, 49, 77, 177, 68, 168, 277, 1277, 268, 458, 59, 159, 666, 368, 259, 1259, 2666, 78, 178, 359, 468, 69, 169, 1169, 2468, 269, 378, 577, 1577, 568, 369, 1369, 88, 188, 79, 179, 288, 469, 279, 1279, 668, 388, 578, 379, 1379, 2388, 569, 1569, 488, 89, 189, 777, 1777, 289, 1289, 2777, 4668, 588, 389, 579, 1579, 2588, 2389, 2579, 4488, 489, 99, 199, 688, 1688, 299, 1299, 2688, 4588, 589, 399, 1399, 3688, 2589, 2399, 12399, 788, 499, 779, 1779, 689, 1689, 2779, 12779, 2689, 3788, 599, 1599, 5688, 3689, 2599, 888, 1888, 789, 1789, 2888, 4689, 699, 1699, 6688, 3888, 2699, 3789, 5779, 15779, 5689, 3699, 4888, 889, 1889, 799, 1799, 2889, 4699, 2799, 12799, 5888, 3889, 5789, 3799, 13799, 23889, 5699, 15699, 4889, 899, 1899, 6888, 16888, 2899, 12899, 26888, 45888, 5889, 3899, 5799, 15799, 25889, 23899, 25799, 7888, 4899, 999, 1999, 6889, 16889, 2999, 12999, 26889, 37888, 5899, 3999, 13999, 36889, 25899, 8888, 18888, 7889, 4999, 7799, 17799, 6899, 16899, 27799, 38888, 26899, 37889, 5999, 15999, 56889, 36899, 25999, 8889, 18889, 7899, 17899, 28889, 46899, 6999, 16999, 58888, 38889, 26999, 37899, 57799, 157799, 56899, 36999, 48889, 8899, 18899, 7999, 17999, 28899, 46999, 27999, 127999, 58889, 38899, 57899, 37999, 137999, 238899, 56999, 78888, 48899, 8999, 18999, 68889, 168889, 28999, 128999, 268889, 378888, 58899, 38999, 57999, 157999, 258899, 88888, 188888, 78889, 48999, 9999, 19999, 68899, 168899, 29999, 129999, 268899, 378889, 58999, 39999, 139999, 368899, 258999, 88889, 188889, 78899, 49999, 77999, 177999, 68999, 168999, 277999, 388889, 268999, 378899, 59999, 159999, 568899, 368999, 259999, 88899, 188899, 78999, 178999, 288899, 468999, 69999, 169999, 588889, 388899, 269999, 378999, 577999, 1577999, 568999, 369999, 488899, 88999, 188999, 79999, 179999, 288999, 469999, 279999, 1279999, 588899, 388999, 578999, 379999, 1379999, 888888, 569999, 788889, 488999, 89999, 189999, 688899, 1688899, 289999, 1289999, 2688899, 3788889, 588999, 389999, 579999, 1579999, 2588999, 888889, 1888889, 788899, 489999, 99999, 199999, 688999, 1688999, 299999, 1299999, 2688999, 3788899, 589999, 399999, 1399999, 3688999, 2589999, 888899, 1888899, 788999, 499999, 779999, 1779999, 689999, 1689999, 2779999, 3888899, 2689999, 3788999, 599999, 1599999, 5688999, 3689999, 2599999, 888999, 1888999, 789999, 1789999, 2888999, 4689999, 699999, 1699999, 5888899, 3888999, 2699999, 3789999, 5779999, 8888888, 5689999, 3699999, 4888999, 889999, 1889999, 799999, 1799999, 2889999, 4699999, 2799999, 12799999, 5888999, 3889999, 5789999, 3799999, 13799999, 8888889, 5699999, 7888899, 4889999, 899999, 1899999, 6888999, 16888999, 2899999, 12899999, 26888999, 37888899, 5889999, 3899999, 5799999, 15799999, 25889999, 8888899, 18888899, 7888999, 4899999, 999999, 1999999, 6889999, 16889999, 2999999, 12999999, 26889999, 37888999, 5899999, 3999999, 13999999, 36889999, 25899999, 8888999, 18888999, 7889999, 4999999, 7799999, 17799999, 6899999, 16899999, 27799999, 38888999, 26899999, 37889999, 5999999, 15999999, 56889999, 36899999, 25999999, 8889999, 18889999, 7899999, 17899999, 28889999, 46899999, 6999999, 16999999, 58888999, 38889999, 26999999, 37899999, 57799999, 88888889, 56899999, 36999999, 48889999, 8899999, 18899999, 7999999, 17999999, 28899999, 46999999, 27999999, 127999999, 58889999, 38899999, 57899999, 37999999, 137999999, 88888899, 56999999, 78888999, 48899999, 8999999, 18999999, 68889999, 168889999, 28999999, 128999999, 268889999, 378888999, 58899999, 38999999, 57999999, 157999999, 258899999, 88888999, 188888999, 78889999, 48999999, 9999999, 19999999, 68899999, 168899999, 29999999, 129999999, 268899999, 378889999, 58999999, 39999999, 139999999, 368899999, 258999999, 88889999, 188889999, 78899999, 49999999, 77999999, 177999999, 68999999, 168999999, 277999999, 388889999, 268999999, 378899999, 59999999, 159999999, 568899999, 368999999, 259999999, 88899999, 188899999, 78999999, 178999999, 288899999, 468999999, 69999999, 169999999, 588889999, 388899999, 269999999, 378999999, 577999999, 888888899, 568999999, 369999999, 488899999, 88999999, 188999999, 79999999, 179999999, 288999999, 469999999, 279999999, 1279999999, 588899999, 388999999, 578999999, 379999999, 1379999999, 888888999, 569999999, 788889999, 488999999, 89999999, 189999999, 688899999, 1688899999, 289999999, 1289999999, 2688899999, 3788889999, 588999999, 389999999, 579999999, 1579999999, 2588999999, 888889999, 1888889999, 788899999, 489999999, 99999999, 199999999, 688999999, 1688999999, 299999999, 1299999999, 2688999999, 3788899999, 589999999, 399999999, 1399999999, 3688999999, 2589999999, 888899999, 1888899999, 788999999, 499999999, 779999999, 1779999999, 689999999, 1689999999, 2779999999, 3888899999, 2689999999, 3788999999, 599999999, 1599999999, 5688999999, 3689999999, 2599999999, 888999999, 1888999999, 789999999, 1789999999, 2888999999, 4689999999, 699999999, 1699999999, 5888899999, 3888999999, 2699999999, 3789999999, 5779999999, 8888888999, 5689999999, 3699999999, 4888999999, 889999999, 1889999999, 799999999, 1799999999, 2889999999, 4699999999, 2799999999, 12799999999, 5888999999, 3889999999, 5789999999, 3799999999, 13799999999, 8888889999, 5699999999, 7888899999, 4889999999, 899999999, 1899999999, 6888999999, 16888999999, 2899999999, 12899999999, 26888999999, 37888899999, 5889999999, 3899999999, 5799999999, 15799999999, 25889999999, 8888899999, 18888899999, 7888999999, 4899999999, 999999999, 1999999999, 6889999999, 16889999999, 2999999999, 12999999999, 26889999999, 37888999999, 5899999999, 3999999999, 13999999999, 36889999999, 25899999999, 8888999999, 18888999999, 7889999999, 4999999999, 7799999999, 17799999999, 6899999999, 16899999999, 27799999999, 38888999999, 26899999999, 37889999999, 5999999999, 15999999999, 56889999999, 36899999999, 25999999999, 8889999999, 18889999999, 7899999999, 17899999999, 28889999999, 46899999999, 6999999999, 16999999999, 58888999999, 38889999999, 26999999999, 37899999999, 57799999999, 88888889999, 56899999999, 36999999999, 48889999999, 8899999999, 18899999999, 7999999999, 17999999999, 28899999999, 46999999999, 27999999999, 127999999999, 58889999999, 38899999999, 57899999999, 37999999999, 137999999999, 88888899999, 56999999999, 78888999999, 48899999999, 8999999999, 18999999999, 68889999999, 168889999999, 28999999999, 128999999999, 268889999999, 378888999999, 58899999999, 38999999999, 57999999999, 157999999999, 258899999999, 88888999999, 188888999999, 78889999999, 48999999999, 9999999999, 19999999999, 68899999999, 168899999999, 29999999999, 129999999999, 268899999999, 378889999999, 58999999999, 39999999999, 139999999999, 368899999999, 258999999999, 88889999999, 188889999999, 78899999999, 49999999999, 77999999999, 177999999999, 68999999999, 168999999999, 277999999999, 388889999999, 268999999999, 378899999999, 59999999999, 159999999999, 568899999999, 368999999999, 259999999999, 88899999999, 188899999999, 78999999999, 178999999999, 288899999999, 468999999999, 69999999999, 169999999999, 588889999999, 388899999999, 269999999999, 378999999999, 577999999999, 888888899999, 568999999999, 369999999999, 488899999999, 88999999999, 188999999999,

799999999999, 1799999999999, 2889999999999, 4699999999999,
2799999999999, 12799999999999, 5888999999999, 3889999999999,
5789999999999, 3799999999999, 13799999999999, 8888889999999,
5699999999999, 7888899999999, 4889999999999, 899999999999,
1899999999999, 6888999999999, 16888999999999, 2899999999999,
12899999999999, 26888999999999, 37888899999999, 5889999999999,
3899999999999, 5799999999999, 15799999999999, 25889999999999,
8888899999999, 18888899999999, 7888999999999, 4899999999999,
999999999999, 1999999999999, 6889999999999, 16889999999999,
2999999999999, 12999999999999, 26889999999999, 37888999999999,
5899999999999, 3999999999999]

Pada barisan A di atas, terlihat bahwa sebagian besar suku-sukunya berakhir dengan digit 9 maka kita dapat menyederhanakan program di atas dengan menemukan sebuah pola sehingga didapatkanlah nilai n terkecil sedemikian sehingga $T(n) = y$ untuk setiap $y \in \mathbb{N}$. Perhatikan program berikut:

```
> value := false;
  for n from 900 to 1 by -1 while value = false do
      if A[n] mod 10 < 9 then print(n, A[n]); value := true ; end if;
  end do;
```

$$value := false$$

$$448,\ 8888888 \tag{24}$$

Hal ini menunjukkan bahwa nilai $A[n]$ terbesar dengan digit terakhir 8 adalah $A[448] = 8888888$.

```
> value := false;
  for n from 900 to 1 by -1 while value = false do
      if A[n] mod 10 < 8 then print(n, A[n]); value := true ; end if;
  end do;
```

$$value := false$$

$$151,\ 2777 \tag{25}$$

Hal ini menunjukkan bahwa nilai $A[n]$ terbesar dengan digit terakhir 7 adalah $A[151] = 2777$.

```
> value := false;
  for n from 900 to 1 by -1 while value = false do
      if A[n] mod 10 < 7 then print(n, A[n]); value := true ; end if;
  end do;
```

$$value := false$$

$$112,\ 2666 \tag{26}$$

Hal ini menunjukkan bahwa nilai $A[n]$ terbesar dengan digit terakhir 6 adalah $A[112] = 2666$.

```
> value := false;
  for n from 900 to 1 by -1 while value = false do
```

```
        if A[n] mod 10 < 6 then print(n, A[n]); value := true ; end if;
    end do;
```

$$value := false$$

$$48,\ 444 \tag{27}$$

Hal ini menunjukkan bahwa nilai $A[n]$ terbesar dengan digit terakhir 5 adalah $A[48]=444$.

```
> value := false;
    for n from 900 to 1 by -1 while value = false do
        if A[n] mod 10 < 5 then print(n, A[n]); value := true ; end if;
    end do;
```

$$value := false$$

$$48,\ 444 \tag{28}$$

Hal ini menunjukkan bahwa nilai $A[n]$ terbesar dengan digit terakhir 4 adalah $A[48]=444$.

```
> value := false;
    for n from 900 to 1 by -1 while value = false do
        if A[n] mod 10 < 4 then print(n, A[n]); value := true ; end if;
    end do;
```

$$value := false$$

$$23,\ 1233 \tag{29}$$

Hal ini menunjukkan bahwa nilai $A[n]$ terbesar dengan digit terakhir 3 adalah $A[23]=1233$.

```
> value := false;
    for n from 900 to 1 by -1 while value = false do
        if A[n] mod 10 < 3 then print(n, A[n]); value := true ; end if;
    end do;
```

$$value := false$$

$$12,\ 222 \tag{30}$$

Hal ini menunjukkan bahwa nilai $A[n]$ terbesar dengan digit terakhir 2 adalah $A[12]=222$.

```
> value := false;
    for n from 900 to 1 by -1 while value = false do
        if A[n] mod 10 < 2 then print(n, A[n]); value := true ; end if;
    end do;
```

$$value := false$$

$$3,\ 111 \tag{31}$$

Hal ini menunjukkan bahwa nilai $A[n]$ terbesar dengan digit terakhir 1 adalah $A[3]=111$.

Dari Program (24) - (31) di atas, dapat dilihat bahwa jika $y > 448$ maka nilai n terkecil sedemikian sehingga $T(n) = y$ untuk setiap $y \in \mathbb{N}$ memiliki digit terakhir yaitu 9, sehingga dapat dibuat sebuah program yang lebih sederhana lagi yaitu:

```
> minimalN := proc(n)
    local q, r, k, ans;
```

```
if n < 486 then ans := A[n];
else q := iquo(n, 81, 'r');
    ans := A[n - (q - 5)·81]·10^(q - 5) + (10^(q - 5) - 1);
end if;
ans;
end;
```

minimalN := **proc**(n) **(32)**

```
    local q, r, k, ans;
    if n < 486 then
        ans := A[n]
    else
        q := iquo(n, 81, 'r');
        ans := A[n - 81 * q + 405] * 10^(q - 5) + 10^(q - 5) - 1
    end if;
    ans
end proc
```

> A[899]

589999999999 **(33)**

> minimalN(899)

589999999999 **(34)**

> minimalN(2532)

2779999999999999999999999999999999 **(35)**

>

Lampiran f. Program mencari bilangan terkecil dari barisan bilangan *happy* terurut dengan panjang enam.

```
> f := n → n^2;  #f adalah suatu fungsi dari n ke n^2.
                              f := n → n^2                                   (1)
> bs := 10;   #bs adalah basis dari n.
                              bs := 10                                       (2)
> onestep := proc(n1)
      #onestep adalah suatu prosedur untuk mencari nilai dari happy function
    local ans, n, d;
    n := n1;
    ans := 0;
    while n > 0 do
    d := n mod bs;
    ans := ans + f(d);
    n := (n − d)/bs;
    end do;
    ans;
    end;
onestep := proc(n1)                                                          (3)
    local ans, n, d;
    n := n1;
    ans := 0;
    while 0 < n do
        d := mod(n, bs); ans := ans + f(d); n := (n − d)/bs
    end do;
    ans
end proc
> onestep(112233)
                              28                                             (4)
> happy := proc(n)
      #happy adalah suatu prosedur dimana outputnya adalah
      -1 (bilangan unhappy) atau r (bilangan happy) untuk suatu r ∈ ℕ.
    local m, j, height;
    m := n;
    height := -1;
    for j from 1 to 100 while (m > 1 and m ≠ 4) do
      m := onestep(m);
    end do;
    if m = 1 then height := j;
    end if;
    height;
    end;
happy := proc(n)                                                             (5)
    local m, j, height;
```

```
    m := n;
    height := -1;
    for j to 100 while 1 < m and m<>4 do m := onestep(m) end do;
    if m = 1 then height := j end if;
    height
end proc
> happy(5)
                                  -1                              (6)
> happy(7)
                                   6                              (7)
```

Pada Lampiran e, telah didapatkan nilai n terkecil sedemikian sehingga $T(n) = y$ untuk setiap $y \in \mathbb{N}$. Adapun 485 nilai n petama dapat dilihat pada barisan A berikut:

A := [1, 11, 111, 2, 12, 112, 1112, 22, 3, 13, 113, 222, 23, 123, 1123, 4, 14, 33, 133, 24, 124, 233, 1233, 224, 5, 15, 115, 1115, 25, 125, 1125, 44, 144, 35, 135, 6, 16, 116, 1116, 26, 45, 145, 335, 226, 36, 136, 1136, 444, 7, 17, 117, 46, 27, 127, 1127, 246, 227, 37, 137, 1137, 56, 156, 1156, 8, 18, 118, 337, 28, 128, 356, 1356, 66, 38, 57, 157, 266, 238, 257, 1257, 48, 9, 19, 119, 248, 29, 129, 1129, 466, 58, 39, 139, 1139, 258, 239, 1239, 448, 49, 77, 177, 68, 168, 277, 1277, 268, 458, 59, 159, 666, 368, 259, 1259, 2666, 78, 178, 359, 468, 69, 169, 1169, 2468, 269, 378, 577, 1577, 568, 369, 1369, 88, 188, 79, 179, 288, 469, 279, 1279, 668, 388, 578, 379, 1379, 2388, 569, 1569, 488, 89, 189, 777, 1777, 289, 1289, 2777, 4668, 588, 389, 579, 1579, 2588, 2389, 2579, 4488, 489, 99, 199, 688, 1688, 299, 1299, 2688, 4588, 589, 399, 1399, 3688, 2589, 2399, 12399, 788, 499, 779, 1779, 689, 1689, 2779, 12779, 2689, 3788, 599, 1599, 5688, 3689, 2599, 888, 1888, 789, 1789, 2888, 4689, 699, 1699, 6688, 3888, 2699, 3789, 5779, 15779, 5689, 3699, 4888, 889, 1889, 799, 1799, 2889, 4699, 2799, 12799, 5888, 3889, 5789, 3799, 13799, 23889, 5699, 15699, 4889, 899, 1899, 6888, 16888, 2899, 12899, 26888, 45888, 5889, 3899, 5799, 15799, 25889, 23899, 25799, 7888, 4899, 999, 1999, 6889, 16889, 2999, 12999, 26889, 37888, 5899, 3999, 13999, 36889, 25899, 8888, 18888, 7889, 4999, 7799, 17799, 6899, 16899, 27799, 38888, 26899, 37889, 5999, 15999, 56889, 36899, 25999, 8889, 18889, 7899, 17899, 28889, 46899, 6999, 16999, 58888, 38889, 26999, 37899, 57799, 157799, 56899, 36999, 48889, 8899, 18899, 7999, 17999, 28899, 46999, 27999, 127999, 58889, 38899, 57899, 37999, 137999, 238899, 56999, 78888, 48899, 8999, 18999, 68889, 168889, 28999, 128999, 268889, 378888, 58899, 38999, 57999, 157999, 258899, 88888, 188888, 78889, 48999, 9999, 19999, 68899, 168899, 29999, 129999, 268899, 378889, 58999, 39999, 139999, 368899, 258999, 88889, 188889, 78899, 49999, 77999, 177999, 68999, 168999, 277999, 388889, 268999, 378899, 59999, 159999, 568899, 368999, 259999, 88899, 188899, 78999, 178999, 288899, 468999, 69999, 169999, 588889, 388899, 269999, 378999, 577999, 1577999, 568999, 369999, 488899, 88999, 188999, 79999, 179999, 288999, 469999, 279999, 1279999, 588899, 388999, 578999, 379999, 1379999, 888888, 569999, 788889, 488999, 89999, 189999, 688899, 1688899, 289999, 1289999, 2688899, 3788889, 588999, 389999, 579999, 1579999, 2588999, 888889, 1888889, 788899, 489999, 99999, 199999, 688999, 1688999, 299999, 1299999, 2688999, 3788899, 589999, 399999, 1399999, 3688999, 2589999, 888899, 1888899, 788999, 499999, 779999, 1779999, 689999, 1689999, 2779999, 3888899, 2689999, 3788999, 599999, 1599999, 5688999, 3689999, 2599999, 888999, 1888999, 789999, 1789999, 2888999, 4689999, 699999, 1699999, 5888899, 3888999, 2699999, 3789999, 5779999, 8888888, 5689999, 3699999, 4888999, 889999, 1889999, 799999, 1799999, 2889999, 4699999, 2799999, 12799999, 5888999, 3889999, 5789999, 3799999, 13799999, 8888889, 5699999, 7888899, 4889999, 899999, 1899999, 6888999, 16888999, 2899999, 12899999, 26888999, 37888899, 5889999, 3899999, 5799999, 15799999, 25889999, 8888899, 18888899, 7888999, 4899999] :

Jika $y \geq 486$ maka untuk menentukan nilai n terkecil sedemikian sehingga $T(n) = y$, dapat digunakan prosedur berikut:

```
> minimalN := proc(n)
   local q, r, k, ans;
   if n < 486 then ans := A[n];
   else q := iquo(n, 81, 'r');
        ans := A[n - (q − 5)·81]·10^(q − 5) + (10^(q − 5) − 1);
   end if;
   ans;
   end;
```

```
minimalN := proc(n)                                                    (8)
    local q, r, k, ans;
    if n < 486 then
        ans := A[n]
    else
        q := iquo(n, 81, 'r');
        ans := A[n − 81 * q + 405] * 10^(q − 5) + 10^(q − 5) − 1
    end if;
    ans
end proc
```

```
> minimalN(1234)
                    6899999999999999                                   (9)
```

Selanjutnya akan dicari bilangan terkecil yang mengawali barisan bilangan *happy* terurut dengan panjang enam. Misalkan N adalah contoh terkecil yang mengawali barisan tersebut dimana

$$N = N_1 . d_0 \quad \text{dengan } d_0 \text{ adalah digit terakhir } N \text{ dan } N_1 \in \mathbb{N}.$$

Karena $N = N_1.d_0$ maka $T(N) = T(N_1.d_0) = T(N_1) + d_0^2$.

Misalkan $M_1 = T(N_1)$ maka $T(N) = M_1 + d_0^2$ dengan $0 \leq d_0 \leq 9$.

Selanjutnya, akan ditentukan kemungkinan-kemungkinan dari d_0.

```
> d0 := 0;
  for M1 from 1 to 3000 do
  if happy(M1 + d0^2) > 0 and happy(M1 + (d0 + 1)^2) > 0
      and happy(M1 + (d0 + 2)^2) > 0 then print(M1, happy(M1
      + (d0 + 3)^2)) end if;
  end do;
```

$$d0 := 0$$
129, -1
1029, -1
1121, -1
1184, -1
1211, -1
1299, -1
1474, -1
1574, -1
1744, -1
1754, -1
1814, -1
1929, -1
2111, -1
2571, -1
2751, -1 **(10)**

Jika d0=0 maka terdapat paling banyak 3 buah bilangan *happy* terurut yaitu $N, N+1, N+2$.

```
> d0 := 1;
  for M1 from 1 to 3000 do
  if happy(M1 + d0^2) > 0 and happy(M1 + (d0 + 1)^2) > 0
      and happy(M1 + (d0 + 2)^2) > 0 then print(M1, happy(M1
      + (d0 + 3)^2)) end if;
  end do;
```

$$d0 := 1$$
382, -1
903, -1
1751, -1
1843, -1
2843, -1 **(11)**

Jika d0=1 maka terdapat paling banyak 3 buah bilangan *happy* terurut yaitu $N, N+1, N+2$.

```
> d0 := 2;
  for M1 from 1 to 3000 do
  if happy(M1 + d0^2) > 0 and happy(M1 + (d0 + 1)^2) > 0
     and happy(M1 + (d0 + 2)^2) > 0  and happy(M1 + (d0 + 3)^2)
     > 0
  then print(M1, happy(M1 + (d0 + 4)^2)) end if;
  end do;
```

$$d0 := 2$$
$$628,\ -1$$
$$640,\ -1 \tag{12}$$

Jika d0=2 maka terdapat paling banyak 4 buah bilangan *happy* terurut yaitu $N, N+1, N+2, N+3$.

```
> d0 := 3;
  for M1 from 1 to 3000 do
  if happy(M1 + d0^2) > 0 and happy(M1 + (d0 + 1)^2) > 0
     and happy(M1 + (d0 + 2)^2) > 0  and happy(M1 + (d0 + 3)^2)
     > 0
  then print(M1, happy(M1 + (d0 + 4)^2)) end if;
  end do
```

$$d0 := 3$$
$$1708,\ -1$$
$$2894,\ -1 \tag{13}$$

Jika d0=3 maka terdapat paling banyak 4 buah bilangan *happy* terurut yaitu $N, N+1, N+2, N+3$.

```
> d0 := 4;
  for M1 from 1 to 3000 do
  if happy(M1 + d0^2) > 0 and happy(M1 + (d0 + 1)^2) > 0
     and happy(M1 + (d0 + 2)^2) > 0  and happy(M1 + (d0 + 3)^2)
     > 0
  then print(M1, happy(M1 + (d0 + 4)^2)) end if;
  end do;
```

$$d0 := 4$$
$$277,\ -1$$
$$2532,\ -1 \tag{14}$$

Jika d0=4 maka terdapat paling banyak 4 buah bilangan *happy* terurut yaitu $N, N+1, N+2, N+3$.

```
> d0 := 5;
  for M1 from 1 to 3000 do
  if happy(M1 + d0^2) > 0 and happy(M1 + (d0 + 1)^2) > 0
     and happy(M1 + (d0 + 2)^2) > 0 then print(M1, happy(M1
```

```
+ (d0 + 3)^2)) end if;
end do;
```

$$d0 := 5$$
$$277, -1$$
$$343, -1$$
$$811, -1$$
$$863, -1$$
$$874, -1$$
$$1735, -1$$
$$2163, -1$$
$$2532, -1$$
$$2977, -1 \quad (15)$$

Jika d0=5 maka terdapat paling banyak 3 buah bilangan *happy* terurut yaitu $N, N+1, N+2$.

Jika d0=8 maka terdapat paling banyak 5 buah bilangan *happy* terurut yaitu $N, N+1, N+2, N+3, N+4$.

Jika d0=9 maka terdapat paling banyak 4 buah bilangan *happy* terurut yaitu $N, N+1, N+2, N+3$.

```
> d0 := 6;
  for M1 from 1 to 3000 do
  if happy(M1 + d0^2) > 0 and happy(M1 + (d0 + 1)^2) > 0
     and happy(M1 + (d0 + 2)^2) > 0 then print(M1, happy(M1
     + (d0 + 3)^2)) end if;
  end do;
```

$$d0 := 6$$
$$1239, -1$$
$$1408, -1$$
$$1839, 4 \quad (16)$$

Asumsikan $N = N_1 d_0$ dengan $d0 = 6$ merupakan bilangan terkecil dari barisan bilangan *happy* terurut dengan panjang enam.
Misalkan $P = N + 3 = N_1.6 + 3 = N_1.9$
Misalkan $N_1 = N_2.d.99 \ldots 99$.
Misalkan $M_1 = T(N_1)$ dan $M_2 = T(N_2)$ maka

$$T(N_1) = T(N_2) + d^2 + 9^2 \cdot k$$
$$T(N_1) = M_1 = M_2 + d^2 + 9^2 \cdot k$$
$$T(N_2) = M_2 = M_1 - d^2 - 9^2 \cdot k$$

dimana k adalah banyaknya digit 9 yang mengakhiri nilai N_1 dan $0 \leq d \leq 8$, $d \in \mathbb{N}$.

Perhatikan bahwa

$P = N_2.d.99 \ldots 99.9$

$P + 1 = N_2.(d+1).00 \ldots 00.0$ maka
$T(P+1) = T(N_2) + T(d+1) + 0^2 + \ldots + 0^2 = T(N_2) + (d+1)^2 =$
$M_1 - d^2 - 9^2 \cdot k + d^2 + 2d + 1 = M_1 + (2d+1) - 9^2 \cdot k.$

$P + 2 = N_2.(d+1).00 \ldots 00.1$ maka
$T(P+2) = T(N_2) + T(d+1) + 0^2 + \ldots + 0^2 + 1^2 = T(N_2) + (d+1)^2 + 1 =$
$M_1 - d^2 - 9^2 \cdot k + (d+1)^2 + 1^2 = M_1 + (2d+1) - 9^2 \cdot k + 1^2.$

Selanjutnya akan ditentukan nilai k, d, M_1 sehingga $P + 1$ dan $P + 2$ adalah bilangan *happy*.

```
> M1 := 1839;
  ktop := iquo(M1, 81);
  for k from 0 to ktop do
  for d from 0 to 8 do
  if happy(M1 + (2·d + 1) − 9^2·k) > 0 and happy(M1 + (2·d + 1) − 9^2
     ·k + 1^2) > 0 then print(M1, k, d, happy(M1 + (2·d + 1) − 9^2·k
     + 2^2),
  happy(M1 + (2·d + 1) − 9^2·k + 3^2), M1 − d^2 − 9^2·k)   end if;
  end do; end do;
```

$$M_1 := 1839$$

$$ktop := 22$$

1839, 1, 6, -1, -1, 1722
1839, 5, 6, -1, -1, 1398
1839, 9, 5, 4, -1, 1085
1839, 15, 6, -1, -1, 588
1839, 19, 0, -1, 3, 300 **(17)**

Dari nilai di atas terlihat bahwa:

```
> k := 1; d := 6; minimalN(1722); evalf(log10(%) + k);
```

$$k := 1$$
$$d := 6$$
27799999999999999999999
23.44404479 **(18)**

```
> k := 5; d := 6; minimalN(1398); evalf(log10(%) + k);
```

$$k := 5$$
$$d := 6$$

$$277999999999999999$$
$$23.44404480 \qquad (19)$$

> $k := 9;\ d := 5;\ minimalN(1085);\ evalf(\log10(\%) + k);$

$$k := 9$$
$$d := 5$$
$$78999999999999$$
$$22.89762709 \qquad (20)$$

> $k := 15;\ d := 6;\ minimalN(588);\ evalf(\log10(\%) + k);$

$$k := 15$$
$$d := 6$$
$$277999999$$
$$23.44404480 \qquad (21)$$

> $k := 19;\ d := 0;\ minimalN(300);\ evalf(\log10(\%) + k);$

$$k := 19$$
$$d := 0$$
$$57899$$
$$23.76267106 \qquad (22)$$

>

Sehingga didapatkanlah kemungkinan terkecil untuk *k*, *d*, dan $T(N_2)$ adalah 9, 5, dan 1085.
Nilai N_2 terkecil sedemikian sehingga $T(N_2) = 1085$ adalah 78999999999999.
Nilai N yang terkecil untuk $d_0 = 6$ adalah

$$N = N_1.d_0 = N_2.d.99\ldots9.6 = 7899999999999959999999996.$$

Hal ini menunjukkan jika d0=6 maka terdapat paling banyak 6 buah bilangan *happy* terurut yaitu $N, N+1, N+2, N+3, N+4, N+5$.

```
> d0 := 7;
  for M1 from 1 to 3000 do
  if happy(M1 + d0^2) > 0 and happy(M1 + (d0 + 1)^2) > 0
     and happy(M1 + (d0 + 2)^2) > 0 then print(M1, happy(M1
     + (d0 - 1)^2)) end if;
  end do;
```

$$d0 := 7$$
$$568,\ -1$$
$$574,\ -1$$
$$1839,\ 7$$
$$2493,\ -1$$
$$2522,\ -1 \qquad (23)$$

>

Dari hasil yang diperoleh pada Program (17) - (22), untuk M1 = 1839 diperoleh bahwa nilai N terkecil dari barisan bilangan *happy* terurut dengan panjang enam adalah

$$N = N_1.d_0 = N_2.d.99 \ldots 9.7 = 7899999999999959999999997.$$

Hal ini menunjukkan bahwa jika d0=7 maka terdapat paling banyak 6 buah bilangan *happy* terurut yaitu $N, N+1, N+2, N+3, N+4, N+5$.

Dari Program (10) - (23) diperoleh bahwa bilangan terkecil dari barisan bilangan *happy* terurut dengan panjang enam adalah 7899999999999959999999996.
Perhatikan bahwa

```
> N := 7899999999999959999999996;
  for j from -1 to 7 do
  print(N + j, happy(N + j));
  end do:
                N := 7899999999999959999999996
                7899999999999959999999995, -1
                7899999999999959999999996, 8
                7899999999999959999999997, 8
                7899999999999959999999998, 7
                7899999999999959999999999, 5
                7899999999999960000000000, 8
                7899999999999960000000001, 4
                7899999999999960000000002, 5
                7899999999999960000000003, -1
```
(24)

sehingga
7899999999999959999999996, 7899999999999959999999997,
7899999999999959999999998, 7899999999999959999999999,
7899999999999960000000000, 7899999999999960000000001
adalah barisan terkecil dari barisan bilangan *happy* terurut dengan panjang enam.
Dari hasil yang diperoleh, ternyata 7899999999999959999999996 juga merupakan bilangan terkecil dari barisan bilangan *happy* terurut dengan panjang tujuh.